DEGREE PROJECT IN MATHEMATICS
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2023

Trajectory Optimisation of a Spacecraft Swarm Maximising Gravitational Signal

RASMUS MARÅK

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ENGINEERING SCIENCES

Author

Rasmus Maråk <marak@kth.se>
Degree Project in Optimization and Systems Theory
Degree Programme in Industrial Engineering and Management
KTH Royal Institute of Technology

Swedish title:
Banoptimering av en Rymdfarkostsvärm för att
Maximera Gravitationsignalen

Place for Project

Advanced Concepts Team
European Space Agency
TEC-SF, ESTEC
Noordwijk, The Netherlands

Examiner

Xiaoming Hu
Department of Mathematics, Optimization and Systems Theory
KTH Royal Institute of Technology

Supervisors

Pablo Gómez and Emmanuel Blazquez
Advanced Concepts Team
European Space Agency

Xiaoming Hu
Department of Mathematics, Optimization and Systems Theory
KTH Royal Institute of Technology

Abstract

Proper modelling of the gravitational fields of irregularly shaped asteroids and comets is an essential yet challenging part of any spacecraft visit and flyby to these bodies. Accurate density representations provide crucial information for proximity missions, which rely heavily on it to design safe and efficient trajectories. This work explores using a spacecraft swarm to maximise the measured gravitational signal in a hypothetical mission around the comet 67P/Churyumov-Gerasimenko. Spacecraft trajectories are simultaneously computed and evaluated using a high-order numerical integrator and an evolutionary optimisation method to maximise overall signal return. The propagation is based on an open-source polyhedral gravity model using a detailed mesh of 67P/C-G and considers the comet's sidereal rotation. We compare performance on various mission scenarios using one and four spacecraft. The results show that the swarm achieved an expected increase in coverage over a single spacecraft when considering a fixed mission duration. However, optimising for a single spacecraft results in a more effective trajectory. The impact of dimensionality is further studied by introducing an iterative local search strategy, resulting in a generally improved robustness for finding efficient solutions. Overall, this work serves as a testbed for designing a set of trajectories in particularly complex gravitational environments, balancing measured signals and risks in a swarm scenario.

The codebase is publicly available at: `https://github.com/rasmusmarak/TOSS`.

Keywords

Spacecraft Swarms, Trajectory Optimisation, Impulsive Manoeuvres, Evolutionary Algorithms, Polyhedral Gravity Model, Small Body Exploration, Discrete Spherical Grids.

Sammanfattning

En korrekt modellering av de gravitationsfält som uppstår runt irreguljärt formade asteroider och kometer är en avgörande och utmanande del för alla uppdrag till likartade himlakroppar. Exakta densitetsrepresentationer tillhandahåller viktig information för att säkerställa säkra och effektiva rutter för särsilt närgående rymdfarkoster. I denna studie utforskar vi användningen av en svärm av rymdfarkoster för att maximera den uppmätta gravitationssignalen i ett hypotetisk uppdrag runt kometen 67P/Churyumov-Gerasimenko. Rymdfarkosternas banor beräknas och utvärderas i parallella scheman med hjälp av en högre ordningens numerisk integration och en evolutionär optimeringsmetod i syfte att maximera den totala uppmätta signalen. Beräkningarna baseras på en öppen källkod för en polyhedral gravitationsmodell som använder ett detaljerat rutnät av triangulära polygoner för att representera 67P/C-G och beaktar kometens egna rotation. Vi jämför sedan prestanden för olika uppdragscenarier med en respektive fyra rymdfarkoster. Resultaten visar att svärmen uppnådde en förväntad ökning i täckning jämfört med en enskild rymdfarkost under en fast uppdragsvaraktighet. Dock resulterar optimering för en enskild rymdfarkost i en mer effektiv bana. Påverkan av dimensionshöjningen hos oberoende variabler studeras vidare genom att introducera en iterativ lokal sökstrategi, vilket resulterar i en generellt förbättrad robusthet samt effektivare lösningar. Sammantaget fungerar detta arbete som en testbädd för att studera och utforma rymdfarkosters banor i särskilt komplexa gravitationsmiljöer, samt för att balansera uppmätta signaler och risker i ett svärmscenario.

Projektets kodbas finns tillgänglig på: `https://github.com/rasmusmarak/TOSS`.

Nyckelord

Svärmande rymdfarkoster, Banoptimering, Impulsiva manövrar, Evolutionära Algoritmer, Polyhedral gravitationsmodell, Utforksning av små kroppar i solsystemet, Diskreta sfäriska rutnät.

Acknowledgements

After months of diligent work and steady focus, I am proud to finally present my thesis for the degree of Master of Science in Mathematics at the KTH Royal Institute of Technology. It has been a wondrous finale to my engineering studies and allowed me to connect with researchers and fellow space enthusiasts worldwide. I am grateful and honoured for the opportunity to work on this thesis with the Advanced Concepts Team at the European Space Agency, which ultimately resulted in a presentation at an internationally renowned conference. First of all, I would like to thank my parents for their endless support and confidence. Without you, I would have never reached this far. Secondly, I want to express my deep gratitude to my industrial supervisors, Pablo Gómez and Emmanuel Blazquez, for allowing me to take on this exciting project and guiding me in both the work and the process leading to the conference. Your diligent support, extensive knowledge of the subject, continued encouragement, and invaluable input have been indispensable for completing the thesis. In addition, I want to direct a special thanks to Pablo Gómez for his commitment, investment of time and never-ending faith in my ideas and abilities. Thirdly, I want to thank my academic supervisor and examiner, Xiaoming Hu, for assisting me with my many administrative inquiries and providing valuable advice on mathematical details throughout the thesis. I also want to thank the Advanced Concepts Team and AI Sweden for providing computational resources and letting me be a part of two very inspiring teams. Furthermore, I sincerely thank KTH Space Center and Christer Fuglesang for the expeditious support and financial aid realising my contribution to ICATT 2023. I also want to direct a word of appreciation to Filip Berendt and Ingemar Markström from the Visualisation Studio, VIC, at KTH, who are credited for the exceptional illustration of a spherical grid encapsulating a comet. Finally, I want to thank my friends for supporting me throughout my academic journey.

Stockholm, 2023
Rasmus Maråk

List of Acronyms

AI	Artificial Intelligence
ACO	Ant Colony Optimiser
ACT	Advanced Concepts Team
ANOVA	Analysis of Variance
AU	Astronomical Units
CA	Cell Automata
CFL	Courant-Friedrichs-Lewy
cPDF	Continous Probability Density Function
CPU	Central Processing Unit
DART	Double Asteroid Redirection Test
DE	Differential Evolution
dPDF	Discrete Probability Density Function
EA	Evolutionary Algorithm
EACO	Extended Ant Colony Optimiser
ESA	European Space Agency
FDM	Finite Difference Method
FEM	Finite Element Method
FLOSS	Free, Libre and Open-Source Software
FVM	Finite Volume Method
GA	Genetic Algorithm
IVP	Initial Value Problem
LBM	Lattice Boltzmann Method
LLG	Longitude-Latitude Grid
MHACO	Multi-Objective Hypervolume Based Ant Colony Optimiser
MINLP	Mixed-Integer Nonlinear Program
NASA	National Aeronautics and Space Administration

NEO	Near-Earth Objects
NLP	Nonlinear Programming
ODE	Ordinary Differential Equations
OSIRIS-REx	Origins, Spectral Interpretation, Resource Identification, and Security Regolith Explorer
PaGMO	Parallel Global Multiobjective Optimiser
PDE	Partial Differential Equations
PDF	Probability Density Function
PSO	Particle Swarm Optimisation
PyGMO	Python Parallel Global Multiobjective Optimiser
SDOG	Spheroid-Degenerate Octree Grid
67P/C-G	67P/Churyumov-Gerasimenko

List of Symbols

$\boldsymbol{a}$	Spacecraft acceleration vector $[\frac{m}{s^2}]$
C	Vertices on a graph $[-]$
$\mathcal{C}$	Collision constraint $[-]$
D	Normalized direction $[-]$
dim	Number of decision variables $[-]$
E	Binary-valued event function $[-]$
E_{tol}	Local error tolerance $[-]$
$\boldsymbol{F}$	Vector field $[-]$
$\mathcal{F}$	Body-fixed reference frame $[-]$
f	Equations of motion (system dynamics) $[-]$
f	Objective value $[-]$
f_0	Running cost function $[-]$
G	Universal gravitational constant $[\frac{m^3}{kg\ s^2}]$
G	Fully connected graph $[-]$
G	Gaussian probability density function $[-]$
G_p	Polygonal line defining plane p $[-]$
G_{pq}	Line segment representing edge q on plane p $[-]$
$\mathcal{G}$	Longitude-latitude grid $[-]$
g	Gaussian function $[-]$
g	Inequality constraint $[-]$
h	Path constraint $[-]$
h	Equality constraint $[-]$
h	Step size (numerical integration) $[-]$
h_p	Distance between a point P and a plane p $[m]$
I	Ideal simulation run time $[s]$
$\mathcal{I}$	Inertial reference frame $[-]$

$\hat{\boldsymbol{I}}$	Identity quaternion $[-]$
J	Cost function $[-]$
J	Number of active spacecraft $[-]$
$\boldsymbol{J}$	Hamiltonian system $[-]$
$\dot{\boldsymbol{J}}$	First time derivative of the Hamiltonian system $[-]$
K	Solution archive size $[-]$
L	Edges on a graph $[-]$
l	Distance between two computational points $[m]$
ℓ	Distance from origin to a plane intersection $[m]$
m_{eq}	Number of equality constraints $[-]$
m_{ineq}	Number of inequality constraints $[-]$
$\boldsymbol{m}$	State derivative $[-]$
$\boldsymbol{N}$	Outer unit normal $[-]$
$\boldsymbol{N_p}$	Outer unit normal to plane p $[-]$
$\boldsymbol{n_p}$	Outer unit normal to closed polygonal line G_p $[-]$
n_c	Number of continuous variables $[-]$
n_d	Number of discrete variables $[-]$
n_G	Number of generations $[-]$
n_m	Number of impulsive manoeuvres $[-]$
n_{pop}	Population size $[-]$
$n_{threads}$	Number of computational threads $[-]$
$NGenMark$	Standard deviation convergence speed parameter $[-]$
$\mathcal{O}$	Origin of a coordinate system $[-]$
P	Polynomial function $[-]$
P	Penalty score $[-]$
P	Continuous probability density function $[-]$
P	Computational point in cartesian elements $[m]$
P'	Projection of point P $[m]$
P''	Projection of point P' $[m]$
p	A geometric plane $[-]$
p	Probability $[-]$
p	Penalty value $[-]$
$\boldsymbol{p}$	Point on grid $\mathcal{G}$ $[-]$

Q	Positive model parameter $[-]$
Q	Discrete probability density function $[-]$
${}^{j}\mathcal{Q}_k$	Transformation of a spacecraft's position from frame k to j using quaternions $[-]$
q	Convergence speed parameter $[-]$
q_i	Individual component of a quaternion $[-]$
$\boldsymbol{q}$	Quaternion vector component $[-]$
$\hat{\boldsymbol{q}}$	Conjugate quaternion $[-]$
$\hat{\boldsymbol{q}}^{\star}$	Quaternion (standard, pure, unit) $[-]$
$\mathcal{R}$	Orthogonal cosine (rotation) matrix $[-]$
${}^{j}\mathcal{R}_k$	Transformation of a spacecraft's position frame k to j using rotation matrices $[-]$
r_i	Radius of bounding sphere i $[m]$
$\boldsymbol{r}$	Spacecraft position vector in cartesian elements $[m]$
$\dot{\boldsymbol{r}}$	First time derivative of $\boldsymbol{r}$ $[\frac{m}{s}]$
$\ddot{\boldsymbol{r}}$	Second time derivative of $\boldsymbol{r}$ $[\frac{m}{s^2}]$
${}^{j}\boldsymbol{r}$	A spacecraft's position in frame j $[m]$
${}^{j}\hat{\boldsymbol{r}}$	Pure quaternion for a spacecraft's position in frame j $[-]$
S	Closed-surface computational mesh $[-]$
S	Solution archive $[-]$
S_p	Individual polygonal plane p on S $[-]$
$\mathcal{S}$	A smooth manifold $[-]$
$\mathcal{S}_i$	Sphere i $[-]$
s	Solution vector $[-]$
T	Measured simulation time $[s]$
t	Time $[s]$
t_{sim}	Simulation run time $[s]$
$threshold$	Threshold parameter $[-]$
U	Volume $[m^3]$
$\mathcal{U}$	Control domain $[-]$
$\hat{u}$	Unit control vector (directional) $[-]$
$\hat{\boldsymbol{u}}$	Euler axis $[-]$
$\boldsymbol{u}$	Spacecraft control vector (manoeuvre) $[\frac{m}{s^2}]$

V	Gravitational potential $[\frac{J}{kg}]$
V	Vertex on a graph $[-]$
V_{x_i}	Partial derivative of the gravitational potential $[-]$
V_r	Gradient of the gravitational potential $[-]$
V_{ij}	Second-order partial derivative of the gravitational potential $[-]$
V_c	Coverage score $[-]$
$\hat{v}$	Spacecraft unit velocity vector $[-]$
v_{max}	Maximum velocity $[\frac{m}{s}]$
$\boldsymbol{v}$	Spacecraft velocity vector $[\frac{m}{s}]$
$\boldsymbol{v_I}$	A spacecraft's inertial velocity (cartesian elements) $[\frac{m}{s}]$
x	Analytical solution $[-]$
$\hat{x}$	Numerical approximation of solution $[-]$
$\boldsymbol{x}$	Spacecraft state vector $[-]$
$\boldsymbol{x}_0$	Initial state vector $[-]$
$\boldsymbol{x}_f$	Final state vector $[-]$
$\dot{\boldsymbol{x}}$	Equations of motion (spacecraft dynamics) $[-]$
$(\boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3})$	Basis terms for a local coordinate system $[-]$
$[x_1, x_2, x_3]$	Cartesian coordinates $[m]$
α	Weight balance parameter $[-]$
α	Angle of right-ascension $[rad]$
α	Accuracy parameter
β	Dynamic bias term $[-]$
Γ	Control magnitude $[\frac{m}{s^2}]$
ΔL	Local length interval $[m]$
Δt	Spacecraft engine burn time $[s]$
Δt	Local time step $[s]$
Δv	Velocity increment $[\frac{m}{s}]$
δ	Estimate of local error ϵ $[-]$
δ	Angle of declination $[rad]$
ϵ	Residual error $[-]$
ϵ_{abs}	Absolute error tolerance $[-]$
ϵ_{rel}	Relative error tolerance $[-]$
θ	Spin angle $[rad]$

λ	Penalty scaling factor $[-]$
μ	Mean value $[-]$
ξ	Principle rotation angle $[rad]$
ρ	Small-body mass density distribution $[\frac{kg}{m^3}]$
ρ	Evaporation speed parameter $[-]$
σ_p	Binary control variable for plane p $[-]$
σ	Standard deviation $[-]$
τ	Pheromone strength $[-]$
τ	Taylor expansion model $[-]$
ϕ	Precession angle $[rad]$
ϕ	Hermite basis function $[-]$
ψ	Nutation angle $[rad]$
ψ	Hermite basis function $[-]$
Ω	Instantaneous rotation vector $[\frac{rad}{s}]$
$\dot{\Omega}$	First time derivative of Ω $[-]$
ω	Angular velocity (small body rotation) $[\frac{rad}{s}]$
ω	Oracle parameter $[-]$
ω	Weight factor $[-]$

Contents

Proceedings

R. Maråk, E. Blazquez, P. Gómez. “Trajectory Optimization of a Swarm Orbiting 67P/Churyumov-Gerasimenko Maximising Gravitational Signal”. In: *Proceedings of the 9th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 2023.

Chapter 1

Introduction

Over the last decades, small-body exploration has gained a significant role in space research. With a relative proximity to Earth, many smaller solar system bodies, including near-Earth objects (NEOs), main-belt asteroids and comets, centaurs, and trans-Neptunium objects, offer unique opportunities for scientific investigation. It is believed that most of these bodies are remnants from the formation of our solar system and, as such, act as geological time capsules. By studying their gravitation field, one can therefore derive valuable clues on their geological properties, which may provide critical insight into the history of our solar system [1–3]. Another burgeoning application includes planetary defence, where a detailed understanding of mass density distribution represents a crucial aspect for a successful redirection mission [4]. For these reasons, the last two decades of space exploration have seen several small-body missions, including targets of observation, landing and collecting samples [4–8]. However, operating near these bodies is a challenging task. With a relatively low mass rendering a particularly weak gravitational field, smaller bodies like asteroids and comets are often characterised by highly rugged topographies and poorly constrained surface properties. Along with features such as irregular shapes, heterogeneous mass density distribution and a rotating motion [9], this typically results in complex dynamics where ideal Keplerian motion becomes unstable, which can lead to undesired escape or collision trajectories [10]. With often limited a priori knowledge of these characteristics, operating near, on, or inside these bodies requires a high level of autonomy and a robust gravitational field model.

In recent years, however, more work has been seen investigating the use of distributed approaches for improving the learning process of small-body characteristics [11–13]. A prominent example is ESA's upcoming mission, Hera, to the Didymos binary asteroid system. With modern advances in autonomous control and miniaturisation of embedded systems, Hera will be using two 6U CubeSats to measure the two asteroids' sizes, shapes, and compositions as well as the impact of NASA's DART mission [4, 14, 15]. The benefits of swarming spacecraft for these purposes are many: it allows for launching smaller spacecraft more frequently and at a lower cost than larger spacecraft, reduced learning times, simultaneous observations and distributed risks [16]. However, due to the complex dynamical nature of these bodies, designing safe and efficient trajectories for swarm-based missions becomes increasingly challenging. In fact, the main body of this work is dedicated to this task: to determine a suitable trajectory optimisation strategy for maximising the information gained by swarming spacecraft on the gravitational field near an irregularly shaped smaller celestial body.

1.1 Related work

By reviewing related work, we aim to distil insights, best practices and fitting research gaps that will inform the design process of a suitable optimisation strategy. To focus the analysis, we will primarily overview recent advances on the two topics of interest: gravity field modelling and trajectory optimisation of swarming spacecraft.

1.1.1 Modelling Gravitational Fields

In the context of geodesy, the literature typically presents three conventional gravity field representations: spherical (or spheroidal) harmonics, mascon models and polyhedral gravity models. The former approach is a multipole expansion method where the accelerations can be computed using the spatial derivatives of the potential harmonics [10]. Following its spherical nature, the resulting model is particularly suited for bodies with strong axial symmetry and regularity. However, the convergence typically becomes erratic inside the Brillouin sphere - a sphere co-centred with the expansion and precisely covering the entire body - which renders the model unsuitable for highly irregularly shaped celestial bodies. The second method attempts the gravity field representation by filling the body's occupied volume with smaller mass concentrations, or mascons, thus limiting the performance by the detail of the discretisation. In addition, without clearly defined surface boundaries, evaluating points inside the interior field becomes increasingly problematic. In contrast, the polyhedral gravity model introduces a multifaceted polyhedron to approximate the target body's closed surface topography. When assuming an internal mass-density homogeneity, it is then possible to derive an analytical formulation for the regional potential field. In the literature of polyhedral modelling, prominent work includes that of Barnett [17], who derived a closed analytical representation of the gravitational attraction considering a triangular mesh representing the body's surface model, Pohanka [18] who continued the work with general triangulation, Werner and Scheeres [19] who presented the analytical expression for the gravity gradient tensor and Petrović [20] who provided a closed expression of the full gravity tensor considering a polyhedral source, including the gravitational potential and the two consecutive derivatives of the gradient matrix. The final expression represents a twice application of the Gaussian divergence theorem to transition from volume to line integrals along the polyhedron's surface of flux. The model was later refined by Tsoulis and Petrović [21] who derived and managed the singularities that appear at specific compute locations. Consequently, the polyhedral model enables an accurate gravity field representation of highly irregularly shaped celestial bodies even inside the Brillouin sphere [19]. However, the performance is inherently limited by the fidelity of the shape model and the assumption of mass density homogeneity, which is rarely the case for smaller solar system bodies.

With recent progress in machine learning and neural networks, there has been a growing interest in developing new data-driven models to provide more accurate information about these complex bodies. One such model is the neural density field, a versatile tool that can accurately describe the density distribution of a body's mass and its internal and external shape with few prior requirements. GeodesyNets is a neural network that attempts to solve the gravity inversion problem by learning a three-dimensional and differentiable function representing the body's density, referred to as the neural density field [22]. By training on either real or synthetic ground truth data, the body's shape and other geodetic properties can quickly be recovered utilising that its integration leads to a gravitational field model. This representation has several advantages as it requires no prior shape model, converges inside the Brillouin sphere and is extensible even to heterogeneous density distributions. Thus avoiding the limitations of spherical harmonics, mascons and polyhedral gravity models [22]. Previous work includes training on synthetic ground truth data without any shape models and on data recovered from the OSIRIS-REx mission visiting the asteroid 101955 Bennu with an available shape model [23, 24]. Although the former non-differential approach showed comparable performance with a polyhedral gravity model assuming homogenous mascons, the latter differential training on real data did not significantly improve the fidelity of the modelled gravitational field. This result was most likely due to uncertainties of non-gravitational forces in the considered trajectories [23]. Hence, it is of great interest to generate more realistic trajectories based on a polyhedral gravity model to study geodesyNets' performance further.

1.1.2 Trajectory Optimisation of Spacecraft Swarms

The challenge of designing trajectories near smaller celestial bodies can be derived from their complex and unstable dynamic environments, which typically makes traditional analytical solutions of Keplerian motion infeasible. Under these circumstances, mission designers are required to plan robust control strategies, which can be done using principles of optimal control [25]. The core idea is to determine a control sequence for a dynamical system that optimises the transition of states according to some pre-defined measure while also satisfying a set of constraints. According to Betts [26], such optimal control problems can be solved using both indirect and direct approaches. Although indirect methods provide excellent accuracy, we will focus on direct methods since they generally provide more robust search processes and lower computational effort without requiring an initial system intuition.

In the literature on small-body exploration, spacecraft swarms have recently demonstrated promising results regarding scientific returns compared to more conventional monolithic approaches [27]. With the prospective benefits of swarms, numerous papers have proposed methods for designing a set of trajectories around smaller bodies to obtain a high-accuracy mapping of body attributes and efficient spacecraft navigation. In terms of direct approaches, examples include mixed-integer linear programming schemes with gradient-based global search processes [27], sampling-based model predictive control frameworks [28] and two-staged optimisation schemes including a nonlinear optimisation process for trajectory planning and a genetic multi-spacecraft travelling salesman problem [29]. Although these articles have used widely different approaches, a common feature is that swarm missions tend to require the simultaneous optimisation of several competing objectives such as viewing angles, keeping in line of sight for communication purposes, minimising fuel consumption, maximising some defined measure of coverage and avoiding direct impacts. In order to satisfy these objectives, the optimisation space expands significantly, creating an even greater demand for robust and efficient search strategies. For these types of problems, population-based optimisation schemes are well suited. With similar motivation, Pearl, J.M et al. [30] proposes an alternative approach using a curvilinear surface-based gravity model and differential evolution for optimising the initial state of a single-spacecraft trajectory. The performance of each candidate solution is then determined by integrating the system dynamics and computing the fitness along the state propagation. With population-based algorithms demonstrating promising results for designing trajectories around smaller bodies, we will use these findings as a reference for selecting a suitable optimisation strategy.

1.2 Objectives and Limitations

The main objective of this thesis is to develop a method of optimising trajectories for a spacecraft swarm operating near a smaller celestial body with the target of maximising the coverage of its gravitational field. Considering the results from previous work, we propose a method employing a population-based evolutionary optimisation scheme to determine the initial state and control sequence corresponding to each spacecraft in the swarm. In order to solve the equations of motion, we rely on a polyhedral gravity model to represent the gravitational signal. To demonstrate the method, we implement it for a prospective mission to the comet 67P/Churyumov-Gerasimenko and compare the relative performance between a single spacecraft and a swarm considering several quality measures.

The contribution of this work is ultimately threefold. First, we aim at formulating and solving a nonlinear optimisation problem considering a spacecraft swarm for the defined context and objective. The second and subsequent contribution is a data set representing the measured accelerations that can be used to study the training and robustness of geodesyNets further [24, 31]. Finally, even though the method is demonstrated for a specific mission, the underlying framework will be designed as an open-source modular codebase[1] in order to remain relevant for a broader range of mission objectives, such as surface observation, inter-spacecraft communication and training of on-board AI models.

In the numerical study, we limit ourselves to evaluating a single swarm setting. Nevertheless, with the level of generality in the code, we aim to present a method applicable to varying sizes of swarms and bodies of interest.

1.3 Report Outline

The thesis is organised as follows: Chapter 2 acquaints the reader with the general concepts of mechanics and coordinate transformation necessary to define space flight near rotating bodies. In addition, we present the basic formulation of the polyhedral gravity model and the equations of motion adapted for impulsive manoeuvres. In Chapter 3, we define a nonlinear optimisation problem suited for the thesis objective. Section 3.3 is particularly important as it introduces a discretisation of the space around the target body from which a coverage score can be defined to represent the measured gravitational signal. Chapter 4 presents an appropriate method for solving the optimisation problem, focusing specifically on population-based evolutionary methods. Next, in Chapter 5, we introduce the necessary numerical methods and techniques for solving the suggested equations of motion and optimisation problem. To demonstrate the models and their performance, Chapter 6 presents numerical results on scaling properties, hyperparameter tuning and three mission scenarios considering a specific target body. Chapter 7 discusses the acquired results, limitations, computational properties and general performance. Finally, Chapter 8 concludes the thesis with general remarks and recommendations for future work.

---

[1] Thesis codebase (TOSS): https://github.com/rasmusmarak/TOSS

Chapter 2

Spacecraft Dynamics

The second chapter is devoted to familiarising the reader with the dynamics characterising operations near smaller, irregularly shaped rotating bodies. A basic introduction is first given on appropriate reference frames and coordinates transformation. The succeeding two sections introduce the traditional derivation of the equations of motion, focusing mainly on an expression representing the gravitational attraction acting on the spacecraft and a brief overview of the polyhedral gravity model. In the last section, we present the specific set of equations defining the state transition over time using first-order differential equations. Lastly, we introduce impulsive manoeuvres as the system perturbation and define the associated control variable.

2.1 Reference Frames

As customary in the field [3, 9, 25], we introduce two frames of reference to model the motion of a spacecraft near a rotating body. The former is the inertial frame, which defines a non-accelerating coordinate system where any considered object adheres to Newton's first law of motion. In other words, we assume the spacecraft to have a constant velocity unless subject to any perturbing forces. Since the assumption of Newtonian motion enables a more straightforward form of mechanics, it will be the preferred reference frame for defining subsequent trajectories. Furthermore, to model the gravitational attraction acting on a spacecraft by a rotating body, it follows naturally to adopt a non-inertial reference frame that rotates at the same rate and in the same plane as the body's surface. Hence, we introduce a body-fixed frame to enable a representation of the gravitational field. The formal definition of the two frames are:

$\boldsymbol{\mathcal{I}} = (\hat{\boldsymbol{\imath}}, \hat{\boldsymbol{\jmath}}, \hat{\boldsymbol{k}})$: In the inertial frame, the origin $\mathcal{O}$ is fixed at the small-body centre of mass. The $\hat{\boldsymbol{\jmath}}$ axis is defined by a hyperplane parallel with the direction vector between the two points on the body with the greatest distance. To complete the right-hand system, $\hat{\boldsymbol{\imath}}$ and $\hat{\boldsymbol{k}}$ are defined orthogonal to each other and to $\hat{\boldsymbol{\jmath}}$.

$\boldsymbol{\mathcal{F}} = (\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3)$: The origin of the body-fixed frame is located at the small-body centre of mass and rotates synchronously with its surface. Hence, the three axes align with the body's principal moments of inertia. In detail, $\hat{e}_1$ and $\hat{e}_3$ are defined in the direction of the minimum and maximum principal axes, which are expressed by angles of declination δ and right-ascension α relative to the inertial frame. The final axis $\hat{e}_2$ is selected to complete the orthogonality of the new right-hand system. Finally, we assume a spin state with constant angular velocity and principal rotation around $\hat{e}_3$. Notably, the direction of the instantaneous axis can also evolve with time, commonly referred to as precession. However, as an effect of considering a shorter mission duration, we limit the study to the assumption of a fixed rotation axis.

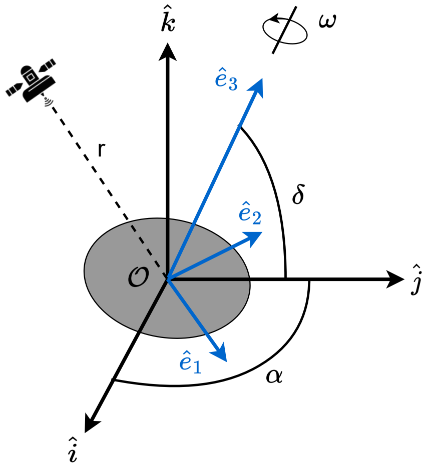


Figure 2.1: Small-body inertial ($\boldsymbol{\mathcal{I}}$) and fixed ($\boldsymbol{\mathcal{F}}$) reference frames.

To simplify the notation, we define a vector $\boldsymbol{v}$ in reference frame $\mathcal{A}$ as ${}^{\mathcal{A}}\boldsymbol{v}$. However, the superscript will remain omitted from the equation whenever the current frame is trivial or extraneous. A coordinate transformation from the inertial frame $\mathcal{I}$ to the body-fixed frame $\mathcal{F}$ can then be expressed as ${}^{\mathcal{I}}\hat{\boldsymbol{v}} = {}^{\mathcal{I}}\mathcal{T}_{\mathcal{F}}\,{}^{\mathcal{F}}\hat{\boldsymbol{v}}$ using corresponding rotation matrices $\mathcal{R}_{x,y,z}(\theta)$ for a rotation of angle θ about each cartesian axis.

