AAS 21-787

ADAPTIVE MESH REFINEMENT SCHEME FOR EFFICIENT PARAMETRIC ANALYSIS AND TRAJECTORY DESIGN

Andrea Pasquale* **and Michèle Lavagna†**

The exploration of dynamical structures in multibody environments is a complex and computationally demanding task. Millions of propagations are usually required to characterize small regions of the state space around either a point, an orbit, or a trajectory of interest. Even though a PC can handle such simulations, limitations associated to the computational times exist. The paper presents a flexible algorithm to cleverly explore the state space for parametric analysis, finer resolving it only in specific regions of interest. The scheme is based on a single-parameter method which refines an initial coarse mesh in case some conditions are meet. Some applicative cases are presented too.

INTRODUCTION

Orbit design in multibody environments usually leans on prior explorations of the dynamical structures of such environments. For the nature of the problem, in fact, it is possible to rely only on numerical scanning, supported by simplified models or approximations, to extract initial guesses of the desired orbit or trajectory. The initial guess is then passed to an optimization scheme which, at the end of an iterative process, retrieves the optimized orbit with the desired constraint. Within this two-steps approach, the most complex and time-consuming part stays the initial guesses generation, more than in the optimization itself. Poincarè maps,[1] parametric analysis and other *dynamics-informed* parameters maps[2–4] are used to accomplish this task. Since the global behavior of the dynamical systems under investigation is usually not unknown *a-priori* when dealing with complex astrodynamical problems, with an evenly distributed mesh, there is no guarantee that the system features can be efficiently resolved during a given analysis. Therefore this work proposes the development of an adaptive mesh refinement algorithm aimed to reduce the computational cost associated to maps generation, capturing the desired features in a faster and efficient way.

Mesh refinement algorithms have been already developed for different fields, including aerodynamics,[5] structural analysis,[6,7] astrophysics[8] and others.[9] Specialized mesh refinement algorithms have been proposed also in the computation of dynamical informed quantities, such as LCS,[10] where problem-specific tracking algorithms are needed. However, most of the available algorithms are usually based on the dynamics and then cannot be transferred to the astrodynamics field easily. In this paper the exploitation of a novel approach, aimed to generalize the way the mesh generation, handling and update for different astrodynamical problems is presented. The proposed algorithm results to be simple, flexible, scalable and easily parallelizable. The computational cost associated to two applications is evaluated and compared with a *brute-force* simulation time, showing the capabilities of the method and the algorithm.

*Ph.D. Student, Dept. of Aerospace Science and Technology, Politecnico di Milano, Italy.
†Full Professor, Dept. of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy.

BACKGROUND

Dynamical Models

In this paper the exploitation of different kind of *maps* generated within two models of increasing complexity, namely the Circular Restricted Three-Body Problem (CR3BP) and the Bicircular Restricted Four-Body Problem (BCR4BP), is presented. Those models are considered to capture the main features of the dynamics within the Earth-Moon-Sun system, which is the *dynamical laboratory* considered here for the application of the presented mesh refinement algorithm.

CR3BP The Circular Restricted Three-Body Problem (CR3BP) describes the dynamics of a small, massless, third body that moves under the gravitational attraction of two massive bodies, called primaries, without influencing their motion, in the hypothesis that the primaries moves in a circular orbit about their center of mass. The equation of motions are expressed in the *synodic reference frame* which is widely used for trajectory design purposes in the context of the CR3BP.[11] In this frame, the position of the primaries is fixed in time on the x (synodic) axis. Thus, the frame rotates with angular velocity equal to that of the primaries (two-body motion). Eq 1 shows the equations of motion in the non dimensional form:

$$\begin{cases} \ddot{x} - 2\dot{y} = \mathcal{U}_x \\ \ddot{y} + 2\dot{x} = \mathcal{U}_y \\ \ddot{z} = \mathcal{U}_z \end{cases} \tag{1}$$

Here $\dot{(\cdot)}$ and $\ddot{(\cdot)}$ denotes the first and the second derivatives with respect to the (non-dimensional) time, while $\mathcal{U}_{(\cdot)}$ indicates the partial derivative of the pseudopotential function with respect to the variable $(\cdot)$. The pseudopotential is, in this case, defined as:

$$\mathcal{U}^{(\mathrm{cr3bp})} = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{2}$$

where r_1 and r_2 represent the distance of the particle from the primaries and μ is the mass ratio between the moonlet and the whole system, which is defined as:

$$\mu = \frac{m_2}{m_1 + m_2} \tag{3}$$

BCR4BP If the motion of the massless particle is assumed to be influenced by the gravitational pull of three bodies instead of two, the Four-Body problem context is considered.[12] In this study it is assumed that: the two primaries P_1 and P_2 revolve in circular orbits about their barycenter, B and B and the third body P_3 moves in circular orbits around the center of mass of the whole system, B_0. In this case, within the primaries synodic frame, the equation of motion have the same form of Eq. 1, except for the fact that the pseudo-potential function is time dependant and has an additional term. In particular, the BCR4BP pseudo-potential can be written as:

$$\mathcal{U}^{(\mathrm{bcr4bp})} = \mathcal{U}^{(\mathrm{cr3bp})} + \frac{\mu_s}{r_s} - \frac{\mu_s(x\cos\alpha + y\sin\alpha)}{a_s^2} \tag{4}$$

with

$$\alpha = \omega_s t + \alpha_0, \qquad \omega_s = n_s - 1, \qquad a_s^3 n_s^2 = \mu_s + 1 \tag{5}$$

where ω_s is the mean angular velocity of the Sun in the synodic frame, n_s the mean angular velocity of the Sun in inertial frame, a_s the Sun to Earth-Moon Barycentre distance, μ_s the Sun mass and r_s the Sun position. All the previous quantities are normalized, as for the CR3BP.

Maps

Dynamical-informed maps are intended to be any kind of map generated as a product of a numerical propagation of a certain dynamics. More specifically, two kind of maps are exploited within this study: apse maps and Trajectory Class Index (TCI) maps.

Apse maps Poincarè maps are commonly used to explore the state space and extract general and local behaviours of the flow associated to a dynamical system. They are also used to reduce the dimensionality of the flow and thus to ease its visualization. The exploitation of Poincarè maps has been already proven to be successful when dealing with multibody trajectory design.[13] A parametrization which can be used to ease the exploration of the state space is and the visualization of certain peculiar characteristics is the Poincarè *apse map*,[14] which represents the locus of points where the position of a test particle relative to a primary P_i is orthogonal to its velocity, or:

$$\mathbf{r}_i \cdot \dot{\mathbf{r}}_i = 0 \tag{6}$$

where $\mathbf{r}_i$ and $\dot{\mathbf{r}}_i$ are respectively the position of a test particle relative to the P_i primary and its velocity. Periapsis and apoapsis then can be then distinguished by means of the sign of the radial acceleration, i.e. if $\ddot{r} \geq 0$ the point is a periapsis and if $\ddot{r} \leq 0$ the point is an apoapsis.

