Applications of Clustering to Higher-Dimensional Poincaré Maps in Multi-Body Systems

Stefano Bonasera* and Natasha Bosanac†
University of Colorado Boulder, Boulder, CO 80304

Poincaré maps are invaluable for rapidly analyzing the complex solution space within multi-body dynamical systems. However, for spatial motion or even planar dynamics in a nonautonomous model that does not admit a constant of motion, the information contained on a Poincaré map is often higher-dimensional and, therefore, challenging to visualize. In this paper, clustering is used to group higher-dimensional crossings on a Poincaré map according to the geometry of the associated trajectories; this procedure is demonstrated for natural and low-thrust-enabled solutions in the circular restricted three-body problem. The value of this clustering approach in reducing the complexity of visualization and analysis is demonstrated within the context of trajectory construction for the Lunar IceCube mission.

I. Introduction

Poincaré maps are often leveraged in astrodynamics and celestial mechanics to rapidly analyze the complex solution space typical of multi-body dynamical systems. A Poincaré map displays the intersections of a set of trajectories with a surface of section, reducing continuous solutions to sequences of individual states. An appropriately constructed map also decreases the dimensionality of the problem while still retaining the characteristics of the solution space. In the context of multi-body gravitational environments, maps have been employed in a variety of scenarios: from studying Earth-to-Moon low-energy transfers to reveal the underlying dynamical structure of the system to uncovering the natural motion of Jovian comets [1, 2]. In dynamical systems such as the Circular Restricted Three-Body Problem (CR3BP), Poincaré maps have been used to facilitate the trajectory design process: arcs of interest are selected from the map and used during initial guess construction to enable rapid recovery of an end-to-end trajectory in a higher fidelity model. Recently, Poincaré maps have been used during the early stages of the trajectory design process for CubeSat and SmallSat missions, where propulsion systems and deployment conditions are severely constrained and initial guess construction is particularly challenging; one prominent example includes the Lunar IceCube mission, involving a CubeSat that is deployed from the upcoming Artemis-1 mission and must reach an eccentric and highly-inclined orbit near the Moon [3].

The benefit of using Poincaré maps to explore the solution space in a chaotic gravitational environment depends on the complexity of the dynamical model and the dimension of the information describing each crossing on the map. For planar motion in a nonautonomous model that does not admit a constant of motion or even spatial motion in an autonomous dynamical system, Poincaré map visualization does not produce a bijective or unique representation of a trajectory via a map crossing; thus, analysis by a human-in-the-loop may be challenging. Typically, in these scenarios, either a multivariate representation of the map crossings is employed or additional constraints are introduced. For instance, Haapala represents four-dimensional map crossings on a two-dimensional projection by leveraging glyphs [4]. Alternatively, Gómez et. al. introduce additional filters to further reduce the dimension of the map crossings associated with spatial motion in the CR3BP [5]. In the CR3BP, applying these techniques to Poincaré map visualization may enable further insight into the solution space via the emergence of patterns in the map crossings or interactive analysis by a human-in-the-loop. However, if the dataset asociated with a higher-dimensional representation of a Poincaré map is dense or associated with a nonautonomous dynamical model, such patterns may not emerge, necessitating alternative approaches to enable effective analysis. In this paper, clustering techniques are used to address the challenges associated with analyzing and visualizing higher-dimensional Poincaré maps. Specifically, clustering is used to group map crossings associated with trajectories that share similar characteristics or geometries.