2.1.1 Coordinate Transformation

There are several ways of describing the angular position of a rotating rigid body. A common choice is to employ the three Euler angles ϕ, θ and ψ, representing the orientation of one basis set relative to another. The body-fixed frame can then be obtained from the inertial frame by a sequence of rotations through each angle

$${}^{\mathcal{F}}\mathcal{R}_{\mathcal{I}} = [R_3(\psi)][R_1(\theta)][\mathcal{R}_3(\phi)], \tag{2.1}$$

where the first rotation is defined by the precession angle ϕ, the second rotation by the nutation angle θ and the third rotation by the spin angle ψ [32, 33]. In detail, the three rotations are explicitly defined as

$$[R_3(\psi)] = \begin{bmatrix} cos\ \psi & sin\ \psi & 0 \\ -sin\ \psi & cos\ \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.2}$$

$$[R_1(\theta)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\ \theta & sin\ \theta \\ 0 & -sin\ \theta & cos\ \theta \end{bmatrix}, \tag{2.3}$$

$$[R_3(\psi)] = \begin{bmatrix} cos\ \phi & sin\ \phi & 0 \\ -sin\ \phi & cos\ \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.4}$$

The resulting orthogonal cosine matrix corresponding to the three Euler rotations can then be defined as

$$^{\mathcal{F}}\mathcal{R}_{\mathcal{I}} = \begin{bmatrix} -sin\ \phi\ cos\ \theta\ sin\ \psi & cos\ \phi\ cos\ \theta\ sin\ \psi + sin\ \phi\ cos\ \psi & sin\ \theta\ sin\ \psi \\ -sin\ \phi\ cos\ \theta\ cos\ \psi & cos\ \phi\ cos\ \theta\ cos\ \psi + sin\ \phi\ sin\ \psi & sin\ \theta\ cos\ \psi \\ sin\ \psi\ sin\ \theta & -cos\ \phi\ sin\ \theta & cos\ \theta \end{bmatrix}. \tag{2.5}$$

However, it should be noted that the traditional Euler angle representation generally suffers from singularities that arise from a so-called gimbal lock. Intuitively, this phenomenon derives from the indistinguishability of precession and spin angle changes that arise when the nutation is at some critical value, such as zero [33]. A common strategy for dealing with these singularities is to apply another transformation representation, such as quaternions, which avoids the singularities altogether. In order to formulate a quaternion rotation, we first recall from Euler's rotation theorem that any two cartesian coordinate frames can be related by a unique rotation about a single line through their common origin. In this definition, the line is referred to as the Euler axis $\hat{\boldsymbol{u}}$ and the angle of rotation as the principal angle ξ. Accordingly, we can obtain the body-fixed frame $\mathcal{F}$ from the inertial frame $\mathcal{I}$ by a single rotation about $\hat{\boldsymbol{u}}$. Using Rodrigues' rotation formula, the unit vector rotation corresponding to the transformation can be defined as

$$\begin{bmatrix} \hat{\boldsymbol{e}}_{\boldsymbol{1}} \\ \hat{\boldsymbol{e}}_{\boldsymbol{2}} \\ \hat{\boldsymbol{e}}_{\boldsymbol{3}} \end{bmatrix} = cos\xi \begin{bmatrix} \hat{\boldsymbol{i}} \\ \hat{\boldsymbol{j}} \\ \hat{\boldsymbol{k}} \end{bmatrix} + (1 - cos\xi)(\hat{\boldsymbol{u}} \cdot \begin{bmatrix} \hat{\boldsymbol{i}} \\ \hat{\boldsymbol{j}} \\ \hat{\boldsymbol{k}} \end{bmatrix})\hat{\boldsymbol{u}} + sin\xi\ (\hat{\boldsymbol{u}} \times \begin{bmatrix} \hat{\boldsymbol{i}} \\ \hat{\boldsymbol{j}} \\ \hat{\boldsymbol{k}} \end{bmatrix}). \tag{2.6}$$

Another approach is to describe the Euler axis in terms of its direction cosines l, m and n along the original inertial frame. By substituting the orthonormal quantities

$$\hat{\boldsymbol{u}} = l\hat{\boldsymbol{i}} + m\hat{\boldsymbol{j}} + n\hat{\boldsymbol{k}} \quad \text{where} \quad l^2 + m^2 + n^2 = 1, \tag{2.7}$$

into Eq. (2.6), performing ordinary vector operations and recalling that the rows of the rotation matrix defined in Eq. (2.5) comprise the direction cosines, we may instead formulate the rotation matrix in terms of the Euler axis direction cosines and the principal angle ξ resulting in

$$
{}^{\mathcal{F}}\mathcal{R}_{\mathcal{I}} = \begin{bmatrix} l^2(1-cos\xi)+cos\xi & lm(1-cos\xi)+nsin\xi & ln(1-cos\xi)-msin\xi \\ lm(1-cos\xi)-nsin\xi & m^2(1-cos\xi)+cos\xi & mn(1-cos\xi)+lsin\xi \\ ln(1-cos\xi)+msin\xi & mn(1-cos\xi)-lsin\xi & n^2(1-cos\xi)+cos\xi \end{bmatrix}. \tag{2.8}
$$

Now, introducing the quaternion representation, we define a quaternion $\hat{\boldsymbol{q}}$ as composed of four numbers

$$
\hat{\boldsymbol{q}} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{q} \\ q_4 \end{Bmatrix}, \tag{2.9}
$$

where $\boldsymbol{q}$ is the vector part and q_4 the scalar part. We refer to $\hat{\boldsymbol{q}}$ as a pure quaternion if the scalar part equals zero. Similarly, a unit quaternion implies that the norm $||\hat{\boldsymbol{q}}|| = \sqrt{||\boldsymbol{q}||^2 + q_4^2}$ is equal to one. Furthermore, quaternions also obey the traditional addition and scalar multiplication rules, where addition is both associative and communicative [33]. The product between two quaternions can be defined as

$$
\hat{\boldsymbol{p}} \otimes \hat{\boldsymbol{q}} = \begin{Bmatrix} p_4\boldsymbol{q} + q_4\boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q} \\ p_4 q_4 - \boldsymbol{p} \cdot \boldsymbol{q} \end{Bmatrix}, \tag{2.10}
$$

where $\boldsymbol{q} \times \boldsymbol{p} = -(\boldsymbol{p} \times \boldsymbol{q})$, thus implying that the multiplication is in general not communicative, that is

$$
\hat{\boldsymbol{p}} \otimes \hat{\boldsymbol{q}} \neq \hat{\boldsymbol{q}} \otimes \hat{\boldsymbol{p}}, \tag{2.11}
$$

with an exception for the identity quaternion

$$
\hat{\boldsymbol{I}} = \begin{Bmatrix} \boldsymbol{0} \\ 1 \end{Bmatrix}. \tag{2.12}
$$

Finally, the conjugate $\hat{\boldsymbol{q}}^*$ is given by multiplying the vector part of $\hat{\boldsymbol{q}}$ by -1, consequently changing the sign of the corresponding vector components. Focusing on the unit quaternion, we can define a new quaternion that considers the Euler axis $\hat{\boldsymbol{u}}$ and principal angle ξ as

$$
\hat{\boldsymbol{q}} = \begin{Bmatrix} sin(\frac{\xi}{2})\hat{\boldsymbol{u}} \\ cos(\frac{\xi}{2}) \end{Bmatrix}. \tag{2.13}
$$

Recalling the quantities presented in Eq. (2.7), we define each component of $\hat{\boldsymbol{q}}$ as

$$
q_1 = lsin(\frac{\xi}{2}), \quad q_2 = msin(\frac{\xi}{2}), \quad q_3 = nsin(\frac{\xi}{2}), \quad q_4 = cos(\frac{\xi}{2}). \tag{2.14}
$$

By calculating the corresponding conjugate quaternion $\hat{\boldsymbol{q}}^*$ and employing standard trigonometric identities, we can then derive a new representation for the direction cosine matrix presented Eq. (2.5) using quaternions:

$$
{}^{\mathcal{F}}\mathcal{R}_{\mathcal{I}} = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2-2q_0q_3 & 2q_1q_3+2q_0q_2 \\ 2q_1q_2+2q_0q_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3-2q_0q_1 \\ 2q_1q_3-2q_0q_2 & 2q_2q_3+2q_0q_1 & 1-2q_1^2-2q_2^2 \end{bmatrix}. \tag{2.15}
$$

We have therefore shown that a unit quaternion represents a rotation about the axis $\hat{\boldsymbol{u}}$ for some angle ξ. In this thesis, however, we will consider transforming a spacecraft's position from the inertial frame to the body-fixed frame, that is ${}^{\mathcal{F}}\mathcal{Q}_{\mathcal{I}}(\boldsymbol{r}) : {}^{\mathcal{I}}\boldsymbol{r} \to {}^{\mathcal{F}}\boldsymbol{r}$. In order to define such a rotation, we first introduce two pure quaternions based on the position in each frame:

$$
{}^{\mathcal{I}}\hat{\boldsymbol{r}} = \begin{Bmatrix} {}^{\mathcal{I}}\boldsymbol{r} \\ 0 \end{Bmatrix} \quad {}^{\mathcal{F}}\hat{\boldsymbol{r}} = \begin{Bmatrix} {}^{\mathcal{F}}\boldsymbol{r} \\ 0 \end{Bmatrix}. \tag{2.16}
$$

Using Eq. (2.10), we compute the product of $\hat{\boldsymbol{q}}$ and ${}^{\mathcal{I}}\hat{\boldsymbol{r}}$ to obtain

$$
\hat{\boldsymbol{q}} \otimes {}^{\mathcal{I}}\hat{\boldsymbol{r}} = \begin{Bmatrix} cos(\frac{\xi}{2})\,{}^{\mathcal{I}}\boldsymbol{r} + sin(\frac{\xi}{2})(\hat{\boldsymbol{u}} \times {}^{\mathcal{I}}\boldsymbol{r}) \\ -sin(\frac{\xi}{2})(\hat{\boldsymbol{u}} \cdot {}^{\mathcal{I}}\boldsymbol{r}) \end{Bmatrix}. \tag{2.17}
$$

Multiplying with the conjugate $\hat{\boldsymbol{q}}^*$ on the right, Eq. (2.17) then results in the desired position quaternion, that is

$$
{}^{\mathcal{F}}\hat{\boldsymbol{r}} = (\hat{\boldsymbol{q}} \otimes {}^{\mathcal{I}}\hat{\boldsymbol{r}}) \otimes \hat{\boldsymbol{q}}^* = \begin{Bmatrix} {}^{\mathcal{F}}\boldsymbol{r} \\ {}^{\mathcal{F}}r_4 \end{Bmatrix}. \tag{2.18}
$$

The operation presented in Eq. (2.18) therefore represents the rotation of ${}^{\mathcal{I}}\boldsymbol{r}$ around the axis $\hat{\boldsymbol{u}}$ through angle ξ. Conversely, the rotation through an angle $-\xi$ is obtained by the much similar operation $\hat{\boldsymbol{q}}^* \otimes (\hat{\boldsymbol{q}} \otimes {}^{\mathcal{I}}\hat{\boldsymbol{r}})$ [33]. This inverse relation can be ensured by the following substitution, making use of the previously presented quaternion properties:

$$
\begin{aligned}
\hat{\boldsymbol{q}} \otimes (\hat{\boldsymbol{q}}^* \otimes {}^{\mathcal{I}}\hat{\boldsymbol{r}} \otimes \hat{\boldsymbol{q}}) \otimes \hat{\boldsymbol{q}}^* &= (\hat{\boldsymbol{q}} \otimes \hat{\boldsymbol{q}}^*) \otimes {}^{\mathcal{I}}\hat{\boldsymbol{r}} \otimes (\hat{\boldsymbol{q}} \otimes \hat{\boldsymbol{q}}^*) && (2.19)\\
&= \hat{\boldsymbol{I}} \otimes {}^{\mathcal{I}}\hat{\boldsymbol{r}} \otimes \hat{\boldsymbol{I}} && (2.20)\\
&= (\hat{\boldsymbol{I}} \otimes {}^{\mathcal{I}}\hat{\boldsymbol{r}}) \otimes \hat{\boldsymbol{I}} && (2.21)\\
&= {}^{\mathcal{I}}\hat{\boldsymbol{r}} \otimes \hat{\boldsymbol{I}} && (2.22)\\
&= {}^{\mathcal{I}}\hat{\boldsymbol{r}}. && (2.23)
\end{aligned}
$$

For this thesis, we limit ourselves by the assumption that the small body's instantaneous axis of rotation coincides with the Euler axis and that the body is rotating about this axis with a constant angular velocity ω. Consequently, the rotation angle is simply obtained by $\xi(t) = \omega t$, where t is the specific time corresponding to position ${}^{\mathcal{I}}\boldsymbol{r}(t)$. The transformation between the two reference frames is expressed as

$$
{}^{\mathcal{F}}\hat{\boldsymbol{r}}(t) = {}^{\mathcal{F}}\mathcal{Q}_{\mathcal{I}}({}^{\mathcal{I}}\hat{\boldsymbol{r}}(t), \omega, t) = \hat{\boldsymbol{q}}^* \otimes (\hat{\boldsymbol{q}} \otimes {}^{\mathcal{I}}\hat{\boldsymbol{r}}). \tag{2.24}
$$

The benefits of using quaternions for this application are many. First, quaternion rotations are generally more intuitive than traditional Euler angles as they only deal with one form of rotation compared to the latter and its complex sequential nature.

Next, quaternions entirely avoid any singularities, such as gimbal lock. Finally, given their compact representation and that they only deal with numeric values rather than trigonometric functions, they are typically more computationally efficient than most other alternatives. However, quaternions also suffer from a notably involved algebra and a lack of unique representation for the attitude.

2.2 Dynamics Near Rotating Bodies

To describe the motion of a spacecraft in the vicinity of a rotating body, we employ a classic representation of a particle's motion in a body-fixed Euclidean space. The formulation is simplified by assuming the particle to have negligible mass following the different order of magnitude between a spacecraft and a smaller solar system body. When considering a body with uniform motion about its instantaneous axis ${}^{\mathcal{F}}\hat{\boldsymbol{z}}$, the equations of motion are most efficiently expressed in the body-fixed frame. The benefits of doing so are unambiguous; by deriving integrated results in a non-inertial frame, there will be no need for further transformations in order to evaluate the gravitational attraction acting on the spacecraft. Furthermore, following the property of uniform rotation, the equations remain time-invariant, simplifying the formulation. Finally, the equations of motion are defined as

$$\ddot{\boldsymbol{r}} + 2\Omega \times \dot{\boldsymbol{r}} + \Omega \times (\Omega \times \boldsymbol{r}) + \dot{\Omega} \times \boldsymbol{r} = V_{\boldsymbol{r}}, \tag{2.25}$$

where $\boldsymbol{r}$ is a body-fixed vector from the small-body centre of mass, $\dot{\boldsymbol{r}}$ and $\ddot{\boldsymbol{r}}$ are the first and second time derivatives of $\boldsymbol{r}$ with respect to $\mathcal{F}$, Ω is the instantaneous rotation vector with magnitude $|\Omega| = \boldsymbol{\omega}$ representing the angular velocity of the rotation, and V_r is the gradient of the gravitational potential $V(\boldsymbol{r})$, which is also time-invariant in $\mathcal{F}$ [34]. We will now focus on studying the conserved quantities of Eq. (2.25) to obtain a richer understanding of its fundamental properties.

First, by the assumption of uniform rotation, it follows that $\dot{\Omega} = 0$, consequently reducing Eq. (2.25) into

$$\ddot{\boldsymbol{r}} + 2\Omega \times \dot{\boldsymbol{r}} + \Omega \times (\Omega \times \boldsymbol{r}) = V_{\boldsymbol{r}}. \tag{2.26}$$

Then, by introducing an explicit definition of the efficient potential, Eq. (2.26) can be reformulated as a complete system, including terms for both the Coriolis force and the potential force [35]. Moreover, by redefining the position and inertial velocity of the spacecraft using canonical variables, $\boldsymbol{q} = \boldsymbol{r}$ and $\boldsymbol{p} = \Omega \times \boldsymbol{r} + \dot{\boldsymbol{r}}$, one can use Legendre transformation to reach the Hamiltonian form of the dynamical system $[\dot{\boldsymbol{q}}, \dot{\boldsymbol{p}}]^T$. We define this quantity as

$$\boldsymbol{J} = \frac{1}{2}\dot{\boldsymbol{r}} \cdot \dot{\boldsymbol{r}} - \frac{1}{2}(\Omega \times \boldsymbol{r}) \cdot (\Omega \times \boldsymbol{r}) - V(\boldsymbol{r}). \tag{2.27}$$

Differentiating $\boldsymbol{J}$ with respect to time results in the equation

$$\dot{\boldsymbol{J}} = \dot{\boldsymbol{r}} \cdot \ddot{\boldsymbol{r}} - (\Omega \times \boldsymbol{r}) \cdot (\dot{\Omega} \times \boldsymbol{r}) - (\Omega \times \boldsymbol{r}) \cdot (\Omega \times \dot{\boldsymbol{r}}) - V_{\boldsymbol{r}}(\boldsymbol{r}) \cdot \dot{\boldsymbol{r}}. \tag{2.28}$$

Now, taking the dot product of Eq. (2.25) with $\dot{\boldsymbol{r}}$, rearranging the resulting equation to equal zero, and then combining it with the differentiated function of $\boldsymbol{J}$ as presented in Eq. (2.28), we obtain a system of equations defined as

$$\dot{\boldsymbol{J}} = \dot{\Omega} \cdot (\boldsymbol{v_I} \times \boldsymbol{r}), \tag{2.29}$$

$$\boldsymbol{v_I} = \dot{\boldsymbol{r}} + \Omega \times \boldsymbol{r}, \tag{2.30}$$

where $\boldsymbol{v_I}$ denotes the inertial velocity of the spacecraft in motion near the body of interest [36]. Consequently, if the asteroid follows the presumption on uniform principal axis rotation, that is, assuming $\dot{\Omega} \equiv 0$, it follows that the quantity $\boldsymbol{J}$ remains conserved and defines the Jacobi integral for Eq. (2.25). Conversely, it also holds that the equations of motion in $\mathcal{F}$ are time-invariant whenever $\dot{\Omega} \equiv 0$. By the uniform rotation property, one can therefore assume the Jacobi integral $J(\boldsymbol{r}, \dot{\boldsymbol{r}})$ to remain constant for the subsequent motion of a spacecraft with a given initial position and velocity [37]. In scalar form, the equations of motion can be reduced to

$$V_x = \ddot{x} - 2\omega\dot{y} - \omega^2 x, \tag{2.31}$$

$$V_y = \ddot{y} - \omega^2 y + 2\omega\dot{x}, \tag{2.32}$$

$$V_z = \ddot{z}. \tag{2.33}$$

In this thesis, we approximate the derivative of the body's gravitational potential, $[V_x, V_y, V_z]^T$, acting on the spacecraft in the body-fixed frame using a series of physical models defining a polyhedral gravity model.

2.3 Polyhedral Gravity Model

The study of the analytical formulation for the gravitational signal considering a polyhedral source is widely known for its long history of active research and steady flux of original contributions improving the existing theoretical and computational foundation [38]. A principal component of the polyhedral framework is the definition of a prismatic solid consisting of a finite number of faces defined by interconnected vertices similar to the general representation of a graph [39]. For a larger number of polygonal faces, the polyhedral model consequently results in a finer approximation of three-dimensional arbitrarily shaped bodies. From these shape models, it is then possible to derive a formulation for the corresponding regional potential fields, including the gravitational potential [20]. However, it was not until the late twentieth century that the geophysical community saw a revitalised interest in adopting polyhedral modelling as a promising tool for the geometric representation of arbitrarily shaped density distributions. Although earlier examples focused primarily on representing buried ore bodies, later work have seen a wide variety of applications in areas of geodesy, geophysics and dynamical astronomy, considering irregularly shaped celestial bodies in particular [19, 40]. When determining the gravity tensor from a polyhedral source, the literature presents two distinct approaches: a numerical solution and the exact analytical solution. Among examples of the former approach, we find several numerical integration schemes, like the one presented by Talwani and Ewing [41], and the computation of spherical harmonics coefficients for polyhedral sources as presented by Tsoulis et al. [42]. For the latter variety, the list of prominent examples is long. The analytical approach is generally classified as either applied geometric triangulation of the considered polyhedral faces or focusing on determining the gradients of the gravitational potential induced by the polyhedral shape model. In this thesis, we will focus on the latter and specifically the line integral method presented by Tsoulis and Petrović [20, 21, 39], and later implemented by Schuhmacher [43], as the primary gravitational field model for the target small body. Hence, the rest of this section is devoted to presenting the fundamental principles of the polyhedral framework, focusing mainly on constructing a field model and its later application as an approximation to Eqs. (2.31-2.33).

To derive an analytical expression for the gravitational signal, let us first consider a polyhedral solid with a finite number of arbitrarily shaped planar faces defined by interconnecting line segments and vertices at the end of each segment. If we then assume the polyhedron to have a homogeneous mass-density distribution ρ and a finite volume U, we can formulate a set of integral equations representing the gravitational field evaluated at an arbitrary point in space P [39]. Thus, we introduce a new coordinate system with origin ($x_i = 0$) located at point P and a suitable direction of each coordinate axis defined by the basis of unit vectors $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$. As in the work by Tsoulis and Petrović [20, 21, 39], we define the explicit expression for the signal with the following functionals

$$V = G\rho \iiint_U \frac{1}{l} \, dU, \tag{2.34}$$

$$V_{x_i} = G\rho \iiint_U \frac{\partial}{\partial x_i} \left(\frac{1}{l}\right) dU, \quad (i = 1, 2, 3), \tag{2.35}$$

$$V_{x_i x_j} = G\rho \iiint_U \frac{\partial^2}{\partial x_i \partial x_j} \left(\frac{1}{l}\right) dU, \quad (i, j = 1, 2, 3), \tag{2.36}$$

where V is the gravitational potential, V_{x_i} is the attraction, $V_{x_i x_j}$ is the second-order derivative of the gravity tensor and $G = 6.6743{\cdot}10^{-11} m^3 kg^{-1} s^{-2}$ is the universal gravitational constant. In this case, the subscript of V denotes the corresponding partial derivative. Furthermore, we introduce l as the distance function

$$l = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + (x_3 - x_3')^2}, \tag{2.37}$$

which refers to a new coordinate system with origin at $x' = [x_1', x_2', x_3']^T = 0_{3\times1}$ corresponding to the orthogonal projection of P onto the polyhedral face S_p and the basis of orthonormal unit vectors $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ located at point P. Given the symmetric properties of linearly independent derivatives that emerge in the gradiometric tensor $V_{x_i x_j}$, and the validity of Laplace's equation for its diagonal terms, the three integral equations ultimately reduce to nine unknown quantities of interest [39]. A complementary geometric perspective on the polygonal plane S_p and the quantities used for representing the corresponding potential can be seen in Fig. 2.2.

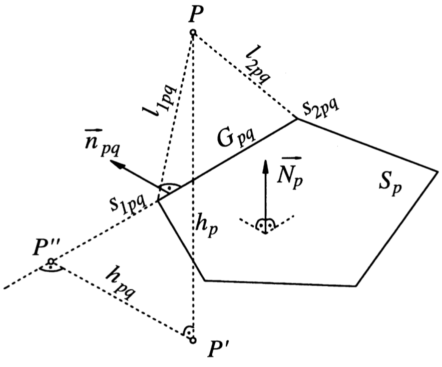


Figure 2.2: Geometric representation of a polyhedral face and the properties expressed in Eqs. (2.45 - 2.47). (Credited to Petrović [20])

The principal idea of the line integral method is to transform the volume integrals in Eqs. (2.34 - 2.36) into a set of closed analytical expressions using the Gaussian divergence theorem while preserving the transcendental functions (logarithms and arctangents) along each line segment on the closed-surface polyhedron S. The transformation consists of two stages. The first step remarks the transition into a set of surface integrals equal to the number of polygonal planes S_p on the closed surface $S = \bigcup_{p=1}^{n} S_p$ using the divergence formula

$$\iiint_U div\, \boldsymbol{F}\, dU = \oiint_S \boldsymbol{F} \cdot \boldsymbol{N}\, dS, \tag{2.38}$$

where $\boldsymbol{N}$ denotes the outer unit normal to S and $\boldsymbol{F}$ a suitable vector field. Although finding three vector fields whose divergence leads to the desired integration represents a problem without a unique solution, a simple example that suffices the divergence property is presented by Pétrovic [20]. The result is the transformation into a set of two-dimensional integral expressions that are algebraically related with the Hessian normal form of the equations of a plane $Ax + By + Cz + D = 0$. For each polygon S_p, the Hessian normal form can be written as

$$\sigma_p h_p = x_1 cos(\boldsymbol{N}_p, \boldsymbol{e}_1) + x_2 cos(\boldsymbol{N}_p, \boldsymbol{e}_2) + x_3 cos(\boldsymbol{N}_p, \boldsymbol{e}_3), \tag{2.39}$$

where the quantity h_p refers to the positive distance between the considered point P and the plane p with a unit normal $\boldsymbol{N}_p$ pointing outside the polyhedral mass. The sign of h_p is then managed by the binary term $\sigma_p = \{-1, 1\}$ which takes a negative value if the unit normal points towards the half-space containing the point P and a positive value otherwise. The equations resulting from the transformation of Eqs. (2.34 - 2.36) are then defined as

$$V = \frac{G\rho}{2} \sum_{p=1}^{n} \sigma_p h_p \iint_{S_p} \frac{1}{l}\, dS, \tag{2.40}$$

$$V_{x_i} = G\rho \sum_{p=1}^{n} cos(\boldsymbol{N}_p, \boldsymbol{e}_i) \iint_{S_p} \frac{1}{l}\, dS, \quad (i = 1, 2, 3), \tag{2.41}$$

$$V_{x_i x_j} = G\rho \sum_{p=1}^{n} cos(\boldsymbol{N}_p, \boldsymbol{e}_i) \iint_{S_p} \frac{\partial}{\partial x_j} \left(\frac{1}{l}\right) dS, \quad (i, j = 1, 2, 3), \tag{2.42}$$

representing the evaluation of each surface plane S_p [39]. The second stage of the line integral approach thus considers the transition from surface integrals to a set of line integral equations along each segment G_{pq} of the polygonal line $G_p = \bigcup_{q=1}^{m} G_{pq}$ in the plane S_p. For this step, we consider the new local coordinate system of the orthogonal projection P' with the three basis terms

$$\boldsymbol{e}_3' = \boldsymbol{N}_p, \quad \boldsymbol{e}_2' = \frac{\boldsymbol{e}_j \times \boldsymbol{N}_p}{|\boldsymbol{e}_j \times \boldsymbol{N}_p|}, \quad \text{and} \quad \boldsymbol{e}_1' = \boldsymbol{e}_2' \times \boldsymbol{e}_3'. \tag{2.43}$$

We can then apply the Gaussian divergence theorem directly in each plane using

$$\iint_{S_p} div\, \boldsymbol{F}\, dS = \oint_{G_p} \boldsymbol{F} \cdot \boldsymbol{n}_p\, dS, \tag{2.44}$$

where $\boldsymbol{n}_p$ refers to the outer unit normal to the closed polygon G_p and $\boldsymbol{F}$ a new suitable vector field. The final formulas are then obtained by evaluating the analytical solution for each resulting line integral equation. In the evaluation step, we introduce a one-dimensional local Cartesian coordinate line along each segment G_{pq} whose origin is located at the orthogonal projection P'' of P' and thus defines the distance to each segment on G_p. Suppose that the point P' falls inside the plane S_p or at any of the points defined on the line G_p. In that case, one can assume that the second transition from surface to line integrals couples with the emergence of singularities. One such case occurs when the position P' implies a zero division, thus resulting in an undefined vector field entering the divergence formula. Tsoulis and Petrović [21] present a method for overcoming these singularities, consequently enabling the line integral formalism to be applicable everywhere in the considered mathematical space, regardless of positions P and P'. The approach leads to an expression for calculating the full gravity tensor at an arbitrary point P, characterised by transcendental expressions whose numerical values are dependent on the exact position of P' on G_p and are necessary for the correct implementation of the analytical formulas. The final expressions are defined as

$$V = \frac{G\rho}{2} \sum_{p=1}^{n} \sigma_p h_p \left[\sum_{q=1}^{m} \sigma_{pq} h_{pq} LN_{pq} + h_p \sum_{q=1}^{m} \sigma_{pq} AN_{pq} + sing_{\mathcal{A}_p} \right], \tag{2.45}$$

$$V_{x_i} = G\rho \sum_{p=1}^{n} cos(\boldsymbol{N}_p, \boldsymbol{e}_i) \left[\sum_{q=1}^{m} \sigma_{pq} h_{pq} LN_{pq} + h_p \sum_{q=1}^{m} \sigma_{pq} AN_{pq} + sing_{\mathcal{A}_p} \right], \tag{2.46}$$

$$\begin{aligned} V_{x_i x_j} = G\rho \sum_{p=1}^{n} cos(\boldsymbol{N}_p, \boldsymbol{e}_i) & \left[\sum_{q=1}^{m} cos(\boldsymbol{n}_{pq}, \boldsymbol{e}_j) LN_{pq} \right. \\ & \left. + \sigma_p cos(\boldsymbol{N}_p, \boldsymbol{e}_j) \sum_{q=1}^{m} \sigma_{pq} AN_{pq} + sing_{\mathcal{B}_{pj}} \right], \end{aligned} \tag{2.47}$$

where $i, j = (1, 2, 3)$, and both the gravitational constant G and the mass density ρ remain from Eqs. (2.34 - 2.36). In Eqs. (2.45 - 2.47), the outer summation acts on the polyhedral faces $p = 1, \ldots, m$ and the inner summation on the corresponding q polyhedral edges defining the particular face. Thus, assuming a closed-surface polyhedron S where each of the polyhedral planes S_p are triangulated, which will be the basis for the mesh representing the target body in this thesis, it holds that $q \in (1, 3)$ and $m = 3$. The value $\boldsymbol{n}_{pq}$ represents the normal to the edge q on face S_p, pointing out from the flat surface. Here, $\sigma_{pq} \in \{-1, 1\}$ depending on whether $\boldsymbol{n}_{pq}$ points at the half-plane that contains P' or not. The value h_{pq} defines the distance between the projected points P' and P''. Furthermore, AN_{pq} and LN_{pq} denote the aforementioned transcendental expressions and are defined as

$$LN_{pq} = ln \frac{s_{2_{pq}} + l_{2_{pq}}}{s_{1_{pq}} + l_{1_{pq}}}, \tag{2.48}$$

$$AN_{pq} = arctan \frac{h_p s_{2_{pq}}}{h_{pq} l_{2_{pq}}} - arctan \frac{h_p s_{1_{pq}}}{h_{pq} l_{1_{pq}}}, \tag{2.49}$$

where both quantities contain the distances $l_{1_{pq}}, l_{2_{pq}}$ and $s_{1_{pq}}, s_{2_{pq}}$, as illustrated in Fig. 2.2. The former distance pair expresses a three-dimensional distance between the computational point P and each set of vertices defining the edge q on face p. Then, assuming that P'' is located at the origin of a one-dimensional local frame on the edge segment pq, the latter distance pair defines the distance between the second projection P'' and the corresponding two vertices. Moreover, depending on the relative position of P'' to the corresponding vertices, the sign of s_i and l_i must be adjusted accordingly [38]. Finally, Eqs. (2.45 - 2.47) also introduces two singularity terms, $sing_{\mathcal{A}_p}$ and $sing_{\mathcal{B}_{pj}}$, that emerge for particular positions of P'' with respect to G_p in the transformation from surface to line integrals resulting in a zero division in Gauss' divergence formula. In detail, the two singularity terms are used to express the specific analytical solution for the limiting values on the line integrals that appear in the partial application of the transformation, specifically for a small circle containing the singularity with a radius tending towards zero. The two terms take the conditional values

$$sing_{\mathcal{A}_p} = \begin{cases} -2\pi h_p & \text{if condition (1) holds,} \\ -\pi & \text{if condition (2) holds,} \\ -\theta h_p & \text{if condition (3) holds,} \\ 0 & \text{if condition (4) holds,,} \end{cases} \tag{2.50}$$

$$sing_{\mathcal{B}_{pj}} = \begin{cases} -2\pi cos(\boldsymbol{N}_p, \boldsymbol{e}_j)\sigma_p & \text{if condition (1) holds,} \\ -\pi cos(\boldsymbol{N}_p, \boldsymbol{e}_j)\sigma_p & \text{if condition (2) holds,} \\ -\theta cos(\boldsymbol{N}_p, \boldsymbol{e}_j)\sigma_p & \text{if condition (3) holds,} \\ 0 & \text{if condition (4) holds,,} \end{cases} \tag{2.51}$$

where the conditions are the following: (1) P'_p lies inside the polygon plane S_p, (2) P'_p is located on edge G_{pq} but not at its corresponding vertices, (3) P'_p is located at one of the vertices corresponding to the edge segment G_{pq}, and (4) when P'_p is located outside the plane of focus S_p. Furthermore, the final expressions, presented in Eqs. (2.45 - 2.47), are independent of the two subsequent coordinate transformations that emerged when applying the Gaussian divergence theorem for each face. This is simply a result of the two summations acting on the same transcendental expressions, LN_{pq} and AN_{pq}, linking to the same polyhedral vertices for each segment pq. However, the critical part for implementing these formulas lies in gaining information about the position of the three different compute points for each face S_p on the closed surface S representing the body of interest [39].

With a brief introduction to the analytical expressions for the potential V, gravitational attraction V_x and the spatial rate of change for the gravitational attraction, $V_{x_i x_j}$, the advantages of the polyhedral method becomes evident. First, by the closed-form expressions in Eqs. (2.45 - 2.47), truncation errors can generally be avoided during computations, consequently relating most errors to the modelling and bias of estimation for the shape model and density distribution. Therefore, the method's accuracy is mainly related to the resolution of the considered polyhedron, that is, the number of defined faces and vertices. Second, the polyhedron's precession is independent of the distance to the considered body, which is uniformly effective even in proximity to the body. Third, with knowledge of the specific positions of the spacecraft, i.e. computing point P, and each of the defined faces on the shape model, the polyhedral method allows for a quick determination of whether the position has entered the closed surface or not, enabling a simple collision detection algorithm. In addition, since the polyhedron is based on a general shape model, it is well suited for practical missions where the body's shape can be obtained and refined with earth-based observations, onboard optical measurements and proximity flight data. Now, with the expression for the gravitational attraction at hand, along with relevant reference frames and an understanding of the dynamics near rotating bodies, we can formulate the specific equations of motion that will be used in this thesis.

2.4 Equations of Motion

In this section, we will focus on defining a set of differential equations that can be used to express the motion of a spacecraft in proximity to the considered body. We will first specify a suitable state representation and then introduce the corresponding control input.