TCI maps When dealing with some kind of astrodynamics problems, can be useful to classify sets of trajectories on the base of some *categorical indexes*. To clarify this concept, consider that a trajectory is built once a set of free parameters $\boldsymbol{\eta}$ are assigned, such that:

$$\mathcal{Z}(\boldsymbol{\eta}) = \{\eta_1, \eta_2, \ldots, \eta_i, \ldots, \eta_q\} \tag{7}$$

More than the free parameters, a certain trajectory may have to respond to some constrains: here they are generally represented by the set $\zeta(\boldsymbol{\lambda})$. Then, a trajectory Γ can be expressed as

$$\Gamma \stackrel{\text{def}}{=} \mathcal{Z}(\boldsymbol{\eta}) \quad \text{subject to} \quad \zeta(\boldsymbol{\lambda}) \tag{8}$$

Then, trajectories belonging to a given dynamical group of propagations may be, for example, classified in one of the following categories:

1. **Impact** group, $\mathcal{I}_i$: Γ do impact a body i.
2. **Escape** group, $\mathcal{E}$: Γ is considered to permanently escape from the gravitational influence of a body or a system if it intersect only once its SOI
3. **Re-entry** group, $\mathcal{R}$: Γ exit and re-enter the SOI.
4. **Confinement** group, $\mathcal{C}$: Γ does belong to any of the other classes.

MESH REFINEMENT ALGORITHM

Overview

In this section an overview of the mesh refinement algorithm developed in this paper is presented. The idea is to have a *general, problem independent* approach to the mesh generation, its topological structure "handing" and refinement method, to cover a large area of applications (e.g. Poincarè sections, apse maps, FTLE maps, ...). Clearly, problem-specific refinement algorithms have better

performances for the specific application but cannot be applied or extended in a simple way to a different scenario.[15,16] However, the presented approach is thought to be flexible enough to have the intrinsic possibility to be tailored on each specific applications while keeping the same backend structure.

The mesh refinement approach is similar to the one presented in literature by different authors,[9] but has some differences in the mesh handling and the possibility to define problem-specific refinement methods and termination criteria. The general features of the algorithm are described hereafter and then detailed in the following paragraphs.

1. An **un-structured** mesh is considered, as composed by a list of m, n-dimensional vertexes, $\boldsymbol{\chi}_i$. This is preferred since the nature of the mesh allows to initialize the algorithm with any kind of structured and un-structured dataset. This allow the initial grid to be generated as simply as possible, with no emphasis to any part of the domain of interest as well as to have the possibility to implement ridges tracking algorithms.

2. **Delaunay triangulation** is used to create the create the initial mesh and handle the new points generated by the mesh refinement algorithm.[17,18]

3. The **refinement method** is specified by:

   - a scalar function, called **refinement-step function**, $\mathcal{J}(\boldsymbol{\chi}, \cdot)$ which depends on the specific problem. Here $\boldsymbol{\chi}$ can include states (e.g. positions, velocities) or other parameters (e.g. TOF, maneuvers magnitudes, ...), while $\mathcal{J}$ can be any non-linear function such that:
   $$\mathcal{J} : \mathbb{R}^n \to \mathbb{R} \tag{9}$$
   - a **binary relation**, $\mathcal{B}(\mathcal{J}, \mathbf{t}, \cdot)$, which is applied to the refinement-step function value, and is the heuristic which decide when the refinement step has to be performed or not. Here $\mathbf{t}$ is a set of parameters or threshold values used in the definition of the binary relation.
   $$\mathcal{B} : \mathbb{R} \times \mathbb{R}^q \to \mathbb{B}, \quad \text{with}\, \mathbb{B} = \{0, 1\} \tag{10}$$
   The following basic binary relations are here considered:
   (a) equality relation:
   $$\mathcal{B}(\mathcal{J}, \mathbf{t}) \stackrel{\text{def}}{=} \mathcal{J} = t_0 \tag{11}$$
   (b) inequality relations:
   $$\mathcal{B}(\mathcal{J}, \mathbf{t}) \stackrel{\text{def}}{=} \mathcal{J} > t_0 \tag{12}$$
   $$\mathcal{J} \geq t_0 \tag{13}$$
   $$\mathcal{J} < t_0 \tag{14}$$
   $$\mathcal{J} \leq t_0 \tag{15}$$
   (c) range relations:
   $$\mathcal{B}(\mathcal{J}, \mathbf{t}) \stackrel{\text{def}}{=} \mathcal{J} \in [t_0, t_1] \tag{16}$$
   $$\mathcal{J} \in (t_0, t_1) \tag{17}$$
   - a **refinement heuristic**, $\mathcal{H}(\mathcal{B}, \cdot)$ which define how the refinement step is performed. In this work the refinement heuristic is based on the *centroid* of the Delaunay n-simplexes defined hereafter.

The n-simplex

In general, the *elements* of the Delaunay triangulation are (topologically) n-simplexes, geometric objects that generalizes triangles and tetrahedra to *any* dimension. For example, a 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron. Specifically, a n-simplex is a n-dimensional polytope determined by convex combination of $n+1$ vertices. In particular, consider the $n+1$ points $\boldsymbol{\chi}_0, \ldots, \boldsymbol{\chi}_n \in \mathbb{R}^n$ affinely independent, the simplex is determined by:[19]

$$\mathcal{S}_n = \left\{ \theta_0 \boldsymbol{\chi}_0 + \cdots + \theta_n \boldsymbol{\chi}_n \;\middle|\; \sum_i \theta_i = 1 \quad \text{and} \quad \theta_i \geq 0 \quad \forall i \right\} \tag{18}$$

This geometrical object has some peculiar properties and characteristics, as well as different homological derived objects can be shown to exist depending on the value of n. In this paper, the 2D (n=2, 2-simplex or triangle) and 3D cases (n=3, 3-simplex or tetrahedron) are explored, but the algorithm can be extended to any arbitrary dimension.