Clustering techniques have been leveraged in a wide variety of disciplines to extract fundamental insights from large datasets; recently, they have been employed in the disciplines of astrodynamics and applied mathematics. For example, Hadjighasem, Karrasch, Teramoto, and Haller apply spectral clustering to locate coherent structures in several nonlinear

*Ph.D. Pre-Candidate, Ann and H.J. Smead Aerospace Engineering Sciences, 3775 Discovery Dr., stefano.bonasera@colorado.edu
†Assistant Professor, Ann and H.J. Smead Aerospace Engineering Sciences, 3775 Discovery Dr., natasha.bosanac@colorado.edu, Member AIAA

flow regimes [6]. In astrodynamics, Nakhjiri and Villac leverage clustering algorithm $k-$means to autonomously detect regions of stability in planar maps with a focus on the region near distant retrograde orbits.[7]. Additionally, Villac, Anderson, and Pini employ $k-$means clustering to organize periodic orbits near small bodies [8]. More recently, Bosanac has employed a hierarchical and density-based clustering algorithm, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), to group the trajectories associated with the crossings on a general periapsis map in the planar CR3BP according to geometry; the result is a set of fundamental solution geometries and a representation of the regions of existence of each type of solution [9, 10]. Bosanac demonstrates this approach for two similarity measures and exploits a density-based cluster validity index to merge clusters of similar solutions in scenarios where the computational effort associated with the clustering step must be reduced [10]. In addition, Smith and Bosanac leverage a variety of clustering algorithms to summarize a continuous family of periodic orbits in the Earth–Moon CR3BP via a finite set of representative solutions [11]. In each of these studies, clustering has successfully supported a data-driven process for extracting new insights into the structure of the solution space in a complex dynamical system.

This paper builds upon the previous work of Bosanac by extending and applying clustering to higher-dimensional maps of the solution space in multi-body gravitational environments: capturing either spatial trajectories in the CR3BP or in a low-thrust-enabled CR3BP that is nonautonomous. The goal of this clustering-based approach is to reduce the complexity of visualizing and analyzing a higher-dimensional map that reflects a complex solution space. To achieve this goal for maps constructed in each of the dynamical models of interest, continuous trajectories are first summarized via a finite set of parameters or geometric characteristics. The resulting dataset is input to the HDBSCAN algorithm following the approach presented by Bosanac [10]. This clustering algorithm is used to group map crossings by the properties of the associated trajectories – producing a representative set of solutions for each cluster as well as insight into the region of existence of each arc. To extend this clustering approach to higher-dimensional maps and reduce the computational complexity of clustering the large dataset associated with a more complex solution space, this analysis also exploits cluster validity measures [12]. Specifically, the large dataset is coarsely partitioned: for spatial motion in the CR3BP, partitions corresponds to several hyperplanes with distinct definitions; and in the low-thrust-enabled CR3BP, partitioning is performed via high-level geometric criteria. Once the dataset is coarsely partitioned, clustering is performed on each partitioned dataset. Then, inspired by the approach presented by Bosanac, a scalar cluster validity measure is used to merge the clusters containing similar solutions across distinct partitions, effectively connecting the partitions in a computationally-efficient manner. The resulting clusters, generated without a priori definition of analytical expressions to govern the grouping, facilitate rapid and informed analysis of the solution space. To demonstrate this clustering-based approach to Poincaré map visualization and analysis, the outlined strategy is explored in the context of the trajectory construction process for the Lunar IceCube mission [3]. In particular, this paper focuses on using clustering to analyze Poincaré maps constructed in two dynamical models that predominantly predict the motion of the spacecraft during two distinct segments of the Lunar IceCube trajectory: (1) spatial motion in the Sun–Earth CR3BP corresponding to the phasing and energy adjustment transfer segment and (2) low-thrust-enabled motion in the Earth–Moon CR3BP, to study the lunar approach phase [3]. In both of these applications, the data contained on a Poincaré map possesses a higher dimension than that of the map representation. Thus, clustering is used to discover groupings of solutions that aid the human analyst during the trajectory construction process.

II. Mission Application: Lunar IceCube

Although CubeSats offer a rapid and low-cost platform for targeted science missions, the trajectory design process is significantly influenced by their form factor and associated operational constraints. Since CubeSats often ride as secondary payloads, there are significant uncertainties associated with their highly-constrained deployment conditions. With limited maneuvering capabilities for trajectory adjustments, trajectory design for a CubeSat destined for locations well beyond low Earth orbit may be a challenging task. Furthermore, regular redesign may be required as the mission concept, hardware design or deployment conditions evolve. Together, these challenges necessitate the development of a systematic trajectory design process and further understanding of the solution space.