2.4.1 State-Space Model

To formulate the equations of motion of a spacecraft, one should first introduce a suitable state-space representation to express its position and motion relative to the body of interest. Although numerous state representations are available with varying benefits, ranging from computational performance to singularity-avoidance and more, we limit the study to Cartesian elements for a simple implementation and avoiding singularities altogether. Let $\boldsymbol{r} = [x, y, z]^T$ and $\dot{\boldsymbol{r}} = [v_x, v_y, v_z]^T$ represent the spacecraft's position and velocity, respectively. The state vector at time t can then be defined as

$$\boldsymbol{x}(t) = \begin{bmatrix} \boldsymbol{r}(t) \\ \boldsymbol{v}(t) \end{bmatrix}. \tag{2.52}$$

The dynamical system constituting the mathematical laws explaining the evolution of the state vector over time can subsequently be defined by the first-order differential system

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{\boldsymbol{r}}(t) \\ \ddot{\boldsymbol{r}}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}(t) \\ \boldsymbol{a}(\boldsymbol{r}(t)) \end{bmatrix}, \tag{2.53}$$

where $\boldsymbol{v}(t)$ is the inertial velocity of the spacecraft and $\boldsymbol{a}(\boldsymbol{r}(t))$ the gravitational attraction acting on the spacecraft by the small body. In detail, we define the acceleration as the partial derivative of the gravitational potential in each axis of the body-fixed frame, where $\boldsymbol{a}(\boldsymbol{r}(t)) = [{}^{\mathcal{F}}V_x, {}^{\mathcal{F}}V_y, {}^{\mathcal{F}}V_z]^T$ as provided by the analytical expression for the polyhedral model in Eq. (2.46) in Euclidean space. Recall from Section 2.2 that irregular gravitational fields and body-relative geometric constraints are most realistically expressed in the small-body fixed frame. Hence, the polyhedral model will be evaluated in $\mathcal{F}$ to account for the body's side-real rotation about its principal axis of inertia. To avoid the computational costs of rotating each vertex defined on the polyhedron, we can simplify the transformation by considering the corresponding quaternion rotation for a negative rotation angle $-\xi$. The equations of motion presented in Eq. (2.53) can then be more accurately defined as

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \boldsymbol{v}(t) \\ \boldsymbol{a}({}^{\mathcal{F}}\mathcal{Q}_{\mathcal{I}}({}^{\mathcal{I}}\hat{\boldsymbol{r}}(t), \omega, t)) \end{bmatrix}. \tag{2.54}$$

2.4.2 Impulsive Control

Optimising a trajectory based on an impulsive model is the traditional approach for simulating spacecraft manoeuvres [44]. Impulsive manoeuvres are those in which brief firings of onboard rockets result in an instantaneous change in the magnitude and direction of the corresponding velocity vector. Employing an impulsive scheme is, for that reason, an idealisation where one can avoid having to solve the equations of motion with the rocket thrust included in the model. This form of control is often satisfactory whenever the spacecraft's position only changes marginally during the period in which the rockets fire. In practice, this may be a feasible assumption whenever the burn time is short compared to the coasting time of the spacecraft [33]. In mathematical terms, this is equivalent to considering the variation of control inputs for a system

$$\dot{\boldsymbol{x}}(t) = A(t)\boldsymbol{x}(t) + B(t)\boldsymbol{u}(t) \tag{2.55}$$

to be zero at the time of each input, i.e. $\dot{\boldsymbol{u}}(t_i^m) = 0$, and the manoeuvre to be a change in velocity increments Δv during a burn time of $\Delta t = 0$. The perturbed velocity can be represented as a change in either magnitude, referred to as a pumping manoeuvre, in direction, known as a cranking manoeuvre, or both. As such, thrust arcs are simulated through isolated, singular events, as illustrated in Fig. 2.3, which enables a segmentation of the continuous problem.

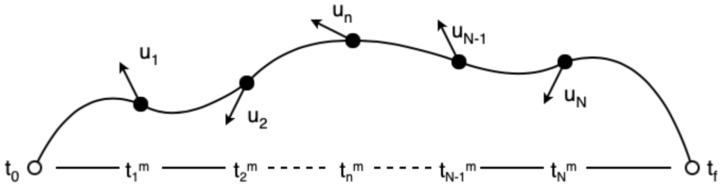


Figure 2.3: Impulsive manoeuvres as a segmentation scheme.

In this thesis, we employ a sequence of impulsive manoeuvres on the form $\boldsymbol{u}(t) = \Gamma(t)\hat{u}(t)$, where each manoeuvre is applied at discrete times $\mathcal{T}_C = \{t_i : t_i < t_{i+1}, i = 0, 1, \ldots, I\}$. The equations of motion presented in Eq. (2.54) can therefore be defined as

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \boldsymbol{v}(t) \\ \boldsymbol{a}({}^{\mathcal{F}}\mathcal{Q}_{\mathcal{I}}({}^{\boldsymbol{\mathcal{I}}}\hat{\boldsymbol{r}}(t), \omega, t)) + \boldsymbol{u}_{t|t\in\mathcal{T}_C} \end{bmatrix}. \tag{2.56}$$

Chapter 3

Problem Definition

In this chapter, we will focus on formulating a suitable optimisation problem for the considered small-body mission. In order to ensure continuity, the first section is devoted to the connection between the main objective of the thesis and mathematical optimisation. The second section will briefly introduce the traditional formulation of a spacecraft trajectory optimisation problem derived from optimal control theory. The subsequent sections will then individually define each part of the considered optimisation problem, leading to the complete formulation presented in Section 3.6. As we will see, following a discrete approximation of the space around the body naturally enables a reformulation of the control problem into a more traditional nonlinear program suited for direct optimisation strategies.

3.1 Mission Objective

As stated in Section 1.2, the main objective of this thesis is to formulate an optimisation problem that results in a set of trajectories corresponding to a spacecraft swarm that maximises the jointly measured gravitational signal around some irregularly shaped, rotating body. With the introduction of impulsive manoeuvres and a state-space model for the governing system dynamics, the mission objective transforms into the determination of a control sequence that optimises the transition of states according to some function representing the measured gravitational signal and a set of constraints maintaining feasibility, such as the avoidance of collisions. To formulate a problem solvable with global optimisation strategies, we emphasise the prominent connection between the considered mission and a conventional trajectory optimisation problem, thus enabling a mathematical definition based on the fundamental principles of optimal control.

3.2 Optimal Control Theory

Optimal control theory is a relatively young yet mature branch of mathematics which mainly concerns controlling dynamical systems. The theory is rooted in the calculus of variations and has several advantages; it provides systematic approaches for designing optimal control inputs for problems that are generally hard to solve using ad-hoc techniques and can efficiently reduce redundancy by identifying the best solutions given a set of requirements and performance measurements [45]. Hence, a typical application includes planning complex spacecraft missions that deal with similar problem formulations. In general, there are many ways to formulate trajectory problems; however, in this thesis, we will limit the focus to single-phase continuous-time trajectory optimisation problems with continuous dynamics throughout the considered time frame. To accustom the reader to the theoretical formulation of these problems, we will briefly describe each of its components and later relate these to the considered spacecraft trajectory optimisation problem.

The objective function is, in general, represented by two C^1 continuously differentiable functions: a terminal cost $\Phi(t_f, \boldsymbol{x_f})$ penalising the deviation from any desired final states and an integral part defining the running cost along the continuous trajectory. When defining an objective composed of both terms, we refer to it as being in *Bolza form*. Similarly, we refer to an objective function with only a terminal cost on $Mayor form$ and an integral term on *Lagrange form*. On *Bolza* form, we define the objective as

$$\Phi(t_f, \boldsymbol{x_f}) + \int_{t_0}^{t_f} f_0(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\, dt. \tag{3.1}$$

Generally, most optimal control problems are defined subject to several constraint functions. One such constraint is the system dynamics, which is typically represented by a set of nonlinear state space equations that describe the state transition over time

$$\dot{\boldsymbol{x}} = f(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) = A(t)\boldsymbol{x}(t) + B(t)\boldsymbol{u}(t). \tag{3.2}$$

In this case, we assume the function f to be continuous with continuous partial derivatives and refer to it as a vector field. The control, however, is assumed to be a piecewise continuous function with a domain $\mathcal{U} \subset \mathbb{R}^m$. In spacecraft trajectory optimisation, we often refer to the system dynamics as the equations of motion and define the equations based on the natural and control forces acting on the spacecraft in motion. Next, to enforce restrictions along the trajectory, such as keeping the spacecraft close to the body, we introduce path constraints defined as

$$h(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \leq 0. \tag{3.3}$$

Another form of constraint is the so-called boundary constraint that enforces the initial and final states to satisfy $\boldsymbol{x}(t_0) \in \mathcal{S}_i$ and $\boldsymbol{x}(t_f) \in \mathcal{S}_f$. An example could be fixing the initial positions of the trajectory to represent a deployment from a mothership. These boundaries are typically represented by smooth manifolds

$$\mathcal{S} = \{\boldsymbol{x} \in \mathbb{R}^n :\ g_k = 0,\ k = 1, \ldots, p\}, \tag{3.4}$$

with linearly dependent gradients ∇g_k. The last considered constraints are the box constraints, which limit each decision variable's domain, such as an interval in magnitudes of thrust that the spacecraft can produce. Finally, the goal of the control problem is to determine the decision variables that minimise the defined total cost. In spacecraft trajectory optimisation, these variables often include the state $\boldsymbol{x}(t)$, control $\boldsymbol{u}(t)$ and time-frame (t_0, t_f). The final problem is formulated as

$$\begin{aligned} \min_{\boldsymbol{x}, \boldsymbol{u}, t_0, t_f} \quad & J(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \Phi(t_f, \boldsymbol{x_f}) + \int_{t_0}^{t_f} f_0(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\, dt \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = f(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \\ & h(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \leq 0, \\ & \boldsymbol{x}(t_0) \in \mathcal{S}_i,\ \boldsymbol{x}(t_f) \in \mathcal{S}_f, \\ & \boldsymbol{u}(t) \in \mathcal{U}(\boldsymbol{x}, t). \end{aligned} \tag{3.5}$$

Furthermore, it should be noted that the set of admissible controls, $\mathcal{U}(\boldsymbol{x}, t)$, is presented here with a state dependency, which is generally uncommon for optimal control problems but relevant for a spacecraft trajectory problem given how the use of manoeuvres are reasonably related to the spacecraft's position and motion. Another characteristic of Eqs. (3.5) is the highly complicated nature of its dynamic constraints, which tends to induce several local optima, making more straightforward numerical methods unsuitable and requiring more robust and accurate global optimisation techniques to cover the search space efficiently [46]. As a consequence, it is generally of interest to further simplify the problem.

3.3 Objective Function

To model the measured gravitational signal, we introduce an objective function representing the spacecraft's spatial coverage throughout its trajectory. Measuring the variations in gravitational acceleration is a continuous process most accurately expressed by integrating a continuous-time function. One approach is to employ spherical harmonics equations to expand a harmonic function within the volume confining the body's gravitational field. Nevertheless, given the convergence limitations of spherical harmonics inside the Brillouin sphere, these equations would likely result in an invalid representation of near-surface coverage. Moreover, the resulting algorithm is also prone to be computationally expensive since the problem formulation would depend on two separate integration schemes.

Another strategy involves considering a discrete approach, thereby coupling the accuracy of coverage with the detail of discretisation. However, defining some discretised volume consistent with the gravitational field for a highly irregularly shaped body is challenging, especially during the early stages of a mission where the geodesy of the body remains unexplored. In this thesis, we consider a mission scenario characterised by initial approaches and non-stable trajectories, where the available shape model is derived from optical measurements, resulting in a polyhedral gravity model. To avoid complicated discretisation procedures, we model the measurements of spatial changes in gravitational acceleration using an indirect representation of coverage in the body-fixed frame. In detail, we assume an encapsulating discrete spherical structure where each cell represents a specific domain of approximately homogeneous gravitational signal. By determining the total number of visited regions on the structure, which is analogous to feasible measurements gathered by the spacecraft, we can formulate a performance measure that is representative of the coverage of the gravitational signal. To define a suitable structure for the simulation, we will first compare existing literature on spatial analysis and common classes of geodesic grids, focusing on three spatial dimensions.

3.3.1 Discrete Spherical Grids

The applications of computational grids are many and include, for instance, constructing geodesic grids for meteorological systems and atmospheric modelling [47], geophysical modelling of magnetic fields [48], and simulations in both solid mechanics and fluid mechanics. When modelling spatial systems, the literature presents two conventional strategies for addressing the geographic space: to treat the experimental space as a whole or divide it into various cells for more precise computations. The most commonly applied numerical techniques for simulating dynamical systems include partial differential equations (PDE), lattice Boltzmann method (LBM) and cell automata (CA). For instance, LBM is a discrete alternative often applied to simulate propagation and collision processes over discrete lattice grids of fictive particles in fluid systems [49]. In contrast, finite element (FEM), finite volume (FVM), and finite difference methods (FDM) are more commonly applied when a discretised grid represents the computational domain. We generally refer to the gridding of a spherical structure as the subdivision of its domain into cells of polygonal volumes with either regular or irregular shapes. In three dimensions, these cells are typically defined as tetrahedrons, pyramids, tri-prisms or hexahedrons (see Fig. 3.1), where each unit could be associated with several variables, such as pressure and temperature [50].

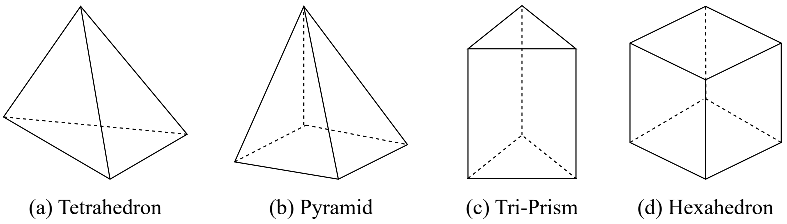

(a) Tetrahedron (b) Pyramid (c) Tri-Prism (d) Hexahedron

Figure 3.1: General shapes on cells in a 3D computational grid.

Furthermore, discrete grids are generally divided into four categories: structured, unstructured, hybrid and Chimera grids. Each category offers a unique set of advantages contributing to the efficacy of grid-based methods in various applications. To motivate the choice of a suitable structure for simulating coverage, we will begin by presenting a concise summary of each category.

Structured Grids: A structured grid is defined by an ordered arrangement of its cells in both vertical and horizontal directions. It then follows naturally that each cell will have a fixed number of first-order neighbours tangent to its edges and vertices, which enables a relatively easy implementation. In applications that focus on solving volumetric problems, structured grids are often designed with a hexahedral cell structure since it naturally enables an embedded tensor product structure, a more considerable tolerance for anisotropy - where either the cell or the model has different physical properties dependent on directions, which then affects numerical stability and accuracy of the simulation - and generally low numerical stiffness [50]. Provided their benefits, various structured grids can be employed depending on the specific computational tasks. For instance, cubed spheres (see Fig. 3.2) are typically applied to model solar wind problems, whereas longitude-latitude grids (LLG) are generally more favoured in global crust modelling and for simulating wave velocities in Earth's mantle [51].

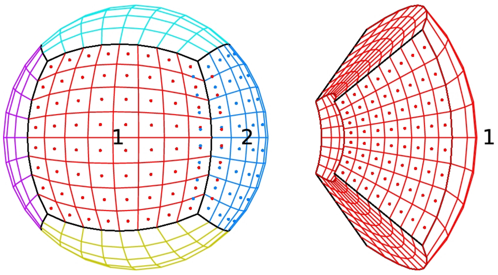


Figure 3.2: A collocated cubed sphere grid [51].

Unstructured Grids: An unstructured grid is defined by arbitrarily shaped cells, where each node and cell can be placed freely to more accurately fit the boundary of the computational domain. However, the flexibility and accuracy often come with the drawback of defining more cells and nodes in random data structures, resulting in a greater demand for computational resources. In three dimensions, unstructured grids are usually constructed with either tetrahedrons, tri-prisms or pyramids as the general cell shape. The resulting grid is therefore well suited for modelling complex geometrical domains with little distortions and often applied in problems such as 3D object modelling, convection of Earth's mantle [52] and electric field modelling [53].

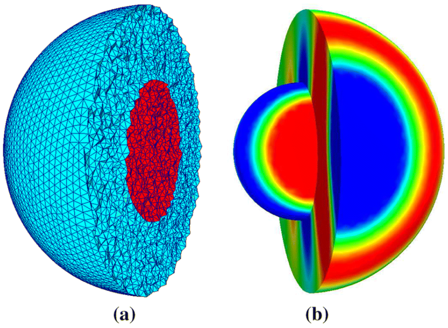


Figure 3.3: Scattering of a plane wave modeled by a coated spherical object with view of (a): an unstructured grid, and (b): electric scattered field [53].

Hybrid Grids: In contrast to the previously mentioned categories, Hybrid grids decompose the computational domain into subregions of independent and adjacent grid structures. The combinations of grid structures are generally classified as either structured-unstructured mixed grids or unstructured mixed grids [50]. As such, three-dimensional hybrid grids are often applied in contexts where properties of the computational domain may differ greatly from one subregion to another. One example of a hybrid grid is the Spheroid-Degenerate Octree Grid (SDOG), first proposed by Yu and Wu in 2009 [54]. With a near-volume preserving property, this grid is highly suitable for ensuring consistent spatial resolutions when projecting the three-dimensional model on a two-dimensional plane for further analysis. However, when applying different grid structures at different subregions, it is equally important to adopt special care regarding data matching across each boundary to account for varying requirements on computational techniques.

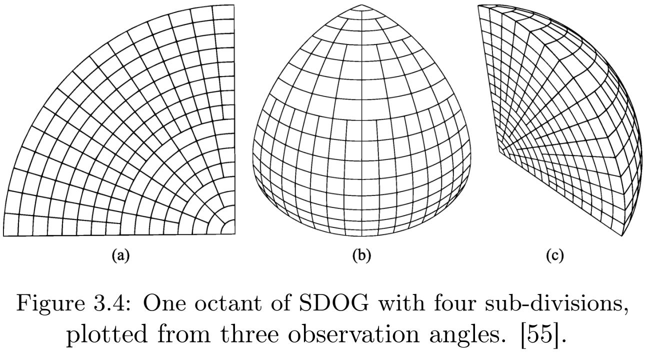


Figure 3.4: One octant of SDOG with four sub-divisions, plotted from three observation angles. [55].

Chimera Grids: Finally, Chimera grids comprise a set of overlapping structured grids called overset grids. Each subgrid is independently adapted for the domain of interest, resulting in an accurate grid system readily available for various numerical techniques. With such qualities, Chimera grids are typically applied in structural optimization [50]. Although this grid offers excellent accuracy and geometrical flexibility for complex domains, interpolating data that stretches over multiple grid structures often becomes challenging since it requires the conservation of physical variables between different sub-regions. One example of an overset grid is the Yin-Yang-Zhong grid, consisting of two overlapping latitude-longitude grids for 90 degrees in latitude and 270 degrees in longitude, a so-called Yin-Yang grid, and some inner decoupled grid. However, provided the structure of the Yin-Yang grid, this method often causes cells to overlap near their borders, resulting in redundancy and ambiguity when assigning data to each of these cells.

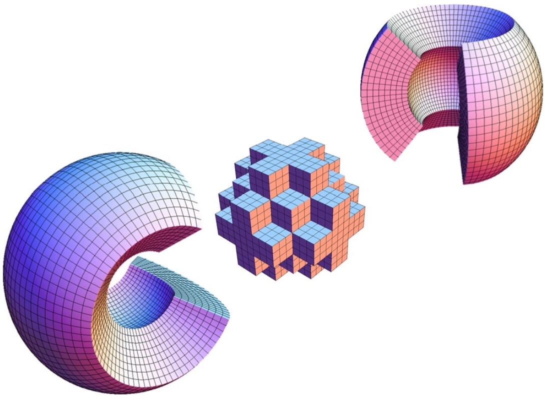
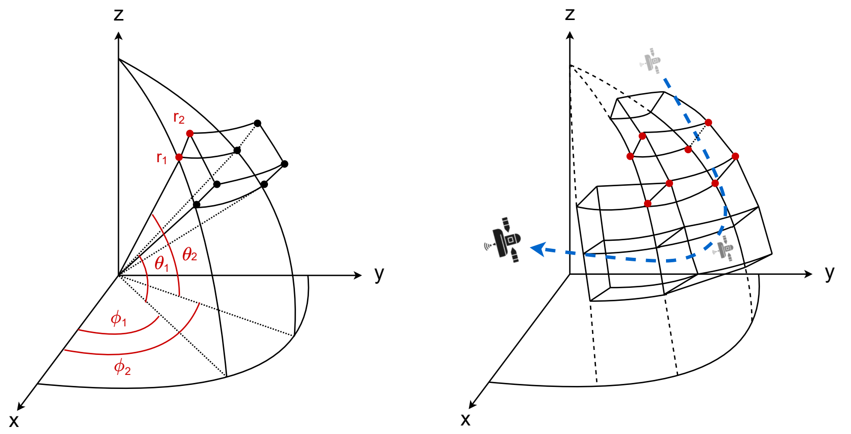


Figure 3.5: Yin-Yang-Zhong overset grid for a spherical shell with a central cuboid [56].

With a brief introduction to each of the four common categories of computational grids, we can now identify the most suitable structure for the simulation of coverage. First, recall that the purpose of the discretization is not to model the gravity field directly but to divide some bounded spherical domain into cells representing regions where the rate of change in gravitational acceleration is small enough to be accurately measured by a visiting spacecraft. Since the gravitational acceleration varies with an inverse relation of the distance to the body, the cells should follow a similar ordered structure in size. Consequently, there is no need for defining subregions with varying structures, as used in hybrid grids. Moreover, following the order of sizes, unstructured grids would not provide any improved accuracy over the more computationally efficient structured grids. Finally, since a polyhedral gravity model already represents the body of interest and is therefore decoupled from the discretization of coverage, it also follows that a multi-structured Chimera grid is unnecessary for this use case. Although both cubed spheres and longitude-latitude grids are suitable for the particular application, we will proceed with the latter since it enables a computationally efficient and easy-to-manipulate data structure with smooth transitions for trajectories crossing multiple cells of varying sizes.

3.3.2 Defining a Longitude-Latitude Grid

We first introduce two new spherical boundaries depicting a non-convex computational domain to define the longitude-latitude grid structure around the body of interest. The inner bounding sphere, $\mathcal{S}_I$, represents a region with greater risk for colliding with the body and is defined by a safety radius $r_\mathcal{I}$. The radius $r_\mathcal{I}$ can be chosen arbitrarily since the grid is decoupled from the polyhedral model. For smaller bodies, a feasible assumption is to consider an inner radius $r_\mathcal{I}$ comparable with the exterior Brillouin sphere since travelling within this region is likely to be risky and should not result in any gain for the objective, even though a near ground-truth signal likely characterises the region. For larger non-spherical bodies, one can instead consider an inner ellipsoidal boundary to reduce the cut-off space. Next, the outer sphere $\mathcal{S}_O$ is defined with radius $r_\mathcal{O} > r_\mathcal{I}$ and represents the space of interest for carrying out measurements. The non-convex region constituting the grid, i.e. the measurable space, is then defined by the manifold on $\mathcal{S}_f = \{(r, \theta, \varphi) \mid r \in [r_\mathcal{I}, r_\mathcal{O}], \theta \in [-\frac{\pi}{2}, \frac{\pi}{2}], \varphi \in [-\pi, \pi]\}$ representing a spherical shell between the two concentric spheres $\mathcal{S}_I$ and $\mathcal{S}_O$. A discrete approximation of $\mathcal{S}_f$ in terms of a longitude-latitude grid $\mathcal{G}$ will then comprise several evenly spaced points $\boldsymbol{p} \in \mathcal{G}$ such that $\mathcal{G} \subset \mathcal{S}_f$. Utilising the tensor structure, we can associate each point by its position, expressed in spherical coordinates, and a numerical value which we refer to as its weight, such that $\boldsymbol{p} = (r_p, \theta_p, \phi_p, w_p)$. As a result, each set of eight adjacent points on $\mathcal{G}$ defines a curvilinear hexahedron, or tesseroid, that is bounded by six differential surfaces as depicted in Fig. 3.6a.



(a) A tesseroid defined on a LLG. (b) Trajectory crossing multiple tesseroids.

Figure 3.6: Isolated components defining a LLG.

In order to ensure stability in the simulation of coverage and produce accurate results, the spacing between the points on the grid must be defined such that each tesseroid satisfies the Courant-Friedrichs-Lewy (CFL) condition, which states that the total numerical space of dependence should include the corresponding physical domain of dependence [57]. In other words, we must ensure that a spacecraft can not cross multiple tesseroids within one local time step. This is done by considering a fixed measurement period corresponding to a local time step Δt and a length interval ΔL defining one of the edges for the corresponding tesseroid. By introducing a courant number C_{CFL}, the condition implies an upper limit on the local time step such that

$$\Delta t \;\leq\; C_{CFL}\,\frac{\Delta L}{{}^{\mathcal{I}}v_{max}}. \tag{3.6}$$

where ${}^{\mathcal{I}}v_{max}$ is the local maximal inertial velocity for a spacecraft travelling inside a tesseroid. However, considering only local quantities does not guarantee that the condition remains satisfied. In fact, for this application, we will consider trajectories that originate from somewhere else on the physical domain but still reach the current tesseroid in time Δt. In order to ensure that the condition is satisfied for all trajectories, we will instead consider a global fixed time step and an upper limit on inertial velocity, which is likely to occur along the contour of the inner bounding sphere where the signal is expected to be maximal. Assuming that the courant number equals one, the condition results in the maximal size for the innermost set of tesseroids and a linear scaling of its outer regions.

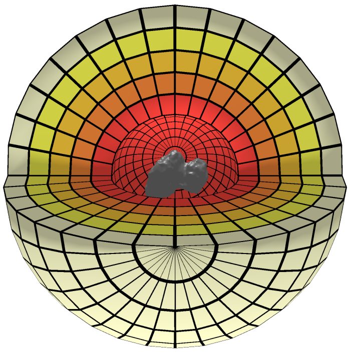


Figure 3.7: A LLG around the comet 67P/Churyumov-Gerasimenko, with one quadrant removed for visibility.

3.3.3 Computing Coverage

With the requirement of using a global fixed time step, we evaluate the coverage of the grid for a given trajectory sampled on the set $\mathcal{T}_T = \{t_0 \leq t \leq t_f : t_{n+1} - t_n = \Delta t,\ n = 0, \ldots, f,\ t_0 \geq 0\}$, ensuring proportionality between the discretisation and the grid spacing. With appropriately chosen values on Δt and ${}^{(\mathcal{I})}v_{max}$, the coverage, V_c, is defined as the weighted ratio of grid points visited by the spacecraft along its trajectory. First, we identify the set of points, $P_{\mathcal{G}} \subset \mathcal{G}$, that minimises the distance to each spacecraft position on the set P_T defined by the trajectory, and thus represents visited regions. For simplicity, we assume that visiting any point on the grid once represents measuring the gravitational signal in that region. The scoring will, therefore, not acknowledge any revisits to specific grid points. Next, in order to ensure that each point represents a signal strength inverse to the radial distance to the body, each point on the grid $p \in \mathcal{G}$ is assigned a weight defined by its radial component, $w_p = 1/r_p$. To avoid ruggedness in the objective, we will instead consider the corresponding normalised weights

$$\hat{w}_p = \frac{w_p}{|w|} = \frac{w_p}{\sum_{p \in \mathcal{G}} w_p}, \quad \text{where } \hat{w}_p \in (0,1). \tag{3.7}$$

The coverage score is then computed as the sum of weights associated with the grid points visited by the spacecraft, that is

$$V_c = \sum_{p \in P_{\mathcal{G}}} \hat{w}_p, \quad V_c \in [0, 1]. \tag{3.8}$$

In addition, since the spherical tensor grid is used to represents measurements of the gravitational field, it should also be consistent with the body's sidereal rotation. However, to avoid the computational cost of rotating the points defined on the grid, the coverage will instead consider evaluating the trajectory adjusted for the body-fixed reference frame (see Eq. (2.24)).

3.4 Path Constraints

In trajectory optimisation, path constraints represent essential conditions that the spacecraft must satisfy throughout its mission duration to ensure compliance with physical, operational, and safety requirements, such as avoiding collisions or meeting specific mission objectives. These constraints are therefore pivotal in guiding the optimisation process to find feasible and efficient trajectories. We will now introduce the two path constraints considered in this thesis, one relating to the coverage and the other to collisions with the body.

3.4.1 Bounded State Space

In this study, we have considered the coverage measurements to be strictly bounded by a longitude-latitude grid. A representative illustration of the shell-like structure can be seen in Fig. 3.7. To maximise the coverage, it then follows that the domain of state variables related to the position should coincide with this region. In practice, this would be equivalent to enforcing a path constraint restricting the radial component of a spacecraft's position expressed in spherical coordinates throughout the entire mission duration:

$$r_{\mathcal{I}} \leq r(t) \leq r_{\mathcal{O}}, \quad \forall t \in [t_0, t_f]. \tag{3.9}$$

However, as an effect of considering a system with particularly complex dynamics and a celestial body with a relatively small mass, it follows by intuition that some solutions could become hyperbolic and highly non-periodic, resulting in positions defined outside $\mathcal{S}_f$ depending on its extent. To avoid over-constraining the problem, we employ penalty methods that relax the constraint functions, transforming the objective into a sum of the previous one and several penalty functions accounting for constraint violations. Since the penalty functions only penalise the objective at an active violation, we refer to the new objective as a cost function. Consequently, we can avoid discarding solutions with good coverage and only slight violations that could potentially guide the search process in finding efficient and mission-compliant solutions. To define suitable penalty functions, we must carefully consider their balancing effect. For instance, too-weak penalties will likely benefit unfeasible solutions, whereas too-strong penalties may prevent finding good solutions. To account for these properties, we will divide the path constraint into two separate functions, one considering positions near the body, where collisions are likely, and one considering those outside the region of interest where the rate of change in gravitational acceleration is minimal.

The former function relates the risk of colliding with the body by defining a penalty based on the normalised squared distance from the most critical position to the safety radius, excluding positions precisely on the safety radius as seen in its range depicted in Fig. 3.8a. The risk penalty is defined as

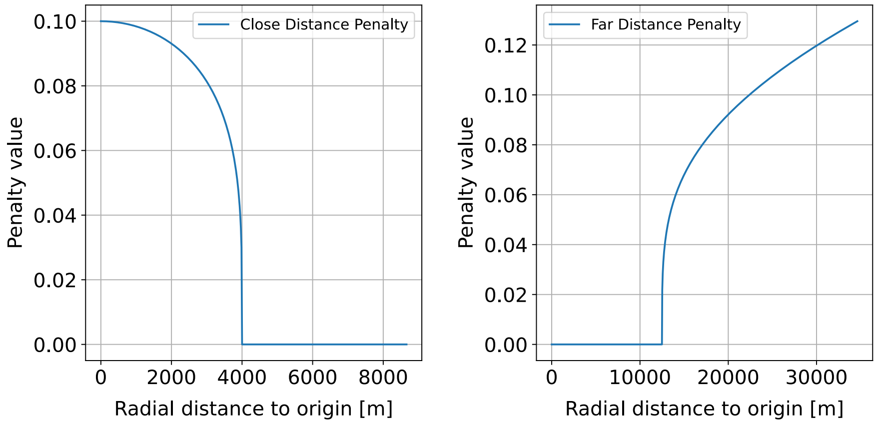
$$P_{\mathcal{I}} = \left[\frac{|r^2_{critical} - r^2_{\mathcal{I}}|}{r^2_{\mathcal{I}}} \right]^{\frac{1}{4}} . \tag{3.10}$$

For positions defined outside the outer bounding sphere, the cost function is penalised by the average squared distance from $r_{\mathcal{O}}$. In contrast to the other components of the cost function, the relevance penalty can generate penalties larger than one for trajectories deviating too far from the grid, thus enabling quick identification of greatly inefficient trajectories during the optimisation process. However, to avoid omitting trajectories that only momentarily leave the region of interest, the penalty assumes a smoother shape near the grid (see Fig. 3.8b). The relevance penalty function, $P_{\mathcal{O}}$, is defined as

$$P_{\mathcal{O}} = \left[\frac{1}{r^2_{\mathcal{O}}} \frac{1}{||P_{T_{\{r>r_{\mathcal{O}}\}}}||} \sum_{r \in P_T} \mathbb{1}_{r>r_{\mathcal{O}}} (r^2 - r^2_{\mathcal{O}}) \right]^{\frac{1}{4}} . \tag{3.11}$$

To avoid ruggedness in the cost function, we introduce lagrangian multipliers to secularize the objectives by proper weighting. We will hereafter refer to these multipliers as penalty scaling factors. Finally, we define the complete cost function as

$$J(\cdot) = -V_c + \lambda_{\mathcal{I}} P_{\mathcal{I}} + \lambda_{\mathcal{O}} P_{\mathcal{O}} . \tag{3.12}$$



(a) Close distance penalty.

(b) Far distance penalty.

Figure 3.8: Two plots depicting the weighted range of the two considered functions penalising risky and distant trajectories.

3.4.2 Collision Avoidance

The second path constraint ensures the safety of the trajectory by excluding collisions with the body. As such, we define a constraint function with a binary output taking a value of zero when there is a presence of a collision between the spacecraft and the celestial body at some point along the trajectory and one otherwise:

$$\mathcal{C}(r_x(t), r_y(t), r_z(t)) = \begin{cases} 0, & \text{when collision detected,} \\ 1, & \text{otherwise,} \end{cases} \qquad \forall t \in [t_0, t_f]. \tag{3.13}$$

3.5 Boundary Constraints

By introducing boundary values for the state transition, the equations of motion in Eq. (2.56) can effectively be solved as a boundary value problem, resulting in a deterministic trajectory. In this case, we will assume a fixed initial state to simulate a deployment from a mothership and a free final state to focus the optimisation on maximising coverage more freely. In detail, we will assume a given initial position and an optimised initial velocity to avoid collisions during swarm deployment, thus allowing the spacecraft to proceed in different directions.