For the sake of clarity, some properties associated to a generic n-simplex are here recalled. In particular, given the vertexes of the simplex, $(\boldsymbol{\chi}_0, \ldots, \boldsymbol{\chi}_n)$, it can be shown that:

- the *centroid* of the simplex can be computed as:

$$C_n = \frac{1}{n+1} \sum_i^n \boldsymbol{\chi}_i \tag{19}$$

- the *volume* of the a n-simplex can be found via Stein rule[20] as :

$$\Sigma_n = \left| \frac{1}{n!} \det \left(\begin{bmatrix} \boldsymbol{\chi}_0 & \boldsymbol{\chi}_1 & \cdots & \boldsymbol{\chi}_n \\ 1 & 1 & \ldots & 1 \end{bmatrix} \right) \right| \tag{20}$$

- the number of k-dimensional faces the n-simplex has can be recovered via:

$$n_k = \frac{(n+1)!}{(k+1)!(n-k)!} \tag{21}$$

Mesh Refinement Algorithm Structure and Specification

In this section the proposed general mesh refinement algorithm is described step-by-step. For the sake of clarity, the notation is firstly introduced. In particular, with $\mathcal{A}_k^i$, it is denoted a set $\mathcal{A}$, composed by k elements, at the i-th iteration of the mesh refinement algorithm. Then, the mesh refinement algorithm is described hereafter:

1. The algorithm is initialized with :
   - an initial discrete set of points, $\mathcal{P}_m^0 = \{\boldsymbol{\chi}_0, \ldots, \boldsymbol{\chi}_n\}$, where $\mathcal{P}_m^0$ represent a set composed by m, n-dimensional points, $\boldsymbol{\chi}_i$;
   - a refinement-step function, $\mathcal{J}(\boldsymbol{\chi}, \cdot)$;
   - a binary relation, $\mathcal{B}(\mathcal{J}, \cdot)$;

- a refinement heuristics, $\mathcal{H}(\mathcal{B}, \cdot)$;
- a termination criteria, $\mathcal{T}(\cdot)$, which can be tailored for the specific problem; in this study, the refinement termination criteria is based on the i-th step mean simplexes *volume* to the initial mean simplexes volume. This ratio is recalled $r_{\mathcal{T}}$ and the refinement steps are stopped whenever $r_{\mathcal{T}}$ is smaller than a given threshold.

2. Delaunay triangulation is performed over $\mathcal{P}_m^0$, $DT(\mathcal{P}_m^0)$, resulting in a l-dimensional set of n-simplexes, i.e.
$$\mathcal{Q}_l^0 = DT(\mathcal{P}_m^0) \tag{22}$$

3. If the termination criteria, $\mathcal{T}$, is not meet, then a refinement step is initialized, otherwise the mesh refinement steps are stopped.

4. The binary relation $\mathcal{B}$ is evaluated for each n-simplex in $\mathcal{Q}_l^0$ and the refinement heuristic $\mathcal{H}$ is applied whether the binary relation is satisfied. The refinement heuristics, together with the initial discrete set form the new set of points, $\mathcal{P}_m^1$.

5. The Delaunay triangulation is performed over $\mathcal{P}_m^1$, obtaining $\mathcal{Q}_l^1$. Step 3 to 5 are repeated until the termination criteria is meet.

For the sake of clarity, an in-depth description of the mesh refinement algorithm is presented for the 2D case, for its conceptual and visual simplicity. With reference to Fig. 1, once assigned the refinement-step function and the binary relation for the specific problem, a **refinement heuristics** based on the simplexes centroids is considered. In particular given any 2-simplex which has to be refined:

- its *centroid* is computed by means of Eq. 19 as:
$$C_2 = \frac{1}{3}\left(\boldsymbol{\chi}_0 + \boldsymbol{\chi}_1 + \boldsymbol{\chi}_2\right) \tag{23}$$
the refinement step function is evaluated at C_2 and the new point is added to the triangulation.

- $k = n - 1 = 1$ dimensional faces (*edges*) of the 2-simplex are recovered. Note that thanks to Eq. 21 it is possible to know a-priori that the 2-simplex has $n_1 = 3$. The centroid of each edge is then computed by means of Eq. 19 as:
$$C_1 = \frac{1}{2}\left(\boldsymbol{\chi}_0 + \boldsymbol{\chi}_1\right) \tag{24}$$
Then either the refinement step function is evaluated at each new point C_1 or its value is computed by linear interpolation. The new points are finally inserted in the triangulation.

Note that procedure here described can be extended to arbitrary dimension simplexes, having as a degree of freedom the number of $k < n$ dimensions the refinement algorithm has to perform. Finally, two kind of termination criteria $\mathcal{T}(\cdot)$ are considered: the first is based on the *mean area* of the elements and is specialized such that given:
$$A_{\text{mean}} = \text{mean}_i\, A_i = \text{mean}_i \left| \frac{1}{2} \det\left(\begin{bmatrix} \boldsymbol{\chi}_0^i & \boldsymbol{\chi}_1^i & \boldsymbol{\chi}_2^i \\ 1 & 1 & 1 \end{bmatrix} \right) \right| \quad \forall i = 1, \ldots, l \tag{25}$$

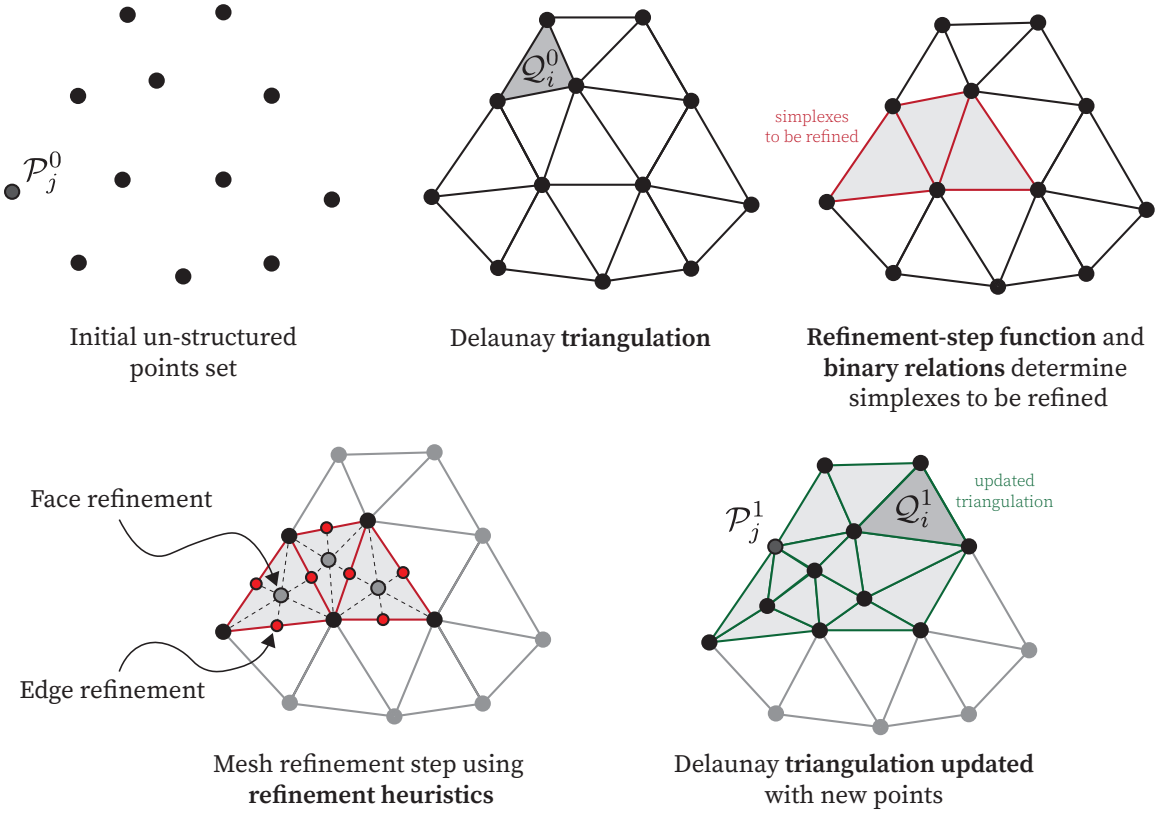


Figure 1: Mesh initialization and refinement step scheme for the 2D case. In this case the domain is represented by a 2-simplex. The refinement heuristics can be composed by a face refinement step (2-simplex), an edge refinement step (1-simplex) or both of them.

