POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Aerospaziale

Sessione di Laurea Marzo/Aprile 2022

Spacecraft trajectory optimization for Earth-crossing asteroid redirection via kinetic impact

Relatori

Professor Lorenzo Casalino

Professor Bruce A. Conway

Candidato

Alessia Speziale

Ai miei nonni

Acknowledgements

Ringrazio la mia famiglia in primis, per avermi sostenuto quando ho scelto di intraprendere questo viaggio, per essermi stata vicina anche quando i problemi non sono mancati, per non avermi mai fatto mancare nulla e per avermi spinto a credere in me stessa e nelle mia capacità.

I deeply thank Prof. Conway for making this possible, for the welcome I received, for sharing his valuable teachings with me, and for the time he dedicated to me. This experience has taught me and made me grow more than I could have imagined. I am very satisfied with my path and what I have learned in a short time and I thank the professor from the bottom of my heart.

Ringrazio il Prof. Casalino per avermi aiutato a intraprendere questa esperienza ed essere stato sempre disponibile in qualità di mio relatore.

Ringrazio le mie amiche per avermi supportato nonostante la distanza. Ringrazio in particolar modo Ludovica, con cui ho condiviso questa esperienza. Ci siamo fatte forza a vicenda e siamo riuscite a tirare fuori il meglio da questi mesi, nello studio così come nel resto della vita a Chambana. I thank also my new friends for these five months of endless memories and adventures.

Ringrazio i miei nonni e Artù. Anche se non consapevoli, sono sempre stati tra i miei pensieri e mi hanno spinto a dare il meglio ogni giorno.

Per aspera ad astra

Abstract

The heart of this thesis consists of the trajectory optimization of a kinetic impactor spacecraft, which is sent to collide with a threatening near-Earth asteroid. As a result of the impact, the subsequent path of the asteroid is very modestly changed. The goal is to maximize the perigee radius of the deflected asteroid at its closest approach to Earth, with the important variables such as the date of Earth departure, the direction of the departure, the thrust program for the low-thrust motor, and the date of the collision all being optimization parameters. This continuous optimal control problem has been transcribed into a nonlinear programming (NLP) problem and is solved in the MATLAB environment. This transcription required the development of a Runge-Kutta (RK) parallel shooting code - implemented in MATLAB for the first time. It was also necessary to solve the problem approximately using Particle Swarm Optimization (PSO) which solution then became the required initial guess for the NLP problem.

Abstract

[Italiano]

Il cuore di questa tesi consiste nell'ottimizzazione della traiettoria di un veicolo spaziale a impatto cinetico, inviato a scontrarsi con un minaccioso asteroide la cui orbita si avvicina a quella terrestre. Come risultato dell'intercettazione, il successivo percorso dell'asteroide viene modificato in modo molto modesto. L'obiettivo è la massimizzazione del raggio di perigeo dell'asteroide, il cui moto è stato deviato, in corrispondenza del suo passaggio più vicino alla Terra. Sono stati selezionati diversi parametri di ottimizzazione quali la data di partenza dalla Terra, la direzione della partenza, il programma di spinta per il motore a bassa spinta e la data della collisione. Tale problema di controllo ottimale continuo è stato trascritto in un problema di programmazione non lineare (NLP) e risolto in ambiente MATLAB. La trascrizione ha richiesto lo sviluppo di un codice "*Runge-Kutta (RK) parallel shooting*", implementato in MATLAB per la prima volta. Inoltre, è stato necessario risolvere il problema approssimativamente tramite *Particle Swarm Optimization* (PSO), la cui soluzione è stata adottata come ipotesi iniziale richiesta dal problema NLP.

Table of Contents

List of Figures

List of Tables

Nomenclature

$\overrightarrow{x}$	State vector
$\overrightarrow{u}$	Control vector
(x, y, z)	Cartesian coordinates
r	Distance from Sun
r_p	Perigee radius
$\delta\vec{r}$	Change in position vector relative to a reference
θ	Angle from position vector to first point in Aries, in the ecliptic plane
z	Component of position vector normal to ecliptic plane
$\overrightarrow{v}$	Velocity vector
$\delta\overrightarrow{v}$	Change in velocity vector relative to a reference
$\Delta\overrightarrow{v}$	Change in velocity vector applied by an outside force
v_∞	Hyperbolic excess velocity
m	Spacecraft mass
μ	Gravitational constant
a	Orbit semi-major axis
e	Orbit eccentricity
i	Orbit inclination
ω	Orbit argument of periapsis
Ω	Orbit longitude of ascending node (RAAN)

f	Orbit true anomaly
E	Orbit eccentric anomaly
M	Orbit mean anomaly
t	General symbol for epoch
t_0	Start of the launch window
t_{flight}, t_f	Flight time
t_L	Epoch of spacecraft launch
t_I	Epoch of spacecraft intercept
β	In-plane thrust pointing angle
γ	Out-of-plane thrust pointing angle
a, b, ω, ϕ	Coefficients within a sinusoidal function
β_L	In-plane launcher thrust pointing angle
γ_L	Out-of-plane launcher thrust pointing angle
T_{max}	Maximum available thrust
c_{exh}	Engine exhaust velocity
A	Acceleration
$a_x\ a_y, a_z$	Perturbing acceleration components
AU	Astronomical unit
TU	Time unit
$f(\overrightarrow{x}, \overrightarrow{u}, t)$	Set of differential equations describing optimal control problems
$J, \tilde{J}$	Objective/fitness function
$\Psi(\overrightarrow{x}(t_f), t_f)$	Terminal conditions

$\phi(\overrightarrow{x}(t_f), t_f)$	Terminal cost function
$L(\overrightarrow{x}, \overrightarrow{u}, t)$	Cost function integrated along a trajectory
H	Hamiltonian system
N	Number of particles; number of time segments
n	Number of variables
$\overrightarrow{\chi}$	Particle position vector
$\overrightarrow{w}$	Particle velocity vector
$d_r(\overrightarrow{\chi}), ceq$	Equality constraint function
α_r, c_1	Coefficient multiplying the equality constraints
$c(x)$	Inequality constraint function
P	NLP parameter vector
$f(P)$	Set of nonlinear constraints for nonlinear programming problem
$J(P)$	Objective function for nonlinear programming problem
$A_L(P)$	Set of linear constraints for nonlinear programming problem
h	Length of each time segment
u	Control value specified at the boundary between time segments
ν	Control value specified within a time segment
p	Number of points
Δ_i	Defect at the righthand node of each segment
COV	Calculus of variations
EA	Evolutionary algorithms
GA	Genetic algorithm

PSO	Particle Swarm Optimizer
NLP	Nonlinear Programming problem
SQP	Sequential quadratic programming
RK4	4th order Runge-Kutta integration rule
DTRK	Direct Transcription with 4th order Runge-Kutta integration rule
ODE	Ordinary Differential Equations
EOM	Equations of motion
SOI	Sphere of Influence
STM	State Transition Matrix
NEO/A	Near-Earth Object/Asteroid
PHO/A	Potentially Hazardous Object/Asteroid

Many of the quantities listed above are attached to different bodies in the Solar System (e.g. the Earth, the Sun, the target asteroid, and the interceptor spacecraft). The following subscripts are used to identify to which body the property refers:

$X_{\odot}$	Property X of the Sun
$X_{\oplus}$	Property X of the Earth
$X_{♀}$	Property X of Venus
$X_{♂}$	Property X of Mars
$X_{♃}$	Property X of Jupiter
$X_{\oplus ☽}$	Property X of the Earth-Moon system
$X_{⚳}$	Property X of an asteroid
$X_{\mathrm{s/c}}$	Property X of the spacecraft

Chapter 1: Introduction

1.1 Spacecraft Trajectory Optimization

Optimization problems have longly been a subject of interest, not only from an academic standpoint but also from a pragmatic point of view in several important scientific applications, such as trajectory optimization of space vehicles. Apollo missions (*Figure 1.1*) are an example of an actual numerical optimization study, conducted to investigate optimal performance limits considering maneuver capability and entry heating during lunar return [1]. These boundaries are useful as guides in vehicle and mission design studies.

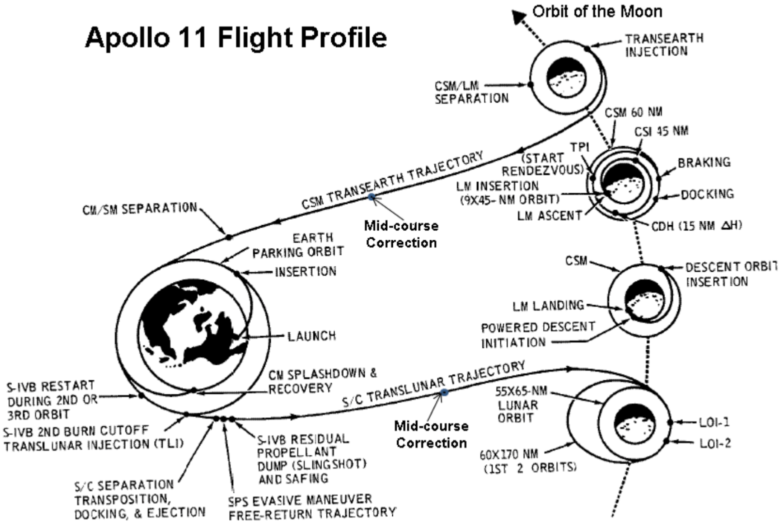


Figure 1.1 Apollo 11 Flight Profile [Credit: NASA]

A spacecraft trajectory optimization problem seeks a trajectory that fully satisfies some criteria, including specified initial and final conditions, within a required mission. Many factors contribute to the achievement of a successful solution, such as the mathematical modeling of the problem, the objective function definition, the development of a certain approach or the implementation of a method or an algorithm [2]. The objective function is a relevant numerical value to be minimized; in particular, the most frequent choices are the propellant minimization - which for continuous thrust applications is a synonym of minimizing the time of flight - or also the payload fraction maximization. Therefore, lower and upper bounds usually delimit the domain of each optimization parameter in order to get practical solutions, e.g. time variables are limited by the date of the projected impact in the problem discussed in Chapter 3.

The problem is usually a complicated continuous optimization problem [3], except in a few special integrable cases which reduce to parameter optimization problems. The complications mainly articulate in the non-linearity of the dynamical system, in possible discontinuities in the state variables (e.g. instantaneous velocity changes or "ΔV's" from rocket motors usage or planetary flybys), non-explicit terminal conditions (function of optimization variables) or presence of time-dependent forces (e.g. planetary perturbations), and the initially unknown basic structure of the optimal trajectory. The latter is subject to optimization, for instance in this work the spacecraft is launched only when Earth and asteroid reflect the optimal "geometry" found by the optimization solver according to the objective function. Of course, to make the solution extremely accurate also perturbations and other minor effects should be considered.

1.2 Low-Thrust Trajectory Optimization

Even if the potential benefits of low-thrust electric propulsion have been largely known for many years, only since few decades this technology has been employed for space missions, in particular for asteroids and comets missions such as Deep Space 1 (1998) [4] or NEAR in (2001) [5].

Low-thrust propulsion is advantageous for the propulsive mass required compared to the fuel that would be burned by a chemical rocket for a given mission; low-thrust electric motors, which provide a much higher specific impulse, are thus more efficient. Besides, Conway [6] inferred that the greater efficiency would yield a payload mass for the mission of approximately 12% of launch vehicle mass, instead of only about 3% using conventional chemical propulsion. However, since electric motors provide a much smaller thrust - typical spacecraft acceleration is on the order of 10^{-5} g - they operate continuously or nearly so. Therefore, the general continuous thrust trajectory optimization problem differs qualitatively from the impulsive one, mainly because of the absence of integrable arcs and the continuous-time histories of the controls themselves (e.g. the thrust pointing angles). Thus, there is no analytic solution and numerical methods must be employed.

Since space trajectories are controlled by the thrust vector, maximizing a specific performance index - by satisfying the boundary constraints - is achievable by finding an optimal control law for the thrust magnitude and direction, especially in high power missions where the spacecraft is equipped with more than one electric thruster. In fact, the result is often significantly affected by how the thrust is exploited, both in terms of throttle adjustment and optimal power partitioning among the available thrusters. The latter aspect has been studied by Casalino and Vavrina [7] who, based on several methods (indirect, direct and evolutionary algorithms), have deduced that the optimal power partitioning strategy provides relevant savings in terms of propellant consumption for example in missions of large sample return from a near-Earth asteroid. A throttling parameter was not introduced in the present work, but certainly could be the subject of further future study.

A low-thrust departure from Earth would require much time because of the many revolutions of the Earth needed. In this work, an instantaneous velocity change given by the upper stage of the launch vehicle ($\Delta \vec{v}_L$) at the initial departure is followed by continuous low-thrust propulsion. Based on initial intuition, the combination of the two could increase the spacecraft energy without exploiting any fuel, so without affecting its mass, which is more significant at the impact, and augmenting its speed of typically 1-2 km [6]. This could be very beneficial at the impact since it increases the momentum imparted to the asteroid by the interceptor spacecraft, but also because it could reduce the time and the consumed fuel, moving away from Earth faster than would be possible with only electric motors.

1.3 Optimization methods available

Trajectory optimization problems can be formulated as optimal control problems, described by a dynamical system of differential equations [3]:

$$\dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t), t) \tag{1.1}$$

where $t_0 \leq t \leq t_f$, $\overrightarrow{x}$ represents an n-dimensional state and $\overrightarrow{u}$ an m-dimensional control. Both those vectors are problem dependent, so for example the type of coordinates is chosen according to the mathematical model used. The initial conditions are specified as

$$x_i(0) \text{ given for } i = 1, 2, \ldots, k \text{ with } k \leq n. \tag{1.2}$$

The terminal conditions may be described implicitly:

$$\Psi(\overrightarrow{x}(t_f), t_f) = 0. \tag{1.3}$$

The objective function to be minimized can be written in the Bolza form as

$$J = \phi(\overrightarrow{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\overrightarrow{x}, \overrightarrow{u}, t)\, dt \tag{1.4}$$

where ϕ is a terminal cost function and the integral denotes a cost incurred during the whole trajectory.

Only in simplified cases, e.g. impulsive-thrust or continuous-thrust orbit transfer, the optimal trajectory is an *analytical* solution. Most researchers rely on numerical optimization methods to solve continuous optimal control problems; these methods are classified as direct or indirect. *Indirect* methods utilize n-costate variables (or adjoint variables or Lagrange multipliers) and their governing equations in addition to analytical necessary conditions from the calculus of variations (COV) or Pontryagin's Minimum Principle, consequently doubling the size of the dynamical system. *Direct* methods convert the continuous optimal control problem into a parameter optimization problem; the system equations are satisfied because of the stepwise integration according to implicit or explicit (e.g. Runge-Kutta) rules. The parameters represent a discretization of the state and control time histories and must satisfy the nonlinear constraint equations of the generated nonlinear programming problem (NLP) [3]. Both direct and indirect methods can converge only to locally optimal solutions, indeed they require an initial guess close to the global minimum. The improvement of the direct methods has led to the development of qualitatively different approaches, consisting of *evolutionary* and *heuristic* algorithms.

1.3.1 Analytical Methods

The original approach for space trajectory optimization was based on analytical methods, still used for elementary but useful examples as the Hohmann transfer. Sufficient and necessary conditions for optimality usually derive from the calculus of variations (COV). In order to derive the conditions for an extremum of (1.4) subject to the system (1.1) and the boundary conditions (1.3), a system Hamiltonian is introduced as $H = L + \lambda^T f$, so that the necessary conditions become [3]

$$\dot{\lambda} = -\left(\frac{\partial H}{\partial x}\right)^T \text{ with boundary condition } \lambda(T) = \left[\left(\frac{\partial \phi}{\partial x}\right) + \nu^T\left(\frac{\partial \Psi}{\partial x}\right)\right]_t^T \tag{1.5}$$

$$\frac{\partial H}{\partial u} = 0. \tag{1.6}$$

where $t = \left\{t_0, t_f\right\}$. If the final time is a variable to be optimized, a scalar equation is added to the system:

$$\left[\left(\frac{\partial \phi}{\partial t}\right) + \nu^T\left(\frac{\partial \Psi}{\partial t}\right) + \left(\left(\frac{\partial \phi}{\partial x}\right) + \nu^T\left(\frac{\partial \Psi}{\partial x}\right)\right)f + L\right]_t = 0. \tag{1.7}$$

The set of equations (1.1) - (1.7) describes a two-point-boundary-value problem (TPBVP), which is solved analytically only in elementary cases, e.g. very low-thrust orbit raising even with some perturbations. Generally, those solutions are either very challenging or impossible, thus numerical solutions are adopted. Some improved analytical solutions rely on Lawden's *primer vector* theory, *shape-based* trajectories [8] or Hamilton-Jacobi theory.

1.3.2 Numerical conversion to NLP problems

A *numerical* solution is a solution obtained not explicitly applying the analytical necessary conditions of the COV and, for instance, adjoint variables or the H-J equation. It is usually preferred to avoid the use of the necessary conditions, although they ensure that the optimal trajectory is a local minimum, are sensitive to terminal conditions and can guide to a better solution. The main reasons for the difficulties often encountered in the *indirect* approach lie in the lack of robustness in dealing with the nonlinearities of the problem and the lack of a systematic means of approximating the initial adjoint variables, which furthermore have not the physical significance of the state variables and may have discontinuities at the junctions of constrained and unconstrained arcs in the solution [3]. Hargraves and Paris ("*Direct Trajectory Optimization Using Nonlinear Programming and Collocation*", 1987) proposed in 1987 to remove the adjoint variables as long as discrete control variables were adopted as additional NLP parameters, improving the robustness of the solution and halving the problem size. Thereby, NLP solvers enable to efficiently solve larger problems.

The majority of optimal space trajectories are determined numerically through *direct* solution methods. They can parametrize in different ways the equation of motion, originally defined as function-space constraints; discretizing a continuous problem so that state and control variables are known in discrete times is probably the most successful approach. A stepwise explicit or implicit rule integrates the equations, which thus are satisfied; consequently, the NLP problem consists of a large number of nonlinear constraints. Canon et al. ("*Theory of Optimal Control and Mathematical Programming*", 1970) named this approach *direct transcription*. Implicit integration rule constraints commonly used are the trapezoidal, Hermite-Simpson, higher degree rules of the Gauss-Lobatto family or Gauss-pseudospectral method.

Since the 1960s, many direct solutions were developed with the aim of converting the continuous problem into a parameter optimization problem and substituting the *shooting* approach with a method where all the free parameters are optimized contemporaneously; some of the most implemented direct transcription methods are the finite differences and especially the collocation [9] method. Collocation involves discretizing the state and control variables in a mesh; in its basic form, the state history between mesh points is represented by Hermite cubic polynomials, while the control history by a cubic polynomial function of time. The satisfaction of the system differential equation is assured by enforcing the equations of motion at the mesh points while minimizing the objective function at the same time, thanks to the NLP problem solver. This is also supported by the collocation scheme: a collocation point is placed at the center of each mesh segment, thus Hermite cubic polynomials are constrained to satisfy the set of equations at mesh points as well as between them. If these constraints are satisfied, the system is *implicitly* integrated. Other constraints, e.g. initial or terminal values of the state's and control magnitudes, can be added.

Another direct transcription method using implicit integration, but without assuming that state histories are described by polynomials, is the Runge-Kutta (RK) explicit integration using parallel shooting (Sect. 2.3.1 for insights) [9][10], very beneficial for cases where controls are allowed to change more rapidly than the states. I adapted this method for the MATLAB computing environment, testing its effectiveness with a simple optimization example (Sect. 2.3.2) and then employing it for the asteroid mitigation problem studied in this work (Sect. 2.3.3 and 3.4).

Direct transcription methods are advantageous since there are no costate variables, reducing considerably the problem size, and since the structure of the problem does not have to be specified in detail *a priori*, enabling to make fewer assumptions (e.g. the number of coast/thrust arcs needed, which if not used can be automatically removed). Moreover, these schemes are characterized by better robustness and the ability to converge to an optimal trajectory surprisingly when the initial guess provided to the NLP solver is a "worse" guess, compared to other numerical optimization methods. From personal experience, during the current research getting numerical solutions to problems has often been an "art" and not always a systematic, straightforward, logical process. Sometimes it is recommended to increase the number of nodes used or to re-evaluate the bounds of some parameters as a strategy to improve the problem solution.

Methods such as collocation have been particularly successful for space optimization problems, including low-thrust orbit raising, Earth-Moon transfer and interplanetary transfers. However, despite their robustness, a reasonable initial guess of the parameters has to be supplied in a discretized form of state and control time-histories along the optimal trajectory to be found by the NLP solver. Evolutionary or heuristic methods can be efficiently used to generate a solution that will be employed as the initial guess of a much more accurate method (i.e. a direct transcription method).

1.3.3 Evolutionary and heuristic methods

Evolutionary and heuristic algorithms are structurally simpler and easier to learn than other extant methods, also they are better suited to locate global minima compared to conventional optimizers. The Genetic Algorithm (GA) and the Particle Swarm Optimizer (PSO) are respectively the most used evolutionary and heuristic methods.

Evolutionary algorithms (EA) optimize numerically a set of discrete parameters, relevant to the problem solution, by using mechanisms inspired by biological evolution. They randomly generate a population of initial solutions, so that an actual initial guess is not needed. Yet, the vector of parameters cannot be as large as it could using a nonlinear program. In a spacecraft optimization problem [3], this might be possible if, for instance, the trajectory is naturally described by a finite set (e.g. times, magnitudes and directions of the thrust for an impulsive trajectory). If the trajectory contains non-integrable arcs (e.g. low-thrust arcs), a small number of parameters (e.g. departure and arrival times for each arc and polynomial coefficients for thrust pointing time-history) would be also sufficient to describe it. Another case is using a *shape-based* method (Sect. 3.3.1) [8], where the "shape" of the optimal trajectory is defined by few parameters which describe a solution of the system equations of motion; however, the thrust pointing history associated can be derived only *a posteriori*. So, an EA chooses the parameters that will satisfy the boundary conditions and will minimize the cost. If the result is not satisfactory enough, it might be used as the initial guess of a direct method for instance.

The genetic algorithm (GA), developed by John Holland in the 1960s [11][12], is a model of biological evolution based on Charles Darwin's theory of natural selection. In its simplest form, the parameters are defined as a sequence of numbers, representing a chromosome, but usually converted to binary form. Every string can be decanted yielding to a cost and corresponding trajectory. At first, an initial population is randomly generated and the costs may be really large or even infeasible. The main steps yielding to the solution consist of selection, combination and mutation. Through selection, the worst sequences are removed, but the élite sequence survives and is kept in the next iterations. The remaining sequences are the "parents" of "new individuals", obtained by partially combining two parents' sequences. Mutation requires that a randomly chosen bit be changed for a part of the population. The process is repeated until a termination condition, such as a fixed number of iterations or when the cost reaches a plateau. A great advantage of this method is that the objective cannot worsen since the best sequence is always preserved, and usually a rapid improvement can be experienced since the first generations. Although, those termination conditions do not ensure that a minimum has been found nor necessary conditions for optimality have been imposed or boundary conditions enforced. A “penalty function” is generally added to the cost because the GA (or PSO) has no direct way of accounting for constraints (e.g. rendezvous), unless the conditions are met. By including this additional term, the solution may be less accurate than in other conventional methods. Thus, GA may provide initial guesses for indirect or direct methods capable of more precise solutions. Furthermore, multi-objective GAs have been developed for solving multi-objective problems, where the purpose is to investigate a set of solutions that singularly satisfies the objectives at an acceptable level without being dominated by any other through specialized fitness functions and promoting solution diversity [13]. A GA routine is provided by MATLAB.

A qualitatively different method is the heuristic method called PSO. This method will be further explored in Chapter 2, where some simple examples will also be provided, and in Chapter 3, where it was employed for the final problem covered by this dissertation.

1.4 Objective

The present research has been developed starting from a very basic knowledge of numerical solutions for trajectory optimization, which has been improved progressively throughout this work. Simple optimization problems have been solved in order to learn step-by-step the orbital mechanics tools needed for solving more sophisticated problems. Direct transcription and heuristic methods have been deepened and implemented, as well as other required tools such as NASA’s Mice package, especially the SPICE routines. The ultimate objective of this dissertation is the full simulation of optimal asteroid mitigation (Apophis is the chosen asteroid) using a low-thrust spacecraft and kinetic impact. The goal is to maximize the perigee radius of the deflected asteroid at its closest approach to Earth, and it is reached by optimizing important variables such as the date of Earth departure, the direction of the departure, the thrust program for the low-thrust motor, and the date of the collision. This continuous optimal control problem has been transcribed into a NLP problem, whose initial guess is provided by approximately solving an equivalent PSO program.