3.6 Problem Formulation

With the definitions presented in Sections 3.3 to 3.5, the optimal control problem of maximising the measured gravitational signal can be simplified to solving the following continuous minimisation problem

$$\begin{aligned}
\min_{v_0, \hat{v}_0, \Gamma_0, \ldots, \Gamma_I, \hat{u}_0, \ldots, \hat{u}_I} \quad & -V_c + \lambda_{\mathcal{O}} P_{\mathcal{O}} + \lambda_{\mathcal{I}} P_{\mathcal{I}} \\
\text{s.t.} \quad & \mathcal{C}(r_x(t), r_y(t), r_z(t)) = 1, && t \in [t_0, t_f] \\
& \Gamma_{min} \leq \Gamma_i \leq \Gamma_{max}, && i = 0, \ldots, I \\
& ||\hat{u}(t_i)|| = 1, && t_i \in \mathcal{T}_C \\
& ||\hat{v}(t_0)|| = 1, \\
& v_{min} \leq v_0 \leq v_{max}, \\
& \boldsymbol{r}_0 \in \mathcal{S}_f, \\
& \boldsymbol{r}_f, \boldsymbol{v}_f \in \mathbb{R}^3 \quad free.
\end{aligned} \tag{3.14}$$

Chapter 4

Optimisation

In previous chapters, we have focused on thoroughly introducing the considered problem through a concise discussion on related spacecraft dynamics and problem characteristics. The outcome was a definition of a suitable trajectory optimisation problem that, when solved correctly, results in an optimised initial state and control sequence for maximising coverage. In this chapter, we will focus on deriving a sufficient optimisation strategy for solving the proposed problem.

4.1 Trajectory Optimisation

In Section 3.2, we presented the fundamental properties associated with a traditional trajectory optimisation problem. In the subsequent sections, each characteristic was then related to the problem definition to derive a suitable formulation for the considered problem. The next step is to develop a strategy for solving the suggested problem. Since trajectory optimisation is known for its extensive research and references, we focus the analysis on concepts sufficient to acquaint the reader with the field and motivate the choices for selecting an appropriate solution method. We begin by making a distinction between the terms *approaches*, *methods* and *techniques*, and design the introduction accordingly.

4.1.1 Numerical and Analytical Approaches

Although trajectory optimisation involves many solution approaches, they are generally of either an analytical or numerical nature, simplifying the taxonomy into a binary classification. In the former case, the goal is to define an exact solution using purely analytical or semi-analytical strategies employing various ad-hoc techniques and intuitive formulations. Common applications include low-thrust orbit raising and Hohmann orbit transfers. However, analytical solutions generally become unachievable when the problem is characterised by more complex dynamics and difficult constraints, resulting in challenging domains and problem formulations. Another strategy is to employ a numerical approach, which allows for determining efficient solutions by discrete approximations. Thus, given the particularly complex nature of the problem considered in this thesis, we will focus on the latter.

4.1.2 Direct and Indirect Methods

Like the approaches, most numerical methods employ one of two strategies. In the former, known as the indirect method, the problem considers the dualised form of equations, including state and co-state variables. Although these methods benefit from exceptional accuracy and reliable error estimates, they introduce several challenges. First, as part of the dualisation, the problem requires analytical expressions of the necessary and sufficient optimality conditions. These include formulating the Hamiltonian, co-state differential equations and transversality conditions, which are seldom trivial for black box models [45]. Another challenge includes managing the increasing problem size following the introduction and discretisation of co-state variables. However, the major disadvantage is generally related to robustness. In detail, to solve the new dynamic equations, the user must provide a guess on the initial co-state variables, which is often difficult due to their non-physical nature. In fact, even with good initial guesses, the equations still risk becoming ill-conditioned.

In contrast, direct methods attempt to solve a discrete approximation of the control problem through transcription, parametrising its state and control vectors. By integrating the dynamics step-wise, one can obtain a set of nonlinear constraint equations that must be satisfied by the parameters representing state and control values over time. Thus effectively converting the original problem into a traditional nonlinear programming problem suitable for a new domain of techniques. As such, direct methods allow for greater flexibility in finding efficient solutions without requiring analytical derivations of optimality conditions, but at the cost of losing verifiability unless other optimality criteria are introduced. The method is generally robust since it only requires initialisation on the physical state variables, which are easier to guess than co-state variables. Besides their ease of implementation, direct methods generally reduce dimensionality and retain a larger domain of convergence resulting from the globalised strategy for minimising the objective [25, 44].

From Sections 3.3.3 and 3.4, it should be noted that both the cost and constraint functions assume a discrete approximation of the candidate trajectory. Thus, it follows naturally to employ a direct method for determining the decision variables that maximise coverage and minimise constraint violations. Furthermore, considering a direct method is also favourable given the use of a polyhedral force model, where deriving analytical expressions for the derivatives of the system dynamics is demanding. Finally, when further expanding the problem to consider a swarm scenario, it is essential to minimise the dimensionality to avoid potential combinatorial challenges. Therefore, it is wise to avoid introducing a dual formulation as well.

4.1.3 Numerical Techniques

In this section, we will briefly introduce several traditional numerical techniques for solving the suggested trajectory optimisation problem using direct methods. The first technique is differential inclusions, which use inequality constraints on the state derivatives to ensure that the equations of motion are satisfied at every discrete time. These inequality constraints can be obtained by substituting the control vector in the dynamic system with its defined boundary values. The main benefit of the technique is that it removes the explicit dependence on control values at each node but at the risk of becoming numerically unstable. For that reason, differential inclusions will not be considered in this thesis.

In contrast, collocation techniques enforce the system dynamics using quadrature rules or interpolation. The goal is to find a reasonable interpolant that matches the state values and derivatives at the nodes spanning some subset of the time interval. At each collocation point, i.e. the points defined between the nodes at which we evaluate the interpolant, an equality constraint is formed, equating the interpolant derivative to the state derivative function and thus ensuring that the dynamics hold (approximately) true across the time interval. The technique is based on a segmentation of the considered trajectory and a near-uniform discrete approximation of thrust directions by a set of thrust profiles and inequality constraints for each segment. The problem is then to minimise the total characteristic velocity with terminal conditions. Hence, the technique can be considered a sequential nonlinear programming algorithm defined by three steps. The first step is transforming the dynamic system into a problem with a finite set of variables. The second step is solving the finite-dimensional problem using a parameter optimisation method (i.e., the nonlinear programming subproblem). Finally, the third step is to check the accuracy of the finite-dimensional approximation and, if necessary, repeat the transcription and optimisation steps.

The last considered technique is the direct shooting method, which works by parameterising the control vector and integrating the motion equations with a time-marching algorithm, effectively reducing the required number of constraints. One can then ensure consistency between the objective and the numerical integrator by determining the corresponding cost function using a quadrature approximation. Consequently, the method results in a problem of minimising a specific cost function subject to any path and interior constraints, as well as a reduced problem size.

Since both the coverage and constraint functions assume the simple form of evenly spaced evaluation points along the trajectory, where the integration occurs segment-wise between manoeuvres, we draw inspiration from the shooting method as the fundamental principle of the selected optimisation strategy. The final stage of the optimisation process is to formulate a global solution method based on the suggested approach, method and technique.

4.2 Global Optimisation

With computational power and resource management in constant development, powerful direct global optimisation techniques have lately gained a prominent role within trajectory design, given their ability to solve large-scale numerical problems remarkably cheaply and online [44]. Generally, we speak of two categories of algorithms: deterministic and stochastic. Deterministic algorithms avoid randomness in their search process by instead using readily available gradient information of the problem, which enables a well-defined convergence criterion for a global solution. Although these methods allow for a rigorous search process, they typically require an initial guess on the decision parameters, which can be challenging depending on the complexity of the problem. Since the problem considered in this thesis involves particularly complex dynamics and swarming spacecraft, traditional nonlinear programming algorithms will generally become difficult to initialise.

An alternative is a stochastic algorithm, which roots its search process in the random generation of decision parameters, consequently treating the problem as a black-box model which imposes fewer constrictions for selecting appropriate numerical techniques and approaches. A common trait of most stochastic methods is that they employ a metaheuristic: an iterative master process that directs the operations of auxiliary heuristics to efficiently cover a complex search space and determine a high-quality solution [44]. As such, these methods provide a robust tool for obtaining qualitative solutions in complex domains when considering reasonable time horizons and computational resources. However, given their use of randomness instead of gradient information, they naturally lack the theoretical foundation for verifying global convergence. On the other hand, given their proven robustness in rugged search spaces and the advantage of self-initialising decision variables, stochastic methods are generally favoured for solving trajectory optimisation problems.

Heuristic methods generally employ one of two strategies: a single solution, such as local search, simulated annealing and greedy heuristics, and population-based techniques, such as evolutionary algorithms. Since the latter category tends to be particularly suitable for problems considering impulsive control [44], it will be of focus from here on. Among notable evolutionary algorithms (EA), we distinguish between the most common methods applied to trajectory optimisation, namely Genetic Algorithms (GA), Particle Swarm Optimisation (PSO), Differential Evolution (DE) and Ant Colony Optimization (ACO). During the last two decades, the European Space Agency (ESA) has conducted extensive research on the application and sufficiency of these methods for solving trajectory optimisation problems, where DE traditionally has demonstrated promising results [58]. However, in recent years, the Ant Colony Optimiser has been shown to outperform most of these algorithms in several benchmark trajectory problems, making it an interesting candidate algorithm to be considered in this thesis as well [59–62].

4.3 Ant Colony Optimisation

With recent refinements of the ant-based metaheuristic showing promising results for several benchmark trajectory optimisation problems, including missions with impulsive control strategies, we will employ a single-objective ant-colony optimiser (ACO) to solve the problem presented in Eq. (3.14). In detail, we will specifically implement the extended ant-colony optimiser (EACO) first presented by Schlüter et al. [63] and later modified by Acciarini [62]. Apart from its proven performance, the motivation for choosing an ant-based algorithm is also related to the structure of the considered problem. First, recall that we aim to find optimal paths through constrained yet large multidimensional spaces. To counter such challenges, ACOs typically use sophisticated large-search exploration mechanisms, hybridising local and stochastic methods. These methods are essential for refining the solution, accelerating convergence and avoiding local optima, which is crucial for finding global or near-global solutions in complex landscapes. As such, EACO becomes particularly efficient for solving large combinatorial problems where the search includes selecting discrete events such as impulsive manoeuvres. Finally, EACO also possesses a computational advantage of a high level of parallelisation, making it suitable for optimising trajectories involving great computational effort. Since each suggested solution results in a deterministic boundary value problem when considering the dynamics, we can use parallelisation to evaluate large quantities of candidate solutions, consequently reducing computational time and enabling a more extensive search process. Due to the high level of relevance for the chosen optimisation strategy in this thesis, we will now introduce the fundamental principles of the optimiser on a practical level. To better understand these techniques, we will first introduce the original ACO algorithm and then further expand the description to the modified version of EACO.

4.3.1 Single-Objective Ant Colony Optimiser

The core concept of ACO is to imitate the natural behaviour of ants searching for food. To find food, ants typically explore their surroundings in a random manner, starting from their nest. Successful ants then return to the nest, leaving a chemical pheromone trail to mark its path, which attracts colony members to follow the same path. However, as time passes, the pheromones gradually start to evaporate, leaving only well-updated paths appealing and eventually only the shortest path. Using the ant-based indirect communication strategy as a heuristic to cover complex search spaces was first introduced to mathematical optimization by Dorigo et al. in 1992 [64]. At that time, ACO was primarily applied to combinatorial problems using artificial ants to traverse a graph to find an optimal path based on factors like length or other defined costs, much like the travelling salesman problem. By employing a parameterized probabilistic model, a pheromone table, the artificial ants select a path across a fully connected graph $G(C, L)$ of vertices C and connecting edges L. The set of vertices C stands for the solution's components, incrementally chosen by each ant to construct its path. Each pheromone value, τ_i, representing the pheromone strength of path i, then guides the colony in choosing subsequent paths with the probability

$$p_i = \frac{\tau_i}{\sum_i \tau_i}, \quad i = 1, 2, 3, \dots \tag{4.1}$$

By adjusting the pheromone values according to information gathered during the search process using the update rule $\Delta\tau_i + \frac{Q}{l_i}$ for some positive model parameter Q, the search process will eventually lead to the discovery of the shortest path due to dominant probability. Furthermore, to avoid favouring the early discovered paths through greater aggregate probabilities, each pheromone trail is assumed to decrease over time according to an evaporation rule $(1 - \rho)\tau_i$ considering an evaporation speed $\rho \in (0, 1]$.

4.3.2 Extended ACO for Non-Convex MINLP

Although the example of ants traversing through a graph-like domain is generally adapted for combinatorial problems, it can also be extended to continuous domains without changing its conceptual principles. The extension to mixed-integer search domains was first proposed by Schlüter et al. [63] and has been tested extensively on various benchmark space applications since then [59, 65]. To properly convey the functional principles of EACO and its differences compared to ACO, we will first recall a set of necessary mathematical definitions.

In general, EACO assumes the problem to be in the form of a mixed-integer nonlinear program (MINLP), which includes the nonlinear program considered in Eq. (3.14). A MINLP problem can effectively be formulated as

$$\begin{aligned} \min \quad & f(x, y) \\ \text{s.t.} \quad & h(x, y) = 0, \qquad && i = 1, \dots, m_{eq} \\ & g(x, y) \leq 0, && i = m_{eq} + 1, \dots, m \\ & x \in \boldsymbol{X},\ y \in \boldsymbol{Y}. \end{aligned} \tag{4.2}$$

where $\boldsymbol{X} \subseteq \mathbb{R}^{n_c}$ is a convex compact set of continuous values, $\boldsymbol{Y} \in \mathbb{Z}^{n_i}$ is a polyhedral set of integer points, where both sets have given bounds, $f : \mathbb{R}^{n_c + n_i} \to \mathbb{R}$ and where h and g are the corresponding constraint functions. Now, in contrast to ACO, the EACO framework considers probability samples from both a continuous probability density function (cPDF) $P : \mathbb{R} \to \mathbb{R}_0$ and a discrete probability density function (dPDF) $Q : \mathbb{Z} \to \mathbb{R}_0$ for the two set of variables as the guiding probabilistic during the path selection. In detail, we adopt a n multi-kernel Gaussian probability density function $G^{h=1,\dots,n}$ where $n = n_c + n_d$ since it allows for making use of its fast sampling property while also overcoming issues of retaining focus on several promising disjoint areas of the search space. The multi-kernel function can be formulated as a weighted sum of K one-dimensional Gaussian density functions corresponding to each dimension of the domain, that is

$$G^h(t, \omega, \mu, \sigma) = \sum_{k=1}^{K} w_k^h \cdot g_k^h = \sum_{k=1}^{K} w_k^h \frac{1}{\sigma_k^h \sqrt{2\pi}} e^{\frac{-(t-\mu_k^h)^2}{2(\sigma_k^h)^2}}, \quad h = 1, \dots, n. \tag{4.3}$$

The continuous and discrete probabilities can then be sampled from their respective density functions, where the discrete function considers the cumulative probability of the multi-kernel function around some integer d:

$$P^i(t) = G^i(t, \omega, \mu, \sigma), \qquad i = 1, \dots, n_c, \tag{4.4}$$

$$Q^i(d) = \int_{d-1/2}^{d+1/2} G^{n_c+j}(t, \omega, \mu, \sigma)dt, \qquad j = 1, \dots, n_d. \tag{4.5}$$

The pheromone triplets $(w_k^h, \sigma_k^h, \mu_h^k)$ are thus central for the path selection process. To update each pheromone, we use information from a solution archive $S := \{(x, y)^1, \dots, (x, y)^K\}$ of the previously K most promising n-dimensional solution vectors s_k. We will now define each of the three triplets in the suggested order. The first parameter, w_k^h, indicate the importance of an individual ant by weighting the multi-kernel PDF accordingly, thus establishing a linear order of priority of the solution archive where s_1 and s_K are the most and least promising solution vectors, respectively. For every new solution value found, its penalty function value, or attraction, is compared to each vector in the archive. If the new solution vector is more promising than s_k, we identify its position in S and shift the subsequent solution vectors backwards with one index, discarding the least promising solution. The process can be compared to the phenomenon of evaporation of a trail. As such, the individual weights are computed by linear proportion

$$\omega_k^h = \frac{K - k + 1}{\sum_{j=1}^{K} j}, \quad \text{where} \quad \sum_{k=1}^{K} \omega_k^h = 1. \tag{4.6}$$

The mean value μ_k^h can be directly obtained from the single solution components in the solution archive, that is

$$\mu_k^h = \begin{cases} x_k^h, & h = 1, \dots, n_c, \\ y_k^{h-n_c}, & h = n_c + 1, \dots n_d. \end{cases} \tag{4.7}$$

The standard deviations σ_k^h can be computed using the maximal and minimal distance between all variable dimensions in the solution archive:

$$\sigma_k^h = \frac{D_{max}(h) - D_{min}(h)}{n_G}, \quad \begin{cases} D_{max}(h) = max\{|s_h^p - s_h^q| : \ p, q \in [1, \dots, K], \ p \neq q\}, \\ D_{min}(h) = min\{|s_h^p - s_h^q| : \ p, q \in [1, \dots, K], \ p \neq q\}, \end{cases} \tag{4.8}$$

where $h = 1, \ldots, n_c, n_c + 1, \ldots, n_d + n_c$ and n_G is the number of generations passed so far. To avoid zero-valued standard deviations, which can occur when all of the solution vectors stored in the archive have converged for the same integer values, thus resulting in the sampling at the explicit means stagnating the improvement, we introduce two new lower bounds for the integer case:

$$\sigma_k^h = max \left\{ \frac{D_{max}(h) - D_{min}(h)}{n_G}, \ \frac{1}{n_G}, (1 - \frac{1}{\sqrt{n_i}})/2 \right\}, \quad \text{if } h = n_c + 1, \ldots, n_c + n_d. \tag{4.9}$$

In summary, the evolutionary process begins with creating the first generations of ants based on a sampling from two uniform continuous and discrete PDFs. We refer to the colony as a population and each candidate solution vector as a chromosome. After individually evaluating the objective function and constraint violations, we select the K most promising chromosomes to define the ordered solution archive S. Since each evaluation is inherently independent, we can parallelise the operation on multiple processing threads to improve the computational performance. We define the next generation by choosing the mean value μ_k^h for every h in the archive. The choice is not completely random as it depends on the corresponding weight w_k^h, which ensures the linear priority established in the solution archive. New chromosomes are then effectively created by sampling new random numbers for each dimension's mean value and standard deviation. As such, we have defined a new variable combination dependent on the previously most promising set of chromosomes, thus forming the new generation. The selection and sampling process then repeats until we have reached a specific stop criterion, which in this case will be the number of generations.

Oracle Penalty Method

A common strategy for managing difficult constraints is to replace the objective with a new penalty function defined as a weighted combination of the original objective function and constraint violations. This strategy is notably similar to the relaxation of the domain-restricting path constraint in Section 3.4. In general, simple penalty methods, like death and static, are often favoured for their ease of use but may not always perform well on more challenging problems. In contrast, advanced methods such as annealing offer improved performance but at the condition of intricate parameter selection. Hence, focusing on high-potential methods with minimal parameter needs is of interest to avoid these shortcomings. In this case of EACO, we assume a single-term high-potential penalty method based on the oracle parameter Ω representing the estimated preferred objective function value. Predicting a reasonable value for Ω is a non-trivial task due to the inherent complexity of the trajectory problem. To ensure robustness, we initialise this variable with a random estimate and refine it using insights from an initial optimiser test run. Let $z := (x, y)$ denote a vector of the considered continuous and discrete decision variables, x and y, representing an ant. The oracle selection method is then based on a ℓ^1-norm function, or residual, over all violations of the given m_{eq} equality constraints $h(z)$ and m_{ineq} inequality constraints $g(z)$ in the considered optimisation problem:

$$||z||_1 = \sum_{i=1}^{m_{eq}} |h_i(z)| - \sum_{j=1}^{m_{ineq}} min\{0, g_j z\}. \tag{4.10}$$

The penalty $p(z)$ is specifically defined as

$$p(z) = \begin{cases} \alpha \cdot |f(z) - \Omega| + (1-\alpha)||z||_1 - \beta, & \text{if } f(z) > \Omega \text{ or } ||z||_1 > 0, \\ -|f(z) - \Omega|, & \text{if } f(z) \leq \Omega \text{ and } ||z||_1 = 0, \end{cases} \tag{4.11}$$

where the parameter α balances the weight between the objective function and the residual, and β acts as a dynamic bias term, which increases over the generations. Thus, both terms enable the penalty method to balance the search process towards either the objective function or the residual. The parameters have conditional definitions depending on the relation between the objective value $f(z)$, residual $||z||_1$ and oracle Ω. For further details on the definitions of α and β, we refer to the paper presented by Schlüter et al. [63]. Moreover, since the oracle is selected as either the best, equal to or slightly improved optimal objective value, finding a good oracle value is analogous to solving the original problem to its global optimum. Thus, a good value can efficiently be determined using the self-tuning property by updating the oracle dynamically with the knowledge gained from several consecutive runs of the given optimisation problem. With the algorithm's structure in mind, EACO can be applied to both box-bounded single-objective, constrained and unconstrained optimisation problems with continuous and integer variables. As such, it is also suitable for the problem presented in Eq. (3.14), which in this case can be considered as a box-bounded single objective optimisation problem when the collision constraint is decoupled from the problem formulation and instead implemented as an infeasible solution marker during the integration process.

4.3.3 Modified EACO

Even with the boundaries introduced in Eq. (4.9), related studies have shown a tendency for EACO to converge prematurely at local optima for several simpler benchmark problems. To counter this shortcoming, Acciarini [62] presents three novel modifications demonstrating promising results. The first modification relates to the computation of standard deviations. When using the original formulation presented in Eqs. (4.8-4.9), the standard deviation values were observed approaching a zero value, resulting in a convergence characterised by sampling near the explicit means. To avoid the issue of rapidly diminishing deviations, a new standard deviations convergence speed parameter $NGenMark$ is introduced, replacing the previous number of generations n_G. The new parameter works much similarly to n_G, but for a defined upper limit. Once this limit has been reached, the value of $NGenMark$ resets to one, and the process continuous. Consequently, as long as $D_{max}(h) - D_{min}(h) \neq 0$ for the set of continuous variables, σ_k^h can be ensured to not reach small values by instead choosing a low $NGenMark$ value. We can then ensure a suitable magnitude of $NGenMark$ by introducing two new parameters to monitor the spread of chromosomes in the solution archive. In detail, these two parameters describe the flatness in each dimension for the variables, dx, and the penalties, dp, when considering the respective best and worst chromosomes, s_1 and s_k, in the solution archive for each generation:

$$dx = \sum_{i=1}^{n} (|x_{K,i} - x_{1,i}|), \tag{4.12}$$

$$dp = |p_K - p_1|. \tag{4.13}$$

The second modification concerns how each weight is computed and allows adjusting the focus of the chromosome selection by introducing a new convergence speed parameter q regulating the convergence towards a local minimum. As such, the weights are reformulated as

$$\omega_k^h = \frac{1}{qK\sqrt{2\pi}} e^{-\frac{(k-1)^2}{2q^2K^2}}, \quad k = 1, \ldots, K. \tag{4.14}$$

If q is then chosen to be relatively large, the weights will become more uniform. In contrast, if q is chosen to be small, the weights will instead favour the best individuals, similar to the original formulation in EACO. As such, the sampling of new offspring will be done considering only the k:th singular Gaussian function in Eq. (4.3) with the probability

$$p_k^h = \frac{\omega_k^h}{\sum_{k=1}^{K} \omega_k^h}. \tag{4.15}$$

The process of sampling the chosen Gaussian function g_k^h begins with each chromosome first being assigned its respective cumulative value from the computed probability p_k^h. Then, by sampling a number from a uniform distribution $U(0, 1)$, the specific Gaussian function can be chosen accordingly. In order to begin with a broad search process and then focus on a subset of the solution archive after an increasing number of generations, the value of q is adjusted dynamically after a certain threshold number of generations. Finally, the last modification includes the addition of an accuracy parameter α, which ensures that the considered penalty functions in the solution archive remain diverse enough. This is done by enforcing the criteria demanding the new chromosomes to outperform the K:th solution in the archive and to have a penalty function value with the minimal difference of α.

Chapter 5

Methods and Software

In order to simulate and determine optimal trajectories for the considered high-dimensional problem, we must first define a suitable model that can efficiently exploit the available computational resources. The model should include both the propagation of the state through numerical integration, a sufficient method for detecting collisions and a general optimisation scheme for considering both single and multi-spacecraft settings. In this chapter, we will therefore focus on presenting the numerical methods, techniques, and software used to account for these considerations.

5.1 Numerical Methods

Recalling from Chapter 4, we know that each generation of the extended ant colony optimiser (EACO) results in a population representing a set of chromosomes, or candidate solutions, for the considered trajectory problem. In this case, these solutions present numeric values for the initial velocity $\boldsymbol{v}_0$ and a complete control sequence $\mathcal{U}(x, \mathcal{T}_C)$ over the considered mission duration. For a given initial position $\boldsymbol{r}_0$, we can subsequently define the initial state of the problem as $\boldsymbol{x}(t_0) = [\boldsymbol{r}_0^T \ \boldsymbol{v}_0^T]^T$. Under the assumption of an impulsive control scheme, the equations of motion consequently result in a series of coupled non-stiff initial value problems (IVP) defining a segmentation of the complete trajectory as seen in Fig. 2.3. For each segment i, the corresponding IVP takes the form

$$\dot{x}(t) = f(t, x, u), \qquad x(t_0^i) = x_0^i \tag{5.1}$$

where $f : \mathbb{R} \times \mathbb{R}^N \to \mathbb{R}^N$ is a sufficiently smooth function such that each time $t_0^i \in \mathbb{R}^+$ and initial state $x(t_0^i) \in \mathbb{R}^N$ yields in a unique trajectory $x(t) : [t_0^i, t_f^i] \to \mathbb{R}^N$. Given the complex nature of these equations, which use impulsive controls, a polyhedral gravity model and quaternion rotations, finding a closed-form expression of an integral solving the differential system remains challenging. Therefore, we will approach solving Eq. (5.1) using numerical approximation where the state is propagated using Cartesian elements. Although numerous other alternatives may improve the computational efficiency and accuracy of the approximation, we favour Cartesian elements given their simplicity and lack of singularities.

When selecting an appropriate numerical method, we recognise that the integration should balance two different objectives. First, to accurately capture the dynamics of the problem, we aim at minimising the residual error $\epsilon = \bar{x}(t) - x(t)$ between the analytical solution $x(t)$ and the numerical approximation $\bar{x}(t)$. Second, since the optimiser will employ the integrator for each defined chromosome and generation, it should also reduce the computational complexity as far as possible. Among the commonly favoured numerical integrators in dynamical astronomy and spacecraft trajectory optimisation, two prominent options emerge: Adams-Bashforth integrator and embedded Runge-Kutta methods. Here, it should be noted that the former method naturally favours smaller step sizes during the integration process, thus preventing it from accurately capturing the dynamics of highly eccentric trajectories [66]. Given that we have relaxed the constraint bounding the states to the spherical grid, possible solutions will likely range from circular trajectories where $e = 0$ to even hyperbolic trajectories where $e > 1$. As such, we omit the Adams-Bashforth integrator from the study leaving the Runge-Kutta embeddings for further investigation.

5.1.1 Embedded Runge-Kutta Methods

The conventional embedded Runge-Kutta scheme uses two formulae of orders p and q to solve the considered initial boundary value problem [67]. In most cases, these orders hold the property that $q > p$ and $q = p + 1$. Adopting a standard notation, let $\hat{x}_{n+1}$ and x_{n+1} represent two estimates of the state $x(t_{n+1})$ where the state update equation is given as $x_{n+1} = x_n + h_n$. Then, the two formulae can be defined as

$$\begin{cases} \hat{x}_{n+1} = \hat{x}_n + \sum\limits_{i=1}^{S} \hat{b}_i k_i, \\ x_{n+1} = \hat{x}_n + \sum\limits_{i=1}^{S} b_i k_i, \\ k_1 = h_n f(t_n, \hat{x}_n), \\ k_i = h_n f(t_n + c_i h_n,\ \hat{x}_n + \sum\limits_{j=1}^{i-1} a_{ij} k_j), \end{cases} \tag{5.2}$$

where $i = 2, 3, 4, \ldots, S$ are the number of stages used to indicate the q:th order formula and h_n the current step size [68]. The coefficients $a_{ij},\ b_{ij},\ \hat{b}_{ij},\ c_{ij} = \sum_{j=1}^{i-1} a_{ij}$ (for $i > j$ and $i, j = 1, 2, 3, \ldots, S$) are chosen such that they satisfy the corresponding equations of condition obtained from Taylor expansions of order k

$$\tau_j^k = 0, \quad k = 1, 2, \ldots, p \tag{5.3}$$

$$\hat{\tau}_j^k = 0, \quad k = 1, 2, \ldots, q \tag{5.4}$$

$$j = 1, 2, \ldots, r_k. \tag{5.5}$$

Considering smaller rotating bodies as the major component exerting a gravitational attraction on the spacecraft, it follows by the corresponding low mass and rotating motion that the irregular body will result in varying complexity of dynamics for the equations of motion, with smaller variations further away from the body and maximal dynamics at some point inside the inner bounding sphere. As such, using an adaptive step integrator is advantageous to save computational time where the dynamics are low and greater steps can be taken without losing accuracy. To control the adaptive step, we use the estimate $\delta_{n+1} = x_{n+1} - \hat{x}_{n+1}$ of the local error ϵ_{n+1} in the p:th order formula presented by Hull et al. [69]

$$h_{n+1} = 0.9 h_n \left[\frac{E_{tol}}{||\delta_{n+1}||_\infty} \right]^{\frac{1}{p+1}}. \tag{5.6}$$

The step size control presented in Eq. (5.6) is used even if $||\delta_{n+1}||_\infty$ exceeds the tolerated local error E_{tol} or not, thus representing both the re-computation of some previously rejected step and the prediction of a new successive step, respectively [68].

5.1.2 Dormand-Prince 8(7)-13M

Among the different orders of Runge-Kutta formulae, Dormand-Prince 8(7)-13M has consistently demonstrated a preferable balance between reliable, accurate and computationally competitive performance considering a diverse set of problems with complex spacecraft dynamics [70–72]. With extensive research on its convergence properties and computational complexity [68, 73], we refer to previous work to motivate it as a suitable integrator.

The algorithm extends the popular fourth-order Runge-Kutta formulae by utilising $S = 13$ stages and formulae equations of orders $q = 8$ and $p = 7$. As a result, it efficiently combines the benefits of high-order polynomial accuracy and dynamic error control using the adaptive step size formula presented in Eq. (5.6). By introducing these two elements, the integrator becomes particularly proficient at managing challenging ordinary differential equations, including those that mix stiff and non-stiff behaviour. However, due to the higher order of q and p, the formulae equations must satisfy a total of 285 equations of condition for both orders. To reduce the number of equations, Dormand and Prince introduce in their original paper the following extra relations

$$a_{i2} = 0, \quad i = 4, \dots, 13 \tag{5.7}$$

$$a_{i3} = 0, \quad i = 6, \dots, 13 \tag{5.8}$$

$$\hat{b}_i = b_i = 0, \quad i = 2, \dots, 5 \tag{5.9}$$

$$\sum_{i=1}^{13} \hat{b}_i a_{ij} = \hat{b}_j (1 - c_j), \quad j = 1, \dots, 13 \tag{5.10}$$

$$\sum_{i=1}^{12} b_i a_{ij} = b_j (1 - c_j), \quad j = 1, \dots, 12 \tag{5.11}$$

$$\sum_{j=1}^{12} a_{ij} c_j^k = \frac{c_i^{k+1}}{k+1}, \quad k = 1;\ i = k+2, \dots, 13,\ k = 3;\ i = 6; \dots, 13 \tag{5.12}$$

$$\sum_{j=1}^{13} \hat{b}_i (1 - c_i) a_{ij} = 0, \quad j = 4, 5 \tag{5.13}$$

$$\sum_{i=1}^{13} \hat{b}_i c_i (1 - c_i) a_{ij} = 0, \quad j = 4, 5 \tag{5.14}$$

$$\sum_{i=1}^{13} \sum_{j=1}^{12} \hat{b}_i (1 - c_i) a_{ij} a_{jk} = 0, \quad j = 4, 5 \tag{5.15}$$

$$\sum_{i=1}^{13} (b_i - \hat{b}_i)(1 - c_i) a_{ij} = 0, \quad j = 4, 5 \tag{5.16}$$

resulting in ten degrees of freedom through a_{84}, b_{12}, $\hat{b}_{13}$, c_2, c_3, c_6, c_{10} and c_{11} [68]. Finally, the coefficients used to construct the two formulae equations are presented in butcher form as seen in Fig. A.1 in the Appendix.