Lunar IceCube is an upcoming CubeSat mission that will study water on the lunar surface; development of the Lunar IceCube mission is led by Morehead State University and is supported by NASA Goddard Space Flight Center, Busek and Catholic University of America. The mission will leverage a 6U CubeSat that will ride as a secondary payload on the upcoming Artemis-1 mission. The CubeSat is expected to possess an initial wet mass of 14 kg and is equipped with a propulsion system that admits an estimated thrust level of approximately 0.9 mN with a specific impulse of 2500s [13]. To fulfill the mission requirements, LunarIce Cube will perform observations from a highly-inclined, elliptical lunar orbit with a semi-major axis of $a = 4287$ km, an eccentricity of $e = 0.5714$, an inclination of $i = 89$ deg and periapsis located over the equator [13]. Prior work by Bosanac, Bosanac, Cox, Howell and Folta and Folta, Bosanac, Cox and Howell has focused on developing a framework for designing a feasible trajectory for the Lunar IceCube spacecraft to reach this desired lunar orbit following deployment [3, 13, 14]. In their works, the authors divide the transfer trajectory into three segments: an Earth outbound leg, a phasing and energy adjustment segment and a final lunar capture phase [3]. During the phasing and energy adjustment segment, the gravitational pull of the Sun is leveraged to decrease the spacecraft energy and ballistically drive the spacecraft towards the Earth–Moon L_2 gateway. In the final approach phase, the low-thrust engine is employed to lower the spacecraft energy until lunar capture. In their work, Poincaré mapping is employed to gain insights into the solution space within each segment via low-fidelity, yet representative, dynamical models; this insight is then used to construct an initial guess for a trajectory in a low-fidelity model [3, 13, 14]. The Poincaré maps used to facilitate this trajectory design process are revisited in this current work via the application of clustering, with the goal of gaining additional insight into the distinct geometries of solutions in each segment of the trajectory as well as their regions of existence.

III. Background: Dynamical Models

The CR3BP is employed to describe the dynamics governing the motion of a spacecraft due to the point-mass gravitational influence of two primary bodies. In this model, two primaries, P_1 and P_2, are assumed to follow circular orbits about their mutual barycenter. The third body, representing the spacecraft, is assumed to possess a negligible mass with respect to the two primaries [15]. Three characteristic quantities are introduced to nondimensionalize mass, length and time quantities, respectively: m^*, equal to the sum of the masses of the primaries; l^*, equal to the constant distance between the two primaries; and t^*, to set the mean motion of the primaries to unity. This nondimensionalization scheme also enables definition of the parameter μ, which represents the ratio between the mass of the smaller primary and the system mass. In the Sun–Earth system, the mass parameter is $\mu \approx 3.00348064 \times 10^{-6}$, while in the Earth–Moon system, $\mu \approx 0.01215058$. Then, the nondimensional state of the spacecraft is expressed in an orthogonal reference frame, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, that rotates with the two primaries; this frame definition enables the construction of equations of motion that are autonomous. The $\hat{\boldsymbol{x}}$-axis of this rotating frame is directed from the larger to smaller primary, the $\hat{\boldsymbol{z}}$-axis is aligned with the total angular momentum of the system, while the $\hat{\boldsymbol{y}}$-axis completes the right-handed coordinate frame. The nondimensional state of the spacecraft is written in this frame relative to the system barycenter as $\boldsymbol{x} = [x,\ y,\ z,\ \dot{x},\ \dot{y},\ \dot{z}]^T$. The nondimensional equations of motion for a spacecraft in the CR3BP are expressed as:

$$\ddot{x} - 2\dot{y} = \frac{\partial U}{\partial x}, \qquad \ddot{y} + 2\dot{x} = \frac{\partial U}{\partial y}, \qquad \ddot{z} = \frac{\partial U}{\partial z} \tag{1}$$