This project was developed in collaboration with L. Malagni, with whom I worked synergistically on the skeleton of the program before specializing in specific tasks. In terms of personal effort, particular attention will be paid to the Runge-Kutta (RK) parallel shooting code, developed for the first time in the MATLAB environment, and to the Particle Swarm Optimization (PSO) program. While I took care of these tasks, my colleague L. Malagni evaluated the use of the State Transition Matrix (STM) and implemented it for the asteroid trajectory propagation post-interception. A more detailed description of the full program will be provided in the next chapters.

1.5 Thesis Outline

Chapter 2 consists of the description of the numerical optimization methods implemented, which are the Particle Swarm Optimization (PSO) and two MATLAB NLP solver routines (*fmincon*, *SNOPT*) using different numerical integration methods (MATLAB *ode45* and *ode113*, Euler step, 4th degree RK rule). The chosen methods are analyzed in detail through simple examples, approaching the final problem step by step.

Chapter 3 is entirely dedicated to the optimal asteroid mitigation problem. At first, an overview of space missions towards asteroids, deflection strategies and prior work are shown. Then, the Chapter develops by getting to the heart of the problem subject of this dissertation, i.e. describing the simulation and the optimization parameters involved. Moreover, it is explained how the initial guess for the NLP problem was derived and the outcomes. Finally, the results of the global program are illustrated and commented on.

The conclusive chapter is aimed at summarizing the main results obtained from this work and proposing possible further steps that could be considered in the future.

Chapter 2: Numerical Optimization Methods

This chapter describes in detail the numerical optimization methods this work is based on, namely the PSO approach, the use of an NLP solver and the direct RK parallel-shooting method. Some applications for each solving method will be shown, as they have been developed to familiarize with these algorithms as well as to create the building blocks for the final problem.

2.1 Particle Swarm Optimization of a dynamic system

2.1.1 The PSO method

Particle Swarm Optimization (PSO) is a heuristic optimization technique developed by Eberhart and Kennedy [14] in 1995. PSO is inspired by the collective behavior of simple individuals interacting with their environment and each other. It is often referred to as *swarm intelligence* because it emulates the social pattern of bird flockings [3][15][16]. Regarding space trajectories, the interest is generally in minimizing a given objective function which depends on the time history of a dynamical system, ruled by differential or algebraic equations.

PSO is quite similar to evolutionary computation methods such as Genetic Algorithm (GA) (Sect. 1.3.3) in that the system is initiated with a population (e.g. $N = 100$ individuals) of solutions randomly distributed in a decision parameter space or, in some cases, given by the user - if a well-suited population is already known, for instance, from a similar case of optimization problem. However, unlike a GA, each potential solution i - called *particle* - is associated with a position vector $\vec{\chi}(i)$ and a velocity vector $\vec{w}(i)$, with which it moves through hyperspace. The optimization process seeks the optimal values of n unknown parameters, which constitute the elements of the position vector

$$\vec{\chi}(i) \equiv [\chi_1(i)\ldots\chi_n(i)]^T, \;\; with\ i = 1, \ldots, N \tag{2.1}$$

constrained to their respective ranges

$$a_k \leq \chi_k \leq b_k, \;\; with \;\; k = 1, \ldots, n. \tag{2.2}$$

The velocity vector, defined similarly to the position vector, identifies the position update and is consequently bounded to suitable ranges

$$-d_k \leq w_k \leq d_k \quad if \; d_k \equiv b_k - a_k, \;\; with \;\; k = 1, \ldots, n. \tag{2.3}$$

The PSO algorithm keeps track of the coordinates of the particles, which are assigned a fitness value derived from the objective function. If the position vector of particle *i* at the *j*th time step is denoted by $\overrightarrow{\chi}^{(j)}(i)$ and the particles take a step $\overrightarrow{w}^{(j+1)}(i)$ in the parameter space, the new position of the *i*th particle becomes [15]

$$\chi_k^{(j+1)}(i) = \chi_k^{(j)}(i) + w_k^{(j+1)}(i) \tag{2.4}$$

with

$$w_k^{(j+1)}(i) = c_I w_k^{(j)}(i) + c_C \, [y_k^{(j)}(i) - \chi_k^{(j)}(i)] + c_S \, [\hat{y}_k^{(j)} - \chi_k^{(j)}(i)] \tag{2.5}$$

where $w_k^{(j)}(i)$ is the step for component *k* of the *i*th particle at the *j*th time step, $\chi_k^{(j)}(i)$ is the *k*th component of the position of the *i*th particle at the *j*th time step, and *c*’s values represent stochastic weights. $y_k^{(j)}(i)$ is the “personal” best position, that is the best position that the *i*th particle has ever reached since its very first time step; $\hat{y}_k^{(j)}$ is the “global” best position, i.e. the best position that any particle of the swarm has ever detected so far. Three main components can be distinguished in equation (2.5): the first is the “inertia” which for each particle is proportional to its velocity in the preceding iteration, so as to drive the particle to move in the same direction in which it had previously been moving; the second is “cognitive” term that promotes the direction towards its own personal best; finally, the third is the “social” component which attracts the particle toward the most satisfying position ever experienced. The importance of each component is weighed by the *inertial* (c_I), *cognitive* (c_C) and *social* (c_S) constants, assessed by experience or by conventional definitions. In particular, widely used definitions are [15][17]

$$c_I = \frac{1 + r_1(0,1)}{2} \quad c_C = 1.49445 r_2(0,1) \quad c_s = 1.49445 r_3(0,1) \tag{2.6}$$

where $r_1(0,1)$, $r_2(0,1)$ and $r_3(0,1)$ are independent random numbers uniformly distributed within the interval [0, 1]. In this work, the three constants were simply tuned to $\{0.65,\ 2,\ 2\}$ as they were proved effective. Moreover, if a particle hits a boundary, the d_k value is either subtracted or added depending on whether the upper or the lower bound has been crossed.

In general, space optimization problems involve equalities and/or inequalities related to the unknown parameters. Equality constraints are more difficult to deal with since the degrees of freedom of the problem are reduced by their number. Indeed, $m \leq n$ are the admissible number of equality constraints

$$d_r(\vec{\chi}) = 0 \quad with\ r = 1, \ldots, m. \tag{2.7}$$

Usually, equality constraints are added to the objective function as a "penalty":

$$\tilde{J} = J + \sum_{r=1}^{m} \alpha_r |d_r(\vec{\chi})| \tag{2.8}$$

where the coefficients α_r are problem-dependent. Instead, inequality constraints just reduce the search space of feasible solutions and, so, they are less troublesome; inequalities can be treated by assigning a fictitious infinite value to the fitness function if at least one of those constraints are not satisfied. Also, the corresponding velocity can be set to zero, so that the inertial term does affect the subsequent velocity update. The PSO algorithm can be terminated, as for GA, imposing a fixed number of iterations or a condition that stops it when the global best has not changed for several iterations.

Therefore, the PSO method is quite robust, simple and is particularly suited for locating global minima when the solution space contains many local minima.

2.1.2 Application of the PSO approach to the Hohmann transfer problem

Published in 1925, Hohmann's conjecture concerns the two-impulse elliptical transfer between two coplanar circular orbits with the minimum fuel consumption [18][19][20]. Given the assumption of two-body-problem and absence of perturbations, the transfer is Keplerian and tangent to both of the terminal orbits in its periapsis (r_1) and apoapsis (r_2). This trajectory is usually optimal because it requires the minimum $\Delta v_{tot} = |\Delta v_1| + |\Delta v_2|$, except in cases of very large ratios of $R = r_2/r_1$ (≥ 1). Simply solving Lambert's problem by inspection, the semimajor axis can be determined as

$$a_H = \frac{r_1 + r_2}{2} \quad . \tag{2.9}$$

The transfer time is half of the Hohmann ellipse:

$$t_H = \pi \left(\frac{a_H^3}{\mu} \right)^{1/2} . \tag{2.10}$$

The eccentric can also be easily calculated as

$$e_H = \frac{r_2 - r_1}{r_1 + r_2} \quad . \tag{2.11}$$

From the conservation of energy equation, the velocities of the transfer orbit at perigee and apogee are

$$v_p^2 = \mu \left(\frac{2}{r_1} - \frac{2}{r_1 + r_2} \right) \tag{2.12}$$

$$v_a^2 = \mu \left(\frac{2}{r_2} - \frac{2}{r_1 + r_2} \right) \tag{2.13}$$

Given the velocities of the initial and final circular orbits $v_{c1} = \sqrt{\mu / r_1}$ and $v_{c2} = \sqrt{\mu / r_2}$, the required impulses and the perigee and apogee are respectively

$$\Delta v_1 = v_p - v_{c1} = \sqrt{\frac{\mu}{r_1}} \left(\sqrt{\frac{2r_2}{r_1 + r_2}} - 1 \right) \tag{2.14}$$

and

$$\Delta v_2 = v_{c2} - v_a = \sqrt{\frac{\mu}{r_2}} \left(1 - \sqrt{\frac{2r_1}{r_1 + r_2}} \right) . \tag{2.15}$$

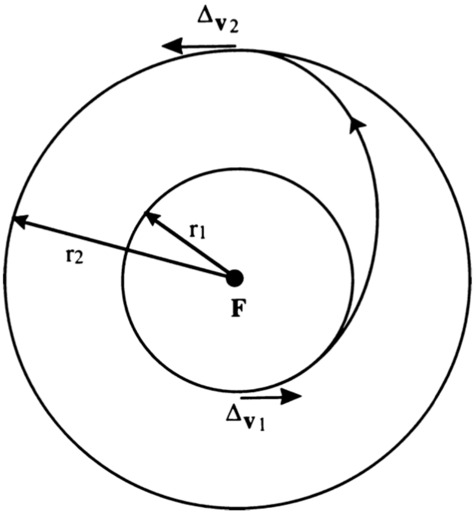


Figure 2.1 Hohmann Transfer Orbit [19]

In the case chosen (*Figure 2.1*) the initial orbit has a smaller radius than the final orbit, but the same strategy can be followed if it were larger, adopting “negative” impulses (i.e. opposite to the direction of motion) first at apogee and then at perigee to decelerate the satellite.

The equations of motion governing the dynamic system are formulated in polar coordinates as

$$\begin{cases} \dot{r} = v_r \\ \dot{\theta} = v_\theta / r \\ \dot{v}_r = \frac{v_\theta^2}{r} - \frac{\mu}{r^2} \\ \dot{v}_\theta = -\frac{v_r v_\theta}{r} \end{cases} \tag{2.16}$$

where $\{r, \theta, v_r, v_\theta\}$ is the state vector. The equations are integrated by the MATLAB solver *ode45* for non-stiff differential equations, which is sufficiently accurate. The integration stops when the event of reaching the final orbit occurs.

Five parameters are optimized, i.e. the two Δv magnitudes, their pointing angles (β_1 and β_2) and the time of flight, according to the following cost function:

$$J = \Delta v_1 + \Delta v_2 + c_1 |ceq| \quad \text{with } ceq = |v_{r2}| + |v_{t2} - v_{c2}| + |r_2 - r_{2*}| \tag{2.17}$$

where ceq is the penalty function multiplied by a constant (c_1) and represents the constraints to be satisfied. While the first two terms in the ceq expression represent the radial and tangential velocity conditions for a satellite that is inserted in a circular orbit, the third righthand term is the difference between the final radius derived from the integration and the radius chosen for the final orbit.

This simple problem is implemented in normalized (or canonical) units (e.g. $\mu = 1\ DU^3/TU^2$, where DU is a distance unit and TU a time unit) since the PSO solver works better if its optimization parameters are of about the same order of magnitude. Velocities are normalized with respect to the circular velocity v_{c1}. Setting $r_1 = 1\ DU$ and $R = 3, c_1 = 200$, 100 swarm particles and 500 iterations, the results are shown in *Table 2.1* - where they are compared with the analytical results - and in *Figure 2.2*.

Table 2.1 Hohmann Transfer Orbit problem

Variable	PSO result	Analytical result
Δv_1	0.2247	0.2247
β_1	10^{-5}	0
Δv_2	0.1691	0.1691
β_2	0.0079	0
time of flight	8.909	8.886

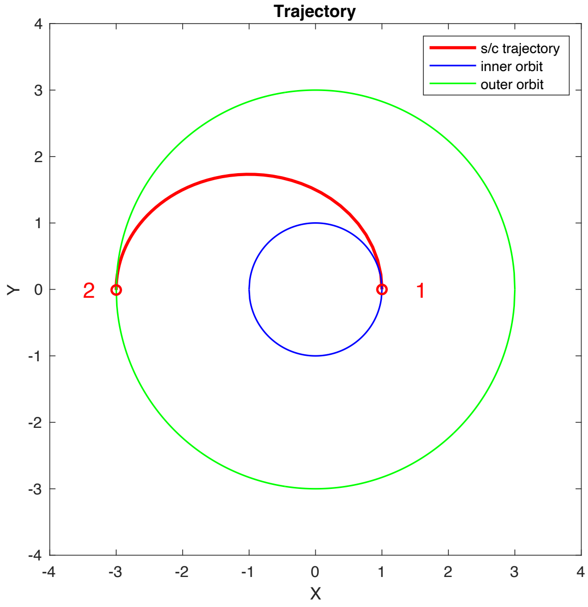


Figure 2.2 Hohmann Transfer Orbit resulting from PSO

From the comparison, it is clear that the PSO solution is very encouraging; in fact, $ceq \approx 10^{-17}$ so that $J = 0.3939 \approx \Delta v_{tot}$. The accuracy of the result can be improved, for instance, by increasing the number of iterations and adjusting the number of particles in the PSO algorithm.

2.1.3 Low-thrust problem of final energy maximization using PSO

The example under investigation is a continuous low-thrust orbit transfer problem, where the acceleration (i.e. the thrust divided by the mass of the spacecraft) imparted to the spacecraft is constant and with a magnitude equal to $A = 0.025\ DU/TU^2$. The objective is to maximize the final energy of the satellite in a specified time, $t_f = 16\ TU$, starting from an orbit with a radius $r_1 = 1.1\ DU$. The only control variable is the thrust angle (β), measured from the local horizontal. The equations of motion that describe the dynamical system are the same as in Sect. 2.1.2 (equations (2.16)) - so written in polar coordinates and their origin is located at the center of mass of the attracting body -, but with the addition of the thrust component:

$$\begin{cases} \dot{r} = v_r \\ \dot{\theta} = v_\theta / r \\ \dot{v}_r = \frac{v_\theta^2}{r} - \frac{\mu}{r^2} + Asin(\beta) \\ \dot{v}_\theta = -\frac{v_r v_\theta}{r} + Acos(\beta) \end{cases} . \tag{2.18}$$

The cost function is formulated as

$$J = -\left(\frac{V^2}{2} - \frac{1}{r_f}\right) \tag{2.19}$$

where the negative sign is put so that the expression in the parenthesis is maximized. In this example, there are no particular constraints to be enforced in the objective function.

The optimal control variable can be parameterized in different ways. The strategy adopted is to discretize the trajectory into (e.g. 17) control points, to which a certain β value is associated, and then observe the time trend to approximate it with a function; in particular, the following sinusoidal waveform has been used:

$$\beta = (a + bt) \cdot sin(\omega t + \phi) \tag{2.20}$$

where a, b, ω and ϕ are the new optimization parameters. Although the function (2.20) does not exactly correspond to the optimal evolution of the thrust pointing angle - which could be obtained by increasing the number of points of the trajectory discretization -, it gives an idea of its trend with fewer parameters, allowing the PSO solver to work more efficiently (it works well up to about 30 parameters) as well as providing a smoother representation than in the case with a few points.

The programs are written in normalized units and the results are compared in *Figures 2.3, 2.4* and *2.5*. The PSO uses 100 particles and 500 generations.

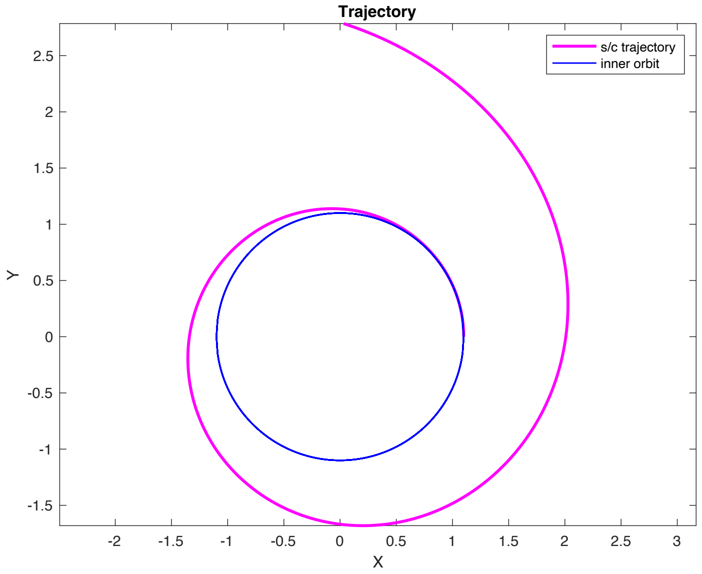
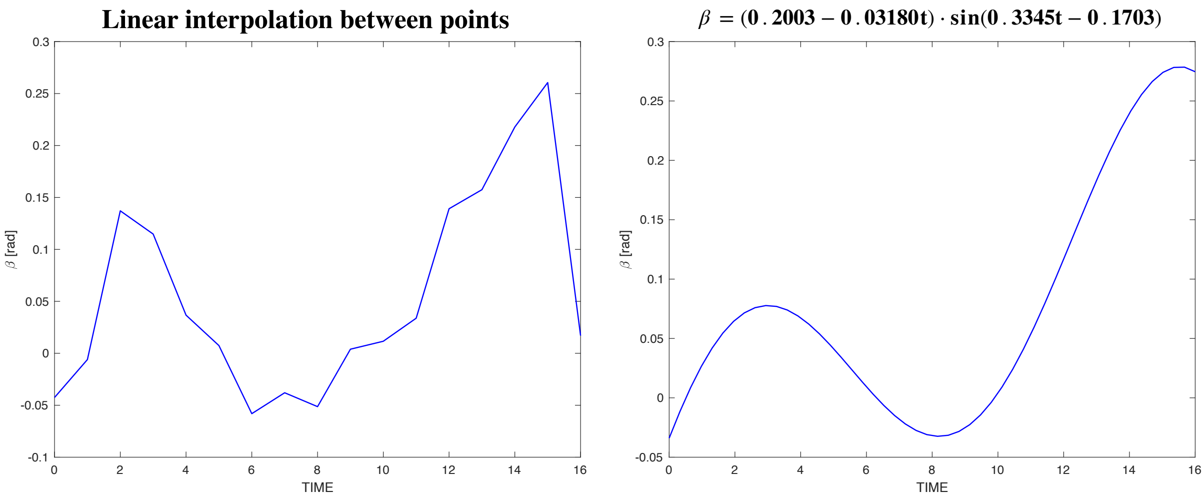


Figure 2.3 Low-thrust trajectory for final energy maximization

Figure 2.3 shows the spacecraft trajectory that results from the case using equation (2.20) to approximate the thrust pointing angle time-history. The result of the other case is essentially the same as that shown in Figure 2.3, but the trajectory is segmented due to poor discretization of β (only 17 control points); by increasing the number of points the solution approaches that shown.


Linear interpolation between points

$\beta = (\mathbf{0.2003} - \mathbf{0.03180t}) \cdot \mathbf{sin(0.3345t} - \mathbf{0.1703)}$

Figure 2.4 Thrust pointing angle trend for final energy maximization using PSO

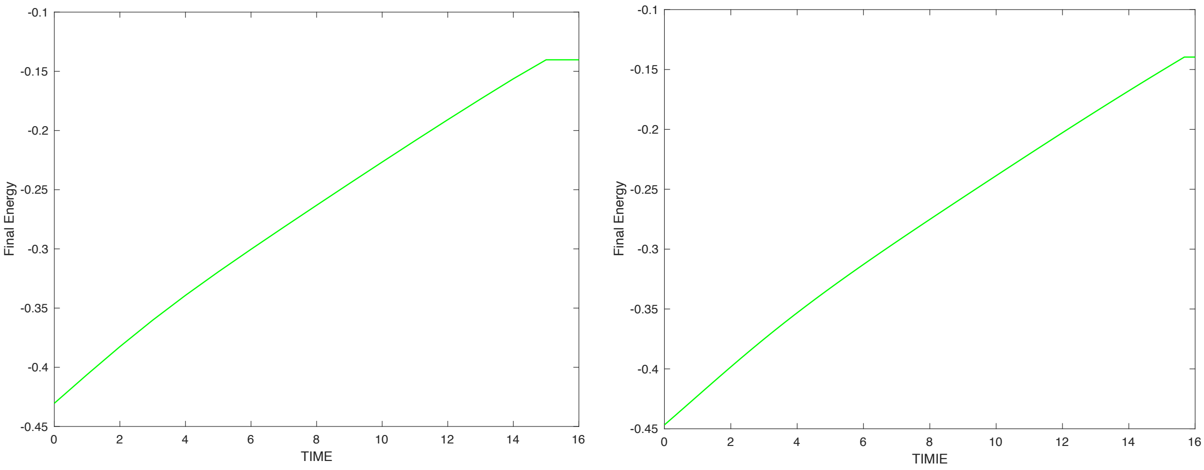


Figure 2.5 Spacecraft energy trend

In *Figures 2.4* and *2.5*, the left side shows the result derived from linear interpolation between the control points, while the right side shows the result obtained from using the sinusoidal waveform (2.20) to describe β. In *Figure 2.5* the flat segment depends only on how the control variable is parametrized. The second case well approximates the first one, indeed the objective functions are respectively equal to $|J| = 0.1403$ and $|J| = 0.1396$.

In order to check the validity of the results, a code has been developed to solve the opposite problem, that is to find the minimum final time corresponding to the value of final energy previously obtained; indeed, the resulting final time is $t_f = 15.9998\ TU$ with the energy constraint satisfied to the order of 10^{-6}.

2.2 Optimization of a dynamic system using NLP solver

Although the PSO solver ensures the achievement of the optimal solution by keeping the best swarm score at each iteration, the validity of the solution strongly depends on how the objective function has been formulated. In order to force the satisfaction of the equations of motion and constraints, it is necessary to introduce a nonlinear solver. Such an optimization problem can have extreme accuracy but needs an initial guess close to the global minimum. This initial guess can be obtained, for example, from a PSO problem.

2.2.1 Nonlinear optimization programs

The main nonlinear programming solver employed is the MATLAB function *fmincon* [21], which finds the minimum of a constrained nonlinear multivariable function as

$$\min_{x} \ f(x) \ such \ that \begin{cases} c(x) \le 0 \\ ceq(x) = 0 \\ A \cdot x \le b \\ Aeq \cdot x = beq \\ lb \le x \le ub \end{cases} \tag{2.21}$$

where b and beq are vectors, A and Aeq are matrices, $c(x)$ and $ceq(x)$ are constraint functions that return vectors, and $f(x)$ is a function that returns a scalar. Moreover, $f(x)$, $c(x)$ and $ceq(x)$ can be nonlinear functions, whereas the solution x, the lower bounds lb and the upper bounds ub can be passed as vectors or matrices. *Fmincon* is expressed in MATLAB as

$$[x, fval] = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon, options) \tag{2.22}$$

where $fval$ is the result of the objective function specified in an m-file fun, $x0$ is the initial guess vector, $nonlcon$ is an m-file that defines the minimization to the nonlinear inequalities $c(x)$ or equalities $ceq(x)$. Specific $options$, such as display features, can be set.

Among the outputs, it may be useful to visualize the "feasibility", which represents the satisfaction of the constraints and therefore must be as small as possible. Another typical output that must be small is the "first-order optimality," which is a measure of how close the solution is to the optimum.

This routine has the disadvantage of not being able to manage a huge amount of parameters (just up to about 400), since the computational speed would slow down a lot and it would lose precision. This is the reason why another nonlinear programming solver is investigated, namely *snsolve* [22][23] (i.e. the implementation of the program SNOPT in the MATLAB environment) which can handle thousands of parameters.