5.1.3 Continuous Extension

Since the Dormand-Prince 8(7)-13M algorithm uses an adaptive step size control, the resulting trajectory will remain limited to the sequence of time steps chosen during the integration process. In order to correctly evaluate the cost function and corresponding path constraints, the trajectory must be re-sampled for each measurement time defined on the set $\mathcal{T}_T$. Therefore, we define a continuous extension between each state determined by the high-order integrator to obtain the required measurement points. A continuous extension between two times $[t_n, t_{n+1}]$ is, in fact, a polynomial function $P_n(t)$ that approximates the state transition throughout the considered time interval [74]. When we reduce the step multiple times, we refer to the continuous extension as the so-called dense output. The next step is determining an appropriate order for the continuous extension. Given the use of a high-order formula to advance the integration, we can employ a lower-order polynomial function to estimate the intermediate states while still maintaining the global order of accuracy from the integrator, thus saving the computational costs of additional high-order approximations. In this case, we assume a cubic Hermite spline for the interpolation scheme since the two states a $x(t_n)$, $x(t_{n+1})$ and their corresponding derivatives $\boldsymbol{m}(t_n, x_n)$, $\boldsymbol{m}(t_{n+1}, x_{n+1})$ are readily available by the integrator [75]. As such, interpolating the state $x(t)$ in the arbitrary interval (x_n, x_{n+1}) can be done by mapping the interval to $[0, 1]$ through an affine change of variables. The interpolation polynomial function can then be defined as

$$P(t) = \phi_0(k)x_n + \psi_0(k)(t_{n+1} - t_n)\boldsymbol{m}_n + \phi_1(k)x_{n+1} + \psi_1(k)(t_{n+1} - t_n)\boldsymbol{m}_{n+1}, \tag{5.17}$$

where $t \in (t_n, t_{n+1})$, $\boldsymbol{m}_n$ and $\boldsymbol{m}_{n+1}$ are the tangent values, $k = (t - t_n)/(t_{n+1} - t_n)$ and $\phi_0, \phi_1, \psi_0, \psi_1$ are the expanded Hermite basis functions

$$\begin{aligned} \phi_0 &= 2k^3 - 3k^2 + 1, & \phi_1 &= -2k^3 + 3k^2, \\ \psi_0 &= k^3 - 2k^2 + k, & \psi_1 &= k^3 - k^2. \end{aligned}$$

5.2 Collision Detection

In Section 3.4, we presented a path constraint to ensure that no feasible solution would result in a collision between a spacecraft and the body of interest. The function is defined for each state along a continuous trajectory and has a binary output. In this case, we assume the trajectory to be valid up until the point of an active collision event. As such, an event is said to occur at time t^* if $C(t^*, x(t^*)) = 0$ for the given constraint function. To avoid bypassing collisions that occur at an intermediate time step to $\mathcal{T}_T$, we incorporate an *Event-Detection Algorithm* already in the numerical integrator. In this section, we will first present the basic idea of managing collision events and then provide further details on the detection of collisions using the readily available polyhedral shape model presented in Section 2.3.

5.2.1 Event-Detection

In order to accurately locate a discrete event, we require an approximation of the state $x(t)$ for a sufficient discretisation of the mission duration $[t_0, t_f]$, which is not always known in advance. However, with a continuous extension between each state defined by the numerical integrator, we can instead monitor a binary output event function $E(\cdot) = \{0, 1\}$ by following its change of sign between two arbitrary states

$x(t_n)$ and $x(t_{n+1})$. Suppose an event occurs on the interval $[t_n, t_{n+1}]$. In that case, we can use the dense output defined by Eq. (5.17) to accurately pinpoint the time of an event by solving

$$E(x(t^*)) = 0 \tag{5.18}$$

as a root-finding problem using the bisection method defined in Alg. (1) [74]. By implementing the continuous extension between every consecutive step of the adaptive integrator, the event detection algorithm can effectively be configured to either interrupt or monitor the integration process. In the former case, one could let the event function represent the collision constraint directly, that is, $E(x(t)) = \mathcal{C}(x(t))$, and consequently terminate the integration at infeasible solutions. However, to avoid repeatedly evaluating a demanding collision detection function during the integration process, we favour the latter approach by monitoring the event of any spacecraft positions being defined inside the inner bounding sphere $\mathcal{S}_{\mathcal{I}}$. Consequently, we define the event function as

$$E(\boldsymbol{r}(t)) = \begin{cases} 0, & \text{if } ||\boldsymbol{r}(t)|| < r_{\mathcal{I}}, \\ 1, & \text{otherwise,} \end{cases} \tag{5.19}$$

where $\boldsymbol{r}(t)$ is the spacecraft's position vector expressed in Cartesian coordinates and $r_{\mathcal{I}}$ the radius of the inner bounding sphere. In this case, we assume that a sign-change method retains sufficient precision for identifying $\boldsymbol{r}(t) \in \mathcal{S}_{\mathcal{I}}$. The accuracy follows using a high-order integrator, which will inherently advance the state with minimal steps near the body to account for the increasing dynamics. After the integration process has terminated, we proceed by evaluating the stored events using a collision detection function representing the path constraint $\mathcal{C}(\boldsymbol{r}(t))$. In this work, we identify collision using Möller-Trumbore's Ray/Triangle Intersection method.

Algorithm 1 Bisection Method for Root Finding

```
Require: Polynomial $P_n(t)$, interval $[t_n, t_{n+1}]$, Event function $E(x(t))$
Ensure: Approximation of root $x_{\text{root}}$
$t_{\text{low}} \leftarrow t_n$
$t_{\text{high}} \leftarrow t_{n+1}$
$t_{\text{root}} \leftarrow \frac{t_{\text{low}}+t_{\text{high}}}{2}$
while $E(P_n(t)) = 1$ do
    if $E(P_n(t_{\text{low}})) = 1$ and $E(P_n(t_{\text{root}})) = 0$ then
        $t_{\text{high}} \leftarrow t_{\text{root}}$
    else
        $t_{\text{low}} \leftarrow t_{\text{root}}$
    end if
    $t_{\text{root}} \leftarrow \frac{t_{\text{low}}+t_{\text{high}}}{2}$
end while
    return $t_{\text{root}}$
```

5.2.2 Möller-Trumbore Ray/Triangle Intersection

Provided a polyhedral gravity model, we may utilise the knowledge of the specific computing point corresponding to a spacecraft's position and the location of each tetrahedral face on the polyhedron's surface to determine whether the spacecraft has entered the closed surface. Thus effectively representing a collision with the body. To verify the position relative to each face, we adopt the efficient Möller-Trumbore's Ray/Triangle Intersection method [76]. Since the method only utilises the individual vertices of the faces, which are readily available in the polyhedral shape model, it reduces the computational storage significantly compared to similar algorithms that often require storing triangle-plane equations.

First, let $R(t)$ be a ray representing the Cartesian position vector $\boldsymbol{r}(t)$ of a spacecraft at time t with origin $\mathcal{O}$ at the small-body centre of mass. The ray can then be deconstructed for a normalized direction D such that

$$R(t) = O + \ell D. \tag{5.20}$$

Next, let V_0, V_1 and V_2 denote the vertices corresponding to the three line segments defining a tetrahedral face on the polyhedron. The goal is to perform a transformation at the ray's origin such that the resulting vector contains information about the intersection coordinates u, v and the distance to the intersection, ℓ. Provided knowledge of the location of the three vertices corresponding to a face, we define the location of some arbitrary point, $T(u, v)$, on the triangular plane using barycentric coordinates where

$$T(u, v) = (1 - u - v)V_0 + uV_1 + uV_2, \quad u, v \geq 0, \quad u + v \leq 1. \tag{5.21}$$

Thus, computing the intersection between the ray and a face is equivalent to solving the system of equations resulting from $R(t) = T(u, v)$. Rearranging the terms, we can separate the solution for the intersection vector

$$[-D \quad V_1 - V_0 \quad V_2 - V_0] \begin{bmatrix} \ell \\ u \\ v \end{bmatrix} = O - V_0, \tag{5.22}$$

where $M = [-D \quad V_1 - V_0 \quad V_2 - V_0]$. Notably, solving the system in Eq. (5.22) is equivalent to translating an arbitrarily positioned face in space to a state where the vertex V_0 is defined at origin $\mathcal{O}$, and then transforming the face into a unit triangle defined on the y-z plane, perpendicular to the ray defined along the x-axis in the Cartesian frame. A geometric perspective on the translation and the subsequent change of basis is provided in Fig. 5.1. Introducing a more simple notation where $E_1 = V_1 - V_0$, $E_2 = V_2 - V_0$ and $T = O - V_0$, we solve Eq. (5.22) using Cramer's rule

$$\begin{bmatrix} \ell \\ u \\ v \end{bmatrix} = \frac{1}{|-D \quad E_1 \quad E_2|} \begin{bmatrix} |T \quad E_1 \quad E_2| \\ |-D \quad T \quad E_2| \\ |-D \quad E_1 \quad T| \end{bmatrix}. \tag{5.23}$$

Using the scalar triple product for vectors defined in Euclidean space, as well as the corresponding commutative property of the product, that is assuming $|X\ Y\ Z| = -(X \times Z) \cdot Y = -(Z \times Y) \cdot X$, we may simplify Eq.(5.23) into

$$\begin{bmatrix} \ell \\ u \\ v \end{bmatrix} = \frac{1}{(D \times E_2) \cdot E_1} \begin{bmatrix} (T \times E_1) \cdot E_2 \\ (D \times E_2) \cdot T \\ (T \times E_1) \cdot D \end{bmatrix} = \frac{1}{P \cdot E_1} \begin{bmatrix} Q \cdot E_2 \\ P \cdot T \\ Q \cdot D \end{bmatrix}, \tag{5.24}$$

where $Q = T \times E_1$ and $P = D \times E_2$. As such, one can use the information on the distance to the intersection, ℓ, to determine the spacecraft's position relative to the considered face. Furthermore, by iterating the process for all defined faces on the polyhedral shape model, we can identify whether or not the spacecraft is outside the bounded surface of the body and subsequently evaluate the condition of a collision-free state.

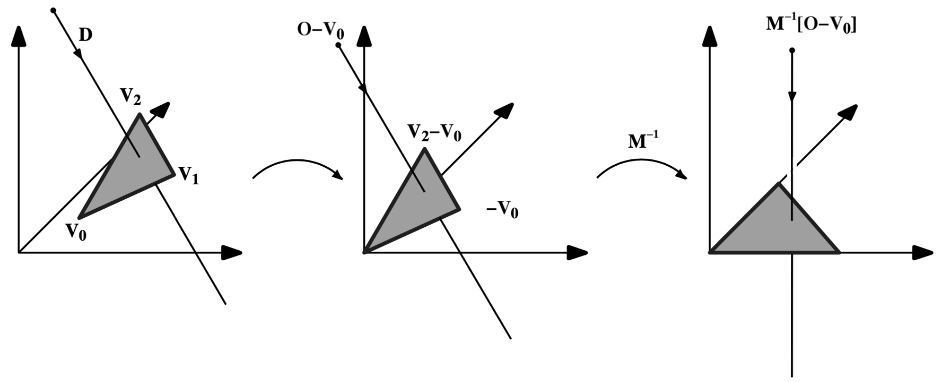


Figure 5.1: Translation and change of basis of the ray $R(t)$ origin [76].

5.3 Optimisation Models

Finally, after having presented and discussed the full problem, including selecting appropriate methods of optimisation and numerical techniques for simulating spacecraft trajectories, we can proceed by completing an optimisation model that integrates the aspects mentioned above to solve the problem in Eq. (3.14). For simplicity, we begin by introducing a simpler setting consisting of only a single spacecraft and then extend the model to consider a swarm.

5.3.1 Minimal Model

The first step of the optimisation procedure is to set up the correct parameter environment. As presented in Fig. 5.2, the simulation input requires the initialisation of the three different modules, including the creation of rotation quaternions for each measurement time, a longitude-latitude tensor with normalised weights and a polyhedral gravity model of the studied body. The fourth group of simulation input is related to providing values for the hyperparameters corresponding to the optimiser, integrator and problem definition.

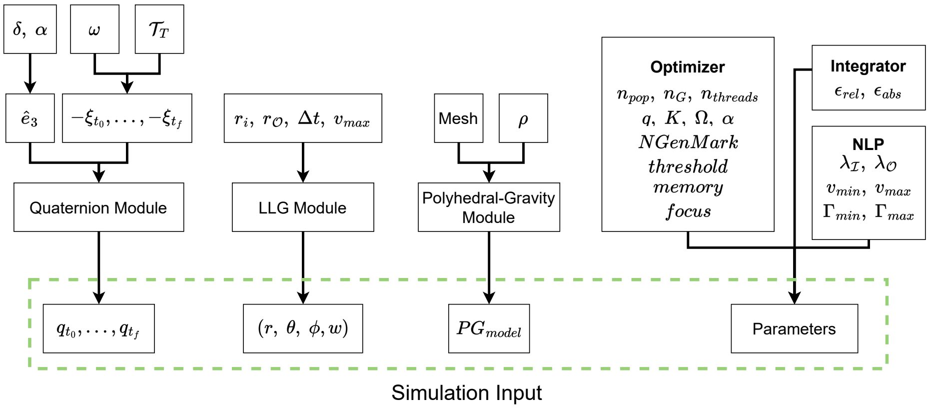


Figure 5.2: Initialisation of simulation input.

Once the parameters have been initialised, we can continue constructing the simulation architecture representing our optimisation model, as depicted in Fig. 5.3. The architecture is primarily based on the structure of EACO, with the addition of an integration module. The first step of the model is to initialise a pool of randomly sampled candidate solutions, referred to as the population. We can effectively ensure feasibility by sampling the solutions within the limits defined by the provided box constraints. For a single spacecraft architecture, we define the chromosome as

$$\begin{aligned} \textit{chromosome} &= [\boldsymbol{x}_0, \mathcal{T}_C, \mathcal{U}(x, t \in \mathcal{T}_C)] \\ &= [r_{0,x}, r_{0,y}, r_{0,z}, v_0, v_{0,x}, v_{0,y}, v_{0,z}, t_1, \Gamma_1, u_{x,1}, u_{y,1}, u_{z,1}, \dots, \\ &\quad t_i, \Gamma_i, u_{x,i}, u_{y,i}, u_{z,i}, \dots, t_I, \Gamma_I, u_{x,I}, u_{y,I}, u_{z,I}], \end{aligned} \tag{5.25}$$

where $t_1, \dots, t_I = \mathcal{T}_C$ are the control times and $\Gamma(t)$ the control magnitudes corresponding to each impulsive manoeuvre $\boldsymbol{u}(t) = \Gamma(t)\hat{u}(t)$ on the set $\mathcal{U}$, $\boldsymbol{r}_0$ is the initial position and $\boldsymbol{v}_0$ the initial velocity. On that note, both the initial velocity and sequence of control vectors have been divided into parts of magnitudes and unit vectors to increase the robustness and stability of the search process. The quality of the individual chromosome is then determined using the resulting trajectory to compute a fitness score according to Eq. (3.12) and by evaluating the collision constraint in Eq. (3.13) to verify its feasibility. The algorithm then modifies the population with subordinate operations, improving each member according to an ant-based search heuristic and re-iterating the process until a maximum number of generations has been reached.

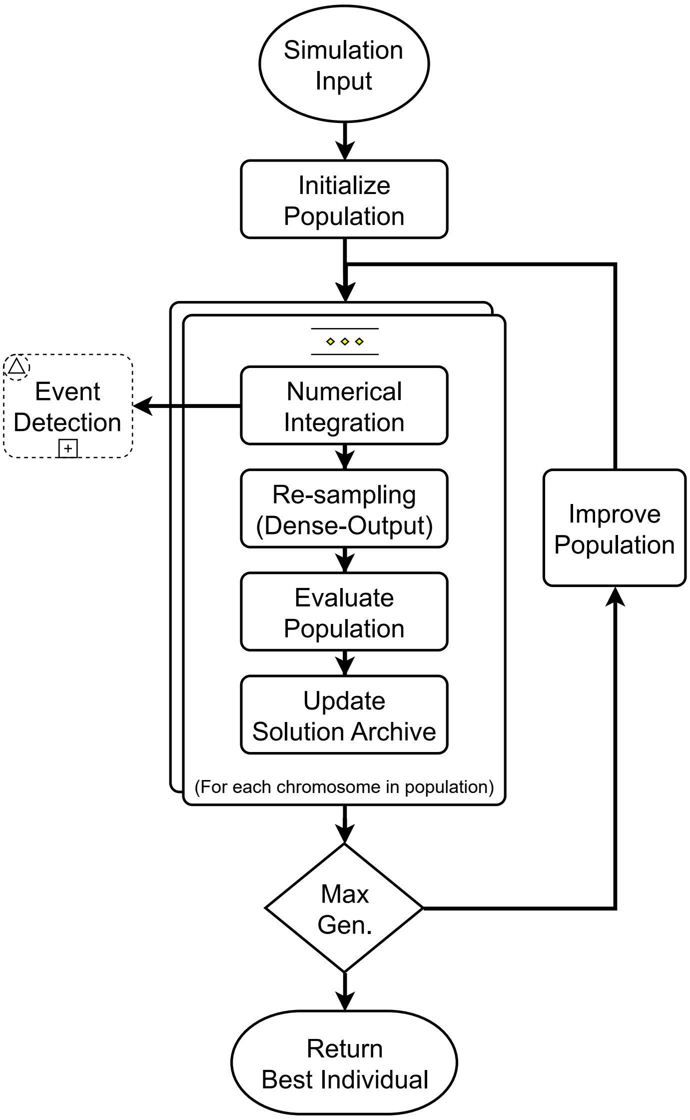


Figure 5.3: Flowchart of global optimization architecture.

5.3.2 Extended Model

To study the performance of a swarm relative to a single spacecraft, the problem is now expanded to consider optimising a set of trajectories simultaneously instead. Formally, this is done by introducing a new notation depicting a set of J active spacecraft, $\mathcal{S}_A = \{s_1, \ldots, s_j, \ldots, s_J\}$. In this work, we primarily consider the fitness function to account for the joint performance of the spacecraft constituting the swarm. As such, we impose a new condition that visiting any specific region on the longitude-latitude grid will only generate a gain for the fitness function once. Similarly, the risk of the corresponding swarm is related to the spacecraft closest to the body at any point on the J simultaneously considered trajectories. The far distance penalty will assume the total average deviation from the region of interest for all spacecraft. With these aspects in mind, the optimisation problem presented in Eq. (3.14) can be further expanded to cover the whole set $\mathcal{S}_A$ by introducing the extended chromosome

$$\begin{aligned} chromosome_{\ swarm} = \ & [x_0^1, \mathcal{T}_C^1, \mathcal{U}(x^1, t \in \mathcal{T}_C^1), \ldots, \\ & x_0^j, \mathcal{T}_C^j, \mathcal{U}(x^j, t \in \mathcal{T}_C^j), \ldots, x_0^J, \mathcal{T}_C^J, \mathcal{U}(x^J, t \in \mathcal{T}_C^J)], \end{aligned} \tag{5.26}$$

which represents a concatenation of decision vectors for each spacecraft $j \in \mathcal{S}_A$.

5.3.3 Guided Local Search

By considering an extended chromosome representation, it follows that the number of dimensions of the problem will scale accordingly with the number of considered manoeuvres and spacecraft:

$$n_{variables} = J \cdot (n_{initial\ state} + 5 \cdot n_{manouvres}). \tag{5.27}$$

To evaluate the robustness and performance of the global optimisation strategy, we will compare it to an iterative local optimisation model as illustrated in Fig. 5.4. The main distinction from the global model is the introduction of a new loop passing the longitude-latitude tensor to the population initialisation process. In detail, we approach the problem by considering the original optimisation architecture for a single spacecraft. However, once an optimal trajectory has been found, we store the best chromosome and update the weights on the tensor grid with a zero value for any visited regions. In that way, the optimisation of the subsequent spacecraft will naturally inherit the results of the previous spacecraft and actively favour measuring new regions without requiring additional path constraints. Thus, the number of variables associated with each optimisation process is instantly reduced to the number of variables in the original chromosome considering only a single spacecraft, which allows for a more manageable optimisation landscape. However, it should be noted that using the serial optimisation approach inherently removes the capability of finding the global optimal solution. For that purpose, we recall that the objective was not necessarily to find the best solution but rather a robust solution sufficient for improving the gravity field model using methods such as GeodesyNets.

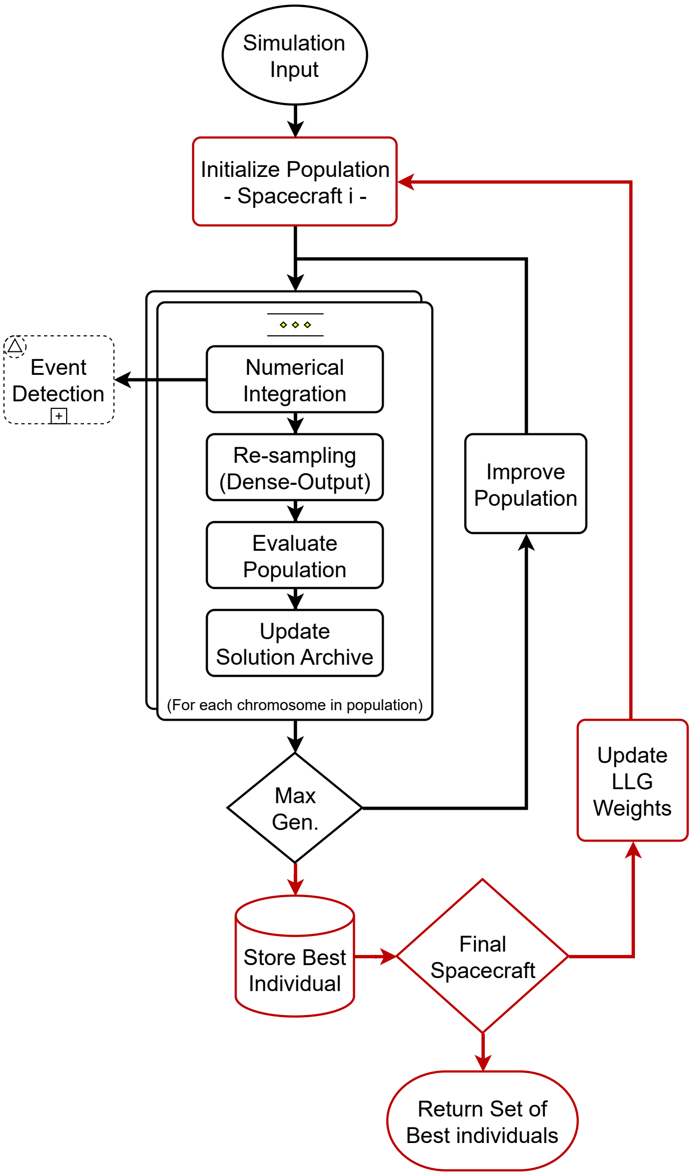


Figure 5.4: Flowchart of guided local optimization architecture.

5.4 External Software

In this work, we integrate several external software packages to build an efficient simulation model. These packages are mainly related to optimisation, quaternions, integration and the polyhedral gravity model. For that reason, we will provide a brief introduction to each software:

PyGMO

The modified extended ant colony optimiser was implemented using the PyGMO library [77]. PyGMO (Python), or PaGMO (C++), is a scientific library for massively parallel optimisation and is built around providing a unified interface to several optimisation algorithms and problems. Consequently, it makes the integration with massively parallel environments straightforward. The library has been extensively utilised for several space applications and offers numerous optimisation algorithms and benchmark problems for users to interact with [78]. PyGMO is typically used to solve constrained, unconstrained, single or multi-objective optimisation problems with either continuous or integer variables. The library is developed in a fully Free, Libre and Open-Source Software (FLOSS) philosophy, and thus allows for external contribution. However, it is actively maintained by the Advanced Concepts Team (ACT) at ESA and employs several tests and verification steps before integrating any new contributions.

DESolver

To solve the equations of motion, we have mainly used DESolver[1] which is a python package for solving initial value problems using various numerical integrators. The library was created and is primarily maintained by Ekin Ozturk, a previous researcher of the ACT at ESA, and comprises several integration routines ranging from fixed step to symplectic and adaptive integrators.

Polyhedral-Gravity-Model

To compute the gravitational attraction by the considered celestial body, we have used the Polyhedral-Gravity-Model package as created and maintained by Jonas Schuhmacher in a collaborative project between the Technical University of Munich and ACT at ESA [43]. The package implements the analytical solution for the model via the line integral approach presented by Tsoulis et al. [39] and relies on an efficient and parallelized backbone in C++, vectorizing expensive computations.

quaternion

To construct the required quaternions we have used the quaternion[2] library which is based on an original code written by Martin Ling and Mark Wiebe and is currently maintained by Mike Boyle (Cornell University) supported in part by the Sherman Fairchild Foundation and by NSF Grants No. PHY-1306125 and AST-1333129. The library remains one of the most popular choices in Python when working with quaternions.

---

[1] DEsolver library: https://github.com/Microno95/desolver (Accessed: 23-02-2023)

[2] quaternion library: https://github.com/moble/quaternion (Accessed: 08-03-2023)

5.5 Validation and Verification

In this thesis we validate the presented framework by motivating the choice of optimiser, methods and numerical techniques based on previous rigorous studies and the current problem structure. Hence eliminating the need for explicit model validation by using already verified sources for methods and optimisation strategies. Furthermore, as a step of verification, we ensure that all implemented software, including modules, algorithms and techniques, works correctly and produces the expected results. To achieve this, we introduce several test functions aimed at maintaining continuity during the development of the code base. For example, for penalty functions, alternative objectives, integration procedures, and the addition of control, we implement multiple tests to ensure that each function yields the expected trajectories and values throughout the code development process.

To ensure a correct implementation of quaternion rotations, we compare the output of ten randomly generated examples to the results of a conventional Euler-Rodrigues rotation formula assuming the rotation of a vector $\hat{\boldsymbol{v}}$ by an angle θ about a unit axis of rotation $\hat{k} = [k_x, k_y, k_z]^T$ [79]. To define the latter formula, we introduce four real-valued numbers

$$a = cos(\frac{\theta}{2}), \quad b = k_x sin(\frac{\theta}{2}), \quad c = k_y sin(\frac{\theta}{2}), \quad d = k_z sin(\frac{\theta}{2}), \tag{5.28}$$

that satisfy Euler's four-square identity. Let $\hat{\omega} = [a, b, c]^T$, we can then express the rotation of $\hat{\boldsymbol{v}}$ on the compact form

$$\hat{\boldsymbol{v}}_{rot} = \hat{\boldsymbol{v}} + 2a(\hat{\omega} \times \hat{\boldsymbol{v}}) + 2(\hat{\omega} \times (\hat{\omega} \times \hat{\boldsymbol{v}})). \tag{5.29}$$

Furthermore, to ensure a correct implementation of the tensor structure representing the longitude-latitude grid, presented in Section 3.3, and subsequent computation of coverage, we introduce three test functions. The former test ensures a consistent and functioning evaluation of large samples of positions, in this case being ten million positions. The second test ensures that evaluating all possible grid points results in a coverage equal to one. Finally, for the last test we set the weights in one of the two hemispheres representing the grid to equal zero and ensure that the computation of coverage results in either a correct marginal gain or no gain when adding a position in either hemisphere.

Chapter 6

Numerical Results

In this chapter, we present the numerical results obtained with the optimisation models proposed in Chapter 5. Specifically, we focus on solving the optimisation problem considering two settings, the former using a single spacecraft and the latter using a swarm of four spacecraft. The results are then reviewed according to several performance measures, such as the quality of measured signals through the evolution of coverage, execution time, distance to the region of interest and fitness convergence rate. To consider a more realistic mission scenario, we use the comet 67P/Churyumov-Gerasimenko as the target small body since its complex shape allows for a challenging search process to define an optimal control law.

6.1 67P/Churyumov-Gerasimenko

The comet 67P/Churyumov-Gerasimenko, often abbreviated 67P/C-G, is a Jupiter family comet believed to originate from the Kuiper Belt and follows an elliptical orbit around the Sun. The comet's aphelion (farthest point from the Sun) is approximately 5.68 astronomical units (AU), and its perihelion (closest point to the Sun) is about 1.24 AU. It primarily orbits in the same direction as the planets, that is, counterclockwise when viewing the solar system's plane from above, and with a slight inclination to the plane. The body has recently gained significant attention due to its visit by the European Space Agency's Rosetta mission between March 2004 and September 2016, which ended with a controlled collision. The purpose of the mission was mainly to perform observation of the comet's nucleus and coma by orbiting around it with the rosetta spacecraft, as well as to deploy the lander Philae, thus making the mission the first of its kind to orbit and land on a comet successfully.

The nucleus of the body consists of two lobes fused by a narrow neck, where the large lobe is of dimensions $4.10 \times 3.52 \times 1.63$ km and the small lobe of dimensions $2.50 \times 2.14 \times 1.64$ km, thus representing a relatively small comet. Furthermore, it has a mass of 1.0×10^{13} kg, a volume of 18.7 km^3 and a density of 533 kg/m^3 [80]. As 67P/C-G approaches the Sun on its highly elliptical orbit, the increasing solar radiation causes the comet's ice to sublimate, turning directly from a solid to a gas. This sublimation process creates a visible coma (a cloud of gas and dust) and a tail that always points away from the Sun due to solar winds and radiation pressure. The sublimation of different ices gives rise to various jets and features on the comet's surface, which are visible in some of the images featured in Fig. 6.1.

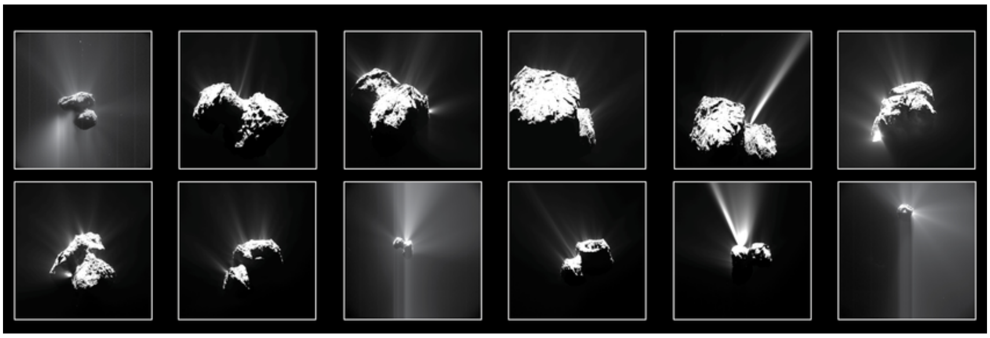
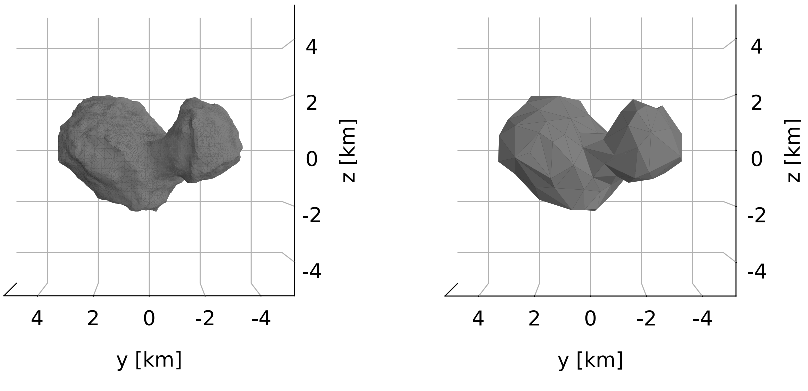


Figure 6.1: Images of 67P/Churyumov-Gerasimenko processed from the ROSETTA Mission. (Accessed: 06-09-2023, Max Planck Institute for Solar System Research, https://www.mps.mpg.de/en/Rosetta)

One of the most notable features of 67P/C-G is its non-uniform rotation compared to perfectly symmetrical objects. The temporal evolution of its rotational parameters is generally believed to be governed by its outgassing torque, irregular shape, varying mass distribution across its bi-lobate nucleus and moments of inertia, resulting in a natural precession. However, after carefully examining the data collected during the Rosetta mission, scientists have concluded that the direction of the rotation axis only changes marginally during its orbit, except for its rotation period rapidly dropping during the perihelion passage. Thus, by considering a relatively short mission duration with prograde motion near its aphelion, we can safely assume the rotation axis to be fixed in the Equatorial direction of approximately 69° in right ascension and 64° in declination. Similarly, we assume a constant rotation period of $T_B = 12.06$ [81–83]. Furthermore, to model the gravitational field using a polyhedral gravity model, we introduce polyhedral mesh consisting of v vertices resulting in f faces as presented in Fig. 6.2. In detail, we adopt a low-poly version of the mesh to minimise the computational cost of the simulation model and as a proof of concept. However, a higher resolution mesh can also be used to retrieve more precise measurements for the training of geodesyNets.