with a pseudo-potential function defined as $U = (x^2 + y^2)/2 + (1 - \mu)/r_1 + \mu/r_2$, and the distances of the spacecraft from the two primaries as, respectively, $r_1 = \sqrt{(x + \mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x - 1 + \mu)^2 + y^2 + z^2}$ [15]. The CR3BP admits one integral of motion, the Jacobi constant, equal to $JC = 2U - \dot{x}^2 - \dot{y}^2 - \dot{z}^2$ [15]. At a single value of the Jacobi constant, a wide variety of fundamental solutions may exist in the CR3BP including: equilibrium points, labelled L_i for $i = [1, 5]$; periodic orbits; bounded quasi-periodic motion and chaos. These solutions are bounded by Zero Velocity Surfaces (ZVS), separating allowable and forbidden regions of motion for a specific value of the Jacobi constant.

For a spacecraft that is equipped with a low-thrust propulsion system, the equations of motion for the CR3BP are augmented by an additional acceleration term. Consider a spacecraft with a mass m and a propulsion system with a dimensional thrust, T, and a specific impulse, I_{sp}; both the thrust and specific impulse are assumed to be constant over time. At a single instant of time, the unit vector describing the thrust direction is written in the velocity-normal-conormal (VNC) frame, defined relative to a primary body to enable the use of heuristics during development of an initial control strategy [14]. This unit vector is then transformed to the rotating frame associated with the CR3BP where it is written as $\hat{\boldsymbol{u}} = [u_x,\ u_y,\ u_z]^T$. Using these definitions, the system of equations in Eq. 1 are then augmented to include the acceleration due to the propulsion system and the time rate of change of the spacecraft mass. The result is the following system of equations for low-thrust-enabled motion in the CR3BP:

$$\ddot{x} - 2\dot{y} = \frac{\partial U}{\partial x} + \frac{T_{lt} u_x}{m}, \qquad \ddot{y} + 2\dot{x} = \frac{\partial U}{\partial y} + \frac{T_{lt} u_y}{m}, \qquad \ddot{z} = \frac{\partial U}{\partial z} + \frac{T_{lt} u_z}{m}, \qquad \dot{m} = -\frac{T t^*}{I_{sp} g_0} \tag{2}$$

where T_{lt} is the thrust normalized only by time and length quantities such that $T_{lt} = T(t^*)^2/l^*$ and $g_0 = 9.81$ m/s^2 is the gravitational acceleration on Earth at sea level. Due to the inclusion of an additional acceleration associated with the low-thrust propulsion system, the solution space is fundamentally different than in the natural CR3BP. Furthermore, the complexity of analyzing the solution space has significantly increased as the thrust direction unit vector evolves.

IV. Background: Poincaré Maps

In dynamical systems theory, Poincaré maps reduce the complexity of analysing a large set of continuous trajectories. This goal is achieved by first defining a surface of section that is transverse to the flow. Useful definitions for a surface of section may include: hyperplanes, e.g. fixing a certain space or velocity coordinate; hyperspheres centered at one primary; and stroboscopic surfaces, capturing the flow at specific times [16]. Then, continuous trajectories are dimensionally reduced to a finite set of states, representing the locations where the continuous arcs pierce the hyperplane. Depending on the problem definition, the definition of the hyperplane and the map configuration, the map crossings associated with each trajectory may provide insight into the characteristics of the underlying dynamics. In fact, the collection of map crossings associated with a large set of trajectories may admit specific patterns that may indicate the existence of specific types of fundamental solutions; the absence of such patterns may also reveal valuable insight in some circumstances. However, in dense, higher-dimensional maps associated with autonomous dynamical models, such patterns may be difficult for a human analyst to detect. Furthermore, in nonautonomous dynamical models, such patterns do not typically exist. Thus, the complexity of analyzing the solution space reflected on a Poincaré map depends on the properties of the dynamical model, the definition of the surface of section and the flow segments of interest.