and given the initial elements mean area A_0, then:

$$\mathcal{T}(A_{\text{mean}}) = r_{\mathcal{T}} \leq r_t \tag{26}$$

where here $r_{\mathcal{T}} = A_{\text{mean}}/A_0$ and r_t is the desired area ratio threshold. The second, depends on the *min area* of the elements:

$$\mathcal{T}(A_{\text{min}}) = r_{\mathcal{T}} \leq r_t \tag{27}$$

where $r_{\mathcal{T}} = A_{\text{min}}/A_0$.

CASE STUDIES

In this section, a group of case studies where the proposed mesh refinement method is used as a basic tool for multibody orbit design is presented and discussed. In order to show the capabilities of the method as well as the possibility to easily extend the presented procedure to different kind of problems, the following applicative scenarios are considered:

- Characterization of Earth-Moon L2 Halo orbit disposal trajectories by means of a single manoeuvre.
- Design of LEO to Earth-Moon L2 Halo orbit transfers by means of Lunar flyby.

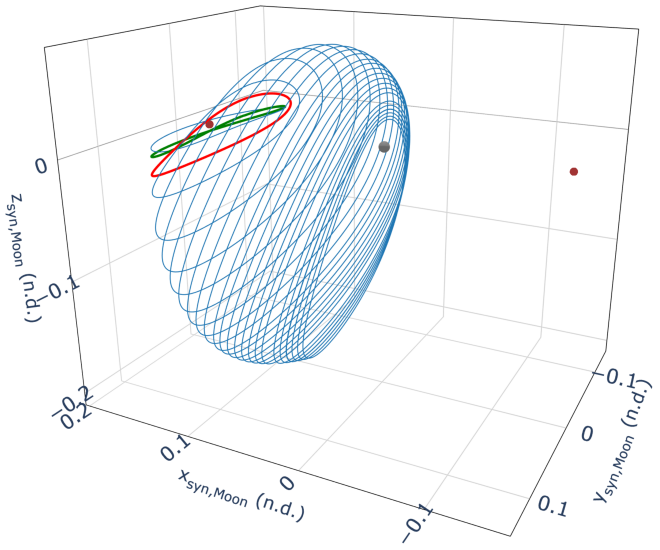


Figure 2: EML2 Halo family in the *synodic* frame, centred at the Moon. In red, the Halo considered in the Mission Scenario 1, while in green the one considered for the Mission Scenario 2.

Mission Scenario 1: Queqiao Disposal

In this mission scenario, the disposal manoeuvre design of a spacecraft on a EML2 Halo orbit is presented. With reference to Fig. 2, a EML2 Southern Halo orbit with $A_z = 14\,800\,\mathrm{km}$, $A_y = 35\,800\,\mathrm{km}$ and a period of $14.8\,\mathrm{days}$ is considered. The selection is driven by the fact that the orbit provide (nearly) continuous visibility of both the lunar Far Side and Earth ground stations, get continuous illumination from the Sun, and requires a little station keeping to be maintained in the long term. Moreover, a similar orbit will be flown by Cheng'e 4 relay satellite (Queqiao).[21] The disposal trajectories of interest are the ones which impact the Moon or the Earth.

Moon Impact Disposal Strategy The effect of a single impulse manoeuvre, applied in the direction of the inertial Moon-centric velocity is considered in this study.[22] The trajectory is modelled within the CR3BP considering two parameters:

- The epoch of the manoeuvre, represented by means of the normalized time on the orbit, ε, varying between (0, 1), such that $\varepsilon = t/P$, where t is the time on the orbit and P its period.
- The signed magnitude of the manoeuvre, ΔV. The direction of the manoeuvre is the inertial velocity direction.

Thus, the j-th disposal trajectories can be defined as:

$$\mathcal{Z}_j(\boldsymbol{\eta}) = (\varepsilon,\, \Delta V) \quad \text{subject to} \quad t \in (0, T_f) \tag{28}$$

The trajectories are propagated for a maximum time of `60 days` and **refinement-step function** is structured in such a way the propagation is stopped whenever the trajectory impact the Moon, the Earth or escape the Earth-Moon system, by assigning an *integer* value to each event: 0 for Moon impacts, 1 for Earth impacts, -1 for confined trajectories and -2 for escape trajectories. Regarding

the **binary relation**, a *range* relation condition is chosen as $\mathcal{B}(\mathcal{J}) \stackrel{\text{def}}{=} \texttt{J is in (0,1)}$. A **termination condition** on the mean area threshold is considered and imposed as $r_{\mathcal{T}} = \texttt{1e-6}$. The free variables are restricted to the ranges $\varepsilon \in [0, 1]$ and $\Delta V \in [-25\,\mathrm{m/s},\, 25\,\mathrm{m/s}]$.

In Fig. 3 the results of the analysis in terms of Moon impact trajectories are presented as a function of the two free variables. Note that the exploitation of the mesh refinement algorithm is extremely beneficial in this case, since the impact trajectories are well organized in distinct groups, resulting in a net reduction of the computational load. In fact, this implies that most of the data computed with a brute-force approach would be useless for this specific application. Two tests have been performed, varying the initial mesh resolution. A multiprocessing pool of 6, 2.80GHz Intel Core i7-7700HQ CPU processors is used for all the analysis.

- An **high-resolution** case, where an initial (`75x75`) mesh is considered. In this case, the termination condition is achieved in `6` refinement step, performed in `44.53 min`. To achieve the same result in terms of mean elements area, without the adaptive mesh algorithm, an analysis of `276.78 min` would be required. Thus the proposed algorithm is capable to *speedup the analysis* of about *6 times* which respect to the brute force approach.
- A **low-resolution** case, where an initial (`20x20`) mesh is considered. In this case, the termination condition is achieved in `7` refinement step, performed in `21.31 min`. Note that this case result in a speedup of *2 times* in comparison to the previous case, and the patterns obtained on the map are really similar to the one of the high-resolution case, as can be seen comparing Fig. 3a with Fig. 3d.