(a) High resolution mesh.

(b) Low resolution mesh (1%).

Figure 6.2: Plot of comet 67P/C-G depicting a) a high resolution mesh with 9149 vertices and 18294 faces and b) a low resolution mesh with 93 vertices and 182 faces.

6.2 Simulation Input

In this section, we present values for a standard set of hyperparameters used to initialise the considered optimisation models. For coherence, we refer to the hyperparameters as the simulation input (see Fig. 5.2). However, it should be noted that the values remain arbitrary and can later be refined depending on either simulated results or targets for assessing isolated computational properties. Furthermore, we assume 67P/C-G to be at its prograde state with a fixed rotation period when determining properties relating to the body's orientation. In addition, for the optimiser, we initially assume a standards set of parameters provided by PyGMO [77].

Table 6.1: Numerical values for the hyperparameters defined in the simulation input.

Parameter	**Description**	**Value**	**Unit**
t_0	Mission start time	0	s
t_f	Mission end time	604800	s
α	Right ascension	69	deg
δ	Declination	64	deg
T_B	Small-body rotation period	12.06	h
ρ	Small-body mass density	533	$\frac{kg}{m^3}$
n_v	Number of vertices	93	–
n_f	Number of faces	182	–
ϵ_{rel}	Relative error tolerance	$1e^{-12}$	–
ϵ_{abs}	Absolute error tolerance	$1e^{-12}$	–
Δt	Measurement period	100	s
K	Solution archive size	63	–
q	Convergence speed parameter	1.0	–
Ω	Oracle parameter	0	–
α	Accuracy parameter	0.01	–
$threshold$	Threshold parameter	1	–
$NGenMark$	q-Standard deviation parameter	7	–
$focus$	Focus parameter	0	–
$memory$	Memory parameter	$false$	–
n_{pop}	Population size	200	–
n_G	Number of generations	1000	–
$n_{threads}$	Number of threads	200	–

Problem Definition

For the nonlinear programming (NLP) module, we assume the two penalty scaling factors to equal $\lambda_{\mathcal{I}} = \lambda_{\mathcal{O}} = 0.1$ in order to ensure a well-balanced cost function. Furthermore, we solve the original problem three consecutive times assuming a single spacecraft, several fixed initial states and no manoeuvres to study reasonable magnitude domains for the initial velocity and impulsive manoeuvres. Due to the relatively low mass of 67P/C-G, the spacecraft rarely reached a velocity above 1 m/s. Hence, we assume a fixed domain of $u \in [0, 2.5]$ m/s and $v_0 \in [0, 4.5]$ to ensure a stable initial trajectory. The chosen ranges are notably comparable to that of a CubeSat, which could represent a feasible assumption on a vehicle architecture for the prospective mission [14, 15].

Longitude-Latitude Grid

In the longitude-latitude grid (LLG) module, the feasible region $\mathcal{S}_f$ is defined, as illustrated in Fig. 6.3, with fixed radii of $r_{\mathcal{I}} = 4$ km and $r_{\mathcal{O}} = 12.5$ km, roughly 0.6 km and 9.1 km above the comet's surface, respectively. These values were primarily chosen for continuity with previous work on the robustness of GeodesyNets [31].

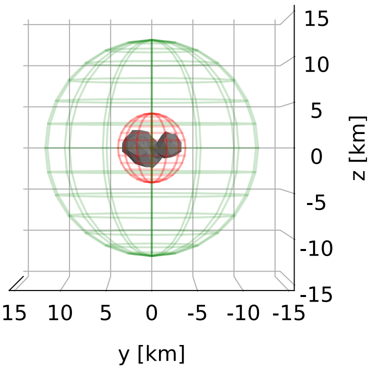


Figure 6.3: A feasible region $\mathcal{S}_f$ around 67P/C-G.

Furthermore, to choose a sufficiently large v_{max}, we recall that its magnitude should ensure that the CFL condition holds for the considered LLG. Since the dynamics increase near the body's surface, where the gravitational signal is considered maximal, the greatest velocity should naturally occur near the radius of the inner bounding sphere. To obtain a feasible estimate on v_{max}, we sample the largest velocity that occurs on a trajectory that minimises a smooth function representing the average squared distance to the safety radius:

$$J(P_T) = \left[\frac{1}{r_{\mathcal{I}}^2} \frac{1}{||P_T||} \sum_{r \in P_T} |r^2 - r_{\mathcal{I}}^2| \right]^{\frac{1}{4}}. \tag{6.1}$$

The new sub-problem was solved assuming a shorter mission duration of only 12 hours, or 43200 s, a variable initial position $x_0 \in \mathcal{S}_f$ and a singular spacecraft with no impulsive manoeuvres resulting in maximal velocity of $v_{max} \approx 1.14$ m/s. However, to guarantee that the problem maintains the CFL condition even with the addition of manoeuvres and to avoid defining a too-sparse LLG, we initialise v_{max} as

$$v_{max} = 2 \cdot (v_{max,\ sample} + u_{max}). \tag{6.2}$$

6.3 Scaling Analysis

Many parallel applications can operate with various thread counts, yet the specific application's relative performance may also decline with an increasing numbers of thread. For instance, while a calculation might experience a nearly twofold speedup as the thread count doubles, another application might deviate more extensively from the anticipated speed increase. Such deviations from expected results are common, implying the necessity of carefully deciding on the number of threads to use for the simulation. Since the appropriate thread count may vary between large and small applications, measuring the relative performance for several problem sizes is essential. In this case, since the parallelisation directly relates to the number of chromosomes that can be evaluated in parallel, where a larger population enables a broader search, it is of interest to assess the simulation model's scaling performance. Hence, we will consider two conventional measurements of scalability: weak and strong scaling. Before assessing any of these measurements, we introduce the two concepts of ideal performance and efficiency. The ideal run time on n threads, I_n, is defined as

$$I_n = \frac{T_{n_{min}} \cdot n_{min}}{n}, \tag{6.3}$$

where $T_{n_{min}}$ is the measured run time at the lowest thread count n_{min}. The ideal computation time can then be used to evaluate the efficiency, E_n, as the ratio of the measured run time using n threads to the expected ideal time:

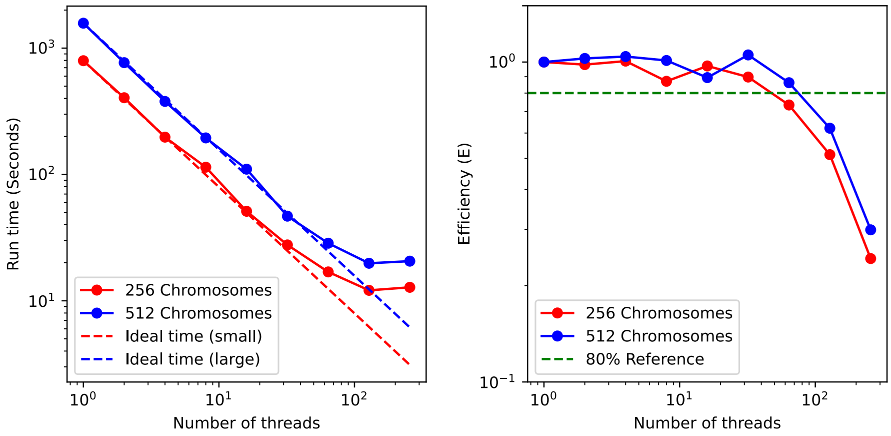
$$E_n = \frac{I_n}{T_n}. \tag{6.4}$$

For that reason, the efficiency can be used to determine an appropriate number of threads for the problem. As a general rule of thumb, we aim at achieving a minimum of 80% efficiency. Since the serial part of the simulation only accounts for a fraction of the total simulation time, we will assume it to be negligible relative to the parallel part and continue with a predominant focus on the latter. Furthermore, in this work, we perform all simulations and tests on two AMD EPYC 7702 CPUs at 2.0 GHz with a total of 128 cores and 256 threads.

6.3.1 Strong Scaling

Strong scaling refers to the simulation's performance when the problem size is kept fixed and the number of utilised threads varies. For this test, we consider the simulation input defined in Section 6.2 and formulate a problem considering a single spacecraft with one manoeuvre, an optimised initial state, a total mission duration of 12 hours and an optimisation process over ten generations. The test problem is then evaluated twice considering a smaller and a larger population size of 256 and 512 chromosomes, respectively.

The results show a nearly ideal strong scaling given how the number of simulated trajectories directly relates to the number of chromosomes, where each thread is assigned a single chromosome at the time. Consequently, resulting in a near-linear speedup as observed in Fig. B.1a. In addition, the two problem sizes also display remarkably similar results. However, whenever the number of threads approaches the population size, the computational time per generation becomes notably more sensitive to differences in integration time between chromosomes, resulting in the deviation seen in Fig. 6.4 for larger thread counts. The varying integration time is a direct consequence of the adaptive step method, which will naturally take smaller steps near the body where the dynamics are higher.

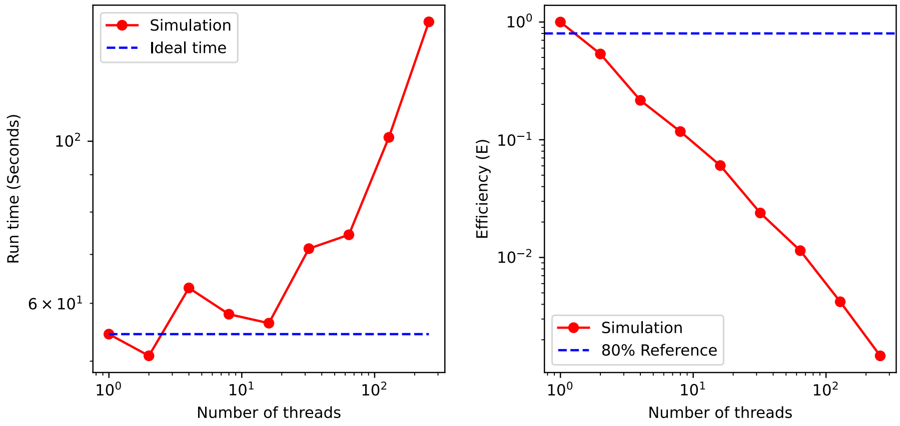


(a) Strong scaling run time. (b) Strong scaling efficiency.

Figure 6.4: Strong scaling test considering two test cases with 256 and 512 chromosomes.

6.3.2 Weak Scaling

To complement the strong scaling analysis, we consider a weak scaling test where the problem size increases relative to the number of threads. Consequently, we can study the effects of communication overhead when considering a larger problem size and thread count. In detail, we consider the following population sizes $n_{pop} \in \{20, 40, 80, 160, 320, 640, 1280, 2560, 5120\}$. Finally, it should be noted that the notion of ideal time in weak scaling directly relates to the fixed value of required computational time when using the lowest possible thread count.



(a) Weak scaling run time. (b) Weak scaling efficiency.

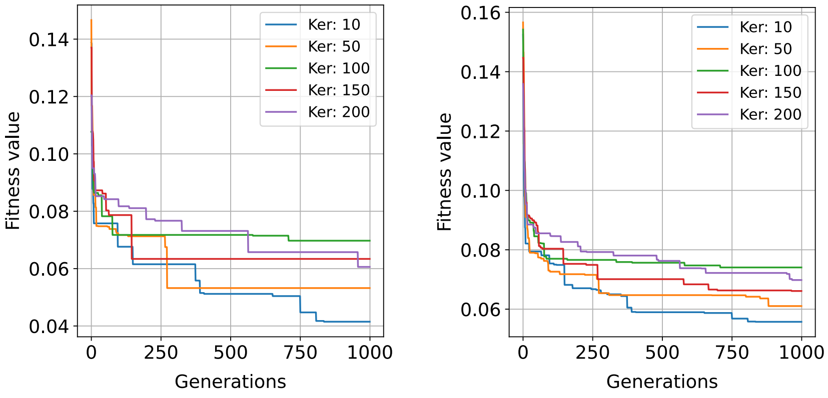
Figure 6.5: Weak scaling test.

The results presented in Fig. 6.5 show an expected near-linear decrease in efficiency and an approximately exponential increase in simulation time. However, judging by the results from both Fig. 6.4 and 6.5, it is evident that the computational performance exhibits a trade-off between efficiency and chromosome count. Since a larger population size favours exploration, we will henceforth use a population size of 200.

6.4 EACO Configuration

Until now, we have assumed a principal implementation of the modified EACO considering a standard set of input parameters provided by PyGMO. In this section, we will focus on a tuning process to adapt the optimiser for the specific problem, thus enabling a more robust search process. Although we will later consider a varying number of spacecraft and manoeuvres, we assume an initial broad tuning process to be sufficient for evaluating the optimisation models' general performance. Since the optimiser has moderately few hyperparameters, we proceed with a parameter-sweep tuning strategy to pinpoint the contribution of each parameter [62]. In detail, we solve the problem using the minimal optimisation model presented in Section 5.3.1 for a mission scenario with a single spacecraft and one manoeuvre. The procedure is repeated three consecutive times for each hyperparameter to compare its best and average performance. We implement control seeds for each simulation to omit randomness from the initial population and guarantee both comparable and reproducible results.

In Section 6.2, we introduced several input parameters that could be adjusted to tweak the modified EACO. Some pertain to the width of the search process (ker, α), whereas others relate to its convergence properties (q, $NGenMark$, $Threshold$). To begin with, we recall that the $focus$ parameter relates to the greediness of the search using local refinement. Selecting a larger value thus implies an increased focus on exploring the local environment for the current best solutions. For a high-dimensional problem, the focus parameter must be tuned cautiously to ensure a broad initial search and to avoid converging prematurely at any local optima. Hence, we assume a standard value of zero. Furthermore, with particularly complex dynamics characterising the considered problem, obtaining an intuition for any optimal solutions becomes increasingly challenging. Therefore, we choose a sufficiently large oracle parameter, $\Omega = 10^9$, as suggested by Schlüter [59].

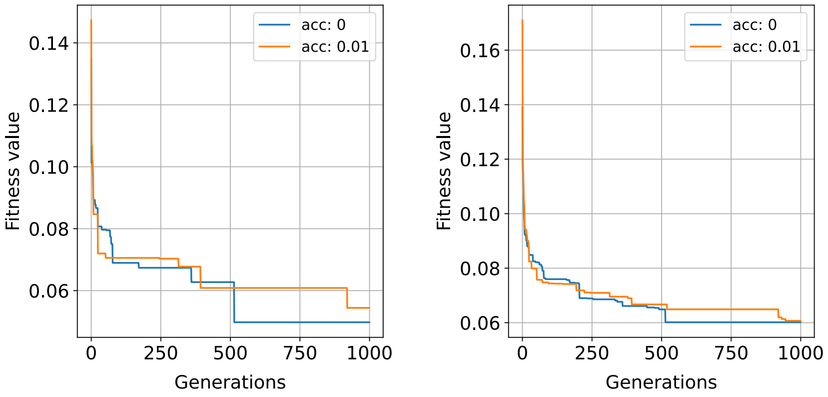


(a) Best fitness.

(b) Average fitness.

Figure 6.6: Best and Average fitness for five different solution archive sizes considering 200 chromosomes and a 1000 Generations.

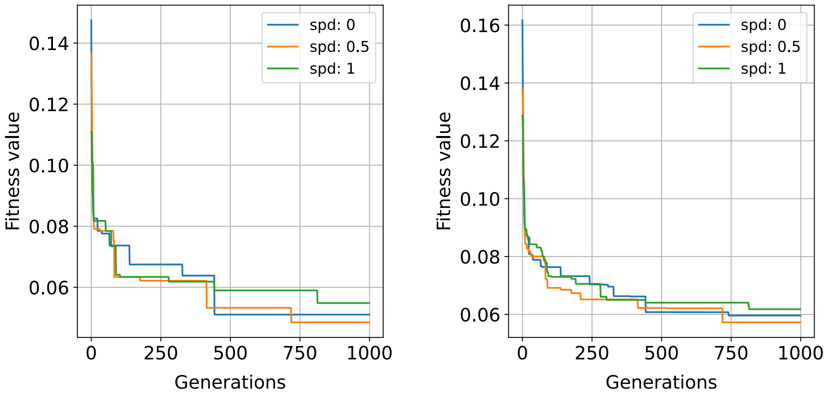
For tuning the number of kernels, K, we benchmark for the values 10, 50, 100, 150 and 200. Reviewing the results in Fig. 6.6, we found that the best and average solutions were achieved with relatively few kernels. However, establishing a specific winner is not straightforward. Since a lower value promotes the exploitation of current solutions, it follows naturally that such values would result in a faster convergence rate. To address the challenges posed by more complex optimisation landscapes involving a greater number of spacecraft and manoeuvres, we prioritise exploration and therefore choose K to equal the population size.



(a) Best fitness. (b) Average fitness.

Figure 6.7: Best and Average fitness for two different values of accuracy considering 200 chromosomes and a 1000 Generations.

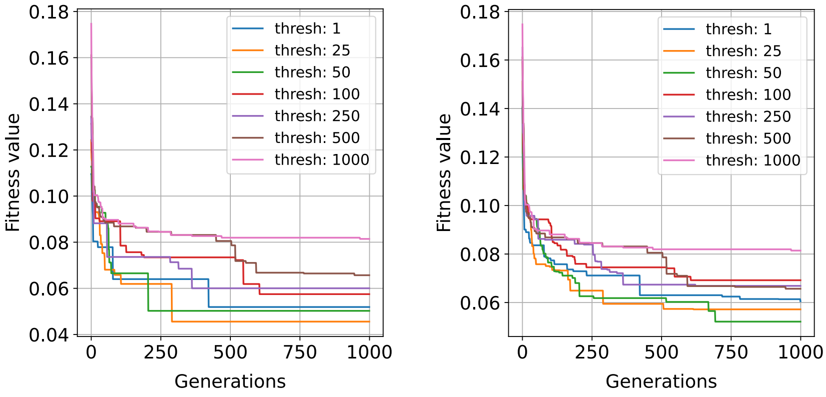
For tuning the accuracy parameter, α, we analyse whether or not it would be beneficial to maintain a distance between the penalty functions in the solution archive by comparing $\alpha = 0$ and $\alpha = 0.01$. We can directly identify a similar convergence behaviour for the two examples without any evident benefit of the larger value. Thus, we favour the former value's quicker convergence and marginally improved solution according to the results presented in Fig. 6.7.



(a) Best fitness. (b) Average fitness.

Figure 6.8: Best and Average fitness for three different values of convergence speed considering 200 chromosomes and a 1000 Generations.

Next, we focus on the convergence speed parameter q and compare results for values of 0, 0.5 and 1. As seen in Fig. 6.8, we obtain comparable performance for each value. However, given that the optimiser will dynamically reset to $q = 0.01$ once it reaches the threshold value, consequently accelerating the convergence from there on, we favour a slower initial convergence rate to balance exploration and exploitation. Thus, we select $q = 0.5$.

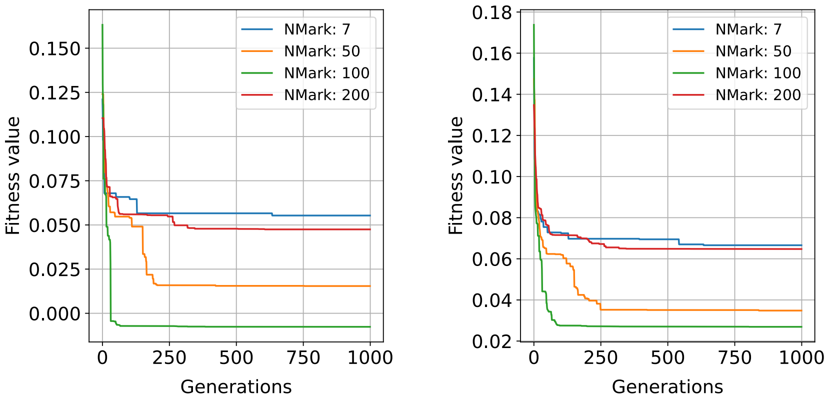


(a) Best fitness.

(b) Average fitness.

Figure 6.9: Best and Average fitness for seven different values of threshold considering 200 chromosomes and a 1000 Generations.

In this work, we study several threshold values ranging from 1 to 1000 generations. As seen in Fig. 6.9, the optimiser converges first for smaller values, which is the expected outcome of increasing the convergence speed at an earlier stage. However, among the larger values, 250 generations seem to provide comparable performance while maintaining an initial broad search. Therefore, we select it as the standard threshold value.



(a) Best fitness.

(b) Average fitness.

Figure 6.10: Best and Average fitness for four different values of NGenMark considering 200 chromosomes and a 1000 Generations.

Finally, we study the effects of $NGenMark$. Recall that the goal of the parameter is to ensure that the standard deviations do not converge too quickly towards the explicit means in the solution archive. Among the four suggested values in Fig. 6.10, $NGenMark = 100$ generated the best solution and, notably, the best fitness throughout the tuning process. As such, we select it as the standard deviation convergence speed value for all subsequent simulations.

6.5 Optimisation Scenarios

In this section, we present the numerical results obtained from three different optimisation scenarios using the models presented in Section 5.3, including a global search for both a single spacecraft and a swarm, and a complementary local search for the latter case. To ensure comparability between each example, we consider the simulation input presented in Section 6.2 and the tuning results in from Section 6.4.

6.5.1 Scenario I: Single Spacecraft

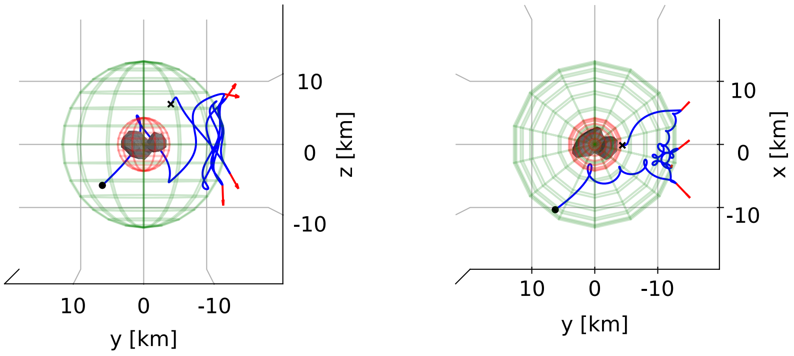
For the first scenario, we simulate a single spacecraft deployed at a certain position defined within the region of interest, $\mathcal{S}_f$. To identify a reasonable starting position, we set up an initial test case considering a single spacecraft with no impulsive manoeuvres. The problem was then solved for numerous starting positions to identify any resulting near-periodic or partially stable trajectories without added control. From these results, we proceed under the assumption of a fixed initial position at $\boldsymbol{r}_0 = [-135, -4090, 6050]^T$. To analyse the effects of the control and an increasing problem dimension, we simulate the first mission scenario for several manoeuvres ranging from one to ten. However, it should be noted that even if we permit ten manoeuvres, the corresponding box constraints still allow for the optimiser to choose any smaller number if it yields a better solution. Consequently, there is an evident trade-off relation between the disposable number of manoeuvres and the problem size. The results corresponding to the first scenario can be seen in Table 6.2.

Table 6.2: Numerical results for a single spacecraft with varying number of manoeuvres n_m.

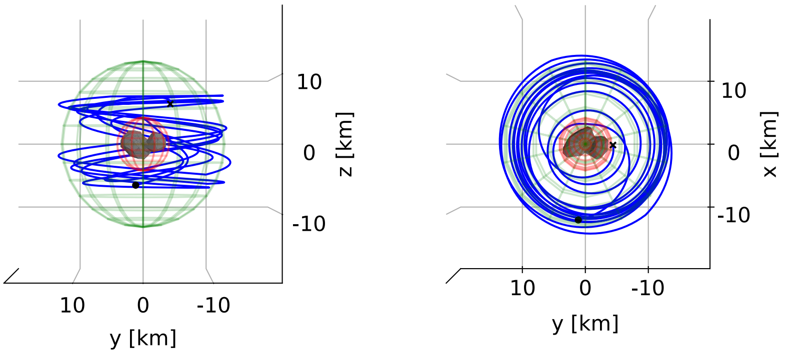
n_m	t_{sim} [min]	f_{best}	V_c	$\lambda_{\mathcal{I}} P_{\mathcal{I}}$	$\lambda_{\mathcal{O}} P_{\mathcal{O}}$	dim	Δv [m/s]
1	48.19	0.0608	0.026	0	0.086	9	0.055
2	53.49	0.0423	0.041	0	0.083	14	0.149
4	61.09	0.0109	0.051	0	0.062	24	0.479
6	64.03	0.0189	0.043	0	0.062	34	1.358
8	67.67	0.0104	0.058	0	0.069	44	2.137
10	67.64	-0.0167	0.071	0	0.055	54	3.263

From Table 6.2, we note that increasing the number of manoeuvres generally induces longer simulation times t_{sim} and an improved fitness f_{best} (smaller fitness value is better). Another interesting development is the total consumed Δv, which increases nearly exponentially when considering adding two control points. Nevertheless, no spacecraft used more than 13% of their disposable Δv. Studying the control distribution in detail, we observe that the optimiser favours minor correction manoeuvres when considering more control points. As a result, the spacecraft starts to bounce between the two spherical boundaries to ensure positions inside $\mathcal{S}_f$. However, as the number of dimensions grows, we can also identify a pattern demonstrating the complexity of finding optimal solutions in a high-dimensional space. This effect becomes evident when reviewing the evolution of coverage and penalty values, which vary more unpredictably for larger problem sizes. To infer general system characteristics, one should therefore consider simulating each scenario multiple times with controlled population seeds to avoid randomness and isolate averages. For future work, we suggest performing a correlation analysis and investigating the robustness of the model by a sensitivity analysis using, for instance, a large scale Monte-Carlo search.

To focus the analysis, we use the example considering four manoeuvres. For this case, the optimisation process converged for a tolerance level of 1e-6 after 289 generations, resulting in the trajectory observed in Fig. 6.11. Throughout the mission duration, the spacecraft visited 399 regions out of 6358 defined on the LLG, resulting in a ratio of 6.27%. In Figs. 6.11a and 6.11b, each manoeuvre is represented by a red arrow in the thrust direction. Immediately, it is noticeable that even though the spacecraft occasionally deviates from $\mathcal{S}_f$, it tends to use its impulsive manoeuvres at its most distant positions to redirect its path towards the body. However, this behaviour is expected given the weak gravitational field at these distances, resulting in particularly linear dynamics. The spacecraft can also be seen following a near zig-zagging course in the vertical plane, which is likely the result of the rotating motion of the body, enabling a hoovering-like strategy to cover a significant part of the LLG. Figs. 6.11c and 6.11d present the corresponding trajectory in the body-fixed frame, which follows the expected circular motion around the body.



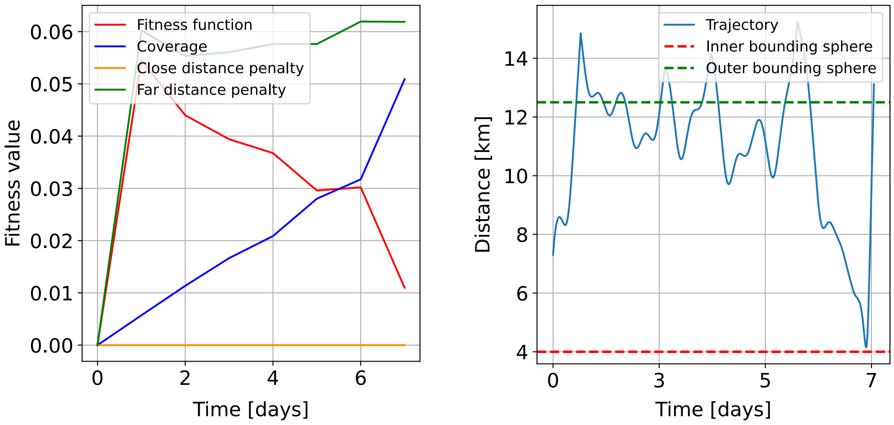
(a) Inertial frame (side view). (b) Inertial frame (top view).



(c) Body-fixed frame (side view). (d) Body-fixed frame (top view).

Figure 6.11: Single spacecraft optimisation results with four manoeuvres depicting a)-b) side and top view of the trajectory in the inertial frame and c)-d) side and top view of the resulting trajectory in the body-fixed frame. (Deployment at the cross).

Recall that minimising the fitness function defined in Eq. (3.12) maximises the coverage and minimises the two penalties. Fig. 6.12a illustrates that the close distance penalty remains at zero, indicating that the spacecraft never enters the inner bounding sphere. The coverage increases monotonically over the mission duration, with only brief pauses whenever the spacecraft exits the region of interest. A similar behaviour can also be seen in Fig. 6.12b, which shows a resonating distance evolution to the comet's centre of mass. Furthermore, the far distance penalty stagnates at a moderately low value after five days, indicating that the spacecraft maintains a close but steady distance to the body over the mission duration. In detail, by comparing with the distance evolution in Fig. 6.12b, we can identify that the spacecraft tends to move in proximity to the radius of the outer bounding sphere and rarely toward the inner parts of the LLG, except for the last day.



(a) Evolution of fitness components.

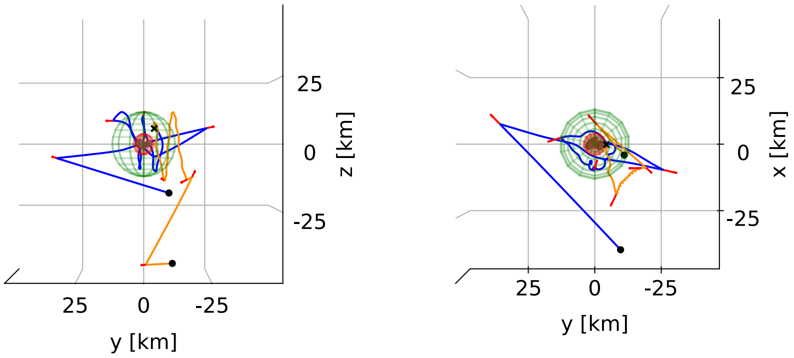
(b) Distance to origin.

Figure 6.12: Single spacecraft optimization results depicting a) evolution of fitness components over the mission duration where the two penalties are minimised and the coverage is maximised, and b) distance from the body's centre of mass over time.

6.5.2 Scenario II: Swarming Spacecraft

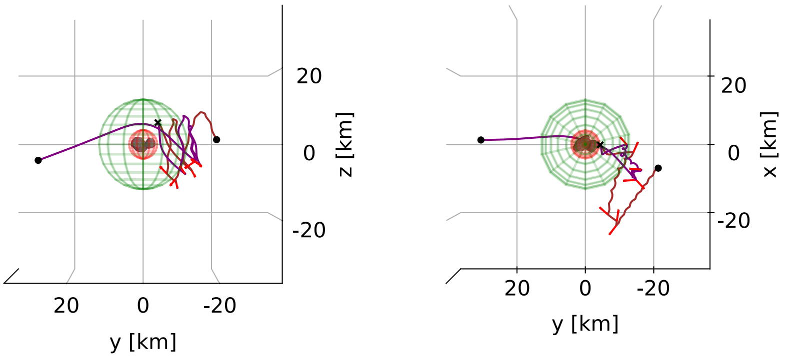
For the second scenario, we consider an equivalent setup to Section 6.5.1 with an initial position at $\boldsymbol{r}_0 = [-135, -4090, 6050]^T$. However, we will now expand the problem by considering a global search for the optimal trajectories corresponding to a swarm, thus adopting the extended chromosome structure presented in Eq. (5.26). Furthermore, we select one of the examples previously considered in the first scenario to ensure comparability between each optimisation model. In detail, to keep the number of dimensions reasonable, we use a setup of four spacecraft with four manoeuvres each, resulting in a chromosome with a total of 96 decision variables.