To supply an overview of the Poincaré map construction process, consider a perigee map for prograde, planar motion in the Sun–Earth CR3BP at a single value of the Jacobi constant; this is the same map presented in Bosanac [10]. First, a periapsis surface of section is defined as:

$$(x - 1 + \mu)\,\dot{x} + y\dot{y} + z\dot{z} = 0 \quad \cup \quad (x - 1 + \mu)\,\ddot{x} + y\ddot{y} + z\ddot{z} + \dot{x} + \dot{y} + \dot{z} > 0 \tag{3}$$

relative to the smaller primary in the Sun–Earth CR3BP, i.e., the Earth [17]. In this example, initial conditions for generating planar trajectories of interest are seeded directly from this surface of section at a single Jacobi constant; for this example, this Jacobi constant is set equal to $JC = 3.00088$. These initial perigees possess states of the form $\boldsymbol{x_{IC}} = [x,\, y,\, z = 0,\, \dot{x},\, \dot{y},\, \dot{z} = 0]^T$. Position coordinates, x and y, for the initial conditions are selected to lie within the Zero Velocity Curves (ZVCs), i.e., the intersection between the ZVS and the plane of the primaries, at the fixed value of the Jacobi constant. In this example, 201 equally-spaced $x-$coordinates are seeded within the ZVCs and between the locations of L_1 and L_2, along with 201 equally spaced $y-$coordinates within the range $[-0.01, 0.01]$. For each combination of x and y, the speed at perigee is calculated from Eq. 2 as $v = \sqrt{2U - JC}$. If this relationship produces a real number for v, the velocity vector at the initial state is defined by multiplying the unit vector that satisfies Eq. 3 by the speed. This unit vector is directed such that the state is prograde relative to the Earth, producing an instantaneous angular momentum vector for the spacecraft relative to the Earth with a positive z-component. Each initial condition that satisfies the prograde periapsis condition is then propagated forward in time in the CR3BP. In this example, up to 20 successive intersections with the hyerplane are recorded. This propagation terminates early if the trajectory passes within a distance of $r_2 < 10^{-5}$ from the Earth or the spacecraft leaves the Earth vicinity through either the L_1 or L_2 gateways. The Poincaré map constructed with this approach is displayed in Fig. 1a). This map reflects the position coordinates $(x,\, y)$ of every intersection of the generated trajectories with the surface of section, plotted in black in Fig. 3. The locations of the Earth, L_1 and L_2 are labeled, with the ZVCs indicating the boundaries of the gray shaded forbidden region. Definition of the map configuration along with constraining the flow in the CR3BP to planar solutions at a fixed value of JC results in each map crossing uniquely representing each trajectory [10].

To demonstrate the complexity of higher-dimensional maps, consider a second Poincaré map constructed at the same Jacobi constant of $JC = 3.00088$, but for spatial motion in the CR3BP. This map is constructed using a similar procedure as in the previous example. However, the set of initial conditions is discretized using 51 equally spaced $x-$coordinates within the locations of L_1 and L_2, 51 equally spaced $y-$coordinates in the range $[-0.01, 0.01]$, 51 equally spaced $z-$coordinates in the range $[-0.01, 0.01]$. Then, when solving for a unit vector that is aligned with the velocity vector, the perigee condition in Eq. 3 admits an additional degree of freedom. Thus, 51 equally spaced values of $\dot{z}$ are considered within the range $[-v, v]$. Each initial state is then propagated forward in time for 5 consecutive intersections with the hyperplane defined in Eq. 3, with the same early termination conditions as in the previous example. Each intersection of the propagated trajectories with the surface of section is then plotted in black in the three-dimensional configuration space in Fig. 1b). This Poincaré map reflects prograde perigees for trajectories near the Earth in the Sun–Earth CR3BP at a Jacobi constant of $JC = 3.00088$. The locations of the Earth, L_1 and L_2 are labeled, with the ZVS displayed as a transparent blue surface; the forbidden region at the selected Jacobi constant lies outside this ZVS. Analysis of this figure reveals that the complexity of this map reflecting spatial motion in the CR3BP has significantly increased. In fact, this three-dimensional projection does not supply a bijective representation of the mapping as each crossing represents four-dimensional data, thereby impeding a rapid and thorough investigation of the available design space by a human-in-the-loop. Furthermore, modifying the data displayed on this map via filtering