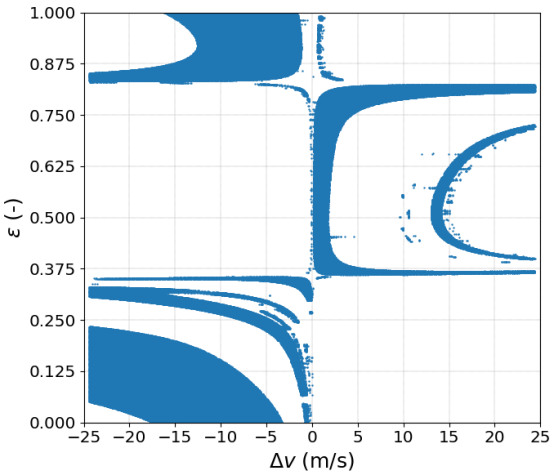
Then the true advantage of the refinement algorithm is, associated to a *coarse* mesh initialization, and its capability to identify the "target regions" also in case the initial mesh is really wide, allowing to *have insights of the dynamics of the problem in a really fast way*. This concept can be pushed to the limit relaxing the termination condition. In fact, comparing Fig. 3b, where an intermediate refinement step of the low-resolution case is presented, and the final refined map in Fig 3d, it is evident that a relaxed value on the termination condition would not impact a lot the global patterns present on the map. In this case, however, a total time of `107 seconds` would be needed to generate the map (it is approximately `1/25` of the high resolution map refinement time and `1/160` the computational time of high resolution map computed by the brute force approach).

Earth Re-entry Disposal Strategy In order to model Earth re-entry trajectories, the effect of a single manoeuvre applied in the direction of the Moon-centric velocity is evaluated within the BCR4BP instead of the CR3BP, to capture WSBs. To ε and ΔV, another parameter is added: ϑ_{Sun}, which represent the Sun angular position about the Earth-Moon barycenter. The j-th disposal trajectories can be defined, in this case, as:

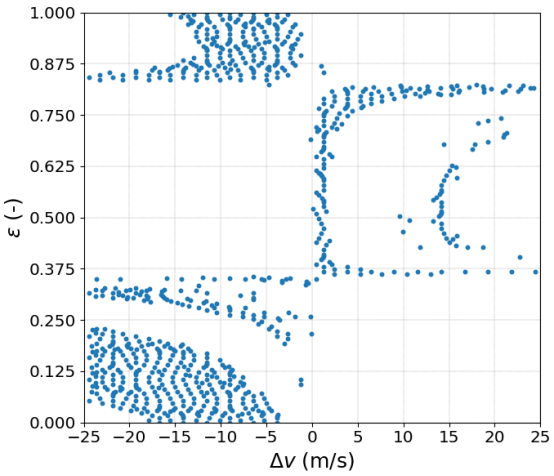
$$\mathcal{Z}_j(\boldsymbol{\eta}) = (\varepsilon,\, \Delta V, \vartheta_{\mathrm{Sun}}) \quad \text{subject to} \quad t \in (0, T_f) \tag{29}$$

The trajectories are propagated for a maximum time of `180 days` in order to account for reversed-WSB. Refinement-step function, binary relation and termination conditions are considered to be the same as before. The initial mesh is considered to be of `30x30x30` points with $\varepsilon \in [0, 1]$, $\Delta V \in [-25\,\mathrm{m/s},\, 25\,\mathrm{m/s}]$ and $\vartheta_{\mathrm{Sun}} \in [0\,\mathrm{deg},\, 360\,\mathrm{deg})$.

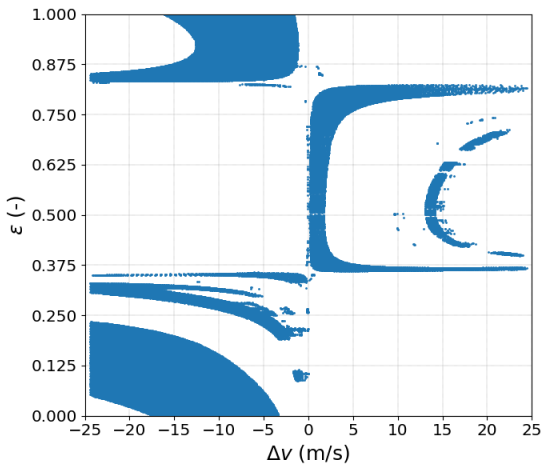
Fig. 4 present the results obtained by refining the initial mesh for `4` steps, with an overall computational time of `3.2 hours`. In this case both Earth & Moon impact trajectories are identified.



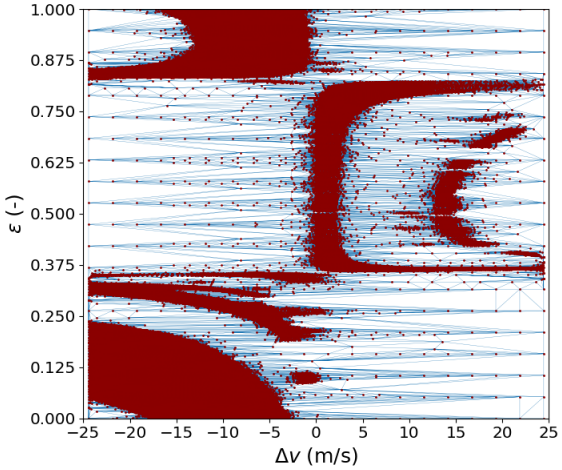
(a) *High* resolution refined map.



(b) *Low* resolution intermediate refinement step map.



(c) *Low* resolution refined map.



(d) *Low* resolution refined mesh.

Figure 3: Moon impact disposal options for the selected EML2 Southern Halo orbit, starting the refinement algorithm with two different initial grid resolution.

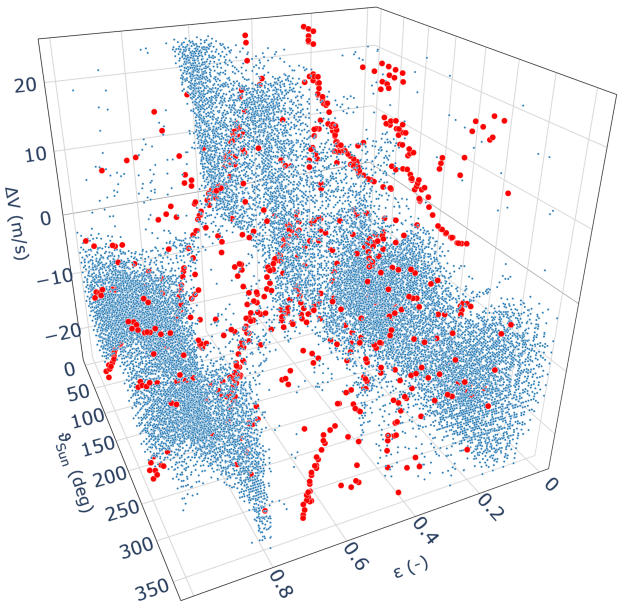


Figure 4: Moon impact (blue) and Earth impact (red) disposal options for the selected EML2 Southern Halo orbit vs free variables.