The simulation finished after 186.69 minutes and converged for an acceptable tolerance level of 1e-6 after 526 generations. Compared to the single spacecraft case, optimising for a swarm resulted in a 24% faster per spacecraft simulation time. The cause for this reduction is likely twofold. First, since each thread is now considering solving the equations of motion for four spacecraft consecutively, it should effectively reduce the total per spacecraft overhead communication between threads. The second cause is likely related to the integration time, where trajectories further away from the body generally result in larger step sizes to account for the near-linear dynamics. In terms of the quality of the solution, the swarm achieved a total fitness value of $f_{best} = 0.0154$ with $P_{\mathcal{I}} = 0$, $P_{\mathcal{O}} = 0.1098$ and $V_c = 0.0945$. When compared with the single spacecraft case, it is evident that the swarm obtained a worse total fitness due to the increased far distance penalty now considering the deviations of all four spacecraft at once. The coverage, however, remarkably achieved only slightly less than twice the single spacecraft score, which was unexpected. When considering four times the number of spacecraft, the relative decline in coverage is likely due to the more complex search space characterised by 96 dimensions, resulting in a more challenging combinatorial problem. However, throughout the mission duration, the four spacecraft jointly visited 702 regions out of 6358 defined on the LLG, resulting in a coverage ratio of approximately 11%, almost twice that of a single spacecraft in Section 6.5.1 which yielded a ratio of 6.27%.



(a) Spacecraft 1 & 2 (side view).

(b) Spacecraft 1 & 2 (top view).

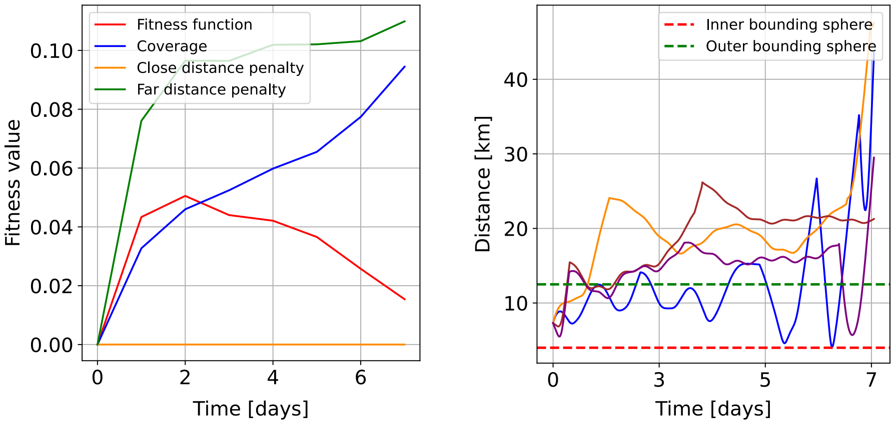


(c) Spacecraft 3 & 4 (side view).

(d) Spacecraft 3 & 4 (top view).

Figure 6.13: Multi-spacecraft optimisation results depicting a) - b) resulting trajectories for Spacecraft 1 and 2 in the inertial reference frame, and c)-d) the corresponding representation for Spacecraft 3 and 4. (Deployment at the cross).

Fig. 6.13 illustrates the four trajectories corresponding to each spacecraft in the swarm. Immediately, it is noticeable that the four spacecraft tend to travel further away from the body than in the case of optimising only a single trajectory. In detail, both Spacecraft 1, as depicted in blue, and Spacecraft 2, in orange, seem to utilise their impulsive manoeuvres with the target of returning to the body from distant positions. Although neither succeeds after their last manoeuvre, it is likely due to the particularly weak gravitational attraction at these distances. Spacecraft 3, in brown, and 4, in purple, seem to follow a similar pattern without deviating to an equally large extent at the end of their respective trajectories. However, by studying the evolution of radial distances to the body's centre of mass presented in Fig. 6.13, one can gain a more detailed insight into the performance of each trajectory. For instance, only the first spacecraft successfully remains within $\mathcal{S}_f$ after the second day, generating coverage for most of the mission duration. Likewise, only the fourth spacecraft successfully returns to $\mathcal{S}_f$ during the last day of the mission.



(a) Evolution of fitness components.

(b) Distance to origin.

Figure 6.14: Global Multi-spacecraft optimisation results depicting a) evolution of fitness components over the mission duration where the two penalties are minimised and the coverage is maximised and b) distance from the body's centre of mass over time.

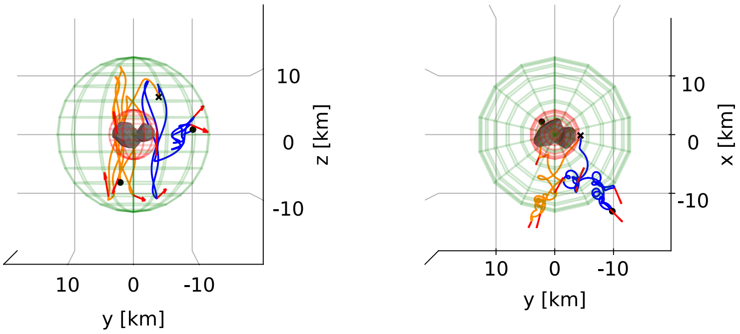
In Fig. 6.14a, we observe that $\lambda_{\mathcal{I}} P_{\mathcal{I}}$ remains at zero, indicating the absence of collisions, $\lambda_{\mathcal{O}} P_{\mathcal{O}}$ changes only marginally after day four, and that V_c increases almost monotonically over the entire mission duration. Furthermore, we also observe that the swarm achieved comparable coverage to that of a single spacecraft after only three days. Finally, Table 6.3 presents the distribution of Δv across each spacecraft and the corresponding set of manoeuvres.

Table 6.3: Distribution of Δv [m/s] for each spacecraft in a swarm (extended model).

$\mathcal{S}_A$	Δv_1	Δv_2	Δv_3	Δv_4	$\sum \Delta v$
1	2.303	0.016	0.918	0.233	3.470
2	0.052	0.979	0.760	2.739	4.530
3	0.095	0.074	1.292	0.292	1.753
4	0.889	1.733	0.006	0.032	2.660

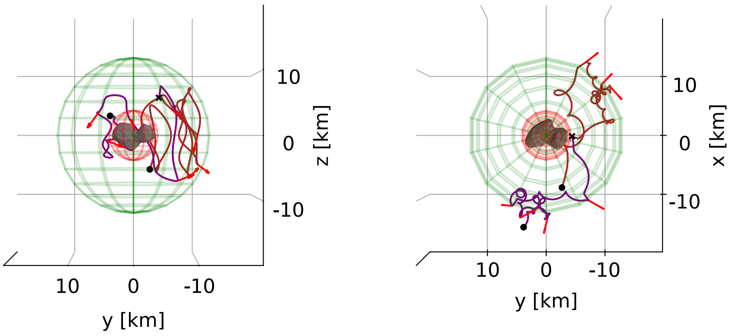
6.5.3 Scenario III: Guided Local Search

For the last scenario, we assume an equivalent setup to that of Section 6.5.2, including simultaneous deployment from position $\boldsymbol{r}_0 = [-135, -4090, 6050]^T$ using four spacecraft and four disposable manoeuvres each. In contrast to Scenario II, we consider the local search model defined in Section 5.3.3. The optimisation process finished after 246.10 minutes and converged for an acceptable tolerance level of 1e-6 after 175 generations. We identify an increased simulation time of nearly 31% compared to the global approach in Section 6.5.2, consequently almost proportional to the results presented for a single spacecraft in Section 6.5.1. The prolonged simulation time is likely due to overhead communication between threads, which are now solving a single trajectory at a time, resulting in greater sensitivity to solutions characterised by complex dynamics.



(a) Spacecraft 1 & 2 (side view).

(b) Spacecraft 1 & 2 (top view).



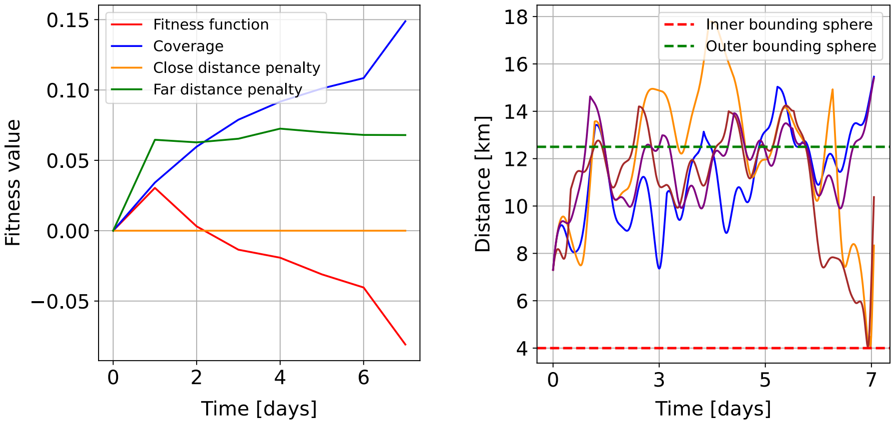
(c) Spacecraft 3 & 4 (side view).

(d) Spacecraft 3 & 4 (top view).

Figure 6.15: Local Multi-spacecraft optimisation results depicting a)-b) resulting trajectories for Spacecraft 1 and 2 in the inertial reference frame, and c)-d) the corresponding representation for Spacecraft 3 and 4. (Deployment at the cross).

In Fig. 6.15, we observe a general improvement over the global approach, with each trajectory now appearing to maintain a closer distance to $\mathcal{S}_f$ over the mission duration. In detail, Spacecraft 1, as depicted in blue, generally employs a strategy similar to hoovering in the inertial frame. In contrast, the second spacecraft, in orange, utilises its manoeuvres to achieve a resonating trajectory in the vertical plane. Spacecraft 3 (in brown) and 4 (in purple) follow a similar pattern, with shorter travel distances in the vertical plane. Nevertheless, to maximise coverage, it follows naturally to favour these characteristics as they take advantage of the rotating motion of the body.

In terms of the quality of the solution, the swarm achieved a total fitness value of $f_{best} = -0.0809$ with $P_{\mathcal{I}} = 0$, $P_{\mathcal{O}} = 0.0679$ and $V_c = 0.1488$. When comparing the fitness performance to previous models, it becomes evident that the local approach achieved an overall improved score, with only the far distance penalty being comparable to that of a single spacecraft. The increased performance is likely a result of the problem size now being reduced from 96 variables to only 24 for each iteration, resulting in a more manageable problem space. Throughout the mission duration, the four spacecraft jointly visited a total of 1193 regions out of 6358 defined on the LLG, resulting in a ratio of approximately 19%, almost three times that of a single spacecraft and slightly less than twice the performance of the global solution for a swarm.



(a) Evolution of fitness components.

(b) Distance to origin.

Figure 6.16: Local Multi-spacecraft optimisation results depicting a) evolution of fitness components over the mission duration where the two penalties are minimised and the coverage is maximised and b) distance from the body's centre of mass over time.

Furthermore, we observe a smaller far-distance penalty compared to the global solution considering a swarm by studying the fitness evolution in Fig. 6.16a. However, these results are expected given how the local solution is able to maintain a general proximity to $\mathcal{S}_f$. Similarly, we acknowledge how the coverage surpasses the performance of a single spacecraft already before the mission's second day, thus scaling well with the now four times as many spacecraft. In addition, the local solution also surpasses the global solution in Section 6.5.2 during the fifth day.

Studying the evolution of the radial distance to the body's centre of mass, we observe that only the second and third spacecraft successfully reach the inner parts of the LLG, with all four spacecraft mainly considering the outer regions of the grid during the first six days. However, this behaviour is expected due to the dynamics being more complex near the body's surface and the general sparsity of LLG points near the outer bounding sphere. As such, most spacecraft can efficiently measure the inner regions in a shorter time, given the general LLG structure and conservation of angular momentum.

Table 6.4: Distribution of Δv [m/s] for each spacecraft in a swarm (local search).

$\mathcal{S}_A$	Δv_1	Δv_2	Δv_3	Δv_4	$\sum \Delta v$
1	0.104	0.095	0.026	0.012	0.237
2	0.061	0.036	0.417	1.230	1.744
3	0.092	0.220	0.008	0.274	0.594
4	0.024	0.005	0.136	0.067	0.232

Finally, the Δv distribution for each spacecraft is presented in Table 6.4. From these results, we observe that the local search converged to a solution requiring less Δv for each spacecraft compared to the solution obtained in Scenario II using a global search. In this case, the smaller control magnitudes represent marginal adjustments and reorientation of the vehicle rather than high-energy redirection manoeuvres, which are required when deviating too far from the region of interest. For the sake of coherence, Table 6.5 presents a comparison of key performance markers for each considered scenario, including problem dimension dim, simulation time t_{sim}, total fitness value f_{best}, ratio of visited grid points, and total consumed Δv. In addition, it should be noted that the maximal amount of control, Δv_{max}, is 10 m/s for Scenario I, considering only one spacecraft, and 40 m/s for the swarm consisting of four spacecraft in Scenario II and III.

Table 6.5: Comparison of key performance markers for each considered scenario.

Scenario	dim	t_{sim} [min]	f_{best}	*Coverage ratio* [%]	Δv_{tot} $[m/s]$
I	24	61.09	0.0109	6.27	0.479
II	96	186.69	0.0154	11.04	12.413
III	96	246.10	-0.0809	18.76	2.807

Chapter 7

Discussion

In this chapter, we present a concise discussion and analysis of the diverse properties characterising the studied optimisation models and the obtained numerical results. The first section analyses several method considerations and their validity for modelling coverage and swarming spacecraft. The second section assesses the model's computational properties and performance by studying convergence rates, execution times and scalability. Finally, the third section discusses the efficiency and usefulness of a swarm compared to a conventional single spacecraft setting.

7.1 Method Considerations

In this work, we present the formulation of a trajectory optimisation problem that results in an initial state and control sequence maximising the measured gravitational signal around an irregularly shaped rotating body when considering either a single spacecraft or a swarm. In detail, we assume a discrete approximation of a bounded space around the body to model the coverage of its gravitational field as the ratio of visited partitions. Using discrete spherical grids for these applications is typically beneficial since they provide a comprehensive and efficient data structure for managing the numerical properties of each resulting tesseroid representing the partition. However, discrete grids also require tuning in order to be sufficient for numerical simulations. For instance, defining a too-sparse grid will likely result in inaccurate simulations, whereas a too-dense grid would become computationally expensive. To adapt the approximation for the specific dynamics characterising the problem, we use the CFL condition and information gained from an initial test presented in Section 6.2. Nevertheless, the coverage is an inherently continuous process and would thus be more accurately defined by the numerical integration of some continuous function. As such, one should also consider the trade-off between accuracy and computational efficiency for a discrete and continuous approach when reviewing the numerical results.

Moreover, to improve the representation of the quality of the measured signal, we assign a weight to each grid point defined by the inverse relation of its radial distance to the body's centre of mass. However, this definition merely simplifies the measurements where we assume an inner homogeneous distribution of gravitational signal and that a single visit at any point inside the tesseroid is sufficient for modelling the specific region. Hence, a new question emerges: to which size of a tesseroid can we feasibly assume a single measurement to be adequate? This is a complex question given the limited knowledge of the body's geodetic properties during the approaching stages of a small-body mission. However, one extension could consider an additional weighting depending on the distance from the spacecraft to the centre point of the tesseroid, which is well-defined along the trajectory, and the number of measurements therein. In that way, one can effectively relate the gravitational coverage to the time spent inside a region and the volumetric quality of the measurement. Nevertheless, studying the numerical results and the presented trajectories in particular (see Section 6.5), we note that the current weighting results in a good dispersion of measurement when considering the rotation of the grid. However, due to the particularly weak gravitational field of 67P/Churyumov-Gerasimenko, it becomes increasingly challenging to maintain positions near the body, resulting in the observed tendency for defining trajectories in the outer regions of $\mathcal{S}_f$. To motivate selecting paths closer to the body, one could also consider studying the results of varying the weighting more intensively according to the importance of each region.

Focusing on the grid structure, we apply a conventional longitude-latitude grid (LLG) and motivate its validity in Section 3.3. To complete the analysis, we will also briefly discuss its limitations. For instance, as seen in Fig. 3.7, the LLG will inherently define smaller tesseroids near its poles, thus creating denser regions with diminutive weights and disproportionate sizes compared to neighbouring tesseroids close to the outer bounding sphere. On the other hand, given the objective of maximising the gravitational signal, the optimiser will intrinsically favour trajectories that visit as many regions as possible. Similarly, it prioritises positions near the body where the CFL condition is guaranteed and the signal approaches ground truth. In fact, by studying the numerical results in Section 6.5, we cannot distinguish any peculiar behaviour near the two poles along the z-axis, thus showcasing the minimal effects of the size inconsistency. However, to ensure that the CFL condition holds everywhere and to define a more realistic order of partitions, one could also consider alternative grid structures such as cubed-spheres and Yin-Yang-Zhong overset grids, which share many qualities with the LLG without introducing additional complexity.

In order to obtain trajectories, we solve the equation of motion using the Dormand-Prince 8(7)-13M numerical integration scheme. The major benefits of the algorithm are its general robustness, ability to manage mixed stiff and non-stiff differential systems, high-order approximation and the use of an adaptive step method to improve computational time without forfeiting significant detail. However, it remains of interest for future studies to further benchmark integrator schemes and their relative performance for the problem in terms of accuracy and computational time. Examples include lower order runge-kutta embeddings and Taylor methods such as Heyoka, which has lately shown promising results for space applications and trajectory optimisation [84].

In this work, we aim to determine trajectories based on impulsive manoeuvres. The optimisation strategy draws inspiration from a direct-shooting method where we compare the performance of several candidate trajectories with defined control sequences and a one-week fixed mission duration. However, in the context of small-body missions (see Rosetta [5] and HERA [14, 15]), one week typically represents a relatively short time frame for carrying out scientific measurements, which we observe by the low count of visited regions on the grid in the presented solutions. However, extending the time frame and number of manoeuvres is not trivial as it couples with the complexity of the problem. An alternative for extending the mission without adding complexity is to consider a multiple shooting method where we instead solve a series of optimisation problems considering subsequent time intervals with the condition of connected boundary states to ensure continuity.

To find optimal trajectories, we rely on a modified version of the extended ant colony optimiser (EACO). The chosen solution method is well motivated in Chapter 4, which highlights several benefits of evolutionary algorithms, particularly EACO. However, with the current chromosome structure, the problem quickly results in a large decision vector. For instance, in the second scenario, where we study the application of four spacecraft with four manoeuvres each, our aim is to solve the problem with 96 decision variables. Consequently, expanding the optimisation space makes it increasingly difficult for EACO to sample the state space adequately, resulting in an evident combinatorial challenge and possibly sub-optimal solutions. This behaviour can also be observed in Appendix D, which presents the convergence rates for the fitness value of each optimisation scenario. To mitigate the effects of adding spacecraft, we adopt an alternative model solving a series of optimisation problems. The results in Section 6.5.3 display a general increase in performance for the local approach but at the cost of losing the global solution. To further improve the conditions for solving the problem, one should consider tuning the hyperparameters related to the search process, integration accuracy and various problem characteristics. In this work, we have limited ourselves to only considering tuning the parameters of EACO, where we use a parameter-sweep strategy to acclimate the search strategy for the considered problem dynamics and fitness function. However, the performance can likely be further improved by considering a case-by-case optimisation of all hyperparameters through a probabilistic Bayesian model.

7.2 Computational Properties

Reviewing the computational properties of the optimisation process, it becomes evident that propagating the state using a polyhedral gravity model results in a computationally expensive task. In fact, evaluating positions to obtain acceleration values at each time step defined by the integrator constitutes almost a third of the total simulation time. For comparison, the second and third most time-consuming tasks, being interpolation and quaternion rotations, make up for less than 10% of the total simulation time, respectively. In order to reduce the time for solving the problem, one can instead use a polyhedral model with a reduced number of vertices, resulting in a faster evaluation process. However, deciding on the appropriate resolution of the mesh should be done considering the effects on accuracy for modelling the gravitational field [31]. Apart from the polyhedral model, fine-tuning parameters such as the tolerance level for integrating the equations of motion and the kernel size for the ant colony optimiser could improve performance both computationally and qualitatively. The effects of these parameters should therefore be studied further in more extensive testing.

Furthermore, we note that the search width can be adjusted by varying the population size. Even though the problem is solvable for smaller workstations, the robustness will likely improve by considering a large initial population favouring exploration, which naturally results in longer simulation times for a limited number of processing threads. However, based on the scaling results, we observe that the problem scales well with the addition of threads. With appropriate resources, future work could also consider benchmarking the convergence for larger populations.

7.3 Efficiency of Swarming Spacecraft

Based on the numerical results in Section 6.5, it is evident that the optimiser often converges to solutions that utilise energetic manoeuvres to rapidly change the spacecraft's course, favoring larger control magnitudes applied outside the region of interest to redirect its path. In the first scenario, the total Δv used was 0.479 m/s, which was evenly distributed over four impulsive manoeuvres and defined within a realistic magnitude range. For the second scenario considering a global approach for a swarm, the four spacecraft utilised a total of 12.413 m/s. Here, Spacecraft 3 used the least amount of Δv which is likely an effect of its proximity to the body, hence being able to utilise the gravitational acceleration and smaller correction manoeuvres to remain on a beneficial trajectory. However, in the case of Spacecraft 2, which traveled farthest away from the body, we also note the most significant use of Δv. Finally, in the third scenario considering a local approach, we observe a rapid decrease in total Δv to 2.807 m/s, with only the second spacecraft deviating from the average use with its last manoeuvre. We generally observe comparable performance regarding minimal control between the first and third scenarios, resulting in similar trajectories and control magnitudes mostly below the escape velocity. To motivate the use of more controlled path corrections, one can either increase the number of available manoeuvres or decrease the corresponding magnitude range. However, increasing the number of manoeuvres also comes with the cost of increasing the search space, affecting the algorithm's convergence rate and ability to find good solutions. For this reason, we evaluated several test cases with larger chromosomes than those considered in the second scenario, which generally resulted in less coverage for a global approach, most likely due to the increasing complexity of the combinatorial problem.

Furthermore, comparing the fitness of the three test scenarios, we observe that the swarm was more effective than a single spacecraft, obtaining a coverage ratio of 11% (extended model) and 19% (local search) compared to a single spacecraft of 6.27% over an equal mission duration. However, the individual performance of the single spacecraft in the first scenario was greater than the individual coverage for the swarm. The behaviour is particularly evident in Figs. 6.11 and 6.15, which show more stable trajectories compared to Fig. 6.13. The results are likely an effect of the second scenario optimising four trajectories simultaneously in a high-dimensional space, thus leading to rapid growth in complexity. In contrast, the relative per spacecraft simulation time was much shorter in the second scenario, where the communication overhead between threads and simulation software is reduced. Although it remains difficult to either convey or deduce any direct improvement in the quality of the measurements, we note that using a swarm scales well for reducing mission time and expanding coverage. In terms of flexibility, the simulation demonstrate promising results for a swarm where smaller spacecraft with less fuel requirements can be fitted with various technologies to simultaneously carry out a diverse set of tasks, including modelling the gravitational field. Finally, in this work, we consider an early stage of the mission where the information on the geodetic properties of the body remains limited. Hence, being able to efficiently cover a large proportion of the body in less time is essential for collecting data to quickly update the gravity field model and determine trajectories for any subsequent mission objectives.

Chapter 8

Conclusions and Outlook

8.1 Conclusions

In view of the rising interest in studying the geodesy and mass-density configuration of smaller solar system bodies, this thesis proposes a model leading to a set of trajectories representing a spacecraft swarm aimed at jointly maximising the measured gravitational signal. One of the most significant contributions from this study includes formulating a trajectory optimisation problem suitable for evolutionary algorithms. In detail, we focus on using a modified extended ant colony optimiser representing a versatile and robust solution method for solving high-dimensional problems in complex dynamical environments. To model the coverage of a spacecraft, we introduce a discrete approach based on a longitude-latitude grid and present several modifications that could further improve the accuracy of measurements. In addition, we adopt a vectorised open-source implementation of the polyhedral gravity model to represent the gravitational field and demonstrate its considerable performance in a large-scale optimisation scenario.

Moreover, we demonstrate the method and present numerical results based on a prospective mission to the comet 67P/Churyumov-Gerasimenko using a detailed triangulated mesh and an initial assumption of homogeneous mass density. The results show improved coverage over a shorter mission duration when comparing a swarm to a single spacecraft. However, optimising the trajectory for a single spacecraft significantly reduces the search space, leading to a more efficient solution. It is therefore motivated to further study how the dimensionality of the problem can be reduced for more complex scenarios and to generate more efficient swarm solutions. Nevertheless, the numerical results suggest the feasibility of employing smaller spacecraft in a swarm setting adapted for dispersed operations to maximise scientific returns, offering reduced costs and enhanced flexibility for pursuing multiple objectives simultaneously and for realising missions to smaller solar system bodies. Finally, the codebase developed for this thesis was designed with modularity in mind, allowing for its application to a broader range of objectives in the context of evaluating the benefits associated with the use of spacecraft swarms.

8.2 Future Research

A fundamental tenet of the scientific methodology involves identifying model inadequacies and opportunities for future work to reiterate and refine the presented solution method. This section is therefore dedicated to underlining areas of improvement for both the model and solution method. The recommendations for future research are divided into three varieties: those that pertain to the dynamical model, those that relate to the optimisation framework and general improvements.

Extended Dynamical Model

A particularly relevant matter for the simulation of trajectories involves defining a dynamical model that captures the principal system characteristics for the considered context. In this thesis, we assume a simplification of reality by a dynamical model based solely on a constant fixed-axis body rotation, gravitational attraction and impulsive control. Hence, future work should extend the model to account for other considerable perturbations to simulate more realistic trajectories. In particular, we recommend that future work consider the following aspects:

- As presented in Section 6.1, most solar system asteroids and comets experience a relatively complex rotational motion. Thus, a reasonable extension includes a model of the body's precession by variables related to its torque, such as sublimation, outgassing, solar radiation pressure and external gravitational forces. These elements would most realistically be introduced using a Hill frame where the precession of the rotation axis is represented by a set of Euler equations similar to the work of Kramer, T. et al. [83].

- Regarding the motion of each spacecraft, we note the existence of several prominent external forces acting on the spacecraft apart from the body of interest. Principal examples include solar radiation pressure, solar tides and gravitational forces exerted by neighbouring solar system bodies. These forces can effectively be added to the equations of motion individually and represented in a spacecraft-specific inertial reference frame. Introducing such a frame also enables a model of attitude, which can be essential for maintaining additional objectives of communication and telemetry.

- Finally, this work assumes a simple model of instantaneous thrusts to control the spacecraft's motion, including variables related to execution time, magnitude and direction of each manoeuvre. A natural progression is extending the control to account for the evolution of the spacecraft's wet and dry mass through the Tsiolkovsky rocket equation. When considering a shorter mission duration, a feasible addition is to include each engine's firing time representing the impulsive manoeuvre. An alternative approach is to adopt continuous low-thrust manoeuvres which may simplify the implementation and dynamical equations.

Optimisation Framework Considerations

Concerning the optimisation framework, we recommend future work to take the following aspects into account:

- This thesis primarily focuses on global models to find optimal solutions, including a heuristic algorithm incorporating steering laws to improve the search process. Nonetheless, it would be interesting to review how local optimisation methods can be coupled with the suggested global scheme to improve the final solution. For this task, PyGMO already provides several appropriate methods which simplify implementation. However, to motivate using a local method, one can also perform an initial sensitivity analysis to acquire a general insight into the current solution's performance. A simple yet efficient method is Monte Carlo sampling, where we study the resulting fitness by varying each decision variable within a specific confidence interval. Since each evaluation is independent, it allows for a similar parallel structure used for the presented optimisation models. One can then distinguish the contribution of each variable by an analysis of variance, ANOVA.

- Tuning problem parameters is essential for adapting any optimisation framework to a specific problem domain. It can enable faster convergence rates, minimally required resources, and an improved quality of the final solution. In this thesis, we present a simple parameter-sweep strategy for tuning parameters related to the optimiser, which resulted in an improved solution. However, since it was only an initial tuning process for a general problem structure, we could not conclude that the selected values represented the optimal parameter group. Additionally, the study also lacks problem-specific tuning for hyperparameters related to the objective, constraints, and numerical techniques. For these values, we referred to previous work. Hence, we advise future work to adopt an efficient strategy for tuning any hyperparameter in the optimisation framework when considering new problem characteristics. Prominent examples of such methods include genetic algorithms, neural networks and Bayesian procedures. Given the current computational complexity of the framework, the last option is likely to represent the most efficient and practical approach.

- In this thesis, we have only considered comparing a limited number of disposable manoeuvres and spacecraft. Therefore, it would be interesting to consider an additional optimisation procedure to determine the best-performing set. Given the integer nature of the two additional variables, the problem is now suitable to be solved as a mixed-integer nonlinear program coupled with the non-integer solution strategy presented in this work. However, given the computational complexity of the presented framework, it remains of interest to further improve the computational properties of the optimisation process. Potential improvements include, for instance, tuning hyperparameters to speed up the convergence of the optimiser and the numerical integrator, redefining the control strategy and dynamical equations, selecting a faster propagator and using a more vectorised code structure.

General Considerations

Finally, we present some general considerations for continued analysis and improvement of the presented work:

- To motivate the choice of suitable numerical techniques and optimisers, we have mainly focused on identifying prominent problem characteristics and referencing the findings of previous work. It is therefore of interest to benchmark the performance of several integrators and optimisers for the specific problem scenario. To accomplish this, DEsolver and PyGMO offer a diverse set of readily available candidate algorithms, simplifying the implementation process given the current modularised state of the codebase.

- Another prominent feature of this work is the polyhedral gravity model. Although it provides a detailed initial gravity field representation and uses a particularly fast vectorised structure, it is evident that the polyhedral model remains one of the most time-consuming components of the considered algorithms and techniques. In order to reduce the simulation time, an alternative is to limit the use of the polyhedral model to inside the Brillouin sphere and adopt a spherical harmonics expansion elsewhere. In other words, minimising computational complexity while maintaining proper convergence close to the body. Furthermore, by assuming a model based on spherical harmonics inside the region of interest $\mathcal{S}_f$, we can now benchmark the resulting precision error between a continuous model of coverage and the discrete approximation presented in this thesis.

- In this work, we assess the quality of the solution by its coverage, relevance in measurements and the risk of collisions with the central body. However, future work should consider expanding the objective to simulate a more realistic scenario. For instance, minimising the total magnitude of control can effectively represent the minimisation of most fuel-related costs. Next, when considering a swarm, maintaining a line of sight representing communication windows is essential to allow for inter-spacecraft cooperation. In practice, we can efficiently represent a line of sight using the algorithm presented in Section 5.2.2 by studying the intersection between inter-spacecraft rays and the polyhedral model. Moreover, since most of these objectives may be competing, a way forward is to adopt a multi-objective optimisation strategy leading to a Pareto optimal solution. For this purpose, PyGMO provides several suitable optimisers, including a Multi-Objective Hypervolume-Based Ant Colony Optimiser (MHACO), which extends on the optimiser considered in this thesis [61]. In order to acknowledge additional operation constraints, we recommend studying how the optimisation framework can take advantage of available software such as PAseos, which simulates the environment for operating multiple Spacecraft [85].

- Lastly, considering that one of the outcomes of the thesis relates to the work on geodesyNets, it would be interesting to evaluate how the experienced accelerations resulting from the considered trajectories affect the robustness and accuracy of the predicted mass-density distribution.

Bibliography

[1] Holger Sierks et al. "On the nucleus structure and activity of comet 67P/Churyumov-Gerasimenko". In: *Science* 347.6220 (2015), aaa1044.

[2] Pingyuan Cui and Dong Qiao. "The present status and prospects in the research of orbital dynamics and control near small celestial bodies". In: *Theoretical and Applied Mechanics Letters* 4.1 (2014), p. 013013.

[3] Kenshiro Oguri and Jay W. McMahon. "Risk-aware Mission Design for In situ Asteroid Exploration under Uncertainty". In: *2021 IEEE Aerospace Conference (50100)*. IEEE, Mar. 2021.

[4] Elena Adams et al. "Double Asteroid Redirection Test: The Earth Strikes Back". In: *2019 IEEE Aerospace Conference*. 2019, pp. 1–11.

[5] Karl-Heinz Glassmeier et al. "The ROSETTA Mission: Flying towards the origin of the solar system". In: *Space Science Review, v.128, 1-21 (2007)* 128 (Feb. 2007).

[6] A. Fujiwara et al. "The Rubble-Pile Asteroid Itokawa as Observed by Hayabusa". In: *Science* 312.5778 (June 2006), pp. 1330–1334.

[7] Takanao Saiki et al. "Overview of Hayabusa2 Asteroid Proximity Operation Planning and Preliminary Results". In: *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan* 19.1 (2021), pp. 52–60.

[8] Michael C. Nolan et al. "Shape model and surface properties of the OSIRIS-REx target Asteroid (101955) Bennu from radar and lightcurve observations". In: *Icarus* 226.1 (Sept. 2013), pp. 629–640.

[9] Stephen B. Broschart and Daniel J. Scheeres. "Control of Hovering Spacecraft Near Small Bodies: Application to Asteroid 25143 Itokawa". In: *Journal of Guidance, Control, and Dynamics* 28.2 (Mar. 2005), pp. 343–354.