SNOPT is a general-purpose routine for constrained optimization, suitable especially for nonlinear programs. In particular, SNOPT uses a sequential quadratic programming (SQP) algorithm on problems where the goal is to minimize a linear or nonlinear function subject to bounds on the variables and sparse linear or nonlinear constraints:

$$\begin{gathered} Minimize\ J(P) \\ subject\ to\ l \leq \begin{pmatrix} P \\ f(P) \\ A_L(P) \end{pmatrix} \leq u \end{gathered} \tag{2.23}$$

where l and u are constant lower and upper bounds, $J(P)$ is a smooth scalar function, $A_L(P)$ represents any linear constraints, and $f(P)$ is the set of smooth nonlinear constraint functions. In general, the solutions found are locally optimal, and for any nonlinear function a gradient should be explicit; otherwise, unknown gradients are estimated by finite differences. QP subproblems provide the search direction so as to minimize a quadratic model of the Lagrangian function subject to linearized constraints; in addition, an augmented Lagrangian merit function is reduced along each search direction to ensure convergence from any starting point. On large problems, SNOPT can be more efficient if many constraints are active. Indeed, unlike *fmincon*, SNOPT requires relatively few evaluations of problem functions, hence number of iterations, and is particularly efficient if the objective or constraint functions (and their gradients) are expensive to evaluate. On the other hand, from experience, it can be inferred that the initial guess chosen has even a more important role in locating the global minimum than using *fmincon*.

SNOPT can be called from a driver program in MATLAB and the call expression is quite the same as for *fmincon*. SNOPT functioning was further explored by my colleague L. Malagni.

2.2.2 Low-thrust problem of final energy maximization using MATLAB *fmincon* routine

The same problem of Sect. 2.1.3 about final energy maximization with low-thrust in a specified time may be solved through the MATLAB NLP solver, *fmincon*. The equations of motion (2.18) - integrated with *ode45* - and the objective function (2.19) are still employed, but the control variable β is discretized into 100 points along the trajectory, linearly interpolated two by two. The trajectory and the energy pattern appear as in *Figures 2.3* and *2.5*, while the thrust pointing angle trend, shown in *Figure 2.6*, is smoother compared to the PSO result with only 17 points but at the same time, no prior assumptions have been made on its evolution. In particular, the last segment of the plot that tends to zero depends only on how the interpolation has been executed.

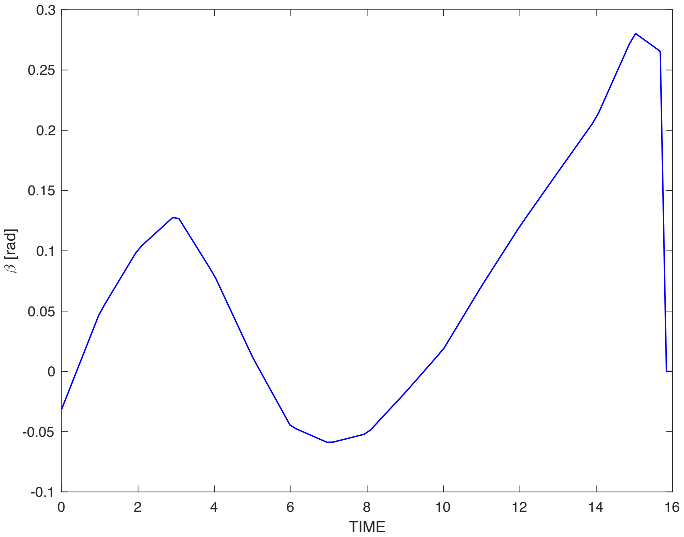


Figure 2.6 Thrust pointing angle trend for final energy maximization using *fmincon* and *ode45*

A local minimum is found for $|J| = 0.1395$, which is quite close to the solution found previously (Sect. 2.1.3) so that it might be a global minimum. The solution is found faster than in the PSO case since *fmincon* does not need a termination condition, indeed it stops when the constraints are satisfied and a minimum is located if the number of iterations or evaluations of the function does not exceed the imposed upper limit.

Since *fmincon* allows the use of more parameters, this problem has also been implemented with another integration rule, which is the Euler step rule:

$$x_{k+1} = x_k + f(x_k, u_k)\Delta t \quad for\ the\ (k+1)^{th}\ step \tag{2.24}$$

where x is the state vector, u the control vector and Δt the time-step. Euler's step rule replaces the simple call to *ode45* to integrate the EOM, leading to a very different structure of the problem. Therefore, the equations of motion are formulated in terms of the rule (2.24) as if they were constraints to be satisfied. Thus the number of NLP optimization parameters increases from 100 to 500, in this example, as four state variables computed at each control point are added to the thrust pointing parameters. This method is quite rough but it provides a good solution for simple problems where the step size is small enough. On the other hand, *ode45* is step-variable and much more accurate than Euler's method, which is generally only first-order accurate and may require several more simulation time-steps to get the same accuracy. The most significant plot to be shown is the thrust pointing angle over time in *Figure 2.7*; given

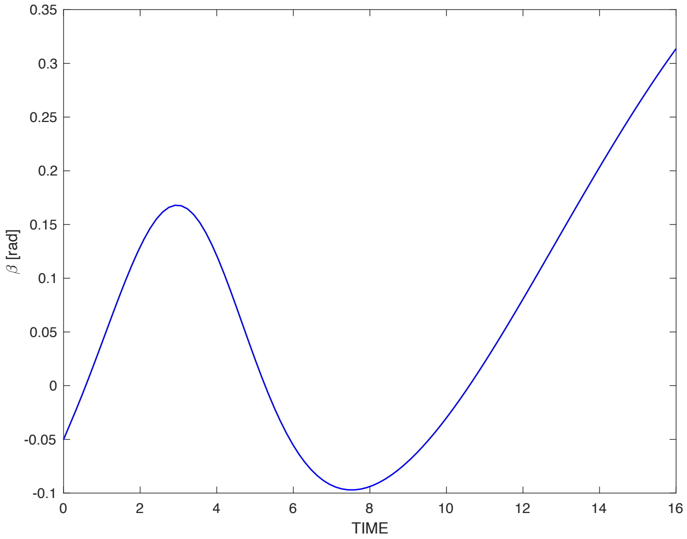


Figure 2.7 Thrust pointing angle trend for final energy maximization using *fmincon* and Euler step

the same number of discretization points, the following graph is much smoother because the time step is constant and therefore it is not necessary to describe the behavior of the variable between two nodes with, for instance, a linear interpolation.

The minimum is located at $|J| = 0.1419$, with the feasibility of the order 10^{-15} and the first-order optimality of 10^{-6}.

2.2.3 Fictional asteroid interception problem in 2D coordinates

The problem under consideration introduces the issue of the interception of an asteroid by an Earth-launched spacecraft. The program can choose the optimal time-variables for the launch and interception events - thus waiting for the optimal "geometry" of the Earth and asteroid in the heliocentric reference - as well as the pointing angles for the low-thrust propulsion of the spacecraft.

The system of units used is based on a two-body problem where the Sun is the central body and the reference orbit is a circular orbit whose radius is the mean distance from the Earth to the Sun, that is one astronomical unit ($a_{\oplus} \approx 1\ AU \approx 1.496 \cdot 10^{8}\ km$) [19][20]. Instead, time variables are given in terms of time units, so that $2\pi\ TU$ is about the time employed by the Earth to revolve around the Sun:

$$2\pi\ TU = 2\pi\sqrt{\frac{a_{\oplus}^{3}}{\mu_{\odot}}} \Rightarrow 1\ TU \approx 5.02 \cdot 10^{6}\ sec. \tag{2.25}$$

Velocities are expressed in AU/TU ($1\ AU/TU \approx 30\ km/sec$) and the Sun gravitational parameter used as reference results equal to $\mu_{\odot} \approx 1.327 \cdot 10^{-11}\ km^{3}/s^{2} \approx 1\ AU^{3}/TU^{2}$. No perturbation is yet taken into account and planetary orbits are Keplerian. For this initial program, Earth's orbit is assumed to be circular - the mean motion is $n_{\oplus} = 1\ rad/TU$ -, but this approximation is acceptable given Earth's small eccentricity ($e_{\oplus} = 0.0167$).

The asteroid that needs to be impacted is non-hazardous and fictional, so its semi-major axis and its eccentricity are arbitrarily assigned as $a = 2.5\ AU$ and $e = 0.2$. In this example, the asteroid is at its perihelion at the opening of the launch window), forming an angle of 30° to the Earth-Sun radius vector at that time. In order to propagate the position of the asteroid, the Kepler equation is applied [18]:

$$M = E - esin(E) = (t - t_p)\sqrt{\frac{\mu}{a^3}} \tag{2.26}$$

where M is the mean anomaly, E is the eccentric anomaly at a time t and $(t - t_p)$ is the time required to reach the position corresponding to the angle E from the periapsis. This equation is implemented through Newton's method, whereby at the (k+1)th time-step

$$E_{k+1} = E_k - \frac{F(E_k) - M}{F'(E_k)} \qquad with \quad E_{k=1} = M, \;\; F(E) = E - esin(E). \tag{2.27}$$

In particular, the mean anomaly can be rewritten as

$$M = (E_0 - esinE_0) + (t - t_0)\sqrt{\frac{\mu}{a^3}} \tag{2.28}$$

where E_0 is the eccentric anomaly at the time t_0 (in this case $E_0 = 0 \; rad$). Given E at a certain time t, the inverse Kepler problem is solved by calculating the correspondent true anomaly (f) and radius (r) as follows:

$$cos(f) = -\frac{e - cos(E)}{1 - ecos(E)} \Rightarrow f(t) \tag{2.29}$$

$$r = \frac{a(1 - e^2)}{1 + ecos(f)} \; . \tag{2.30}$$

The equations of motion that govern the spacecraft dynamics are the same described in Sect. 2.1.2 (2D polar equations (2.18)), with a constant acceleration of magnitude $A = 0.05 \; AU/TU^2$. Regarding the objective function, priority has been reserved to minimizing the time required to intercept the fictional asteroid that might pass close to or impact the Earth; therefore, post-interception events are not accounted for. However, minimizing the time could result in the asteroid coming even closer to Earth rather than intercepting the asteroid in the direction that will deflect it away from it. The interception event is guaranteed by a specific constraint on the position:

$$ceq = \sqrt{(x_{\circledast} - x_{s/c})^2 + (y_{\circledast} - y_{s/c})^2 + (z_{\circledast} - z_{s/c})^2} \, |_{t=t_I} . \tag{2.31}$$

This simple problem is solved using the *fmincon* routine and the Euler step integration rule, so that, as previously stated, interpolation between nodes where the thrust pointing angles are optimized is not required as it would occur using *ode45*; therefore, also the states become optimization parameters. An initial guess is easily found through a converted PSO program, which, instead, employs *ode45*. The trend of the pointing angle is described by a sine function (2.20) whose coefficients constitute the optimization parameters along with the launch and interception time; reasonable bounds and the shape of this function were established through a trial-and-error process. A guess for the states is provided by the output of the *ode45* routine for the ultimate time variables and thrust pointing angles. Results are shown in *Figures 2.8* and *2.9*, where the trend of the guess is compared to the final result. In particular, in order to get the following outcomes, 200 particles and 500 iterations were used in the PSO algorithm and the continuous history of β was discretized into 100 points.

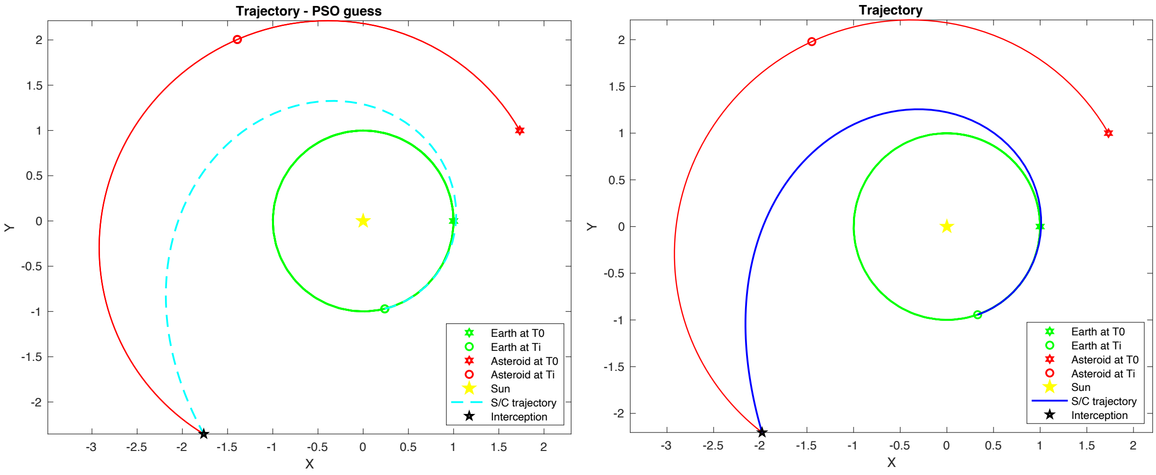


Figure 2.8 Fictional asteroid interception 2D: trajectories

Figure 2.8 shows that since the launch window opened, the Earth travels for about four-fifths of an orbit before the spacecraft is launched. The flight time is 9.7815 TU according to the PSO guess, while it is equal to 9.1866 TU in the final solution. Observing *Figure 2.9*, the trend found with the NLP program (*green*) deviates quite significantly from the evolution of the guess (*red*), especially in the second half, and then it converges at the end. In particular, the constraints are satisfied in the PSO problem to the order of 10^{-6}, while the ultimate feasibility is about 10^{-8}. It can be inferred that the PSO guess is very different from the final solution, but it is "good enough" to allow *fmincon* to converge.

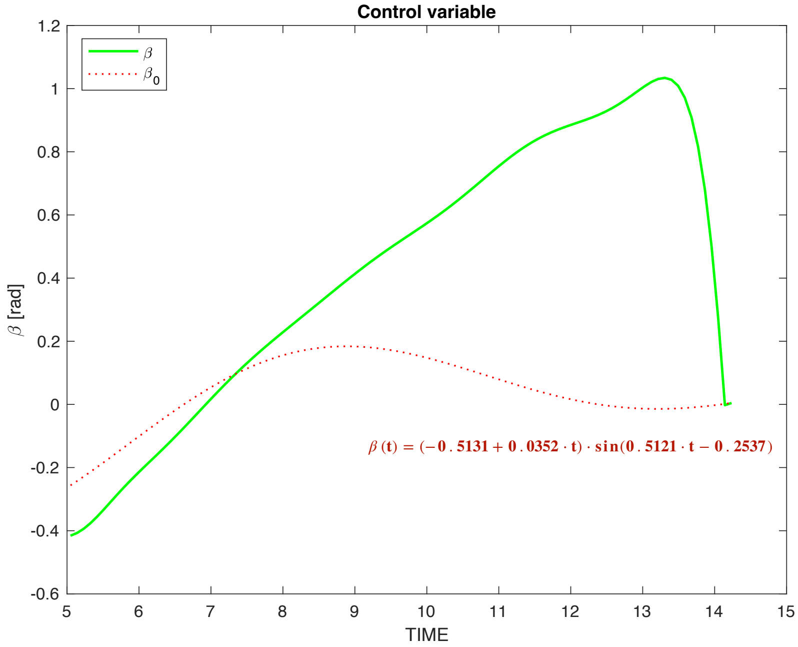


Figure 2.9 Fictional asteroid interception 2D: thrust pointing angle

2.2.4 Fictional asteroid interception problem in 3D coordinates

The problem of the 3D interception of a fictional asteroid by a low-thrust spacecraft is analogous to that discussed in Sect. 2.2.3. In fact, the chosen fictional asteroid is the same as well as the objective function; only the out-of-plane thrust pointing angle is added to the optimization parameters. The orbit of the Earth lies in the ecliptic, whereas the asteroid orbit has an inclination of 10° with respect to it. In particular, in this 3D version, the orbital parameters of the asteroid are shown in *Table 2.1*. The motion of the asteroid is propagated through Kepler's equation and its cartesian coordinates are derived through Proper Euler transformation as follows [20]:

$$\begin{cases} x = r\left(cos(\Omega)cos(\theta) - sin(\Omega)sin(\theta)cos(i)\right) \\ y = r\left(sin(\Omega)cos(\theta) + cos(\Omega)sin(\theta)cos(i)\right) \\ z = rsin(\theta)sin(i) \end{cases} \tag{2.32}$$

where $\theta = \omega + f$. Furthermore, the spacecraft equations of motion (EOM) are formulated both in cylindrical and cartesian coordinates.

Table 2.1 Fictional Asteroid Orbital Elements

Symbol	Description	Value
a	*semi-major axis*	2.5 AU
e	*eccentricity*	0.2
i	*inclination*	10°
ω	*argument of periapse*	90°
Ω	*longitude of ascending node (RAAN)*	300°
f	*true anomaly*	function of time

The EOM in cylindrical coordinates are formulated as

$$\begin{cases} \dot{r} = v_r \\ \dot{\theta} = v_\theta / r \\ \dot{z} = v_z \\ \dot{v}_r = \frac{v_\theta^2}{r} - \frac{\mu_\odot r}{(r^2 + z^2)^{3/2}} + Asin(\beta)cos(\gamma) \\ \dot{v}_\theta = -\frac{v_r v_\theta}{r} + Acos(\beta)cos(\gamma) \\ \dot{v}_z = -\frac{\mu_\odot z}{(r^2 + z^2)^{3/2}} + Asin(\gamma) \end{cases} \cdot \tag{2.33}$$

Using the cylindrical coordinates, it might be possible to exploit the 2D solution as a starting point to construct the initial guess of the 3D problem by gradually increasing the inclination of the asteroid's orbit. This method is called *homotopy* and basically consists in using an already satisfactory solution to generate a convergent series of solutions for nonlinear systems by slightly changing certain parameters of the problem. In this case, the homotopy approach was not sufficient as the solution struggled to converge; therefore, a PSO program with the same principle as in the 2D example was employed. In particular, in the non-linear program, the Euler step method is implemented (eq. (2.24)); consequently, the states also are optimization parameters and the EOM become constraints in addition to the interception event (2.34) expressed as

$$ceq = \sqrt{(x_{\circledast} - x_{s/c})^2 + (y_{\circledast} - y_{s/c})^2 + (z_{\circledast} - z_{s/c})^2} \,\Big|_{t=t_I} \tag{2.34}$$

The results illustrated in *Figures 2.10*, *2.11* and *2.12* were obtained setting 200 particles and 400 iterations in the PSO program, and discretizing the thrust pointing angles with 100 points in the NLP program using *fmincon*.

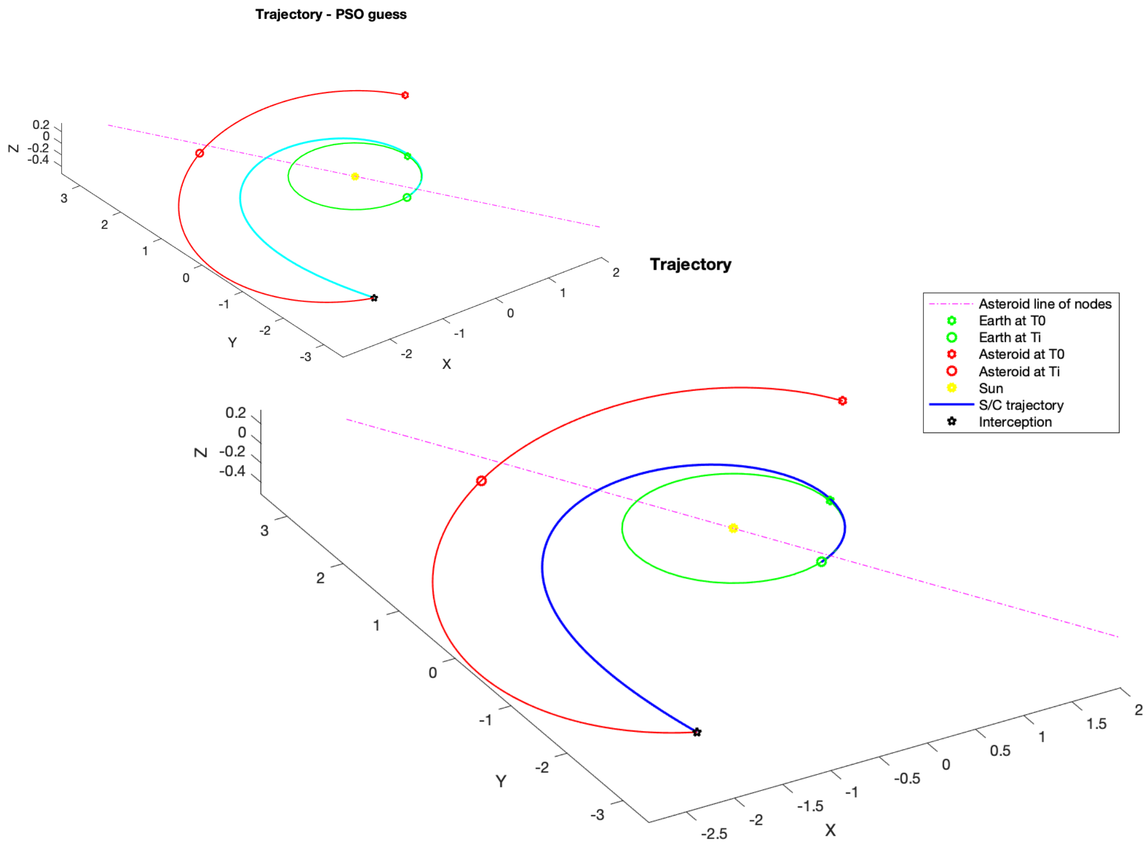


Figure 2.10 Fictional asteroid interception 3D - cylindrical coordinates case: trajectories

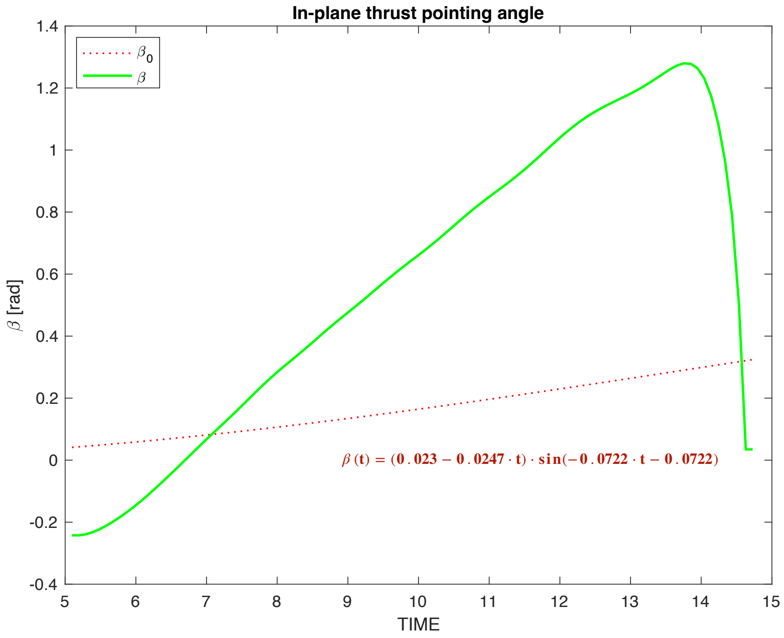


Figure 2.11 Fictional asteroid interception 3D - cylindrical coordinates case: in-plane thrust pointing angle

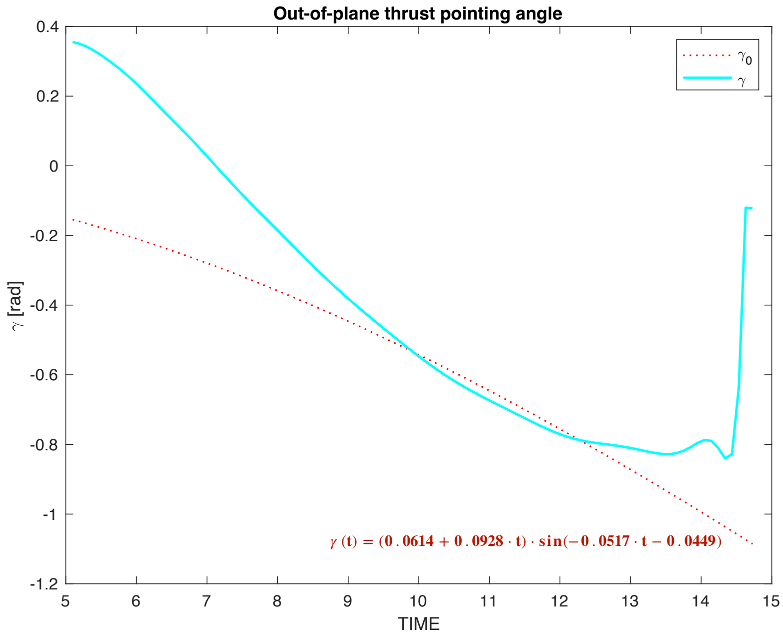


Figure 2.12 Fictional asteroid interception 3D - cylindrical coordinates case: out-of-plane thrust pointing angle

Figure 2.10 shows the trajectory of the Earth, the asteroid and the spacecraft, resembling the 2D result found in Sect.2.2.3. Indeed, the launch and interception times of the final solution are quite close to the previous outcomes, leading to a flight time of 9.6319 TU - which is not too much higher - and a feasibility of 10^{-7}. The in-plane thrust pointing (*Figure 2.11 - green*) angle has also a trend similar to the 2D one and, like the trend of the out-of-plane thrust pointing angle (*Figure 2.12 - cyan*), deviates considerably from the guess (*red*).