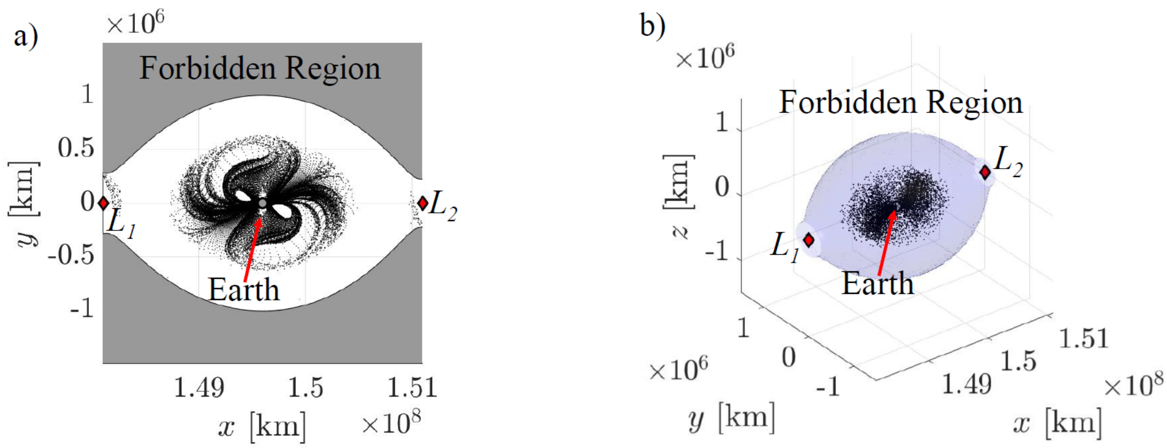

Fig. 1 Example Poincaré maps capturing prograde perigees at $JC = 3.00088$ **in the Sun–Earth CR3BP for a) planar and b) spatial trajectories.**

or higher-dimensional visualization schemes may require significant a priori insight into the structure of the solution space. In fact, constructing exact analytical conditions for further discretizing the data set may not be feasible. Thus, data-mining techniques, such as clustering, offer a potential approach for effectively exploring the solution space and reducing the complexity of analysis.

V. Background: Clustering

Clustering techniques enable an unsupervised grouping of the members of a dataset: data in the same cluster are considered as similar while data in separate clusters are considered dissimilar. This grouping is performed based on a specified representation of the data, labeled the feature vector, in a multi-dimensional space. Then, cluster assignments may be performed via a variety of algorithms, which tend to fall into the following classes: partition-based, hierarchical and density-based [18]. In this paper, the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm, developed by Campello, Moulavi and Sander, is leveraged for cluster assignment following the approach presented by Bosanac [9]. As Bosanac notes, this algorithm is particularly well-suited to the clustering of the crossings on a Poincaré map according to the geometry of the associated solutions since it: can accommodate clusters of arbitrary shape and density; can label data as noise; does not require a priori knowledge of the number of clusters; and can accommodate an unknown or nonconstant distance between data in a cluster [10]. In this section, an overview of the HDBSCAN algorithm is presented, followed by the definition of the feature vector and the similarity measures employed in this paper along with the approach for extracting a cluster representative. Then, density-based cluster validity indices, introduced by Moulavi et al., are discussed as a means for algorithm parameter selection and merging clusters of similar solutions [12].