For comparison, a brute force run with the aim to obtain the same element mean area takes approximately `24 hours`, making the analysis performed using the proposed mesh refinement algorithm *7.5 times faster*. In this case, Earth re-entry trajectories are extracted. It can be shown that they are due to the combined attraction of the Earth, the Moon and the Sun, being in the weak stability region. Note that the R-WSB Earth re-entry trajectories have really *tight* departure windows associated, thus resulting in a much less robust solution than the previous one. However, the possibility to perform a second manoeuvre at the apogee of the trajectory seen from the Sun-Earth *synodic* frame can be exploited to finely tune the Earth re-entry.[23]

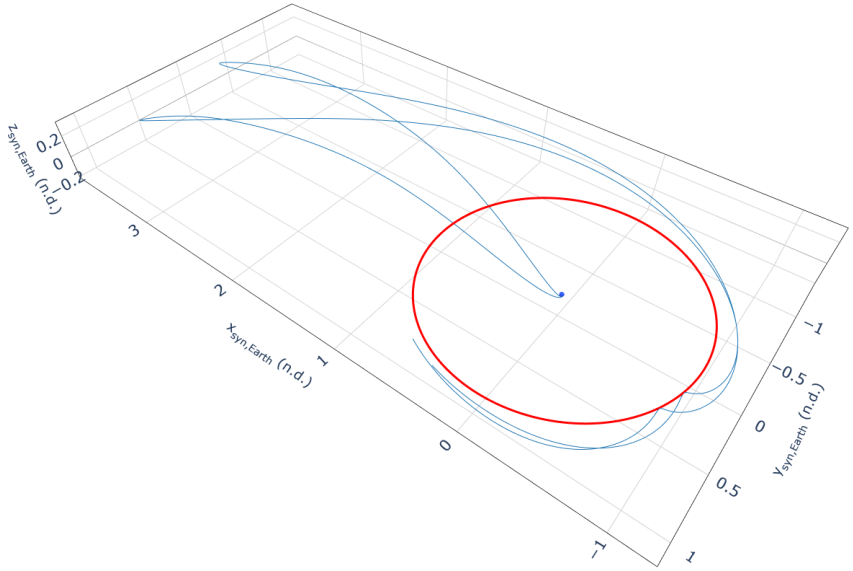


Figure 5: Example of Earth re-entry trajectories.

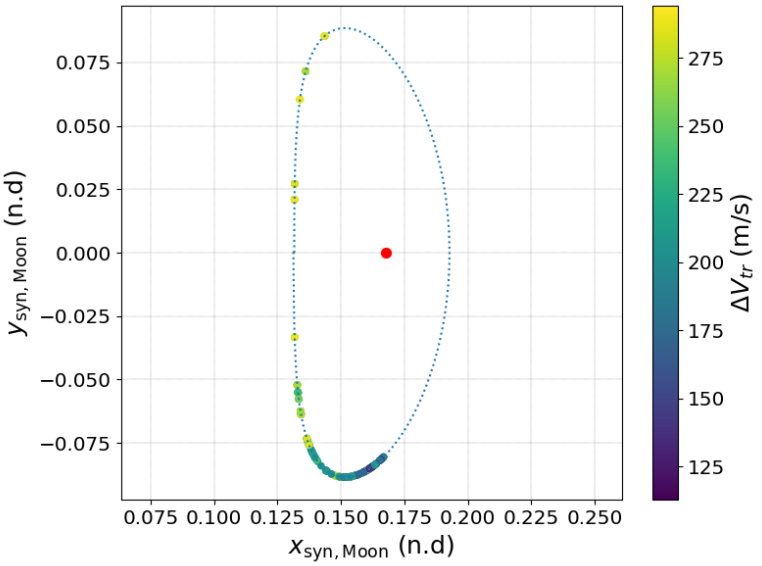


Figure 6: Transfer cost: fly-by and HOI cost.

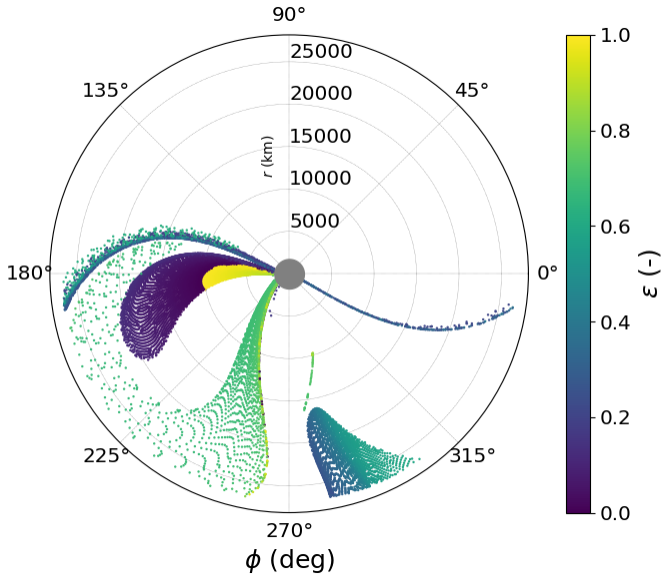
Mission Scenario 2: LEO to Halo Orbit Transfer Design

Earth to Halo orbit transfer design can be performed in different ways.[24,25] In this study an *indirect* insertion strategy is considered: Moon flyby transfers are modelled. With reference to Fig. 2, the selected Halo orbit has a lower out-of-plane component ($A_z = 6000\,\text{km}$) than the one considered for Mission Scenario 1. This simplifies the design of the transfer and enlarges the possible windows to be exploited. Such an orbit can be used as a parking orbit to easily transfer toward the other members of the Halo/NRHO family.