On the other hand, 3D cartesian coordinates are preferred to construct the final problem, where the SPICE routines will be employed to find ephemerides, velocities and perturbing accelerations of celestial bodies within the Solar System. In this second version, a few additional factors increase the complexity of the problem. Firstly, the equation of the variation of the spacecraft mass is added to the EOM describing the dynamics in 3D cartesian coordinates as follows:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = -\frac{\mu_\odot x}{r^3} + \frac{T_x}{m} \\ \dot{v}_y = -\frac{\mu_\odot y}{r^3} + \frac{T_y}{m} \\ \dot{v}_z = -\frac{\mu_\odot z}{r^3} + \frac{T_z}{m} \\ \dot{m} = -\frac{T_{max}}{c_{exh}} \end{cases} \tag{2.35}$$

where the exhaust velocity of the thrusters is set to $c_{exh} = 1\ AU/TU$ ($\approx 30\ km/s$) and the maximum thrust available to $T_{max} = 0.05 \cdot m_0$, with $m_0 = 1 \equiv 100\,\%$. The thrust components - T_x, T_y and T_z - are calculated as functions of the pointing angles β and γ:

$$T_x = \frac{T_{max}x}{r} sin(\beta)cos(\gamma) + \frac{T_{max}}{r^2 v}[(z^2 + y^2)v_x - x(zv_z + yv_y)]cos(\beta)cos(\gamma) + \\ + \frac{T_{max}}{rv}[yv_z - zv_y]sin(\gamma) \tag{2.36}$$

$$T_y = \frac{T_{max}y}{r} sin(\beta)cos(\gamma) + \frac{T_{max}}{r^2 v}[(x^2 + z^2)v_y - y(xv_x + zv_z)]cos(\beta)cos(\gamma) + \\ + \frac{T_{max}}{rv}[zv_x - xv_z]sin(\gamma) \tag{2.37}$$

$$T_x = \frac{T_{max}z}{r} sin(\beta)cos(\gamma) + \frac{T_{max}}{r^2 v}[(y^2 + x^2)v_z - z(yv_y + xv_x)]cos(\beta)cos(\gamma) + \\ + \frac{T_{max}}{rv}[xv_y - yv_x]sin(\gamma) \tag{2.38}$$

Secondly, an impulse $\Delta v = 0.06\ AU/TU$ ($\approx 1 - 2\ km/s$) delivered by the upper stage of the launch vehicle is considered as a contribution to the initial velocity conditions; therefore, the corresponding in-plane (β_L) and out-of-plane (γ_L) pointing angles become part of the set of optimization parameters. These modifications bring the problem closer to the final one making it more realistic, but the number of variables is significant so the *snsolve* solver is preferred. Results of this version are illustrated in *Figures 2.13*, *2.14*, *2.15* and *2.16*.

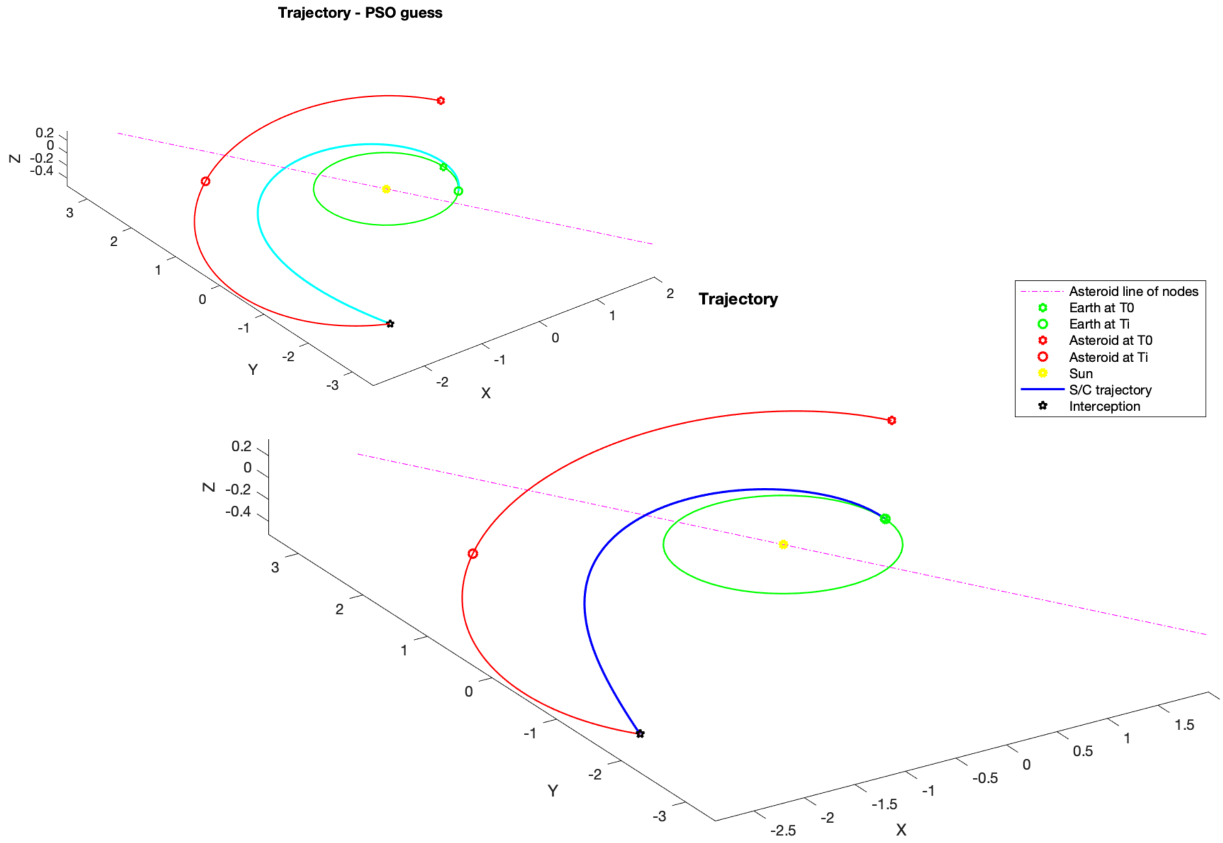


Figure 2.13 Fictional asteroid interception 3D - cartesian coordinates case: trajectories

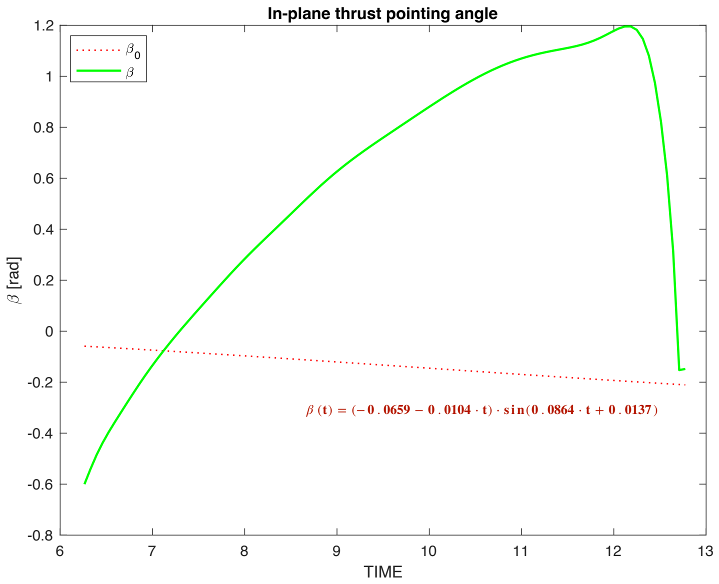


Figure 2.14 Fictional asteroid interception 3D - cartesian coordinates case: in-plane thrust pointing angle

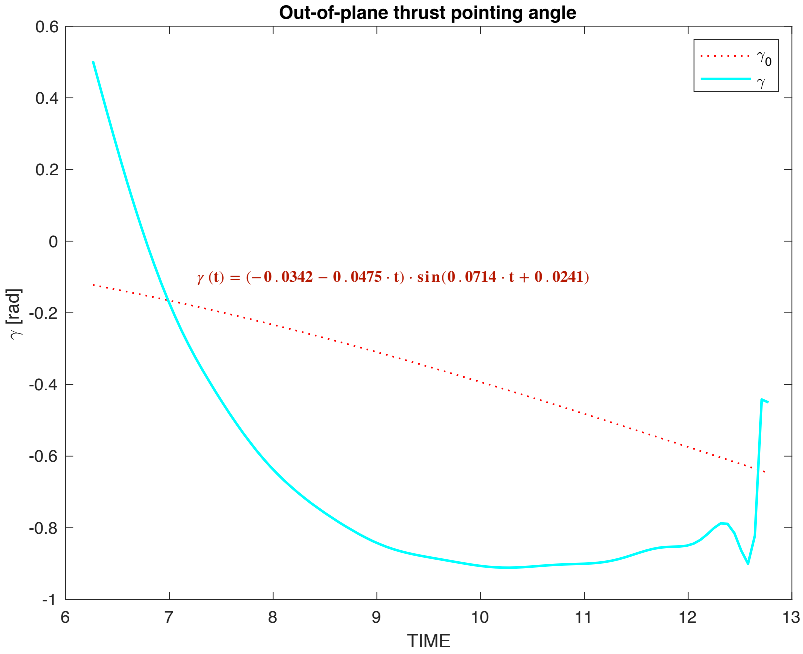


Figure 2.15 Fictional asteroid interception 3D - cartesian coordinates case: out-of-plane thrust pointing angle

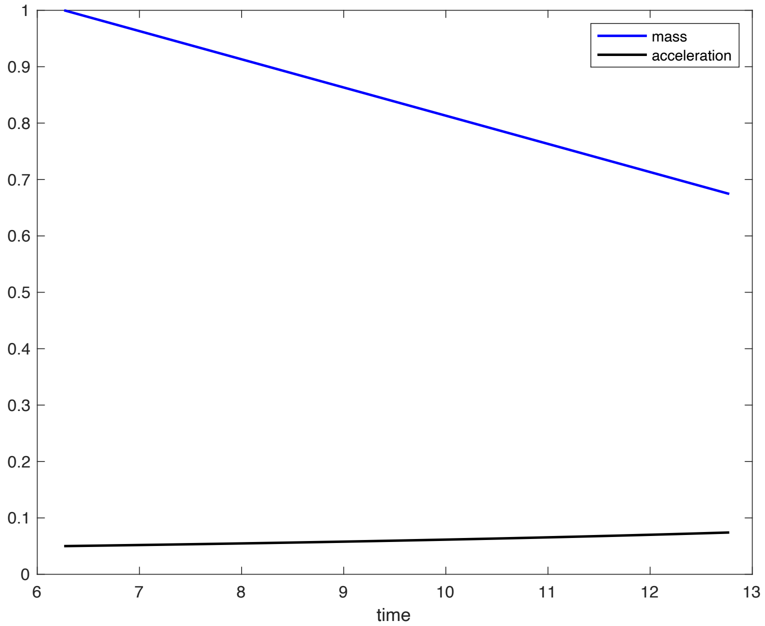


Figure 2.16 Fictional asteroid interception 3D - cartesian coordinates case: mass, acceleration

The spacecraft trajectory in *Figure 2.13* starts later than the previous case; in particular, in the NLP solution, it is launched when the Earth has travelled for about one orbit from its perigee. Nonetheless, the flight time is reduced to 6.5114 TU (with feasibility $\sim 10^{-8}$) due to the beneficial effect of the initial impulse (optimal values: $\beta_L = -3.1352\ rad, \gamma_L = 0.8539\ rad$). Thrust pointing angles (*Figure 2.14* and *2.15* -

respectively *green* and *cyan* for the final solution, while *red* is always used for the PSO solution) have a similar pattern compared to the first case, thus analogous observations can be inferred. *Figure 2.16* shows that the mass of the spacecraft is decreased by about 33% at the interception with the asteroid, consequently its acceleration augments by about 50%. The momentum possessed at the impact is 56% lower than at launch.

2.3 Optimization with a more sophisticated direct solver

Previous examples of optimization problems have employed the MATLAB *ode45* algorithm and the Euler step method to propagate the spacecraft trajectory while satisfying the equations of motion and the constraints. The 3D asteroid interception problem is the basis from which to build the final problem, so given the increasing number of variables to be handled and the overall complexity, the use of a more sophisticated direct solver using Runge-Kutta parallel shooting method is suggested. Like Euler's method, the RK rule involves a fixed time-step but has an accuracy comparable to *ode45*. Some of its advantages are further discussed in Sect. 2.3.1, including a favorable reduction in computational cost.

2.3.1 The 3-Step RK Parallel Shooting method

Direct transcription with Runge-Kutta (RK) integration and parallel shooting [9] [10] is a direct method that converts the optimization problem into a NLP. It is a beneficial choice for low-thrust trajectory optimization problems, which are characterized by a more rapid evolution of the controls than of the states. The time-continuous optimal control problem is discretized into a series of h-long segments, so that $t_0 < t_1 < \ldots < t_N$ and $h_i = t_i - t_{i-1}$ for $i = 1, 2, \ldots, N$ (*Figure 2.17*). Generic *nodes* $[t_{i-1}, t_i]$, namely the mesh points, define a *segment*. Each segment is usually divided into multiple integration steps and, in this case, the number of steps (p) is three as shown in *Figure 2.18*. The state variables (x) are approximated by a parameter at each node, while the control variables (u) are specified at the nodes, at the center points and at the edges of each integration step - $\nu_{ij} = u(t_{i-1} + jh/2p)$ for $j = 1, 2, \ldots 2p - 1$ and for $i = 1, 2, \ldots, N$.

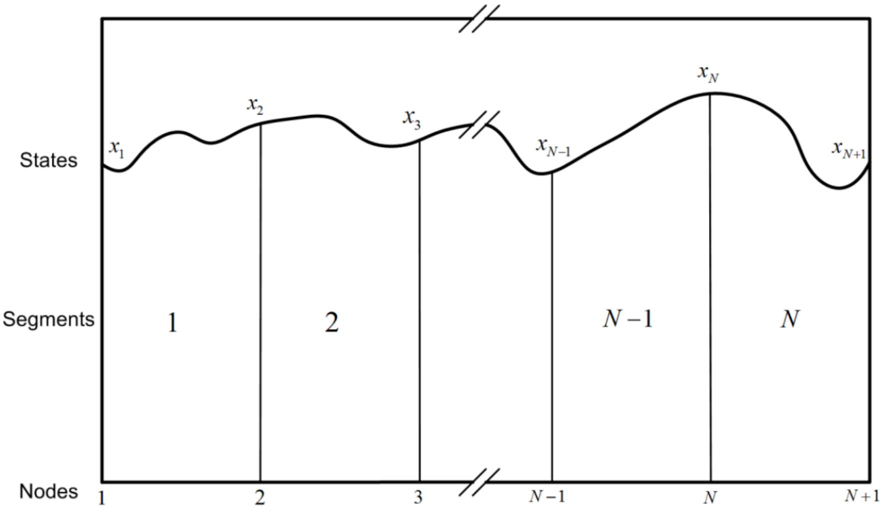


Figure 2.17 DTRK scheme

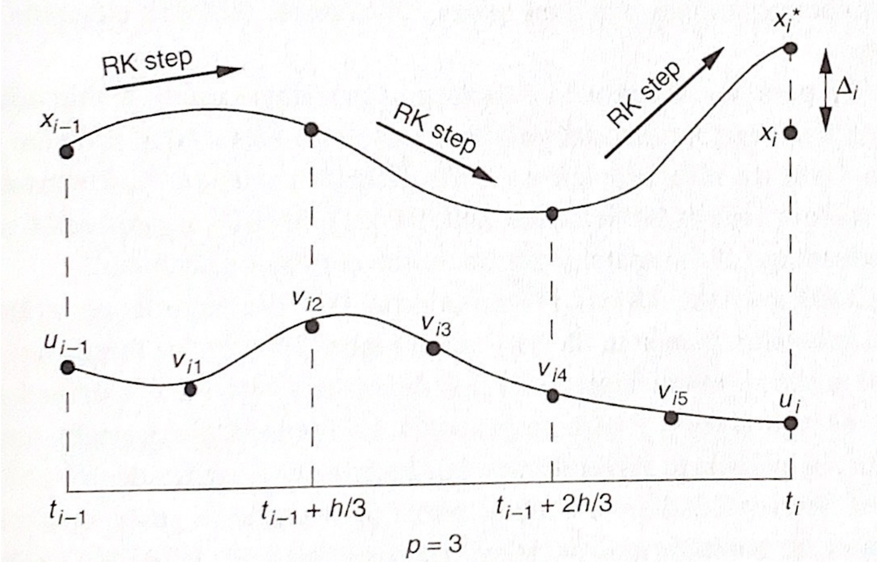


Figure 2.18 3-step RK parallel-shooting integration rule for one time segment [9]

In order to get a feasible solution, both the states and the controls must satisfy the governing equations. Instead of using explicit numerical integration across the problem, implicit integration is used so that the computational cost is less demanding. Therefore, an analytical set of constraints is generated integrating forward the state variables across each time step and exploiting the five interior control points ν_{ij}, in compliance with the equations of motion. The 3-steps adopted formulation performs the integration following the 4th order Runge-Kutta rule. Considering for instance the leftmost step of a segment - from t_{i-1} to $t_{i-1} + h/p$ using the controls u_{i-1}, ν_{i1} and ν_{i2} -, the equations of such process can be written as below:

$$\begin{cases} y_{i1}^{1} = x_{i-1} + \frac{1}{2p} h f(x_{i-1}, u_{i-1}) \\ y_{i1}^{2} = x_{i-1} + \frac{1}{2p} h f(y_{i1}^{1}, \nu_{i1}) \\ y_{i1}^{3} = x_{i-1} + \frac{1}{p} h f(y_{i1}^{2}, \nu_{i1}) \\ y_{i1}^{4} = x_{i-1} + \frac{1}{6p} h [f(x_{i-1}, u_{i-1}) + 2f(y_{i1}^{1}, \nu_{i1}) + 2f(y_{i1}^{2}, \nu_{i1}) + f(y_{i1}^{3}, \nu_{i2})] \end{cases} \tag{2.39}$$

where $f(x, u)$ refers to the system of equations of motion of the dynamical system. Every following step employs the results provided by the equation (2.39) corresponding to the previous step as its initial states and initial controls. Accordingly, the second step integrates from $t_{i-1} + h/p$ to $t_{i-1} + 2h/p$ using the controls ν_{i2}, ν_{i3} and ν_{i4}, then the third one executes the integration from $t_{i-1} + 2h/p$ to t_i using the controls ν_{i4}, ν_{i5} and u_i yielding an approximation of the state (x_i^*) at the rightmost node (t_i) of the segment.

Nonlinear constraints equations derive from the difference between the integrated state at the right node of each time step x_i^* (i.e. y_{i3}^4) and the NLP parameter x_i representing the state at the left node of the following segment. Thus, these defects equation can be expressed as follows:

$$\Delta_i = x_i^* - x_i, \quad \text{for } i = 1, \ldots, N. \tag{2.40}$$

If satisfied, i.e. $\Delta_i = 0$ for $i = 1, \ldots, N$, the equations of motion are successfully integrated across the whole problem using the 4th order Runge-Kutta method.

The advantage of this method lies in the fact that *(p-1)*-estimates of the state vector are computed within each segment but, since these variables are not saved as NLP parameters, the size of the problem is significantly reduced without any loss in accuracy and actually increasing speed in solution achievement. It is notable that system controls are much more frequently specified than system states, therefore different timescales typical of low-thrust problems are well represented.

2.3.2 The development of the RK code in MATLAB

In order to implement the RK parallel shooting method in MATLAB, the usual problem of final energy maximization in a specified time ($t_f = 16\ TU$) using low-thrust propulsion is used as a simple example, since the validity of the results can be verified by comparing them to the previously discussed codes (Sect. 2.1.3 and 2.2.2).

Examining the variables, the problem consists of four states - $\{r, \theta, v_r, v_\theta\}$ - and one control (β). If the RK parallel shooting rule is applied, then each segment has two states and two controls at the edge nodes, and five controls at five equally space intermediate points; so, considering that the rightmost node represents also the starting node of the following segment, the total number of variables is $n = (n_{states} + 6 \cdot n_{controls}) \cdot n_{segments} + (n_{states} + n_{controls})|_{last\ node} = 335$, given 34 points. In fact, fewer points can lead to the same accuracy as the previous cases because of the structure of the RK parallel shooting method. The constraints are calculated through the defects equation (2.40), where the x_i^* vector results from integrating the equations of motion $f(x, u)$ (2.18) with the RK rule.

As an initial guess, the solution from the same problem solved by Euler's step method (2.24) can be used; for this purpose, it is sufficient to assume constant controls and states within each segment. The resulting trajectories, trends of spacecraft energy and thrust pointing angle derived from the *fmincon* solver are shown in *Figure 2.19*, *2.20* and *2.21*. The final energy of the spacecraft (2.19) is equal to $J = 0.1349$ with about 10^{-12} of feasibility of and 10^{-8} of first-order optimality. The fitness function assumes a lower value with respect to the solution achieved with the method of Euler steps and this is noticeable also observing the different final position reached by the spacecraft, while the trend of β follows that obtained previously but is slightly less smooth. Both differences can be improved by increasing the number of points of discretization - thus using the *snsolve* solver - but despite this, an excellent accuracy has been achieved with significantly fewer segments. Frequently, nonlinear programming solvers - such as those presented in Sect. 2.2.1- work best when the initial solution is not too close to the optimum.