[10] Roberto Furfaro et al. "Modeling irregular small bodies gravity field via extreme learning machines and Bayesian optimization". In: *Advances in Space Research* 67.1 (Jan. 2021), pp. 617–638.

[11] Jacopo Villa et al. "Gravity Estimation at Small Bodies via Optical Tracking of Hopping Artificial Probes". In: (2022).

[12] Nathan Stacey, Kaitlin Dennison, and Simone D'Amico. "Autonomous Asteroid Characterization through Nanosatellite Swarming". In: *2022 IEEE Aerospace Conference (AERO)*. IEEE, Mar. 2022.

[13] Nathan Stacey and Simone D'Amico. "Autonomous swarming for simultaneous navigation and asteroid characterization". In: *AAS/AIAA Astrodynamics Specialist Conference*. Vol. 1. 2018.

[14] D. DellaGiustina et al. “OSIRIS-APEX: A Proposed OSIRIS-REx Extended Mission to Apophis”. In: *Apophis T-7 Years: Knowledge Opportunities for the Science of Planetary Defense.* Vol. 2681. LPI Contributions. May 2022, 2011, p. 2011.

[15] Mirko Trisolini, Camilla Colombo, and Yuichi Tsuda. “Target selection for Near-Earth Asteroids in-orbit sample collection missions”. In: *Acta Astronautica* 203 (2023), pp. 407–420. ISSN: 0094-5765.

[16] Julie C. Castillo-Rogez et al. “Approach to exploring interstellar objects and long-period comets”. In: *CaltechAUTHORS* (Jan. 2019).

[17] C. T. Barnett. “Theoretical Modeling of the Magnetic and Gravitational Fields of an Arbitrarily Shaped Three-Dimensional Body”. In: *GEOPHYSICS* 41.6 (Dec. 1976), pp. 1353–1364.

[18] V. Pohanka. “Optimum Expression for Computation of the Gravity Field of a Homogeneous Polyhedral Body”. In: *Geophysical Prospecting* 36.7 (Oct. 1988), pp. 733–751.

[19] Robert A. Werner and Daniel J. Scheeres. “Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia”. In: *Celestial Mechanics and Dynamical Astronomy* 65.3 (1997).

[20] S. Petrović. “Determination of the potential of homogeneous polyhedral bodies using line integrals”. In: *Journal of Geodesy* 71.1 (Dec. 1996), pp. 44–52.

[21] Dimitrios Tsoulis and Sveto Petrović. “On the singularities of the gravity field of a homogeneous polyhedral body”. In: *GEOPHYSICS* 66.2 (Mar. 2001), pp. 535–539.

[22] Dario Izzo and Pablo Gómez. “Geodesy of irregular small bodies via neural density fields”. In: *Communications Engineering* 1.1 (Dec. 2022).

[23] D. S. Lauretta et al. “OSIRIS-REx: Sample Return from Asteroid (101955) Bennu”. In: *Space Science Reviews* 212.1-2 (Aug. 2017), pp. 925–984.

[24] Moritz von Looz, Pablo Gomez, and Dario Izzo. “Study of the asteroid Bennu using geodesyANNs and Osiris-Rex data”. In: *Proceedings of the Eigth International Conference on Astrodynamics Tools and Techniques, ICATT.* 2021.

[25] Danylo Malyuta et al. “Advances in trajectory optimization for space vehicle control”. In: *Annual Reviews in Control* 52 (2021), pp. 282–315.

[26] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, Second Edition.* Second. Society for Industrial and Applied Mathematics, 2010.

[27] Federico Rossi et al. “Communication-Aware Orbit Design for Small Spacecraft Swarms Around Small Bodies”. In: *Journal of Guidance, Control, and Dynamics* 45.11 (Nov. 2022), pp. 2046–2060.

[28] Francesco Capolupo, Thierry Simeon, and Jean-Claude Berges. “Heuristic Guidance Techniques for the Exploration of Small Celestial Bodies”. In: *IFAC-PapersOnLine* 50.1 (July 2017), pp. 8279–8284.

[29] Sumeet G. Satpute, Per Bodin, and George Nikolakopoulos. “Cooperative planning for multi-site asteroid visual coverage”. In: *Advanced Robotics* 35.21-22 (July 2021), pp. 1332–1346.

[30] J Pearl, W Louisos, and D Hitt. "Curvilinear Surface-Based Gravity Model for Evolutionary Trajectory Optimization around Bennu". In: *AAS/AIAA Space Flight Mechanics Meeting.* 2019.

[31] Jonas Schuhmacher et al. "Investigation of the Robustness of Neural Density Fields". In: *Proceedings of the 12th International ESA Conference on Guidance, Navigation & Control Systems (GNC).* 2023.

[32] James Diebel. "Representing attitude: Euler angles, unit quaternions, and rotation vectors". In: *Matrix* 58.15-16 (2006), pp. 1–35.

[33] Howard D Curtis. *Orbital mechanics for engineering students.* 4th ed. Aerospace Engineering. Woburn, MA: Butterworth-Heinemann, Sept. 2020.

[34] D.J. Scheeres et al. "Dynamics of Orbits Close to Asteroid 4179 Toutatis". In: *Icarus* 132.1 (Mar. 1998), pp. 53–79.

[35] Yang Yu. *Orbital Dynamics in the Gravitational Field of Small Bodies.* Springer Berlin Heidelberg, 2016.

[36] D.J. Scheeres et al. "Orbits Close to Asteroid 4769 Castalia". In: *Icarus* 121.1 (May 1996), pp. 67–87.

[37] Yu Jiang and Hexi Baoyin. "Orbital Mechanics near a Rotating Asteroid". In: *Journal of Astrophysics and Astronomy* 35.1 (Mar. 2014), pp. 17–38.

[38] Dimitrios Tsoulis and Georgia Gavriilidou. "A computational review of the line integral analytical formulation of the polyhedral gravity signal". In: *Geophysical Prospecting* 69.8-9 (Sept. 2021), pp. 1745–1760.

[39] Dimitrios Tsoulis. "Analytical computation of the full gravity tensor of a homogeneous arbitrarily shaped polyhedral source using line integrals". In: *Geophysics* 77.2 (Mar. 2012), F1–F11.

[40] Dimitrios Tsoulis. "Terrain modeling in forward gravimetric problems: a case study on local terrain effects". In: *Journal of Applied Geophysics* 54.1-2 (Nov. 2003), pp. 145–160.

[41] Manik Talwani and Maurice Ewing. "Rapid Computation of Gravitational Attraction of Three-Dimensional Bodies of Arbitrary Shape". In: *GEOPHYSICS* 25.1 (Feb. 1960), pp. 203–225.

[42] Dimitrios Tsoulis et al. "Recursive algorithms for the computation of the potential harmonic coefficients of a constant density polyhedron". In: *Journal of Geodesy* 83.10 (Mar. 2009), pp. 925–942.

[43] Jonas Schuhmacher. *Efficient Polyhedral Gravity Modeling in Modern C++.* en. Tech. rep. Technical University of Munich, Dec. 2022. URL: `https://github.com/esa/polyhedral-gravity-model`.

[44] Abolfazl Shirazi, Josu Ceberio, and Jose A. Lozano. "Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions". In: *Progress in Aerospace Sciences* 102 (Oct. 2018), pp. 76–98.

[45] Petter Ögren Ulf Jönsson Claes Trygger. *Optimal Control.* KTH Royal Institute of Technology, 2010.

[46] Dario Izzo. "Global Optimization and Space Pruning for Spacecraft Trajectory Design". In: *Spacecraft Trajectory Optimization.* Ed. by Bruce A.Editor Conway. Cambridge Aerospace Series. Cambridge University Press, 2010, pp. 178–201.

[47] David L. Williamson. "The Evolution of Dynamical Cores for Global Atmospheric Models". In: *Journal of the Meteorological Society of Japan. Ser. II* 85B (2007), pp. 241–269.

[48] E. H. Vestine et al. “Integral and Spherical-Harmonic Analyses of the Geomagnetic Field for 1955.0, PART 2”. In: *Journal of geomagnetism and geoelectricity* 15.2 (1963), pp. 73–89.

[49] Shiyi Chen and Gary D. Doolen. “Lattice Boltzmann Method for Fluid Flows”. In: *Annual Review of Fluid Mechanics* 30.1 (Jan. 1998), pp. 329–364.

[50] Maarten ’t Hart. “The projection point geodesic grid algorithm for meshing the sphere”. In: *Journal of Computational Physics* 454 (Apr. 2022), p. 110993.

[51] Liang Yin et al. “Parallel numerical simulation of the thermal convection in the Earth’s outer core on the cubed-sphere”. In: *Geophysical Journal International* 209.3 (Mar. 2017), pp. 1934–1954.

[52] M. Tabata and A. Suzuki. “A stabilized finite element method for the Rayleigh-Bénard equations with infinite Prandtl number in a spherical shell”. In: *Computer Methods in Applied Mechanics and Engineering* 190.3-4 (Oct. 2000), pp. 387–402.

[53] A. Gansen et al. “A 3D Unstructured Mesh FDTD Scheme for EM Modelling”. In: *Archives of Computational Methods in Engineering* 28.1 (Jan. 2020), pp. 181–213.

[54] Yu Jie-qing and W U Li-xin. “Spatial subdivision and coding of a global three-dimensional grid: Spheoid Degenerated-Octree Grid”. In: *2009 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2009.

[55] Benjamin Ulmer and Faramarz Samavati. “Toward volume preserving spheroid degenerated-octree grid”. In: *GeoInformatica* 24.3 (Jan. 2020), pp. 505–529.

[56] Hiroshi Hayashi and Akira Kageyama. “Yin–Yang–Zhong grid: An overset grid system for a sphere”. In: *Journal of Computational Physics* 305 (Jan. 2016), pp. 895–905.

[57] Nickolay Y. Gnedin, Vadim A. Semenov, and Andrey V. Kravtsov. “Enforcing the Courant-Friedrichs-Lewy condition in explicitly conservative local time stepping schemes”. In: *Journal of Computational Physics* 359 (Apr. 2018), pp. 93–105.

[58] Darren R. Myatt et al. “Advanced global optimisation for mission analysis and design”. In: 2004.

[59] Martin Schlüter, Matthias Gerdts, and Jan-J. Rückmann. “A numerical study of MIDACO on 100 MINLP benchmarks”. In: *Optimization* 61.7 (July 2012), pp. 873–900.

[60] Martin Schlüter, Mohamed Wahib, and Masaharu Munetomo. “Numerical Optimization of ESA’s Messenger Space Mission Benchmark”. In: *Applications of Evolutionary Computation*. Springer International Publishing, 2017, pp. 725–737.

[61] Giacomo Acciarini, Dario Izzo, and Erwin Mooij. “MHACO: a Multi-Objective Hypervolume-Based Ant Colony Optimizer for Space Trajectory Optimization”. In: *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, July 2020.

[62] Giacomo Acciarini Acciarini. *Optimizing a solar sailing polar mission to the Sun: Development and application of a new Ant Colony optimizer*. Nov. 2019.

[63] Martin Schlüter, Jose A. Egea, and Julio R. Banga. “Extended ant colony optimization for non-convex mixed integer nonlinear programming”. In: *Computers & Operations Research* 36.7 (July 2009), pp. 2217–2229.

[64] M. Dorigo. “Optimization, Learning and Natural Algorithms”. In: *Ph.D. Thesis, Politecnico di Milano* (1992).

[65] Martin Schlüter et al. “MIDACO on MINLP space applications”. In: *Advances in Space Research* 51.7 (Apr. 2013), pp. 1116–1131.

[66] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits*. Springer Berlin Heidelberg, 2000.

[67] J. R. Dormand and P. J. Prince. “New Runge-Kutta algorithms for numerical simulation in dynamical astronomy”. In: *Celestial Mechanics* 18.3 (Oct. 1978), pp. 223–232.

[68] P.J. Prince and J.R. Dormand. “High order embedded Runge-Kutta formulae”. In: *Journal of Computational and Applied Mathematics* 7.1 (Mar. 1981), pp. 67–75.

[69] T. E. Hull et al. “Comparing Numerical Methods for Ordinary Differential Equations”. In: *SIAM Journal on Numerical Analysis* 9.4 (Dec. 1972), pp. 603–637.

[70] Jinglang Feng et al. “Sensitivity analysis of the orbital motion around 469219 Kamo’oalewa (2016 HO3) to uncertainties on asteroid mass and solar radiation pressure”. In: *Advances in Space Research* 69.3 (Feb. 2022), pp. 1602–1618.

[71] Brandon A. Jones and Alireza Doostan. “Satellite collision probability estimation using polynomial chaos expansions”. In: *Advances in Space Research* 52.11 (Dec. 2013), pp. 1860–1875.

[72] Jonathan Aziz et al. “A Smoothed Eclipse Model for Solar Electric Propulsion Trajectory Optimization”. In: *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan* 17.2 (2019), pp. 181–188.

[73] J.R. Dormand and P.J. Prince. “A reconsideration of some embedded Runge-Kutta formulae”. In: *Journal of Computational and Applied Mathematics* 15.2 (June 1986), pp. 203–211.

[74] Lawrence F. Shampine and Laurent O. Jay. “Dense Output”. In: *Encyclopedia of Applied and Computational Mathematics*. Ed. by Björn Engquist. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 339–345. ISBN: 978-3-540-70529-1.

[75] Erwin Kreyszig. *Advanced Engineering Mathematics*. en. 10th ed. Chichester, England: John Wiley & Sons, May 2011.

[76] Tomas Möller and Ben Trumbore. “Fast, Minimum Storage Ray-Triangle Intersection”. In: *Journal of Graphics Tools* 2.1 (1997), pp. 21–28.

[77] Dario Izzo. “Pygmo and pykep: Open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization)”. In: *Proceedings of the Fifth International Conference on Astrodynamics Tools and Techniques, ICATT*. Jan. 2012.

[78] Francesco Biscani, Dario Izzo, and Chit Hong Yam. *A Global Optimisation Toolbox for Massively Parallel Engineering Optimisation*. 2010.

[79] Jian S. Dai. “Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections”. In: *Mechanism and Machine Theory* 92 (Oct. 2015), pp. 144–152.

[80] ESA/Rosetta. *Rosetta’s Target: Comet 67P/Churyumov-Gerasimenko*. Accessed 2023.05.01. 2019. URL: `https://sci.esa.int/web/rosetta/-/14615-comet-67p`.

[81] H. U. Keller et al. "The changing rotation period of comet 67P/Churyumov-Gerasimenko controlled by its activity". In: *Astronomy & Astrophysics* 579 (June 2015), p. L5.

[82] L. Jorda et al. "The global shape, density and rotation of Comet 67P/Churyumov-Gerasimenko from preperihelion Rosetta/OSIRIS observations". In: *Icarus* 277 (Oct. 2016), pp. 257–278.

[83] T. Kramer et al. "Comet 67P/Churyumov-Gerasimenko rotation changes derived from sublimation-induced torques". In: *Astronomy & Astrophysics* 630 (Sept. 2019), A3.

[84] Francesco Biscani and Dario Izzo. "Revisiting high-order Taylor methods for astrodynamics and celestial mechanics". In: *Monthly Notices of the Royal Astronomical Society* 504.2 (Apr. 2021), pp. 2614–2628. ISSN: 0035-8711.

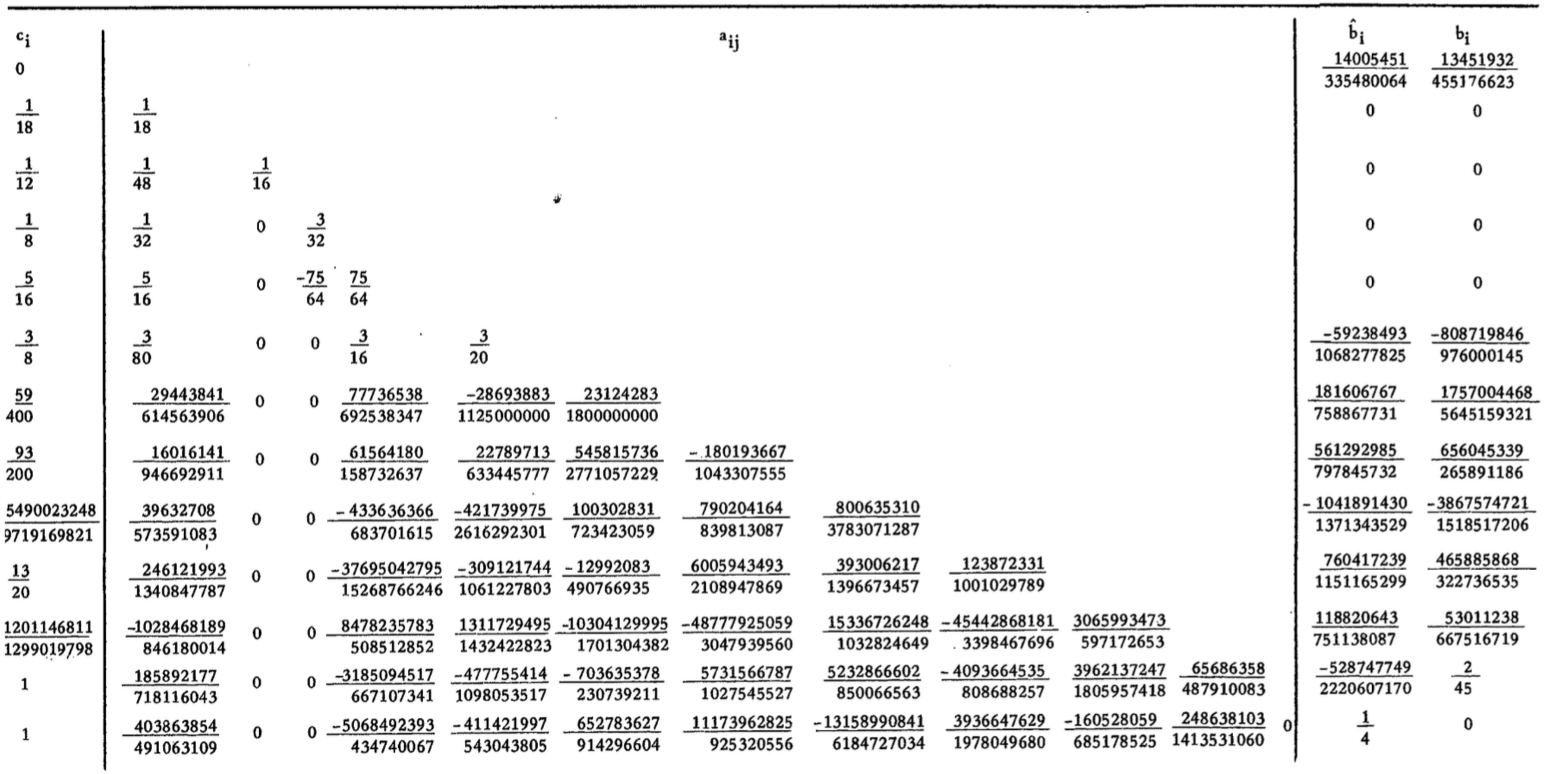
[85] Pablo Gómez et al. *PAseos Simulates the Environment for Operating multiple Spacecraft.* 2023.

[86] Gene M. Amdahl. "Validity of the single processor approach to achieving large scale computing capabilities". In: *Proceedings of the April 18-20, 1967, spring joint computer conference on - AFIPS '67 (Spring).* ACM Press, 1967.

[87] John L. Gustafson. "Reevaluating Amdahl's law". In: *Communications of the ACM* 31.5 (May 1988), pp. 532–533.

Appendix A

Butcher Tableau for Dormand-Prince 8(7)-13M



c_i	a_{ij}												$\hat{b}_i$	b_i
0													$\frac{14005451}{335480064}$	$\frac{13451932}{455176623}$
$\frac{1}{18}$	$\frac{1}{18}$												0	0
$\frac{1}{12}$	$\frac{1}{48}$	$\frac{1}{16}$											0	0
$\frac{1}{8}$	$\frac{1}{32}$	0	$\frac{3}{32}$										0	0
$\frac{5}{16}$	$\frac{5}{16}$	0	$\frac{-75}{64}$	$\frac{75}{64}$									0	0
$\frac{3}{8}$	$\frac{3}{80}$	0	0	$\frac{3}{16}$	$\frac{3}{20}$								$\frac{-59238493}{1068277825}$	$\frac{-808719846}{976000145}$
$\frac{59}{400}$	$\frac{29443841}{614563906}$	0	0	$\frac{77736538}{692538347}$	$\frac{-28693883}{1125000000}$	$\frac{23124283}{1800000000}$							$\frac{181606767}{758867731}$	$\frac{1757004468}{5645159321}$
$\frac{93}{200}$	$\frac{16016141}{946692911}$	0	0	$\frac{61564180}{158732637}$	$\frac{22789713}{633445777}$	$\frac{545815736}{2771057229}$	$\frac{-180193667}{1043307555}$						$\frac{561292985}{797845732}$	$\frac{656045339}{265891186}$
$\frac{5490023248}{9719169821}$	$\frac{39632708}{573591083}$	0	0	$\frac{-433636366}{683701615}$	$\frac{-421739975}{2616292301}$	$\frac{100302831}{723423059}$	$\frac{790204164}{839813087}$	$\frac{800635310}{3783071287}$					$\frac{-1041891430}{1371343529}$	$\frac{-3867574721}{1518517206}$
$\frac{13}{20}$	$\frac{246121993}{1340847787}$	0	0	$\frac{-37695042795}{15268766246}$	$\frac{-309121744}{1061227803}$	$\frac{-12992083}{490766935}$	$\frac{6005943493}{2108947869}$	$\frac{393006217}{1396673457}$	$\frac{123872331}{1001029789}$				$\frac{760417239}{1151165299}$	$\frac{465885868}{322736535}$
$\frac{1201146811}{1299019798}$	$\frac{-1028468189}{846180014}$	0	0	$\frac{8478235783}{508512852}$	$\frac{1311729495}{1432422823}$	$\frac{-10304129995}{1701304382}$	$\frac{-48777925059}{3047939560}$	$\frac{15336726248}{1032824649}$	$\frac{-45442868181}{3398467696}$	$\frac{3065993473}{597172653}$			$\frac{118820643}{751138087}$	$\frac{53011238}{667516719}$
1	$\frac{185892177}{718116043}$	0	0	$\frac{-3185094517}{667107341}$	$\frac{-477755414}{1098053517}$	$\frac{-703635378}{230739211}$	$\frac{5731566787}{1027545527}$	$\frac{5232866602}{850066563}$	$\frac{-4093664535}{808688257}$	$\frac{3962137247}{1805957418}$	$\frac{65686358}{487910083}$		$\frac{-528747749}{2220607170}$	$\frac{2}{45}$
1	$\frac{403863854}{491063109}$	0	0	$\frac{-5068492393}{434740067}$	$\frac{-411421997}{543043805}$	$\frac{652783627}{914296604}$	$\frac{11173962825}{925320556}$	$\frac{-13158990841}{6184727034}$	$\frac{3936647629}{1978049680}$	$\frac{-160528059}{685178525}$	$\frac{248638103}{1413531060}$	0	$\frac{1}{4}$	0

Figure A.1: Butcher Tableu for Dormand-Prince 8(7)-13M
(Credited to Dormand and Prince [68]).

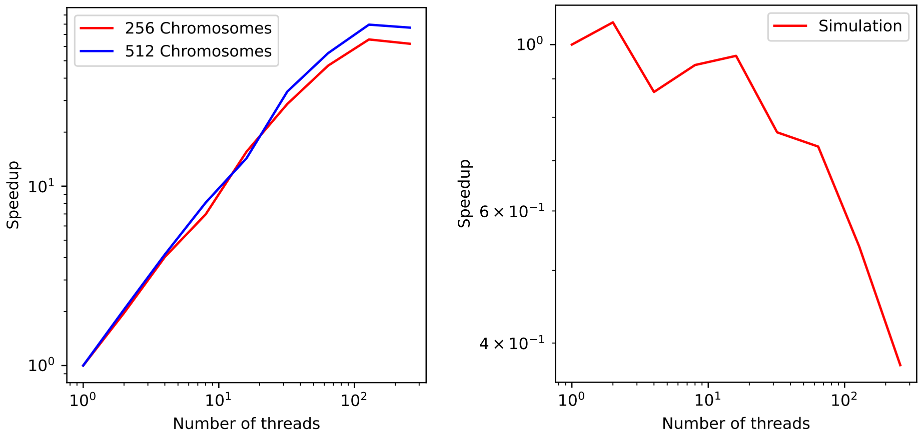
Appendix B

Scaling Test Speedup

Considering both strong a weak scaling, we define the speedup as the improvement of the computational time on n threads when compared to the minimal thread count n_{min}. Hence, speedup is formulated as

$$speedup = \frac{T_{n_{min}}}{T_n} \tag{B.1}$$

where $T_{n_{min}}$ is the required computational time when considering the minimal number of computational threads, which in this case equals to one, and T_n the measured simulation time when using n threads. For further study on the speedup performance of the presented simulation software, we recommend including measurements of the serial part and adopting both Amdahl's law of strong scaling [86] and Gustafson's law of weak scaling [87].



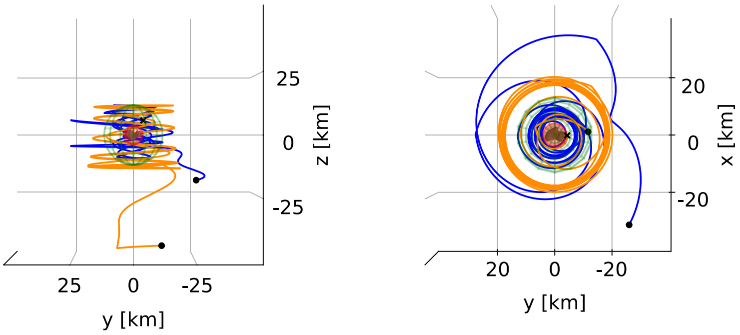
(a) Strong scaling speedup.

(b) Weak scaling speedup.

Figure B.1: Scaling test speedup performance.

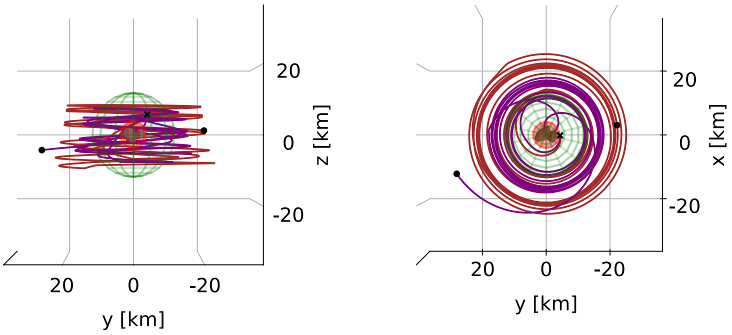
Appendix C

Results in the Body-fixed Frame



(a) Spacecraft 1 & 2
(side view).

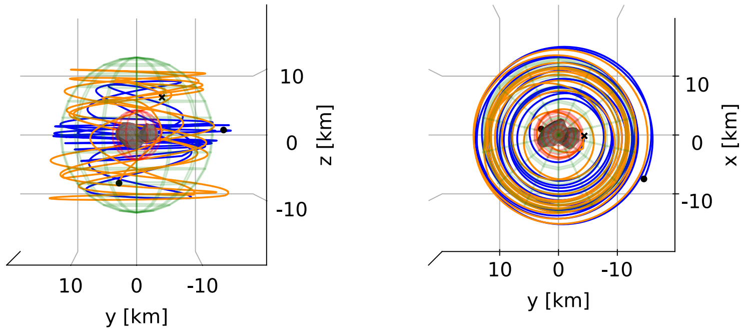
(b) Spacecraft 1 & 2
(top view).



(c) Spacecraft 3 & 4
(side view).

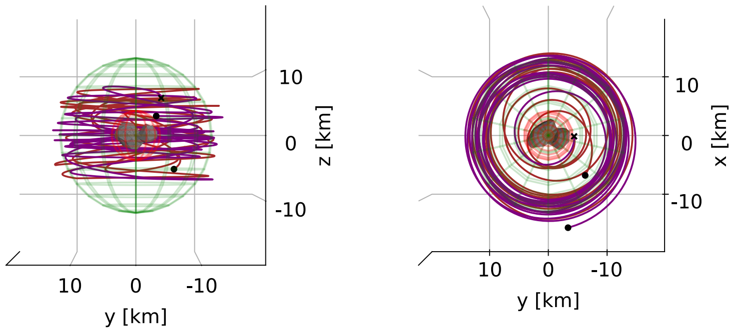
(d) Spacecraft 3 & 4
(top view).

Figure C.1: Global Multi-spacecraft optimisation results depicting a)-b) resulting trajectories for Spacecraft 1 and 2 in the body-fixed reference frame, and c)-d) the corresponding representation for Spacecraft 3 and 4. (Deployment at the cross).



(a) Spacecraft 1 & 2
(side view).

(b) Spacecraft 1 & 2
(top view).



(c) Spacecraft 3 & 4
(side view).

(d) Spacecraft 3 & 4
(top view).

Figure C.2: Local Multi-spacecraft optimisation results depicting a)-b) resulting trajectories for Spacecraft 1 and 2 in the body-fixed reference frame, and c)-d) the corresponding representation for Spacecraft 3 and 4. (Deployment at the cross).

Appendix D

Optimiser Convergence Rates

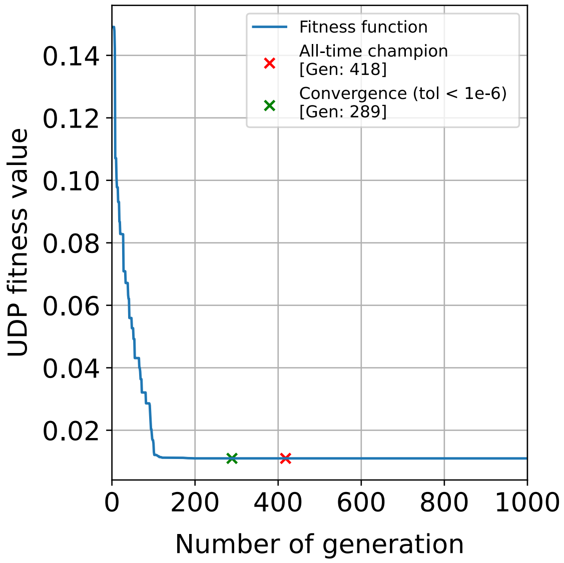


Figure D.1: Convergence rate: Single spacecraft with four manoeuvres.

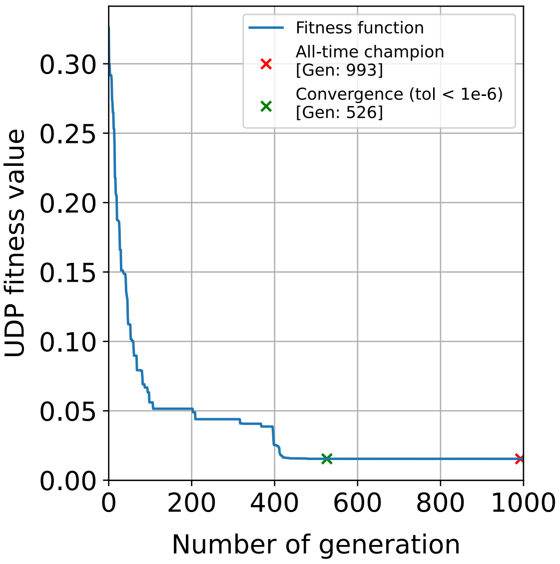


Figure D.2: Convergence rate: Four spacecraft with four manoeuvres each (extended model).

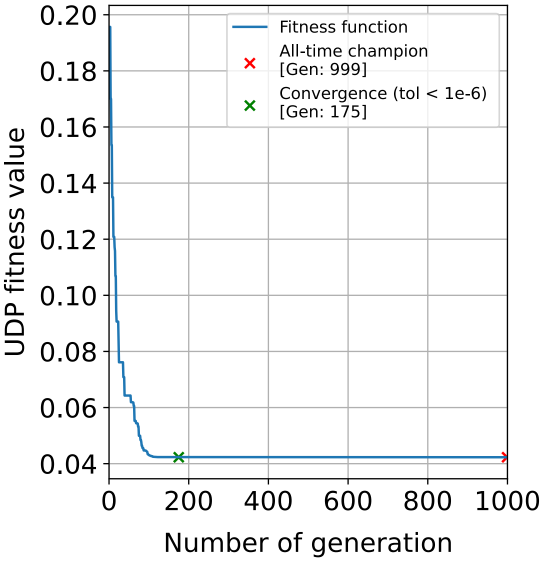


Figure D.3: Convergence rate: Four spacecraft with four manoeuvres each (local search).

TRITA-SCI-GRU 2023:445

KTH Royal Institute of Technology
School of Engineering Sciences
KTH SCI
SE-100 44 Stockholm, Sweden
URL: www.kth.se/sci