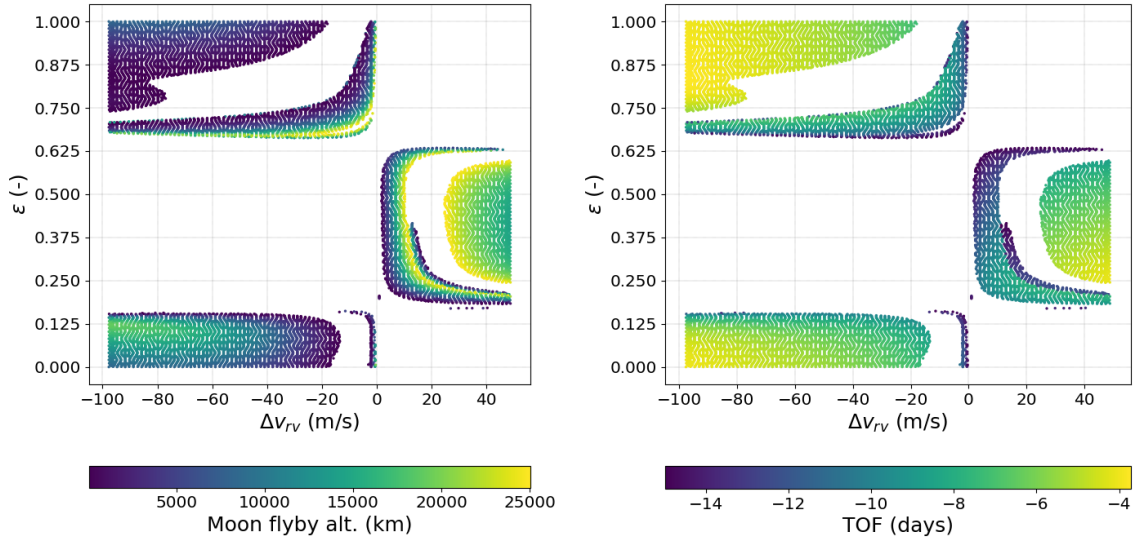
Earth-via-Moon Periapse Map The orbit insertion to the selected Halo orbit is performed via a Moon flyby. To model that, a manoeuvre is placed on the orbit in the Moon-centric velocity direction, defined by ε, with magnitude ΔV_{rv}. This manoeuvre is usually called Halo Orbit Insertion (HOI) in literature and is necessary to enter the Halo orbit from the transfer trajectory. The j-th trajectory belonging to the Moon periapse map can be then defined as:

$$\mathcal{Z}_j(\boldsymbol{\eta}) = (\varepsilon,\, \Delta V_{\text{rv}}) \quad \text{subject to} \quad \zeta = \begin{cases} t \in (0, -T_f) \\ \ddot{r}|_{\text{apse}} \geq 0 \\ h_{\text{Moon}}|_{\text{apse}} \leq h_{\text{threshold}} \end{cases} \tag{30}$$

In this case the trajectories are propagated *backward in time* for a maximum TOF of `20 days` and **refinement-step function** is structured in such a way the propagation is stopped whenever the trajectory arrives on a Moon periapse, impact the Moon, the Earth or escape the Earth-Moon system, by assigning an *integer* value to each event: 1 for Moon periapses *within a given maximum altitude*, -1 for Moon impacts, -2 for Earth impacts, 0 for confined trajectories and -3 for escape trajectories. Regarding the **binary relation**, a *equality* ralation condition is chosen as $\mathcal{B}(\mathcal{J}) \stackrel{\text{def}}{=}$ `J == 1`. A **termination condition** on the mean area threshold is considered and imposed as $r_{\mathcal{T}} =$ `1e-4`. An initial mesh of `40x40` points with $\varepsilon \in [0, 1]$ and $\Delta V \in [-100\,\text{m/s},\, 50\,\text{m/s}]$ is considered. The mesh refinement algorithm is run until the termination is achieved, in `5` refinement steps and `124`



(a) Periapse map in the *synodic* frame, with polar axis centred at the Moon. Here ϕ taken positive counter-clockwise from the x axis.



(b) Flyby altitude map.

(c) TOF map.

Figure 7: Moon periapse map results.

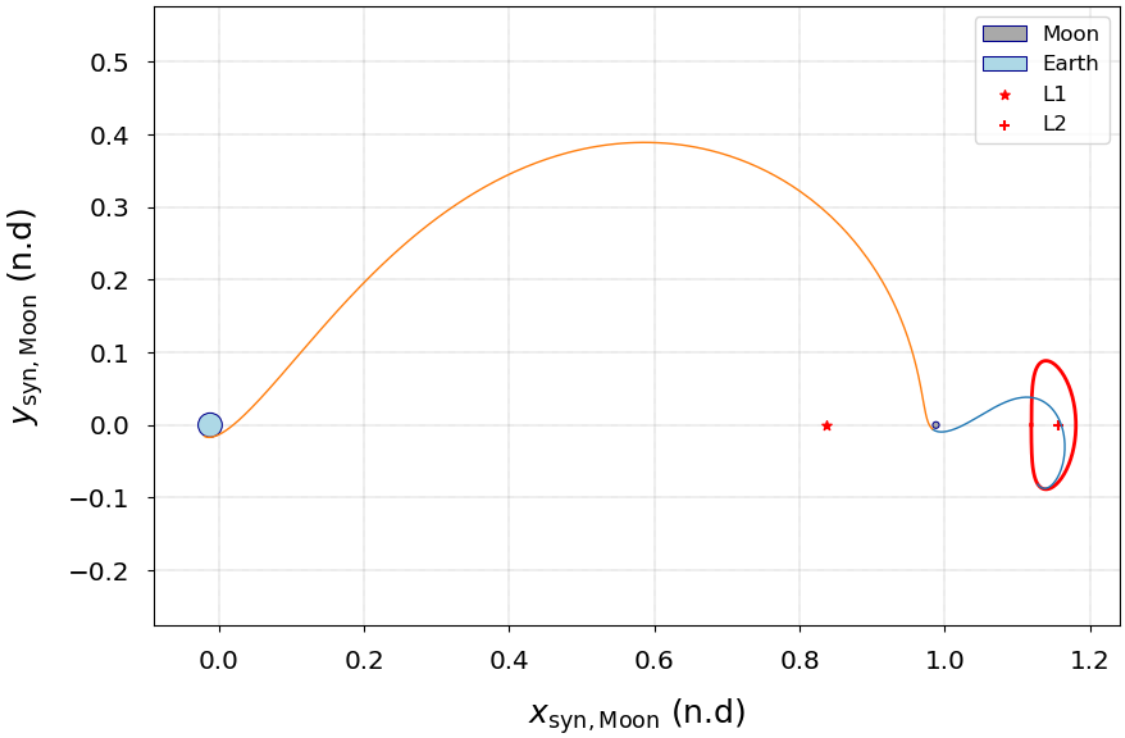


Figure 8: Example of short-period $500\,\mathrm{km}$-LEO to Halo transfer.

`seconds`.

In Fig. 7 the Moon periapse map are presented. Note that, again, the proposed adaptive mesh algorithm is extremely beneficial, since to obtain the same "accuracy" with a brute force approach, a simulation of `32.56 min` would be needed. From the Moon periapses, then, a second manoeuvre is applied: it is perfomed in the Moon-centric velocity direction with a magnitude in the range $-100\,\mathrm{m/s}$ to $250\,\mathrm{m/s}$. This allows the trajectory to reach the LEO region. In particular, in Fig. 6 the sum of the Moon flyby cost and the HOI cost is shown for the points that reaches an Earth perigee lower than $1000\,\mathrm{km}$ altitude: those results are compatible with the one present in literature,[24] which are in the order of magnitude of $400\,\mathrm{m/s}$. Therefore, the exploitation of a low-altitude lunar flyby, instead of the manifold, which is usually used in the design of such kind of transfers, reduces the *insertion* cost in the range $130\,\mathrm{m/s}$ to $300\,\mathrm{m/s}$.