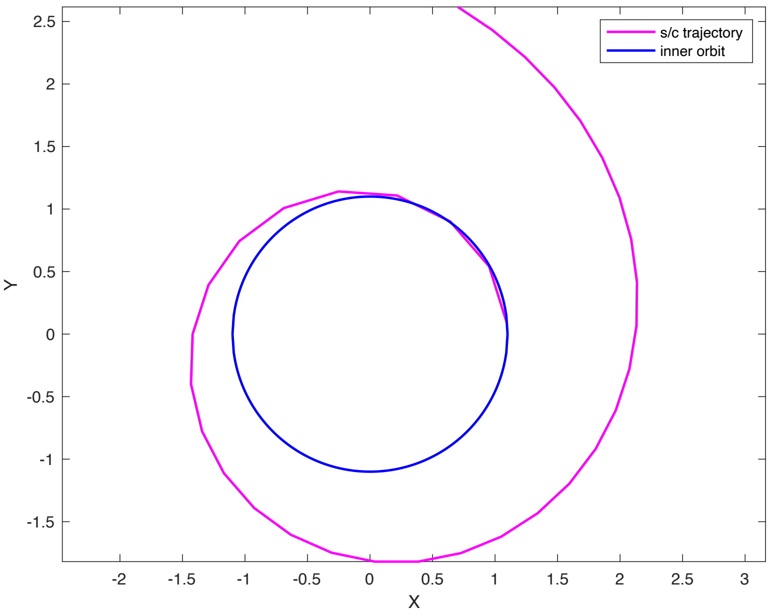


Figure 2.19 Low-thrust trajectory for final energy maximization - *fmincon* + DTRK

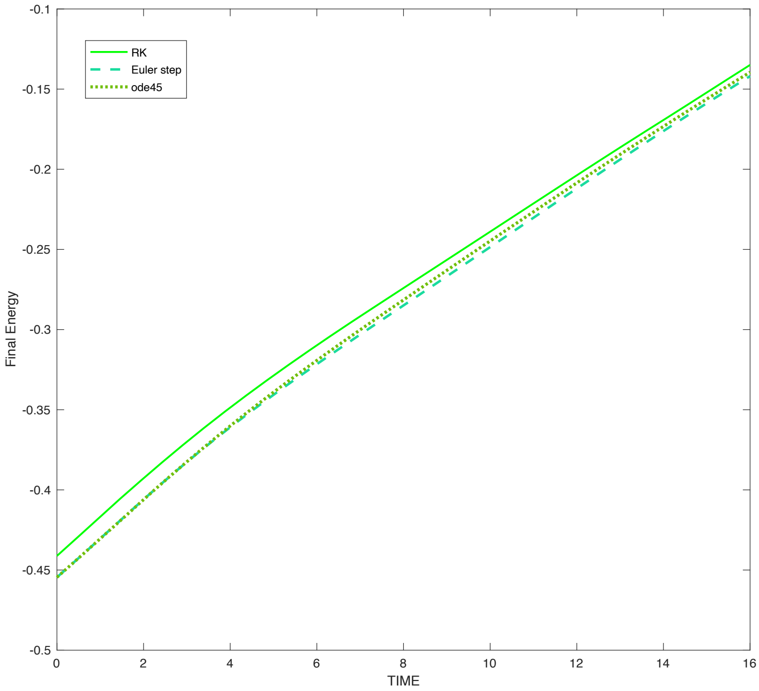


Figure 2.20 Spacecraft energy trend - *fmincon* + DTRK

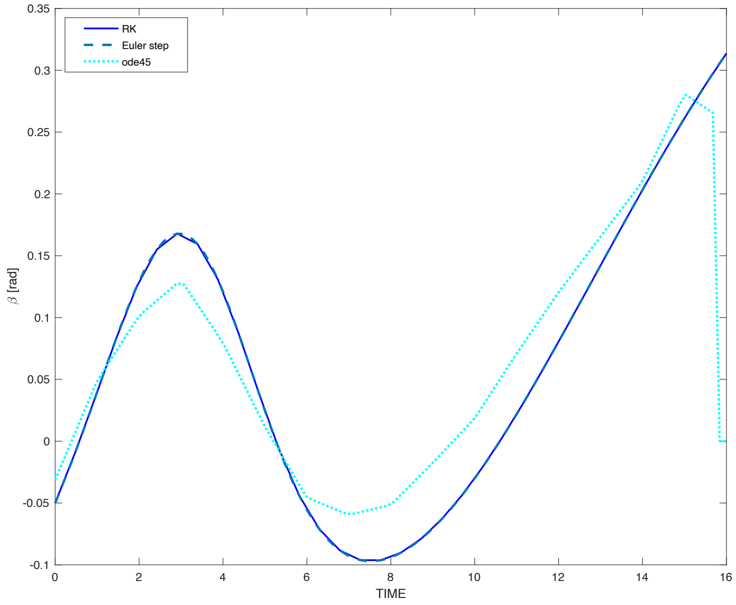


Figure 2.21 Thrust pointing angle trend for final energy maximization using *fmincon* and DTRK

2.3.3 Asteroid interception program using the RK code

The problem of intercepting the fictional asteroid (referred to in Sec. 2.34) via spacecraft impact was solved via the nonlinear solver *fmincon* and the DTRK method. The orbital elements of the asteroid are the same as listed in *Table 2.1* and its unperturbed motion is propagated thanks to Kepler's equation (2.26); the eccentric anomaly of the asteroid at the time t_0 is assumed to be $E_0 = 45°$, but the mutual angular position between the Earth and the asteroid at the time t_0 is not established a priori since the Earth ephemerides are obtained from SPICE routines. Additionally, in order to use such libraries for celestial bodies, Julian days were introduced, converted into seconds or canonical time units as needed. Therefore, given the same optimization parameters chosen earlier - directional angles of the initial impulse given by the launcher, thrust pointing angles, the launch and interception events - the temporal variables were considered from a certain date. For this example, the selected epoch for t_0 is October $12, 2024$ ($\approx 155.7782\ TU$).

As aforementioned, the choice of the coordinate system strongly influences the optimization process [9]. Although cartesian coordinates are the most intuitive choice for a two-body problem, with the origin of the reference system located at the center of mass of the main body, they are not generally the most suitable for describing a trajectory found using a direct transcription and NLP approach. The reason is that NLP solvers are more robust and efficient when the state variable parameters change slowly and within a limited range. Polar or cylindrical coordinates allow for this slow evolution since the radius is always positive and, like angular position, does not change suddenly but in a predictable pattern; angular velocities also do not generally change rapidly and thrust pointing direction controls are primarily tangential, hence small in size. Thus, control parameters maintain, or may even improve, the robustness of the NLP solution. On the other hand, the precision in position, velocity and acceleration provided by the SPICE routines, which work in cartesian coordinates, is a huge advantage, whereby the choice has fallen on the coordinate formulation that conforms to them. This allows to avoid the conversion, which would not affect the accuracy of the results but would increase the complexity of the problem; moreover, the program at issue was found to work well even in the absence of the advantages of spherical or cylindrical coordinates. Therefore, the EOM for this problem are expressed as

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = -\frac{\mu_{\odot} x}{r^3} + \frac{T_x}{m} + a_{x(\text{♀})} + a_{x(\oplus\text{☽})} + a_{x(\text{⛢})} + a_{x(\text{♃})} \\ \dot{v}_y = -\frac{\mu_{\odot} y}{r^3} + \frac{T_y}{m} + a_{y(\text{♀})} + a_{y(\oplus\text{☽})} + a_{y(\text{⛢})} + a_{y(\text{♃})} \\ \dot{v}_z = -\frac{\mu_{\odot} z}{r^3} + \frac{T_z}{m} + a_{z(\text{♀})} + a_{z(\oplus\text{☽})} + a_{z(\text{⛢})} + a_{z(\text{♃})} \\ \dot{m} = -\frac{T_{max}}{c_{exh}} \end{cases} . \tag{2.41}$$

with $c_{exh} = 1\ AU/TU$, $T_{max} = 0.05 \cdot m_0$ with $m_0 = 100\,\%$ and $\Delta v_L = 0.06\ AU/TU$. A generic planetary perturbing acceleration ($\overrightarrow{a}_p$) can be determined as [19]

$$\overrightarrow{a}_p = -\mu_p \left[\frac{\vec{r}_{pv}}{r_{pv}^3} + \frac{\vec{r}_{sp}}{r_{sp}^3} \right] \tag{2.42}$$

where $\vec{r}_{pv}$ is the position vector of the vehicle relative to the planet and $\vec{r}_{sp}$ is the position vector of the planet relative to the Sun. In this program, the differential equations governing the dynamical system include perturbations due to the presence of the gravitational bodies of Venus, the Earth-Moon system, Mars and Jupiter, whose perturbing accelerations are also computed using SPICE routines. In particular, the effect of the Earth-Moon system is considered only when the distance of the spacecraft from its barycenter is greater than $0.01\ AU$, which is about four times the average distance between the center of mass of the Earth and the Moon.

The objective function to be minimized is denoted as follows:

$$J = -\left(m_{s/c}v_{s/c}\right)\big|_{t_I} \tag{2.43}$$

meaning it consists of maximizing the momentum possessed by the spacecraft when it impacts, which is equivalent to maximizing the momentum transferred to the asteroid. This implies that the optimizer can choose whether to prioritize the mass or final velocity of the spacecraft, as well as evaluate which launch and interception time variables are most functional for the most effective momentum exchange. However, this objective compared to the minimization of the flight is not necessarily closer to our goal - which is to have the maximum useful effect when the asteroid will move toward the closest point of its orbit to the Earth - as the direction in which the impulse is given could lead to a large deflection but bring the asteroid closer to Earth. As for the boundary conditions, the constraints are given by the defects equation (2.40) and the definition of the asteroid-spacecraft missing distance which must be zero at the interception (2.34).

In order to formulate an initial guess, as operated previously, a PSO-based program has been written, equivalent to the nonlinear program in terms of parameters, equations, fitness function, etc… However, this program is different from the other examples of asteroid interception because it reproduces more faithfully the nonlinear program without assuming a priori a certain function for describing the trend of the thrust angles. Indeed, the initial guess strongly influences the result, so in the absence of a rigorous method to estimate the evolution of these parameters a linear interpolation between the optimal points is the choice with fewer assumptions, although *ode45* does not always work efficiently with this type of interpolation. The thrust pointing angles were discretized only to 14 points each in this example, for a total of 32 optimization parameters; the RK parallel shooting method was not applied in the PSO program because it would have yielded too many parameters to handle, so that the optimization algorithm would not have worked optimally. The PSO algorithm employed 200 particles and 500 iterations, while 29 segments were used in the nonlinear program so that the total number of variables amounted to $n = 564$, generally quite high for it to be handled by the *fmincon* routine but, instead, the fitness value of x was found with good feasibility ($\sim 10^{-7}$) and first-order optimality ($\sim 10^{-3}$). Results are shown in *Table 2.2* and *Figures 2.22*, *2.23*, *2.24* and *2.25*.

Table 2.2 Asteroid interception problem - results

Variable	Description	Value
β_L	*in-plane* Δv_L *pointing angle*	- 49.7911°
γ_L	*out-of-plane* Δv_L *pointing angle*	45.7506°
t_L	*launch time*	168.1725 TU October 2, 2026
t_I	*interception time*	177.6908 TU April 7, 2028

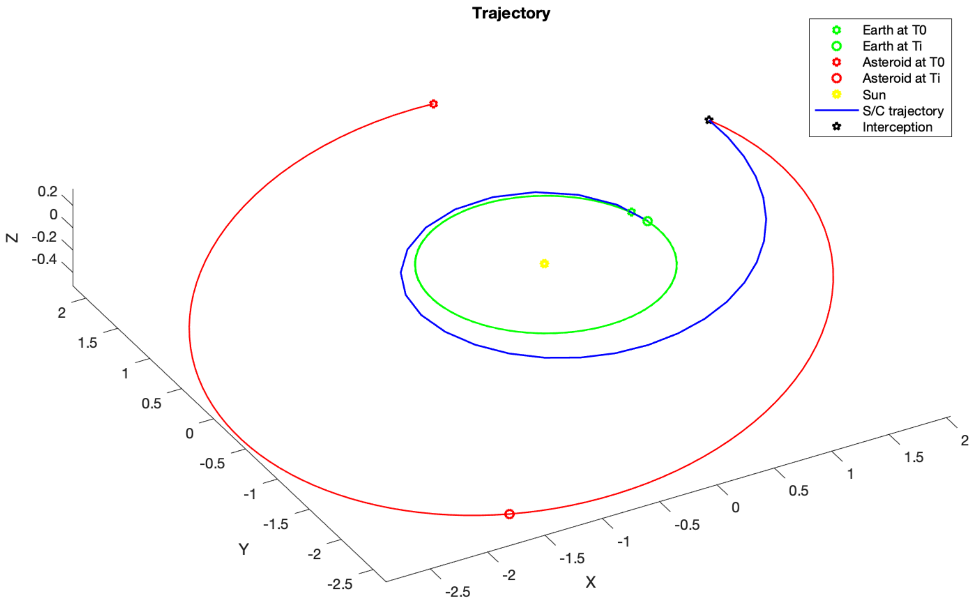


Figure 2.22 Fictional asteroid interception - DTRK + *fmincon*: trajectories

As can be seen from *Table 2.2* and *Figure 2.22* representing the trajectories, launch conditions that allow optimization of momentum exchange occur about two Earth years from t_0, after which the spacecraft travels for about a year and a half before impacting it. The flight time is equal to $9.5183\ TU$, but this result is influenced by the value set for E_0. In particular, the presented example has the minimum flight time; instead, in the other examined cases the flight time could be greater than $1 \div 2\ TU$ compared to the result presented.

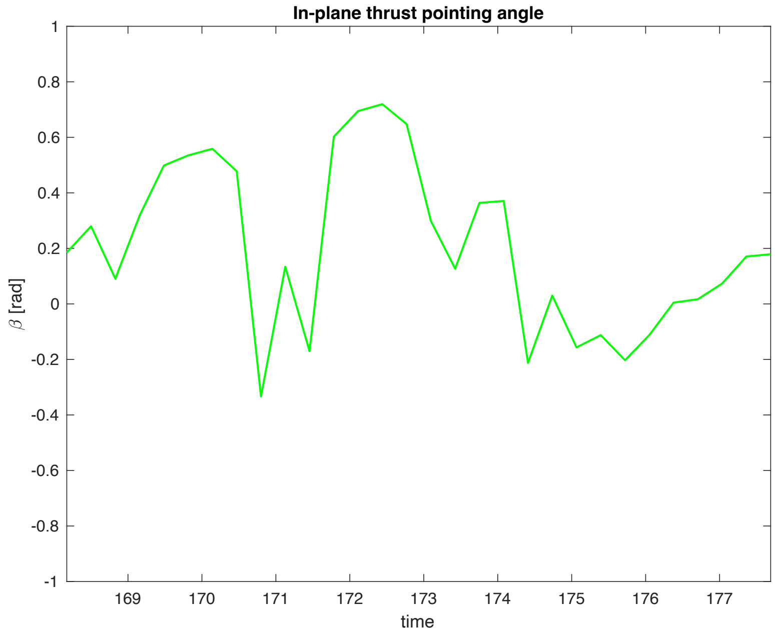


Figure 2.23 Fictional asteroid interception - DTRK + *fmincon*: in-plane thrust pointing angle

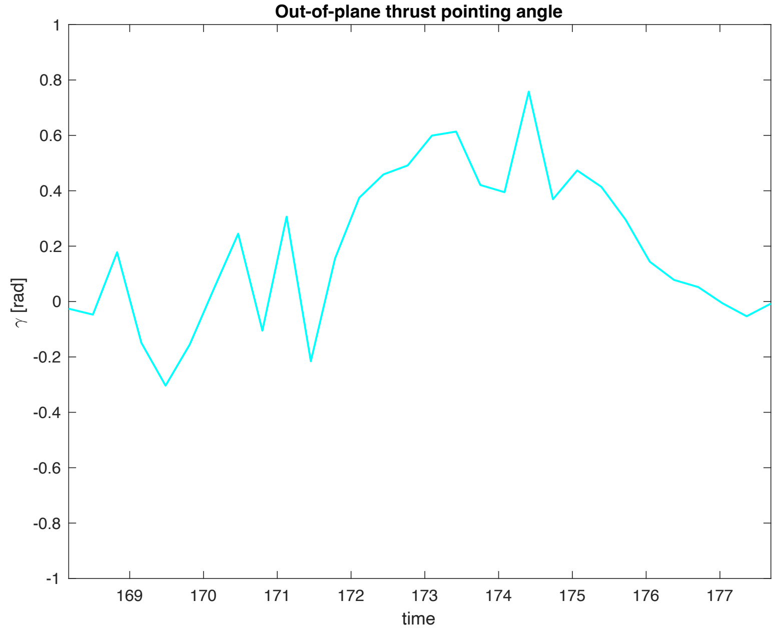


Figure 2.24 Fictional asteroid interception - DTRK + *fmincon*: out-of-plane thrust pointing angle

From *Figures 2.23* and *2.24*, it can be noted that the angles are in a range of $-23°$ to $45°$; moreover, their trend is very different from that encountered in the previous examples that aim to minimize the flight time. The final thrust pointing angles are those associated with the collision, which allow the asteroid to be impacted in the optimal direction: the in-plane angle measures just over $10°$ while the out-of-plane angle is less than half a degree.

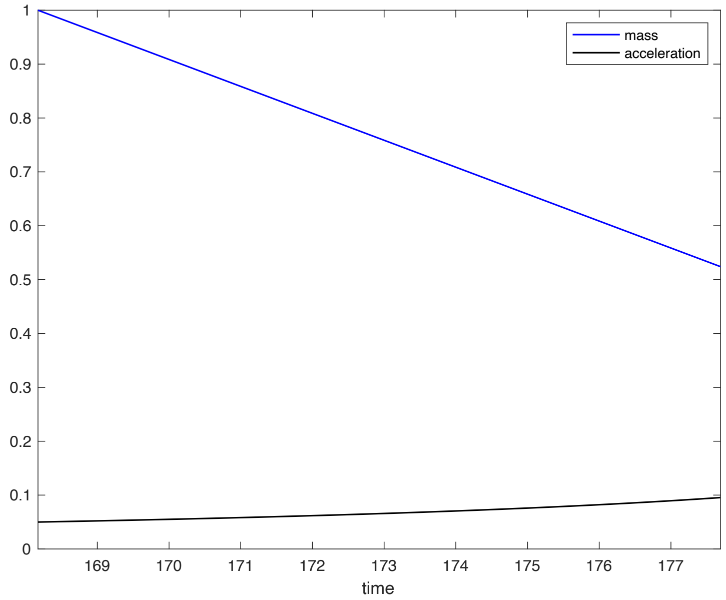


Figure 2.25 Fictional asteroid interception - DTRK + *fmincon*: mass, acceleration

Figure 2.25 shows that the mass was reduced by about 48% at impact compared to initial conditions, while the acceleration increased by 90%. The momentum at impact is 0.4535, which is about 53% lower than the value assumed at the start. Comparing the results with the Cartesian case examined in Sect. 2.2.4, obviously here the flight times have been penalized in favor of momentum at the interception and acceleration, which has undergone a significant gain.

The fictitious asteroid interception problem described above is the most complete version that is being presented, while the next chapter will analyze the overall problem including a real hazardous asteroid and the second, so far neglected, phase of evaluating the deflection resulting from the impact at the closest approach to Earth.

Chapter 3: Optimal asteroid mitigation using a low-thrust kinetic impactor

3.1 Near-Earth-crossing asteroids

Asteroids are remnant debris from the solar system formation process and, therefore, potential molecular precursors to the origin of life. Recently more than ever, these space objects have arisen the interest of the space community for scientific, technology demonstration, human spaceflight and planetary defense purposes. Indeed, Near Earth Asteroids (NEAs) represent an opportunity to uncover the mysteries of the formation, evolution and composition of the solar system being the easiest celestial bodies to reach from Earth, but also a risk since a potential impact could result in catastrophic consequences.

Our solar system bears the marks of asteroids impacts: the spectacular event observed in 1994, when the comet Shoemaker-Levy 9 smashed into Jupiter, is an example [6][24]. All along with its history, Earth has experienced many ground and air impacting events. An asteroid with a diameter of less than 100 m is estimated to impact the Earth every several hundred years, while asteroids greater in size pass once every 10,000 years [25] and may cause local damage, earthquakes and tsunamis. Asteroids with a diameter larger than 1 km are considered to be global killers; in fact, approximately 66 million years ago a kilometer-sized object hit a gypsum deposit near today's Yucatan, in Mexico, and three-quarters of all species went extinct in what is called the K-T extinction event. The largest impact event recorded in Earth's history is the Tunguska event, where a meteor airburst involving a stony meteoroid about only 60 meters in size in 1908 released tens of megatons of TNT equivalent energy [6].

US Government sensors keep track of fireballs and bolides (i.e. exceptionally bright meteors) spectacular enough to be seen over a very wide area. The world map shown in *Figure 3.1* provides a chronological data summary of those events reported between the 15th October 1988 and the 29th September 2021. Each event has a corresponding total impact energy and, among those, the Chelyabinsk Event that

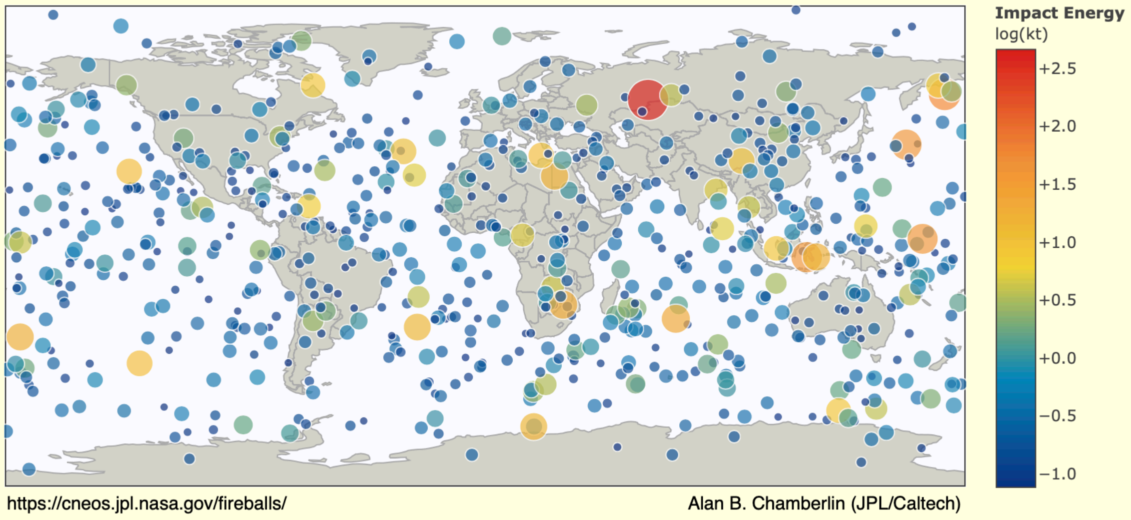

Impact Energy
log(kt)
+2.5
+2.0
+1.5
+1.0
+0.5
+0.0
−0.5
−1.0
https://cneos.jpl.nasa.gov/fireballs/
Alan B. Chamberlin (JPL/Caltech)

Figure 3.1 Fireballs reported from US Government sensors

occurred in 2013 stands out. It involved the largest known natural object (measuring about 20 meters in diameter) to have entered Earth's atmosphere since the 1908 Tunguska Event. The superbolide disintegrated before reaching the ground, yet the large shockwave that followed had an estimated total kinetic energy before atmospheric impact greater than the energy released from the atomic bomb detonated at Hiroshima. Each of these impacts poses risks to our ecosystem and the growing awareness marked the urgency for the international community to detect and track asteroids and comets that constitute a hazard, as well as test technologies to mitigate these events and thus provide planetary protection.

3.1.1 A planetary defense challenge

Although the probability of a massive collision is low, the potential damage is too high to be neglected. Important planetary defense resources have been devoted to hazard detection and possible impact interdiction [26]. NASA’s Near-Earth Object (NEO) Observations Program channel its efforts in finding, tracking and characterizing NEOs. NEOs refer to asteroids and comets with a heliocentric orbit that can bring them into Earth’s proximity, that is within 30 million miles of Earth’s orbit. Currently, the objective is to identify and define at least 90% of the predicted number of NEOs that have a dimension greater than or equal to 140 meters in size and to define a subset representative of the entire population. Despite the low chance to hit the Earth for the next 100 years, this class of objects could cause concerning devastation and that is why it is the main focus of global interest. The Planetary Defense Coordination Office (PDCO) [27] is responsible for coordinating global defense and in particular it provides early detection of potentially hazardous objects (PHOs), i.e. a subset of NEOs with a predicted orbit that passes within 5 million miles of Earth's orbit and large enough (30÷50 meters) to cause significant damage on Earth. Furthermore, the European Space Agency (ESA) has built up the Space Situational Awareness (SSA) system, which tasked a specific NEO segment to support the protection of European critical space and ground infrastructure from threats by potential asteroid impacts [28].

Close-up observations of these objects may drastically increase our knowledge about the NEA population and their characteristics, indeed many space missions of international engagement have been aiming at asteroid exploration in the last decades. In 1999, Deep Space 1 accomplished its primary mission goals of new technologies testing flying by the asteroid 9969 Braille and comet Borrelly [4]. In 2001, the NEAR Shoemaker spacecraft was the first to successfully orbit and land on an asteroid, the near-Earth asteroid Eros [5]. The Dawn spacecraft reached the giant asteroid Vesta and the dwarf planet Ceres in 2011, becoming the first spacecraft to orbit two extraterrestrial bodies [29]. During the same year, Stardust spacecraft was the first to return extraterrestrial material from outside the orbit of the Moon to Earth, in particular a sample belonging to the comet Tempel 1, already visited in 2005 by the Deep Impact spacecraft which released an impactor on its surface [30]. In 2014, ESA's Rosetta mission was the first to rendezvous with a comet, follow it on its orbit around the Sun and deploy a lander to its surface [31].

After the rendezvous with a NEO, some spacecraft are aimed to collect a sample of the object they encountered and then return it to Earth. The ongoing OSIRIS-REx NASA mission [32] has been intended to study the near-Earth asteroid Bennu, in particular the spacecraft has captured a regolith sample from Bennu's surface and in May 2021 has started its return back to Earth. Since asteroid samples may record the earliest history of our planets, collecting them could be the key to knowing more about their natural resources, which in future may be exploited to fuel space exploration. Moreover, the asteroid Bennu is classified as one of the most potentially hazardous asteroids for its high probability of impacting the Earth late in the 22nd century and determining now its characteristics may be essential in the event of an impact mitigation mission.