It can be shown that different families of transfers exists. For the sake of clarity an example of *short-period* transfer from a $500\,\mathrm{km}$ LEO is numerically corrected by means of a multiple shooting algorithm and presented in Fig. 8. The transfer presented is a ΔV optimal solution between the short-period family, with an overall ΔV of $3.156\,\mathrm{km/s}$. Lower ΔV solutions exists in the long-period family,[24] but are not presented in this study.

CONCLUSIONS & FUTURE WORKS

In this paper a novel mesh refinement algorithm based on Delaunay triangulation and a customizable refinement heuristics is presented. Its capabilities are explored by means of two case studies: a Libration Point Orbit disposal and an Earth to EML2 transfer design. In both cases the proposed

method is capable to reduce substantially the computational time, enabling the possibility to explore the state space in a much more efficient way. The algorithm is shown to be extendable to any arbitrary problem dimension, although the computational time increases exponentially with the number of dimensions. The proposed method, although being similar to other methods present in the literature,[9] introduces an innovative approach to the refinement heuristics handling as well as proof to be largely advantageous for different astrodynamics applications.

Future works includes the deployment of the method on GPUs or distributed architectures, to boost further the refinement algorithm performances as well as to give the possibility to explore even more efficiently multibody astrodynamical problems.

REFERENCES

[1] X. Tricoche, W. Schlei, and K. C. Howell, "Extraction and visualization of Poincaré map topology for spacecraft trajectory design," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 27, No. 2, 2020, pp. 765–774.

[2] K. C. Howell, D. C. Davis, and A. F. Haapala, "Application of periapse maps for the design of trajectories near the smaller primary in multi-body regimes," *Mathematical Problems in Engineering*, Vol. 2012, 2012.

[3] E. S. Gawlik, J. E. Marsden, P. C. Du Toit, and S. Campagnola, "Lagrangian coherent structures in the planar elliptic restricted three-body problem," *Celestial mechanics and dynamical astronomy*, Vol. 103, No. 3, 2009, pp. 227–249.

[4] C. R. Short and K. C. Howell, "Lagrangian coherent structures in various map representations for application to multi-body gravitational regimes," *Acta Astronautica*, Vol. 94, No. 2, 2014, pp. 592–607.

[5] J.-C. Jouhaud, M. Montagnac, and L. Tourrette, "A multigrid adaptive mesh refinement strategy for 3D aerodynamic design," *International journal for numerical methods in fluids*, Vol. 47, No. 5, 2005, pp. 367–385.

[6] T. Y. Hoshina, I. F. Menezes, and A. Pereira, "A simple adaptive mesh refinement scheme for topology optimization using polygonal meshes," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 40, No. 7, 2018, pp. 1–17.

[7] C. T. Traxler, "An algorithm for adaptive mesh refinement inn dimensions," *Computing*, Vol. 59, No. 2, 1997, pp. 115–137.

[8] C. Brummel-Smith, G. Bryan, I. Butsky, L. Corlies, A. Emerick, J. Forbes, Y. Fujimoto, N. J. Goldbaum, P. Grete, C. B. Hummels, *et al.*, "ENZO: An adaptive mesh refinement code for astrophysics (version 2.6)," *Journal of Open Source Software*, Vol. 4, No. 42, 2019, p. 1636.

[9] W. Huang and D. K. Tafti, "A parallel adaptive mesh refinement algorithm for solving nonlinear dynamical systems," *The International Journal of High Performance Computing Applications*, Vol. 18, No. 2, 2004, pp. 171–181.

[10] F. Sadlo and R. Peikert, "Efficient visualization of Lagrangian coherent structures by filtered AMR ridge extraction," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, No. 6, 2007, pp. 1456–1463.

[11] V. Szebehely, *Theory of Orbits: The Restricted Problem of Three Bodies*. Academic Press, New York and London, 1967.

[12] D. A. Dei Tos and F. Topputo, "On the advantages of exploiting the hierarchical structure of astrodynamical models," *Acta Astronautica*, Vol. 136, 2017, pp. 236–247.

[13] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross, "Heteroclinic connections between periodic orbits and resonance transitions in celestial mechanics," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Vol. 10, No. 2, 2000, pp. 427–469.

[14] D. E. C. Davis, *Multi-body trajectory design strategies based on periapsis Poincaré maps*. PhD thesis, Purdue University, 2011.

[15] S. S. Barakat and X. Tricoche, "Adaptive refinement of the flow map using sparse samples," *IEEE transactions on visualization and computer graphics*, Vol. 19, No. 12, 2013, pp. 2753–2762.

[16] H. Hang, B. Yu, Y. Xiang, B. Zhang, and H. Liu, "An objective-adaptive refinement criterion based on modified ridge extraction method for finite-time Lyapunov exponent (FTLE) calculation," *Journal of Visualization*, Vol. 23, No. 1, 2020, pp. 81–95.

[17] L. P. Chew, "Constrained delaunay triangulations," *Algorithmica*, Vol. 4, No. 1, 1989, pp. 97–108.

[18] "QHull,"

[19] P. B. Pal, *A Physicist's Introduction to Algebraic Structures: Vector Spaces, Groups, Topological Spaces and More*. Cambridge University Press, 2019.

[20] P. Stein, "A note on the volume of a simplex," *The American Mathematical Monthly*, Vol. 73, No. 3, 1966, pp. 299–301.

[21] S. Gao, W. Zhou, L. Zhang, W. LIANG, D. LIU, and H. ZHANG, "Trajectory design and flight results for Chang'e 4-relay satellite," *Scientia Sinica Technologica*, Vol. 49, No. 2, 2019, pp. 156–165.

[22] L. Bucci and M. Lavagna, "Escape, Disposal and Re-Entry Trajectories from Lunar Non-Keplerian Orbits," *70th International Astronautical Congress (IAC 2019)*, 2019, pp. 1–8.

[23] L. Bucci, *Mission analysis and operational aspects for a lunar exploration architecture*. PhD thesis, Politecnico di Milano, 2020.

[24] D. P. Gordon, "Transfers to Earth-Moon L2 halo orbits using lunar proximity and invariant manifolds," *Purdue University, August*, 2008.

[25] D. Conte, G. He, D. B. Spencer, and R. G. Melton, "Trajectory Design for LEO to Lunar Halo Orbits Using Manifold Theory and Fireworks Optimization," *Advances in the Astronautical Sciences*, Vol. 167, 2018, pp. 1335–1353.