Figure 3.2 Asteroid deflection mission [Credit: C. Daniloff, MIT]

Several studies have been investigating how to perform an asteroid deflection mission (*Figure 3.2*). Already in 2004, ESA presented a mission concept called Don Quijote [28], testing the capability to deflect the course of an asteroid with a high-velocity impact which would be monitored by an observing spacecraft. This study evolved to a NASA/ESA international cooperation mission called AIDA (Asteroid Impact & Deflection Assessment), consisting of two mission elements: the NASA Double Asteroid Redirection Test (DART) mission [33] and the ESA Asteroid Impact Mission (AIM) rendezvous mission, now redesigned as Hera mission. The AIDA mission is the first planetary defense technology demonstration for asteroid impact mitigation via a kinetic impactor (Sect. 3.1.2), that is, the modification of the orbit through momentum transfer; moreover, it will characterize the physical properties of its target and measure the deflection caused by the impact [34]. The target is Dimorphos, the small moonlet (160-m) of the asteroid Didymos (800-m), which will be intercepted in fall 2022. This near-Earth (65803) binary asteroid system has been chosen because it passes close to Earth but it is not an actual threat. In late 2024, Hera will carry two CubeSats that will fly around before touching down; their main goals are to map the impact crater and measure the asteroid's mass.

Among other papers, Housen and Holsapple [35] have proposed a theoretical model to predict the momentum transfer efficiency of kinetic impacts, but the AIDA mission will provide for the first time measurements of momentum transfer, crater size and morphology, and the evolution of an ejecta coma causing a substantial advance in the understanding of asteroid impact processes. In particular, the momentum transferred to the target body will be determined using ground-based telescopic observations of the orbital period change of Didymos and imaging of the DART impact ejecta plume by the Italian LICIACube CubeSat - embarked as a piggyback on the DART vehicle -, along with modeling and simulation of the DART impact [36].

3.1.2 Deflection strategies

Asteroids posing a threat to Earth may be deflected off of an Earth-impacting trajectory by different strategies. The methods considered by the scientific community differ in the type of asteroid-spacecraft interaction [34][37]. The most studied techniques are based on the impulsive change in the linear momentum of the asteroid; examples are kinetic impactors and nuclear blast interceptors. The former relies on a hypervelocity collision with a spacecraft to transfer momentum to the impacted asteroid and deflect its motion; the latter consists of a nuclear device that vaporizes part of the asteroid's surface when it detonates, producing an impulsive change in velocity due to the momentum imparted to the ejected mass. A multi-impulsive change in the linear momentum of the asteroid can also be produced by the ejection of surface material, such as a mass driver. Other techniques of interest actively produce controlled continuous low thrust, e.g. propulsion devices - such as electric/chemical motors and solar sails [38] - or gravitational tugs (i.e. gravity tractors). Moreover, there are ways to generate passive low thrust via an induced change in the thermo-optical properties of the asteroid surface, exploiting the Yarkovsky effect or the emissivity via white paint. A controlled thrust can also be achieved through the ablation of an asteroid surface by irradiating it with laser beams or solar collectors, which avoid a catastrophic fragmentation of the asteroid and eliminate the need to physically land onto its surface [25].

The effectiveness of various techniques was assessed by Sanchez et al. [37] through a multi-criteria quantitative comparison. This analysis was conducted by considering a collection of NEOs with different physical characteristics and employing different mitigation strategies evaluated in terms of four figures of merit, i.e. the achievable miss distance at Earth, the warning time, the total mass into orbit and the current technology readiness level, that is the estimated time to develop the technology required to implement a given mitigation strategy. The first three criteria quantitatively express the ease of deflecting an asteroid by a given method and whether current launch capabilities allow implementation of that strategy. In particular, the warning time not only provides information on how long in advance the impact information is needed to respond, but it is also an indication of the time available to repair a failed deflection operation. The results of this study show that the solar collector and nuclear interceptor options achieve the best results in terms of deflection, mass in space, and warning time, but are strongly influenced by the level of technological readiness, which is subject to current and future political and economic situations. In contrast, the kinetic impactor and low thrust devices both show excellent performance for deflecting asteroids with a mass less than 10^{10} kg and are the most technologically mature, as discussed in the National Research Council report (2010). As for the mass driver, it shows an intermediate performance between the two pairs.

In addition, when evaluating strategies to deflect an incoming asteroid, it is critical to know about the NEO's size, physical properties, probability of striking the Earth and the impact hazard zone. Especially for NEAs defined as potentially hazardous asteroids (PHAs), chemical, physical, and mineralogical composition as well as orbital properties play a key role in the success of a deflection or sample return mission. However, often exhaustive characterization - such as that in situ - may not be feasible due to insufficient timing (a few years to a decade), hence causing uncertainty in those properties [38]. Short timescales certainly limit some deflection options, whereas mischaracterization can significantly alter the effectiveness of a certain strategy. Sugimoto et al. [39] studied the relationship between the need to accurately define the composition of the target asteroid (i.e. porosity, surface materials, precise shape, etc.) and the deflection that a certain method yields. Although there are methods to deal with uncertainties in the composition of NEAs (e.g. via the Dempster-Shafer theory, i.e. the Evidence Theory [39]), deflection methods that have a strong interaction with the target object - e.g., nuclear interceptor or kinetic impactor - are more strongly affected than others. The study by Sugimoto et al. compared the effect of such uncertainties on kinetic impact, nuclear interceptor, and solar sublimation. Among the three deflection methodologies, kinetic impact is superior in robustness and reliability, while nuclear interceptor and solar sublimation, although potentially achieving greater deflection distances, are more dependent on epistemic uncertainty in the composition of the asteroid.

Among the range of concepts for asteroid threat mitigation, it is to be remarked that impulsive methods obtain immediate effects, while the non-impulsive methods may take years of operation to accumulate sufficient deflection and thus require the hazardous object to be discovered multiple years in advance of its threatened collision with Earth [34]. In this light, nuclear devices are probably the only viable option for preventing large or detected asteroids from impacting the Earth at short notice; otherwise, kinetic impact is generally the preferred strategy. Full-scale experiments to test the effectiveness of kinetic impactors have been rare (Sect. 3.1.1); NASA's Deep Impact spacecraft adopted it, but the mission was not intended to cause an observable orbit deflection. Indeed, its main goal was to investigate the nature of the comet, whereas the feasibility of the kinetic impactor technique is going to be demonstrated with the AIDA mission (Sect. 3.1.1) which will provide the first quantitative test of asteroid deflection. These valuable opportunities need to be complemented by a broad numerical treatment of the problem; in fact, future missions focused on asteroid defense will aim precisely at directly measuring the momentum transfer imparted by kinetic impactors.

Given current limitations on spacecraft masses (no more than a few tens of tons), asteroid deflection from kinetic impacts is considered feasible for asteroids up to 1 km in diameter. Assuming that the spacecraft (m_i, v_i) impacts in line with the center of mass of an asteroid (m_a), it will transfer all of its momentum $p_i = m_i v_i$ to the body, changing the asteroid's translation velocity by

$$\Delta v_a = \frac{m_i v_i}{m_a + m_i} \approx \frac{m_i}{m_a} v_i \,. \tag{3.1}$$

Additional momentum transfer occurs in the cratering process, as material is ejected beyond the escape velocity. This additive effect to the momentum transfer can be expressed as [36][34][35]

$$m_a \Delta v_a = m_i v_i + m_{ej} v_{ej} = \beta m_i v_i \,. \tag{3.2}$$

In the collision, the total momentum of the material is conserved. Therefore, the change in the asteroid's momentum is the sum of the momentum of the impactor and the momentum contained in the material ejected permanently backward from the impact site. The amount of ejected material is typically many times the mass of the impactor and has initial ejection velocities greater than the escape velocity. Consequently, the momentum transferred to the target asteroid can be significantly greater than the momentum of the incoming spacecraft. The ratio of the momentum imparted to the asteroid to the momentum of the impactor (β) is commonly referred to as "momentum multiplication factor" and is actually a measure of the deflection efficiency of the impact [35]. The ejecta of the crater are the source of β which depends on the properties of the impactor and the size and composition of the asteroid. The mechanics of mass ejection in an impact is extremely complicated and not yet well known; in particular, for small events, the ejection process is driven by the strength of the surface material, while for larger impacts the gravitational field of the asteroid determines the outcome. The basic approach is to sum the momentum of all ejecta that escape the asteroid, considering that if the material falls back on the asteroid there is no contribution to the momentum, whereas if it escapes the asteroid's gravitational field it is necessary to estimate its velocity and direction at infinity. The ejecta acts like a rocket engine, pushing the asteroid away; this availability of “free thrust” may seem an advantage in terms of maximizing the deflection from the asteroid’s path, hence the resulting missing distance, but it is important to verify that changing the asteroid's path will not expose the Earth to a new impact hazard after a certain period of time.

The kinetic impactor technique is the strategy chosen in this work, where the target is a PHA (Sect. 3.2.2). Quantifying the change in the trajectory of an asteroid post-impact is relevant both in terms of studies of asteroid evolution and in preventing a potential asteroid impact on the Earth. Therefore, a trajectory optimization problem must be numerically solved to obtain feasible trajectories for spacecraft equipped with the chosen propulsion system. However, at first approximation, the ejecta process will not be taken into account as a contribution to the momentum of the asteroid, but future work may certainly develop this aspect based on literature.

3.1.3 Prior work

Optimal control theory has frequently been applied to the problem of PHO interception and deflection. Indeed, a numerical study of kinetic impact deflection can provide guidelines for the design of kinetic impact missions, including pre-impact reconnaissance of a threatening asteroid. In addition, performing numerical simulations for a range of conditions can help define the current limitations of the kinetic impact approach.

A hazardous asteroid deflection mission includes two consecutive phases, the rendezvous or interception phase and the deflection phase. Previous numerical studies of kinetic impact deflection have focused on simulating impacts, such as the one envisioned by the AIDA mission (e.g. Cheng et al. [34]), at relatively low velocities (~6 km/s) and using modest impactor masses (300-400 kg). In particular, the work of Syal et al. [40] attempted to address the limitations of the kinetic-impactor approach by modeling asteroid bodies and properties in more detail and using impact masses at the limit of the current launch vehicle technology (1000-10,000 kg).

However, the majority of published articles in this field focus on deflection strategies and analysis of mitigation effectiveness (e.g. Peloni et al. [38], Sanchez et al. [37], Gibbings et al. [25]). Whereas, few have focused on the optimized design of the low-thrust interception trajectory that is a prerequisite for successfully implementing potentially dangerous asteroid deflection strategies. Among them, Conway has successfully solved the problem of such low-thrust trajectories aimed at minimizing the time-of-flight [6] or maximizing the deflection imparted by the spacecraft to the asteroid [41], via collocation method and nonlinear programming. Instead, relying on analytical formulae derived from proximal motion equations, Vasile et al. [42] addressed the problem of increasing the minimum orbit intersection distance of the asteroid due to an impact with a spacecraft in order to establish the optimal direction of the deviating impulse transferred to the asteroid. Casalino and Simeoni [43] also addressed the problem of optimizing a kinetic impact mission to deflect an Earth-crossing asteroid; they applied an indirect method based on optimal control theory and numerical integration of the spacecraft and asteroid equations of motion, providing an exact solution of the dynamical model. The considered objective is the maximization of the miss distance at the closest approach of the asteroid to the Earth. Furthermore, a procedure is developed to find mission opportunities considering both ballistic and electric trajectories, given a certain departure date. The results, in particular, show that electric propulsion is more advantageous.

The design of low-thrust rendezvous trajectories generally requires the solution of an optimal control problem, which has no general closed-form solution; moreover, the problem formulation depends on which deflection strategy is chosen. A versatile approach for rapid design and optimization of the low-thrust rendezvous/intercept trajectory, applicable to future asteroid deflection missions, has been developed by Li et al. [24]. The proposed problem implements a shape-based method in combination with the genetic algorithm (GA) as the initial guess to the next most accurate optimization; then, the low-thrust trajectory optimization problem, converted into a discrete nonlinear programming problem, can be solved using the pseudospectral Radau method via already developed nonlinear optimization (NLP) algorithms. In particular, sequential quadratic programming (SQP) - exploited in solvers such as *snsolve* (Sect. 2.2.1) - is used to solve the nonlinear programming problem and obtain the optimal trajectories sought.

Having provided some examples of the state of the art in optimizing low-thrust trajectories for Earth-crossing asteroid interception and the deflection strategies currently under study, here follows an in-depth discussion of the project carried out by this thesis. The originality of this work lies in not selecting a priori what is more functional to optimize, but the optimal solution is generated from a tradeoff between impacting the asteroid as early as possible, impacting with a high velocity, impacting from the optimal direction, and impacting with a large enough mass to significantly affect the asteroid, starting from as few assumptions as possible. The explanation of the objective and how it was determined will be further analyzed in Sect. 3.2.3. Moreover, this research takes inspiration from the Master's thesis research of Jacob Englander (2008) [44], who had already addressed this issue in a different way. His conceptual work was revisited for developing the final program from scratch. For instance, Englander used MATLAB GA along with the 2D shape-based trajectory approximation (Sect. 3.3.1) to find a suitable initial guess for the accurate DTRK solver, whereas the present work explored the use of PSO for that purpose. Also, the DTRK method is developed here in MATLAB. Furthermore, the final problem applies also a piece of the code developed by another Master student, Andrew Koehler, for his project. From its code, the propagation of an asteroid's trajectory was extrapolated, including the change in coordinates from heliocentric to planet-centric flight and vice versa, if the asteroid enters or leaves a planet's sphere of influence.

3.2 The simulation of asteroid mitigation

This section discusses in detail the final complete problem of asteroid mitigation using a low-thrust kinetic impactor. First, the scenario is presented, then the asteroid taken as an example is introduced, and finally, the objective is thoroughly described. The previous sections introduce the parameters, equations, constraints, integration methods, and solvers as building blocks of the final program, and are key to interpreting it.

3.2.1 The scenario

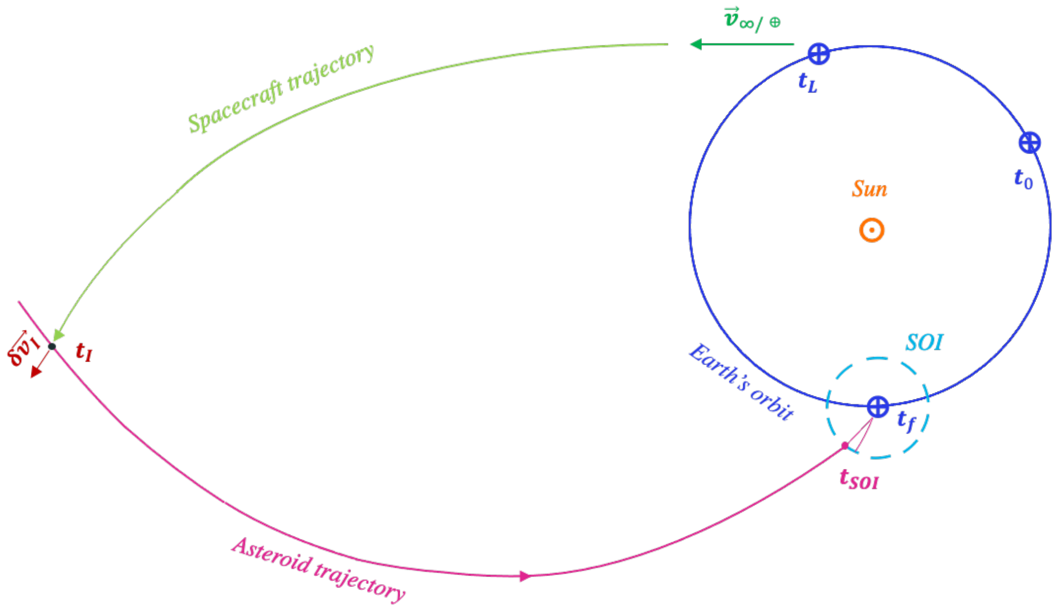


Figure 3.3 Scenario of the final problem

The problem of asteroid mitigation through a low-thrust kinetic impactor is depicted in *Figure 3.3* (note that it is not in scale). The scenario consists of two main parts, the spacecraft trajectory toward a hazardous asteroid and the deflected motion of that asteroid after its interception. By setting a time t_0 of initial observation of the asteroid, the program can choose the best time (t_L) to launch the spacecraft; including t_L as an optimization parameter allows to wait for the geometry between the Earth and the asteroid corresponding to a future optimal at t_I, which is also to be optimized.

The upper stage of the launch vehicle propels the spacecraft out of low Earth orbit with an initial impulse ($\Delta\vec{v}_L$), burning all its fuel. The $\Delta\vec{v}_L$ magnitude is constant and its direction angles β_L, in-plane, and γ_L, out-of-plane, are values to be optimized. Then, the spacecraft switches to low-thrust electric propulsion to travel to the asteroid, assuming that the empty launch vehicle upper stage is carried along to the target asteroid to provide additional impact mass.

Imagining to follow the spacecraft from the first leg of the mission to the impact of the target, its motion is described by equations (2.41), integrated using the RK parallel shooting method (Sect. 2.3.1). The spacecraft is provided with a low-thrust propulsion system, with a fixed maximum thrust magnitude $T_{max} = 0.05 \cdot m_{s/c,0}$, an exhaust velocity of $c_{exh} = 1\ AU/TU$ and in-plane (β) and out-plane (γ) pointing angles to be optimized along the trajectory.

When the spacecraft impacts the asteroid, it transfers momentum, yielding a small change in the asteroid velocity, $\delta \overrightarrow{v}_I$. Using a modified code based on A. Koehler's (Sect. 3.1.3), the asteroid motion is propagated with MATLAB *ode113* solver - similar to ode45 in operation but it can obtain a solution faster and with fewer function evaluations - starting from the interception time (t_I), and stops if the event of entering the Earth's sphere of influence occurs within a certain time period (e.g. three years). The asteroid equations of motion are

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = -\frac{\mu_{\odot} x}{r^3} + a_{x(\text{♀})} + a_{x(\oplus\text{☽})} + a_{x(\text{♂})} + a_{x(\text{♃})} \\ \dot{v}_y = -\frac{\mu_{\odot} y}{r^3} + a_{y(\text{♀})} + a_{y(\oplus\text{☽})} + a_{y(\text{♂})} + a_{y(\text{♃})} \\ \dot{v}_z = -\frac{\mu_{\odot} z}{r^3} + a_{z(\text{♀})} + a_{z(\oplus\text{☽})} + a_{z(\text{♂})} + a_{z(\text{♃})} \\ \dot{m} = -\frac{T_{max}}{c_{exh}} \end{cases} \tag{3.3}$$

where the perturbing accelerations - defined by equation (2.42) - are generated respectively by Venus, the Earth-Moon system (when the spacecraft is farther than 0.01 AU with respect to the system's barycenter), Mars and Jupiter. The positions and velocities of asteroids and planets - including Earth - are found through SPICE routines as usual. The State Transition Matrix (STM) is then used to find the perturbation in position ($\delta \vec{r}$) and velocity ($\delta \overrightarrow{v}$) of the asteroid as a result of the initial impulse ($\delta \overrightarrow{v}_I$)

provided by the spacecraft, at the time of the entry to Earth's sphere of influence (t_{SoI}). The STM is expressed as [18]

$$\begin{bmatrix} \delta\vec{r} \\ \delta\vec{v} \end{bmatrix} = \begin{bmatrix} \tilde{R} & R \\ \tilde{V} & V \end{bmatrix} \begin{bmatrix} \delta\vec{r}_0 \\ \delta\vec{v}_0 \end{bmatrix} \tag{3.4}$$

where $\delta\vec{r}_0$ and $\delta\vec{v}_0$ are respectively the perturbation in position and in velocity experienced by the asteroid at the impact site, and the matrix is time-dependent. In particular, the $\delta\vec{v}_0$ is calculated assuming that the occurred impact is analogous to an inelastic collision, then considering the conservation of the total momentum:

$$\delta v_0 = \frac{m_{s/c}(v_{s/c} - v_{\circledast})}{m_{\circledast} + m_{s/c}} \,. \tag{3.5}$$

Since $\delta\vec{r}_0 \approx 0$, the perturbed position and velocity vectors at a time t can be found as

$$\begin{cases} \delta\vec{r}(t) = [R]\delta\vec{v}_I \\ \delta\vec{v}(t) = [V]\delta\vec{v}_I \end{cases} \tag{3.6}$$

with

$$[R] = \frac{r_0}{\mu}(1 - F)[(\vec{r} - \vec{r}_0)\vec{v}_0^T - (\vec{v} - \vec{v}_0)\vec{r}_0^T] + \frac{C}{\mu}\vec{v}\,\vec{v}_0^T + G[I] \tag{3.7}$$

$$[V] = \frac{r_0}{\mu}(\vec{v} - \vec{v}_0)(\vec{v} - \vec{v}_0)^T + \frac{1}{r^3}[r_0(1 - F)\vec{r}\,\vec{r}_0^T - C\vec{r}\,\vec{v}_0^T] + G_t[I] \tag{3.8}$$

$$F = 1 - \frac{U_2}{r_0} \tag{3.9}$$

$$G = t - t_0 - \frac{U_3}{\sqrt{\mu_\odot}} \tag{3.10}$$

$$G_t = 1 - \frac{U_2}{r} \tag{3.11}$$

$$C = \frac{1}{\sqrt{\mu_\odot}}(3U_5 - \chi U_4) - (t - t_0)U_2 \tag{3.12}$$

$$\chi = \sqrt{a}(E - E_o) \tag{3.13}$$

where a, e and E are the asteroid orbital elements at $t = t_{SoI}$, E_0 at $t_0 = t_I$ and $U_k(\chi, a)$ are the universal functions described by Battin [18]. The code implementing the STM has been developed by L. Malagni, as aforementioned. The resulting $\delta\vec{r}$ and $\delta\vec{v}$ are added to the position and velocity of the asteroid at the time t_{SoI}, then the integration of its EOM continues according to the modified Koehler's code, particularly identifying the asteroid closest approach to Earth (at t_f). Maximizing the perigee radius of the asteroid is the goal of the problem and will be further investigated in Sect. 3.2.3.

To summarize, the optimization parameters are the pointing angles of the $\Delta\vec{v}_L$ provided by the upper stage launcher vehicle (β_L and γ_L), the continuous thrust pointing angles (β, γ) of the spacecraft propulsion system, the launch time (t_L) and the interception time (t_I). The initial time t_0 is set to a specific date, chosen by knowing the predicted date of the closest approach of the unperturbed asteroid to Earth. The constraints that the NLP solver must satisfy relate to the defects equation (2.40) and the interception event (2.34), where the asteroid and the spacecraft position must match.

3.2.2 Apophis

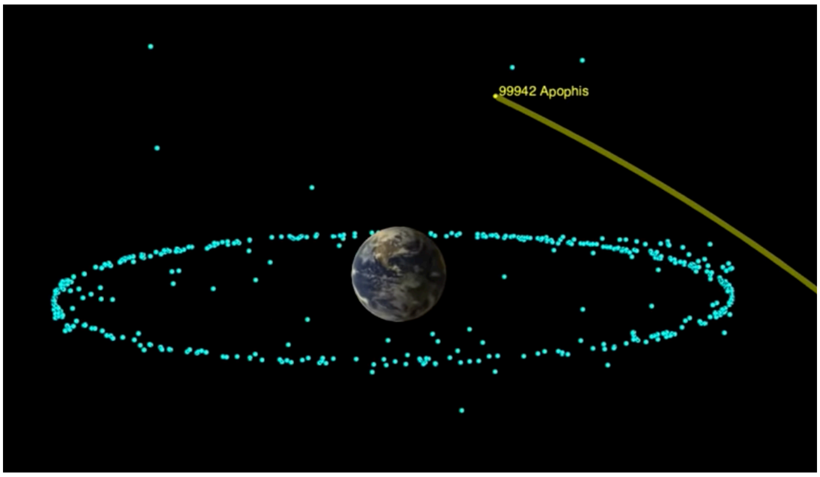


Figure 3.4 Apophis asteroid and Earth at the time of the asteroid's closest approach

The potentially hazardous asteroid chosen for the final problem is (99942) Apophis (2004 MN4), discovered on July 19th, 2004 by astronomers Roy Tucker, David Tholen, and Fabrizio Bernardi at the Kitt Peak National Observatory in Tucson, Arizona

[45]. It is a near-Earth asteroid (NEA) more than 1000 feet (over 300 meters) in size, with a mass of $2.1 \cdot 10^{10}$ kg and it passes close to Earth every seven years.

Upon its discovery, Apophis was briefly estimated to have a 2.7% chance of impacting the Earth on April 13th, 2029 which was the highest impact probability ever recorded (*Figure 3.4* - the blue dots are man-made satellites; credit: NASA/JPL-Caltech). Later astronomical observations and measurements predicted that the asteroid will harmlessly pass about 19,800 miles (31,900 kilometers) from our planet's surface [46], appearing brightly to the naked eye for several hours as its moves rapidly across the sky, over the Atlantic Ocean at its closest approach. Although it is considered a safe distance, Apophis will pass close enough to be between Earth and our Moon; in particular, the flyby will be within 0.1 times the lunar distance which is only a few Earth radii above the Earth's surface, so it will also travel within the orbits of some spacecraft revolving around the Earth [47]. An Earth encounter was also supposed to happen in 2036, but a research refinement in 2013 confirmed that in that year Apophis will quietly pass at a distant 0.388 au (about 150 lunar distances) from Earth [48]. There is still an impact probability in 2068, which is estimated to be low; in particular, the uncertainties that shift probabilities are dominated by non-gravitational perturbations such as the Yarkovsky effect and solar radiation pressure [49].

Smaller asteroids, on the order of 10 meters, flying by Earth at a similar distance are much more common to spot, whereas asteroids the size of Apophis are rarer and do not pass this close to Earth as often. Therefore, Apophis flyby is an incredible opportunity for science, in order to learn about an asteroid's interior and to help gain important scientific knowledge that could one day be exploited for planetary defense.

Apophis has been classified as a "Potentially Hazardous Asteroid" (PHA) by the International Astronomical Union (IAU) Minor Planet Center [47][49]. Even though Apophis is no longer a threat to Earth, it is a valid example of a PHA to study. The final problem focuses on its closest approach on April 13th, 2029 with the goal of increasing its missing distance. This requires only a very small change in Apophis' actual orbital elements, since the 31900 km missing distance is the distance Earth travels in 18 minutes along its orbit. Apophis' orbit elements for the date of its closest approach to the Earth are given in *Table 3.1*.

Table 3.1 Apophis Orbital Elements at Epoch April 13th, 2029

Symbol	Description	Value
a	*semi-major axis*	0.9214 AU
e	*eccentricity*	0.1957
i	*inclination*	3.42°
ω	*argument of*	126.62°
Ω	*longitude of ascending node*	203.79°
f	*true anomaly*	232.78°

3.2.3 The objective function

If a dangerous object was crossing Earth's trajectory, the main focus would intuitively be the prevention of the catastrophic impact intercepting the asteroid at the earliest possible time [6] and then applying a countermeasure that requires its deflection or destruction (Sect. 3.1.2). However, the ideal objective function of the optimal control problem under consideration is not a simple function of the path as it would be minimizing the interception time or maximizing the spacecraft final mass or velocity. The actual focus should be on deflecting the asteroid in the direction that most rapidly moves the impact point off of the surface of the Earth. In this regard, in his early work [41] Conway was interested in the maximization of the deflection magnitude, calculated using the state transition matrix (STM); this is not the best approach since, if the closest approach were near the eastern edge of the Earth, a large deflection that moves its location westward, but still on the Earth, is not as useful as a small deflection that moves it further east and off the Earth's surface (*Figure 3.5*). A slightly different idea was investigated by Englander and Conway (Sect. 3.1.3) [44] and is implemented in this work, that is maximizing the distance of the asteroid impact from the center of Earth but in terms of the difference between the "unperturbed" asteroid's perigee radius ($r_{p,\ unpert}$) and its value post-interception ($r_{p,\ pert}$). The STM method is also applied (Sect. 3.2.1). Also Vasile and Colombo [42], among others, considered the problem of optimal impact strategies to deflect potentially dangerous asteroids, but by maximizing the minimum orbit intersection distance between the asteroid and the Earth, so differently than maximizing the asteroid's closest approach distance on the projected date of the impact.

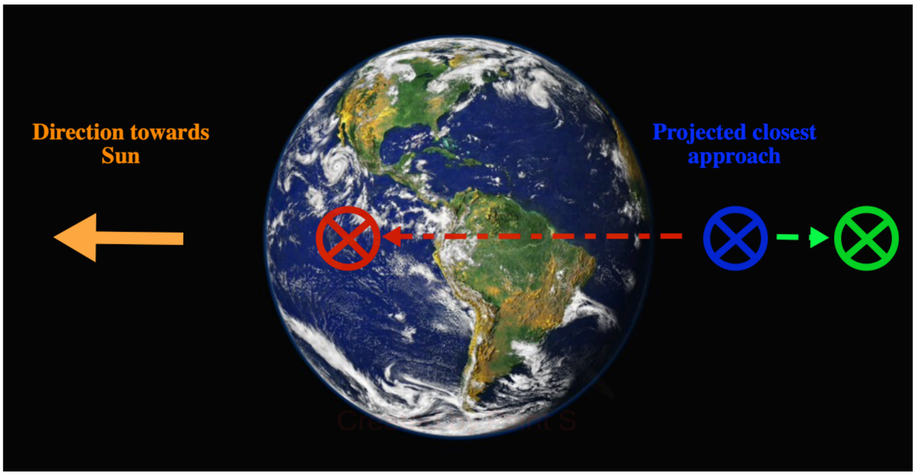


Figure 3.5 Deflection of the closest approach to Earth

Therefore, the cost function of the problem, formulated as an NLP, is

$$J = r_{p,\ unpert} - r_{p,\ pert}\ . \tag{3.14}$$

3.3 PSO solution as the initial guess for a direct transcription solution

Given the overview of the problem outlined above, this section will go into the initial guess sought for the final NLP program.

3.3.1 Initial guess for the NLP problem

The DTRK method employed within the final NLP problem is effective for computational speed and robustness. However, the problem of the asteroid deflection is complex and thus a valid initial guess is not obvious. Creating an initial guess means obtaining a vector containing the discrete-time histories of the state and control parameters, which are necessary for solving NLP problems by direct transcription methods [9]. As a general guideline, a "reasonable candidate" satisfies, at least at the 0th iteration, the EOM so that the initial nonlinear defects are all very small. It should also satisfy all constraints, initial and terminal, and boundary conditions related to the upper and lower bounds of the parameters of the NLP problem. The inherent difficulty in finding such an initial guess makes evolutionary or heuristics methods the ideal candidates.

Prior works (Sect. 3.1.3) investigated, for instance, the use of GA combined with a shape-based method, which did not succeed in locating a global minimum for this problem, emphasizing that the solution is very sensitive to the initial guess and multiple locally optimal solutions exist. The shape-based approach approximates the shape of a low-thrust trajectory typically with a 5th or 6th degree inverse polynomial in terms of polar or cylindrical coordinates; this method works very well for example for rendezvous missions [8]. For the case of this dissertation, it may not be much suited since it requires detailed knowledge of the states at the initial and terminal conditions (e.g. interception point).

The strategy finally adopted was the conversion of the NLP problem of asteroid mitigation into a PSO problem (Sect. 3.3.2). The PSO solution is then provided as an initial guess to the NLP problem. However, the DTRK approach cannot be adopted within the PSO solver because it implies a large number of controls, so the total amount of parameters would be too high for the PSO optimizer to work effectively unless a few segments are used for trajectory approximation, which would lead to poor accuracy. The Sims-Finlayson method [50] was also examined; it basically involves dividing of the trajectory into “legs” and “control nodes”, similarly to the RK parallel shooting approach, and a “match point” is associated with each leg. The trajectory is propagated forward in time from the previous control node of the leg to the match point and backward from the following control node of the leg to the match point. This multiple shooting technique with intermediate control nodes is simple, has robust convergence and can handle many intermediate encounters. This approach would not really solve the problem of handling several parameters, indeed it is better recommended for NLP solvers. Thus, the choice falls into the MATLAB function *ode45* for the integration of the equations of motion, so that the tradeoff between the number of points of the trajectory discretization and the smooth operation of PSO is reasonable.

Another possibility for an initial guess is the homotopy approach (Sect. 2.2.4) [6]. This method is less robust compared to the other presented. However, from experience, an initial guess that allows the optimizer to converge to an optimal trajectory may be exploited in other cases with different initial and/or terminal conditions from the desired case; therefore, new optimal trajectories can result starting from the converged solution as the new initial guess. This approach might be applied in future work for analyzing how the solution evolve if one or more of the fixed parameters change.

3.3.2 Conversion to a PSO problem

The PSO methodology has been proved effective and reliable in optimizing space trajectories, in fact it does not require either differentiability or continuity of the objective function, does not need any initial guess to generate the solution, is accurate in locating global minima and is intuitive to program. Everything mentioned above renders it a good choice for generating an initial guess for the complete NLP problem. An equivalent PSO problem of asteroid deflection is thus needed; in particular, two different versions were developed. Of course, in both of them, the spacecraft dynamics is governed by the equations of motion (2.41).

At first, only the spacecraft trajectory until the interception event was modeled and the objective function was focused on maximizing the momentum transfer from the spacecraft to the asteroid at the impact, in terms of maximum spacecraft momentum at the collision time

$$J = -\, m_{s/c,I} v_{s/c,I} + c_1 \left| ceq \right| \tag{3.15}$$

where ceq is the interception constraint (eq in Sec. 2.2.3) and c_1 is tuned to 500. This version is derived from the PSO program used to develop an initial guess for the asteroid interception problem described in Section 2.3.3, which was adapted for the asteroid Apophis. Incorporating the initial guess thus constructed into the full program resulted in a convergent solution (using the SNOPT solver).

The second PSO version was programmed as a complete conversion of the full NLP problem, including the objective function (3.14) which becomes

$$J = \left(r_{p,\ unpert} - r_{p,\ pert} \right) + c_1 \left| ceq \right| \tag{3.16}$$

where ceq is the interception constraint (2.34) and c_1 is set to 10^4.

The asteroid mitigation problem is a constrained optimization problem, and despite its simplicity and efficiency, the PSO technique can experience difficulties when dealing with the constraints of the problem, particularly equality constraints. Especially in the case of the more complex version using equation (3.16) as the fitness function, the coefficient that multiplies the equality constraints in absolute value must be chosen carefully to assess how much weight that term has with respect to the one representing the change in the perigee radius of the asteroid at t_f. Indeed, a small value may lead to excessive constraints violations, whereas a high value yields an ill-conditioned problem. In this work, c_1 was selected through a coarse but satisfactory trial-and-error approach.

On the other hand, inequalities are here handled similarly as it is explained in Sect.2.1.1: a high value is assigned to J if an inequality constraint is violated so that, among the other particles, the one corresponding to such a value will be discarded in favor of a more suitable “global best”, which is saved until a better solution is encountered by some particle. This approach has been adopted in order to help the algorithm to consider as optimal only feasible values of J and therefore to avoid the above-mentioned consequences - especially the one for which the first term of the expression (3.16) takes over the second. This inequality condition was selected by reasoning on orders of magnitude, i.e. assuming a δv_0 of the order of $[10^{-6}, 10^{-5}]$ km/s and that the Δt between interception (t_I) and asteroid reaching perigee (t_f) is of the order of 10 TU, the STM could result in a deflection of the order of 10^2 km. Consequently, an “if statement” was placed setting J to 10^4 when the first term of equation (3.16) resulted greater than 1000 km or positive - so that the perturbed asteroid would pass even closer to Earth. Furthermore, another "if statement" in the code prevents absurd values of $t_L > t_I$ by bypassing the integration of *ode45* and setting the variables of final position and final spacecraft mass to zero, so that the interception constraint is not satisfied and a value destined to be discarded is associated with J.

3.3.3 PSO solution

The optimization parameters of the PSO program are the in-plane and out-of-plane thrust pointing angles, the Δv_L pointing angles, and the time variables corresponding to the launch and interception events. The launch window opens (t_0) on January 1, 2026 ($\approx$ 163.4543 TU). Other dates for t_0 (e.g., January 1, 2023) were tested to assess whether a different and/or better result could be achieved. Yet it was observed that the spacecraft would have waited about the same date for the launch. Therefore, it is assumed that a kinetic impactor is available at the time t_0; a warning time of at least 1 to 2 years is typically required for smaller asteroids, hence a departure in 2026 is preferred. In general, having available time could also enable a previous mission to study the asteroid up close and send information back to Earth, greatly increasing the chance of success of an asteroid mitigation mission.

The number of particles of the population is set according to the complexity of the problem to be optimized [15]. Generally, a greater number of particles can better handle several unknown parameters, since a higher density of particles allows the search space to be better explored, enhancing the probability of detecting the global optimal solution when multiple local minima exist. In the cases examined, the former used 100 particles, while the latter used only 30 particles because the process was facilitated by inserting the best particle resulting from the first version - containing the optimal parameters to maximize the momentum exchanged at the intercept - into the population of the PSO algorithm of the second version, hence reducing the processing time. The results of the two versions of the program are compared in *Table 3.2*.

Table 3.2 Apophis mitigation problem - PSO results

Variable	Description	1st version (momentum max)	2nd version (deflection max)
β_L	*in-plane Δv_L pointing angle*	- 138.5641°	- 52.6949°
γ_L	*out-of-plane Δv_L pointing angle*	104.2669°	25.4336°
t_L	*launch time*	171.7158 TU April 26, 2027	172.9019 TU July 4, 2027
t_I	*interception time*	176.8390 TU February 18, 2028	176.4785 TU February 21, 2028
$m_{s/c,I}$	*spacecraft mass at t_I ($m_{s/c,0} = 10^4$ kg)*	0.74 $m_{s/c,0}$	0.82 $m_{s/c,0}$
$a_{s/c,I}$	*spacecraft acceleration at t_I*	0.0672 AU/TU^2	0.0609 AU/TU^2

It can be observed that the interception event occurs with a few days difference in the two cases, while the launch event happens with a little more than two months difference so that in the first case the flight time will last longer. The directional angles of the impulse given by the upper stage of the launcher are very different, but indeed at launch the Earth is located at a different point with respect to the asteroid in the two cases. The thrust pointing angles along the trajectory are discretized at 14 points, hence their trends are not smooth, therefore not very indicative, and only the results from the second version were chosen to be reported (*Figure 3.6*).

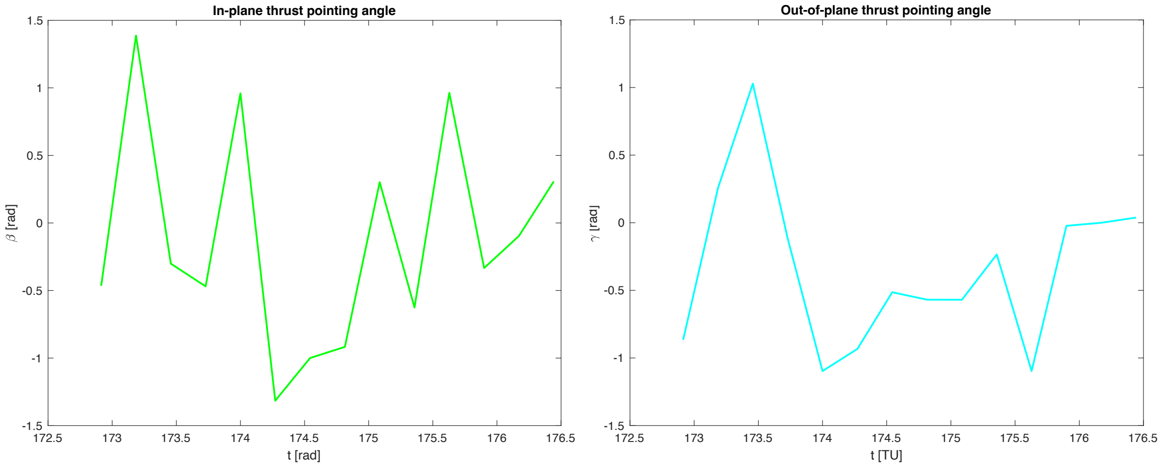


Figure 3.6 Apophis mitigation problem - PSO program: thrust pointing angles

The number of iterations also plays a role in achieving the optimal solution (here 100 iterations were used). Generally, increasing their amount can only yield an unchanged or improved result, given the nature of particles behavior. However, given the complexity of the “complete” PSO program and the inherent difficulty of the PSO algorithm in handling equality constraints, a worsening of the solution, involving a huge distance between Earth and Apophis, has also been experienced. This result is insignificant since the constraint is violated and thus the interception event does not even occur: the second term of expression (3.16) is relaxed in favor of the first one, which is more easily prioritized by the PSO algorithm. In addition to the countermeasures explained above (Sect. 3.3.2) - such as inequality constraints and a proper tuning of the constant c_1 -, a further modification can be made to the program and, in particular, to the optimizer algorithm, thus solving this drawback. It consists in inserting the same "if statement" of inequality related to the feasible deflection range ($[-1000,\ 0]\ km$) in the calculation of the function J, so that when it is not satisfied the inertial term of the particle velocity (eq. (2.5)) becomes zero and the particle tends to pursue only the directions of "personal best" and "global best". This modified version employed an increased number of particles (100) and iterations (200), setting the value of the constant c_1 to 10^5. This value is not too high considering that the real deviation has been expressed in kilometers, while the constraint is in astronomical units. The perigee deflection value closest to the NLP program result (Sect. 3.4) was obtained from this last modified version: the solutions are $-316.6405\ km$ in the first case and $-337.4335\ km$ in the second. In fact, the $ceqs$ are respectively of the order of 10^{-3} and 10^{-7}, reflecting that the last version is closer to the searched optimum. The pointing angles of the $\Delta\vec{v}_L$ differ by 2°÷3° from one other, also the time variables to be optimized are quite close (few days of difference) and the trends of the thrust pointing angles are superimposable to a certain degree of accuracy. However, no convergent solutions were found by the NLP solver that uses the modified PSO result as the initial guess, while a local minimum was located with the other, less accurate version. Often, nonlinear solvers enable a smoother resolution of an optimization problem when the initial guess is not too close to the optimal solution, indeed they tend to move away from it especially in the first iterative steps.

3.4 Optimal Solution

The complete Apophis deflection program includes spacecraft states as optimization parameters, as well as controls on pointing angles and time variables. The initial guess for the states, which is necessary to use the DTRK method, is derived from the output of *ode45* in the PSO program. The two initial guesses, previously illustrated in *Table 3.2*, were inserted in the NLP program using the *fmincon* optimizer leading to two converged solutions (shown *Table 3.3)*.

The number of segments used is respectively 13 and 29, thus the total number of variables is 260 in the first case and 564 in the second case. The constraints to be satisfied are typically the defects equations (2.40) of the DTRK method and the interception condition (2.34). It can be observed from *Table 3.3*. that the optimal launch date occurs in late April and the asteroid-spacecraft impact takes place about 8 months later, with a shorter flight time in the second case. Moreover, the feasibility in both versions is very satisfactory, as well as the deflection magnitude which is more effective in the second case.

However, because the second phase of deflection was not accounted for in the PSO maximum momentum solution, the initial guess experienced gave a deflection in the wrong direction bringing Apophis closer to Earth in the very first iterations.

Table 3.3 Apophis mitigation problem - NLP results

Variable	Description	1st version (max momentum guess)	2nd version (max deflection guess)
β_L	*in-plane* Δv_L *pointing angle*	-137.0425°	- 113.0090°
γ_L	*out-of-plane* Δv_L *pointing angle*	21.4447°	36.1682°
t_L	*launch time*	171.6186 TU April 20, 2027	171.7292 TU April 27, 2027
t_I	*interception time*	176.1818 TU January 10, 2028	175.9608 TU December 29, 2027
$m_{s/c,I}$	*spacecraft mass at* $t_I\,(m_{s/c,0} = 10^4\ kg)$	$0.77\ m_{s/c,0}$	$0.79\ m_{s/c,0}$
$a_{s/c,I}$	*spacecraft acceleration at* t_I	0.0648 AU/TU^2	0.0634 AU/TU^2
	Deflection of Apophis at perigee	- 414.0499 km	- 599.4544 km
	Feasibility	$\sim 7 \cdot 10^{-9}$	$\sim 3 \cdot 10^{-8}$

Probably, under the conditions that were set, the optimal direction of interception that maximizes momentum does not correspond to the one that moves the asteroid as far away from Earth as possible. Therefore, the NLP solver had difficulty locating a local minimum, in fact the solution was obtained by stopping processing at an iteration associated with a good output that was then given as input to the same program.

The graphs resulting from the second version of the PSO program (*Table 3.3*) are shown in *Figures 3.7 - 3.12*.

The semi-major axis (a) during the low-thrust arc (*Figure 3.11*) has been calculated through the vis-viva equation, also referred to as the orbital energy invariance law, describing the motion of orbiting bodies:

$$-\frac{\mu_\odot}{2a} = \frac{v^2}{2} - \frac{\mu_\odot}{r} \quad . \tag{3.17}$$

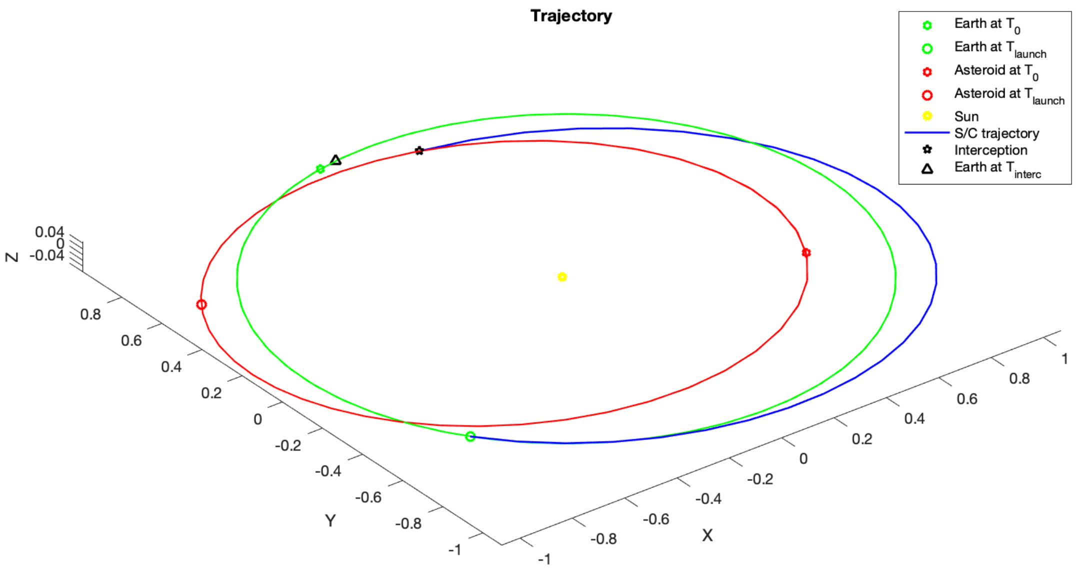


Figure 3.7 Apophis mitigation problem - NLP program: trajectories

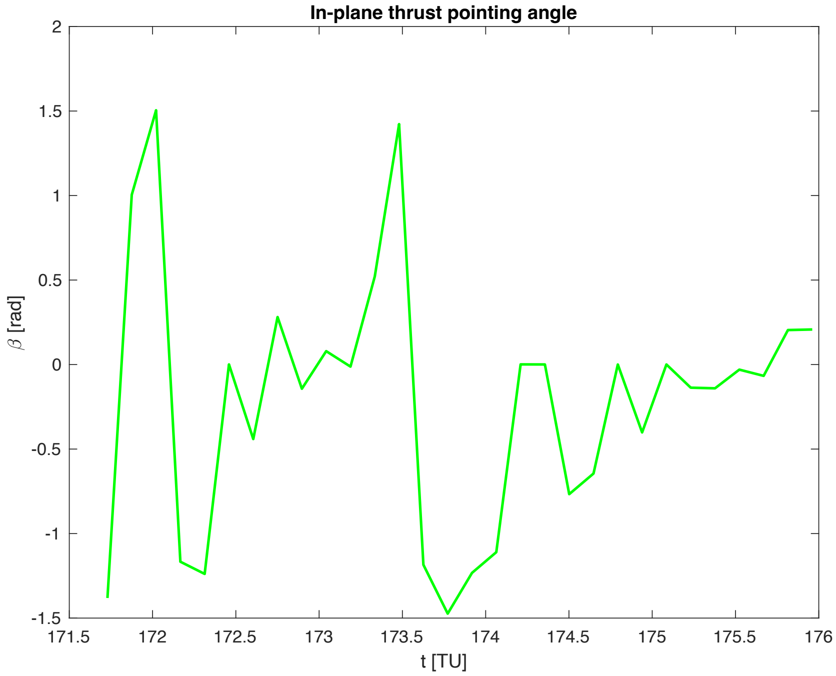


Figure 3.8 Apophis mitigation problem - NLP program: in-plane thrust pointing angle

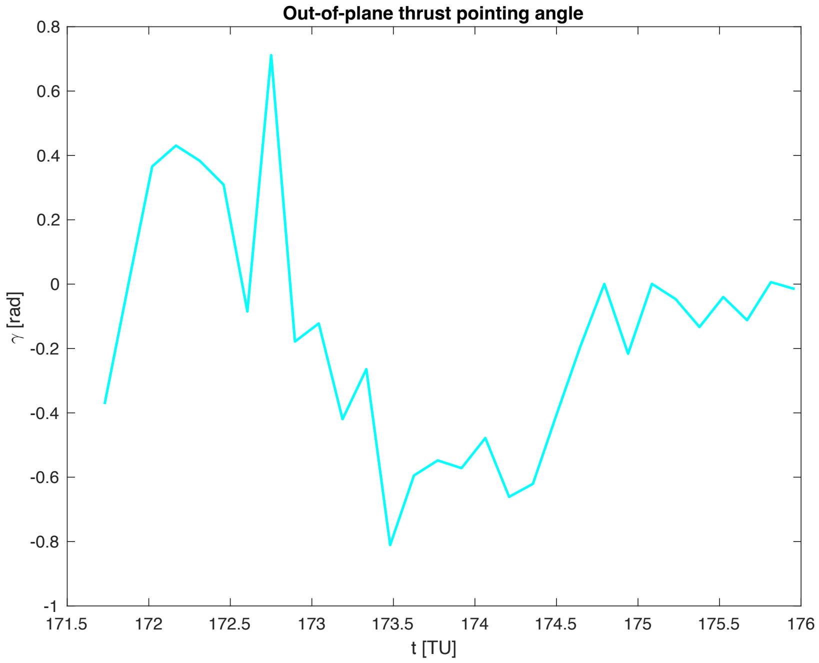


Figure 3.9 Apophis mitigation problem - NLP program: out-of-plane thrust pointing angle

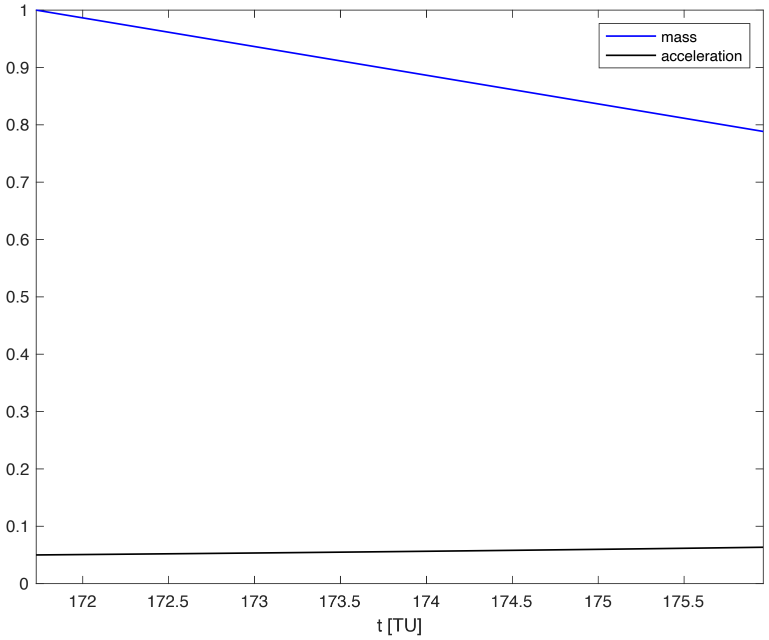


Figure 3.10 Apophis mitigation problem - NLP program: s/c mass and acceleration

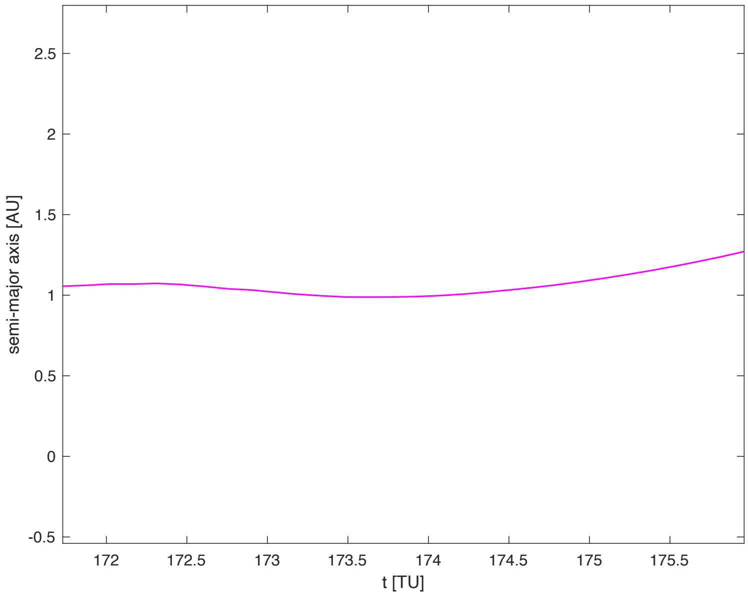


Figure 3.11 Apophis mitigation problem - NLP program: s/c trajectory semi-major axis

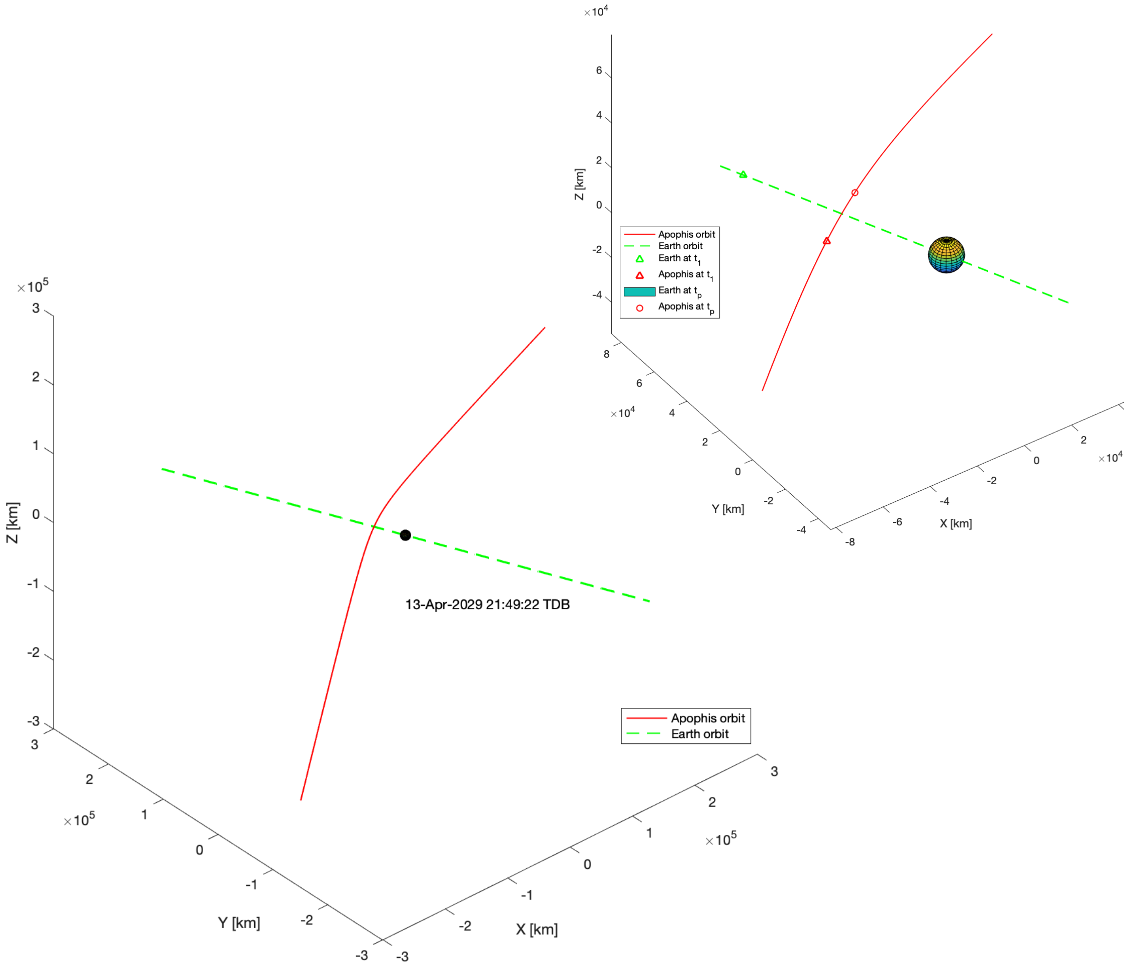


Figure 3.12 Apophis mitigation problem - NLP program: Earth-centered frame

Perturbed Apophis would pass at $32355\ km$ from the Earth surface (considering the equatorial radius of $6378\ km$) on the 13th April, 2029. *Figure 3.12* zooms in on the closest approach of Apophis to Earth in a geocentric reference. The trajectory of the asteroid within the Earth's sphere of influence is considered to be a planetary flyby, so the miss distance from the Earth's center could be calculated through the characteristic equations of such a hyperbolic trajectory:

$$r_{miss} = a_{f/b}(1 - e_{f/b}) \tag{3.18}$$

where

$$a_{f/b} = -\frac{\mu_{\oplus}}{v_{\infty}^{2}} \tag{3.19}$$

with the hyperbolic approach velocity given by

$$v_{\infty} = \left\| \vec{v}_{\circledast}(t_{\infty}) - \vec{v}_{\oplus}(t_{\infty}) \right\| \tag{3.20}$$

and the eccentricity is expressed as

$$e_{f/b} = csc(\delta/2) \tag{3.21}$$

with the turning angle found as

$$\delta = 2\tan^{-1}\left(\frac{\mu_{\oplus}}{\left\| \vec{r}_{\circledast}(t_{\infty}) - \vec{r}_{\oplus}(t_{\infty}) \right\| \cdot v_{\infty}^{2}} \right) . \tag{3.22}$$

The quantity maximized in this study is exactly r_{miss}.

Once the asteroid has been deflected, it is necessary to verify that the Earth is not again exposed to an impact or excessively close encounter in subsequent years. The perturbed motion of Apophis is then propagated as was done previously for the deflection phase assessment. This verification was initially performed over the next two to three years, as some time would be required to prepare for a new mission. Having secured this time frame, a simulation was performed over multiple years. Considering that a low probability of impact of the unperturbed asteroid in 2036 had been found, a time frame of eight years after the closest approach in 2029 was selected. The minimum approach was measured in September 2037 but close to the boundaries of Earth's conventional sphere of influence ($\sim 10^6 \; km$ radius).

The results seen were obtained via the nonlinear solver *fmincon*, which performed better than *snsolve*. SNOPT is a "sparse" optimizer, but it was used in a non-sparse fashion, numerically determining the Jacobian matrix of the very large system. Better performance could be achieved by providing only the few non-zero elements of the Jacobian.

Chapter 4: Conclusions

4.1 Summary

Building the final program step by step allows focus to be placed on one key element at a time, as well as ensuring that each building block works by itself, avoiding carrying around errors that are difficult to identify in a larger and more complex problem. The asteroid hazard mitigation program was developed using this approach to maximize the perigee radius of the Apophis asteroid at its closest approach to Earth by intercepting it with a low-thrust kinetic impactor, so that an eventual impact or less catastrophic, but equally undesirable, consequences are more likely to be avoided.

The problem under consideration is formulated as a nonlinear program using the RK parallel shooting method. The spacecraft was assumed to be ready for launch on January 1, 2026. From this date on, the optimizer can choose when to perform it. The modeled spacecraft has a mass of $10^4\ kg$, and can provide a maximum thrust of $T_{max} = 0.05 \cdot m_0$ and an exhaust velocity of $c_{exh} = 1\ AU/TU$; an additional contribution to the impactor trajectory is made by the $\Delta v = 0.6\ AU/TU$ provided by the upper stage of the launch vehicle. The optimal launch occurs in late April 2027 and the interception event at the end of December 2027. A deviation of about $-600\ km$ is found with respect to the miss distance of the unperturbed asteroid on April 13, 2029. Furthermore, it has been shown that the perturbed asteroid would not become a threat until at least the year 2037.

The solution method is very sensitive to the initial guess. A sub-optimal solution close to the previous one with a deviation of about -400 km was also found using a different initial guess. The initial guesses provided were derived from a program similar to the main one, but employing the heuristic Particle Swarm Optimization algorithm. Two main versions were developed, the first optimizing the momentum exchanged at the spacecraft-asteroid interception and the second maximizing the deflection of the asteroid at perigee, as in the main program. The latter case has been shown to be effective and to have good potential for improvement.

4.2 Future work

The presented results might serve as a guide for future work. The main areas that could potentially be improved are the fidelity of the model, the solution space by further working on the initial guess, testing the model on different targets, and paying more attention to the user interface. In particular, the accuracy of the model can be improved by incorporating more details about the asteroid (e.g. shape, composition, internal structure), the spacecraft (e.g. propulsion system), the launcher (e.g. characteristics of a chosen vehicle) and the kinetic impact itself (i.e modeling the ejecta process and estimate its contribution on the momentum transfer). More perturbations might be considered such as the effect of Earth's non-sphericity (J2 effect), the effect of other planetary gravities, solar radiation pressure and the solar wind. The exit of the spacecraft from the Earth's sphere of influence might be an additional feature of the scenario model.

Currently, the spacecraft uses thrust continuously, as it is assumed that adding an acceleration parameter would not significantly affect the results: the urgency of reaching the asteroid could justify exploiting all available thrust. Future work could evaluate whether using both low-thrust and coasting arcs would ultimately lead to a better solution. For example, it could be the case that, by burning less fuel, the spacecraft possesses more mass at the time of collision with the asteroid, and this feature may be assessed as advantageous by the optimizer. Moreover, future work could incorporate planetary flybys, so that the spacecraft could choose whether to make a close approach to Venus or Mars to gain speed and significantly change its trajectory without expending any propellant. As a result, new optimal interception geometries could result.

The methodology adopted in the Apophis asteroid detour program can lead to several optimal solutions depending on the quality of the initial guess. Indeed, in this type of problem there are several local minima that can be found by an NLP solver. In future work, parameters such as mass, maximum thrust, associated spacecraft discharge velocity, intensity of Δv provided by the launcher, or a combination of these could be set differently in order to analyze how the solution, if any, is varied. In addition, the versatility of the model could be tested by changing targets. Low-inclination asteroids with Earth-like orbits, such as Apophis, are the most likely threat because they spend a lot of time in the vicinity of Earth, but they would impact Earth at a lower relative velocity than a high-inclination asteroid or a short- or long-period comet, which would release more energy on impact and cause more damage.

Finally, the initial guess obtained by exploiting the PSO heuristic algorithm has been shown to be quite effective in solving this type of problem and has a good potential. The performance obtained by the PSO algorithm could be improved in several ways; for example, the number of particles and iterations could be modified to obtain a more satisfactory result, or the weighting coefficients in the velocity update formula (2. - Sect. 2.1.1) could be adjusted more finely, or some experimental definition (e.g., eq. 2. in Sect. 2.1.1) could replace the value given to them. However, it should be noted that, contrary to intuition, an initial guess closer to the optimum does not always produce a better or even convergent solution. Thus, it is inferred that obtaining optimal solutions numerically is not straightforward and deterministic but is still something of an art requiring experience and patience.

References

[1] Young, J. W., Smith, R. E. Jr (1967) *Trajectory optimization for an Apollo-type vehicle under entry conditions encountered during Lunar return*, NASA TR R-258.

[2] Shirazi, A., Ceberio, J., and Lozano, J. A. (2018) *Spacecraft Trajectory Optimization: A review of Models, Objectives, Approaches and Solutions*, Progress in Aerospace Sciences.

[3] Conway, B. A. (2010) “The Problem of Spacecraft Trajectory Optimization”, *Spacecraft Trajectory Optimization*, edited by Conway, B. A., Cambridge University Press, pp. 1-13.

[4] Deep Space 1, JPL, NASA: https://www.jpl.nasa.gov/missions/deep-space-1-ds1

[5] Near Earth Asteroid Rendezvous (NEAR) Mission, NASA: https://nssdc.gsfc.nasa.gov/planetary/near.html

[6] Conway, B. A. (1997) *Optimal Low-Thrust Interception of Earth-Crossing Asteroids*, Journal of Guidance, Control, and Dynamics Vol. 20, No. 5, pp. 995-1002.

[7] Casalino, L. and Vavrina, M. A. (2018) *Optimal Power Partitioning For Electric Thrusters*, AIAA/AAS Astrodynamics Specialist Conference 20-24 August 2017, Stevenson, WA.

[8] Wall, B. J. and Conway, B. A. (2009) Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design, Journal of Guidance, Control, and Dynamics, Vol. 32, No. 1, pp. 95-101.

[9] Conway, B. A. and Paris, S. W. (2010) “Spacecraft Trajectory Optimization Using Direct Transcription and Nonlinear Programming”, *Spacecraft Trajectory Optimization*, edited by Conway, B. A., Cambridge University Press, pp. 37-76.

[10] Enright, P. J. and Conway, B. A. (1992) *Discrete approximations to optimal trajectories using direct transcription and nonlinear programming*, Journal of Guidance, Control, and Dynamics, Vol. 15, No.4, pp. 994-1002.

[11] Holland, J. H. *Genetic Algorithms*: https://www2.econ.iastate.edu/tesfatsi/holland.gaintro.htm

[12] Reeves, C. R. (2005) Chapter 3 *Genetic Algorithms.*

[13] Konak, A., Coit, D. W. and Smith, A. E. (2006) *Optimization using genetic algorithms: A tutorial*, Reliability Engineering and System Safety, Vol. 91, No. 9, pp. 992–1007.

[14] Eberhart, R. and Kennedy, J. (1995) A New Optimizer Using Particle Swarm Theory, IEEE, MHS'95 Sixth International Symposium on Micro Machine and Human Science 4-6 Oct. 1995, Nagoya, Japan.

[15] Pontani, M. and Conway, B. A. (2010) "Swarming Theory Applied to Space Trajectory Optimization", *Spacecraft Trajectory Optimization*, edited by Conway, B. A., Cambridge University Press, pp. 263-295

[16] Sentinella, M. R., PhD thesis (2008) *Development Of New Procedures And Hybrid Algorithms For Space Trajectories Optimisation*, Politecnico Di Torino.

[17] Hu, X. and Eberhart, R. (2002) *Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization*, Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002), Orlando, FL.

[18] Battin, R. H. (1999) *An Introduction to the Mathematics and Methods of Astrodynamics*, Revised Edition, American Institute of Aeronautics and Astronautics.

[19] Prussing, J. E, and Conway, B. A. (1993) *Orbital Mechanics*, Oxford University Press.

[20] Bate, R. R., Mueller, D. D., White, J. E. and Taylor, W. W. (2nd Edition 2020) *Fundamentals of Astrodynamics*, Dover Publications Inc., Mineola, New York.

[21] MathWorks Documentation for *fmincon*: https://it.mathworks.com/help/optim/ug/fmincon.html

[22] Gill, P. E., Murray, W., Saunders, M. A. and Wong, Elizabeth (2018) *User's Guide for SNOPT 7.7: Software for Large-Scale Nonlinear Programming*, CCoM Technical Report 18-1, University of California, San Diego.

[23] Gill, P. E., Murray, W. and Saunders, M.A. (2005) *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Review, Vol. 47, No. 1, pp. 99-131.

[24] Li, S., Zhu, Y. and Wang, Y. (2014) *Rapid design and optimization of low-thrust rendezvous/interception trajectory for asteroid deflection missions*, Advances in Space Research, Vol. 53, No.4, pp. 696–707.

[25] Gibbings, A., Vasile, M., Watson, I., Hopkins, JM. and Burns, D. (2013) *Experimental analysis of laser ablated plumes for asteroid deflection and exploitation*, Acta Astronautica, Vol. 90, No. 1, pp. 85–97.

[26] Planetary Defense, Center for Near-Earth Object Studies: https://cneos.jpl.nasa.gov/pd/

[27] NASA Office to Coordinate Asteroid Detection, Hazard Mitigation, NASA JPL: https://www.jpl.nasa.gov/news/nasa-office-to-coordinate-asteroid-detection-hazard-mitigation

[28] Koschny, D. and Drolshagen, G. (2015) *Activities in Europe related to the mitigation of the threat from near-Earth objects*, Advances in Space Research , Vol. 56, No. 3, pp. 549–556.

[29] Dawn, Solar System Exploration, NASA Science: https://solarsystem.nasa.gov/missions/dawn/overview/

[30] Stardust, JPL, NASA: https://www.jpl.nasa.gov/missions/stardust

[31] Rosetta, Science and Exploration, ESA: https://www.esa.int/Science_Exploration/Space_Science/Rosetta

[32] OSIRIS-REx, Asteroid Sample Return Mission: https://www.asteroidmission.org

[33] Double Asteroid Redirection Test (DART) Mission, Planetary Defence, NASA: https://www.nasa.gov/planetarydefense/dart

[34] Cheng, A. F., Michel, P., Jutzi, M., Rivkin, A. S., Stickle, A., Barnouin, O., Ernst, C., Atchison, J., Pravec, P., Richardson, D. C., AIDA team and AIM team, (2016) *Asteroid Impact & Deflection Assessment mission: Kinetic impactor*, Planetary and Space Science, Vol. 121, pp. 27–35.

[35] Holsapple, K.A. and Housen, K. R. (2012) *Momentum transfer in asteroid impacts. I. Theory and scaling*, Icarus, Vol. 221, No. 2, pp. 875–887.

[36] Cheng, A. F., Stickle, A. M., Fahnestock, E. G., Dotto, E., Della Corte, V., Chabot, N. L. and Rivkin, A. S. (2020) *DART mission determination of momentum transfer: Model of ejecta plume observations*, Icarus, Vol. 352, 113989.

[37] Sanchez, J.P., Colombo, C., Vasile, M. and Radice, G. (2009) *Multi-criteria Comparison among Several Mitigation Strategies for Dangerous Near Earth Objects*, Journal of Guidance, Control, and Dynamics, Vol. 32, No. 1, pp. 1-55.

[38] Peloni, A., Dachwald, B. and Ceriotti M. (2018) *Multiple near-earth asteroid rendezvous mission: Solar-sailing options*, COSPAR publication, Advances in Space Research, Vol. 62, No. 8, pp. 2084-2098.

[39] Sugimoto, Y., Radice, G. and Sanchez, J. P. (2013) *Effects of NEO composition on deflection methodologies*, Acta Astronautica, Vol. 90, No. 1, pp. 14–21.

[40] Syal, M. B., Owen, J. M. and Miller, P. L. (2016) *Deflection by kinetic impact: Sensitivity to asteroid properties*, Icarus, Vol. 269, pp. 50-61.

[41] Conway, B. A. (2001) *Near-Optimal Deflection of Earth-Approaching Asteroids*, Journal of Guidance, Control, and Dynamics, Vol. 24, No. 5: engineering notes, pp. 1035-1037.

[42] Vasile, M. and Colombo, C. (2008) *Optimal Impact Strategies for Asteroid Deflection*, Journal of Guidance, Control, and Dynamics, Vol. 31, No. 4, pp. 858-891.

[43] Casalino, L. and Simeoni, F. (2012) *Indirect Optimization of Asteroid Deflection Missions with Electric Propulsion*, Journal of Guidance, Control, and Dynamics, Vol. 35, No. 2, pp. 423-433.

[44] Englander, J. (2008) Optimal Strategies for Deflecting Hazardous Near-Earth Objects via kinetic impactor, M.S. Thesis, University of Illinois at Urbana-Champaign.

[45] Giorgini, J. D., Benner, L. A. M., Ostro, S. J., Nolan, M. C. and Busch, M. W. (2008) *Predicting the Earth encounters of (99942) Apophis*, Icarus, Vol. 193, No. 1, pp. 1–19.

[46] Apophis, NASA Science:
https://solarsystem.nasa.gov/asteroids-comets-and-meteors/asteroids/apophis/in-depth/

[47] Goldstone Radar Observations Planning: 99942 Apophis in 2021: https://echo.jpl.nasa.gov/asteroids/Apophis/apophis.2021.goldstone.planning.html

[48] 99942 Apophis (2004 MN4), Center for Near Earth Object Studies:
https://cneos.jpl.nasa.gov/doc/apophis/

[49] Hirabayashi, M., Kim, Y. and Brozović, M. (2021) *Finite element modeling to characterize the stress evolution in asteroid (99942) Apophis during the 2029 Earth encounter*, Icarus, Vol. 365, 114493.

[50] Sims, J. A., Finlayson, P. A., Rinderle, E. A., Vavrina, M. A., and Kowalkowski, T. D. (2006) *Implementation of a Low-Thrust Trajectory Optimization Algorithm for Preliminary Design*, AIAA/AAS Astrodynamics Specialist Conference and Exhibit 21-24 August 2006, Keystone, CO.