A. HDBSCAN Overview

HDBSCAN is a density-based hierarchical clustering algorithm developed by Campello, Moulavi and Sander [9]. This algorithm takes as an input the dataset $[T] = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_N\}$, composed of N members that are described by M-dimensional feature vectors. Then, HDBSCAN groups data in sufficiently dense regions of a multi-dimensional space into clusters. Two input parameters govern this clustering algorithm: m_{pts} and m_{clSize}. The first input parameter, m_{pts}, enables calculation of the core distance, d_{core}, of the i-th member of the dataset, defined as the metric-distance of $\boldsymbol{t}_i$ from its nearest m_{pts}-neighbor, including $\boldsymbol{t}_i$, i.e., $d_{core}(\boldsymbol{t}_i) = KNN(\boldsymbol{t}_i, m_{pts})$. The second parameter, m_{clSize}, defines the minimum number of members of the dataset that may form a single cluster. Once the input parameters are set, HDBSCAN populates a distance matrix based on a scalar distance metric, e.g., Euclidean norm, L_∞-norm, or Hausdorff distance. The distance between the i-th and j-th data points reflects the mutual reachability distance (MRD), calculated as $d_{reach}(\boldsymbol{t}_i, \boldsymbol{t}_j) = \max\left\{d_{core}(\boldsymbol{t}_i), d_{core}(\boldsymbol{t}_j), d(\boldsymbol{t}_i, \boldsymbol{t}_j)\right\}$ where $d(\boldsymbol{t}_i, \boldsymbol{t}_j)$ is simply the distance between the two points. A Minimum Spanning Tree (MST) is then constructed by leveraging the computed MRDs as the weights of the edges between each pair of members of the dataset. A self-loop representing the core distance of each member is added at each node to generate an extended MST. HDBSCAN then condenses the MST to produce a dendrogram that supports cluster assignment: clusters are identified as those that both possess at least a minimum number of members and are considered sufficiently stable across the dendrogram. During the clustering process, each member of the dataset is either assigned to a cluster or considered noise [9]. As discussed by Campello, Moulavi and Sander, the HDBSCAN algorithm is $\sim O(MN^2)$ in time and $\sim O(MN)$ in memory storage, when the clustering is performed on an $N \times M$-dimensional dataset [9]. In this analysis, the HDBSCAN algorithm is accessed via the *hdbscan* Clustering Library in Python [19].

B. Feature Vector Representation

Each member of the dataset is described by a feature vector that reflects properties of the map crossing and associated trajectory. In this work, continuous arcs are summarized as a sequence of states. However, the challenge in applying this definition lies in selecting the appropriate states and parameters to balance fidelity of the representation with computational performance and avoiding the well-known curse of dimensionality [10]. Following the previous work of Bosanac, the geometry of a trajectory is summarized via a feature vector, $\boldsymbol{t}_i$, defined using a curve-based approximation. Each trajectory is identified by a sequence of N_a apses – periapses and/or apoapses – calculated with respect to one of the primaries [10]. This feature vector is constructed such that $\boldsymbol{t}_i = \left[\boldsymbol{s}_{i,0},\, \boldsymbol{s}_{i,1},\, \ldots, \boldsymbol{s}_{i,k},\, \ldots,\, \boldsymbol{s}_{i,N_a}\right]^T$, where $\boldsymbol{s}_{i,k}$ represents the k-th apse recorded along the trajectory associated with the i-th map crossing. Each apse, $\boldsymbol{s}_{i,k}$, is then represented by the time and state components at that apse such that:

$$\boldsymbol{s}_{i,k} = \left[\tau_{i,k}, x_{i,k},\, y_{i,k},\, z_{i,k},\, \dot{x}_{i,k},\, \dot{y}_{i,k},\, \dot{z}_{i,k}\right]^T \tag{4}$$

where $\tau_{i,k}$ is the time at which the k-th apse occurs normalized by the total propagation time along the trajectory and $\left(x_{i,k},\, y_{i,k},\, z_{i,k},\, \dot{x}_{i,k},\, \dot{y}_{i,k},\, \dot{z}_{i,k}\right)$ are the state components of the k-th apse in the rotating frame. If the trajectory terminates early and prior to reaching the k-th apse, the vector $\boldsymbol{s}_{i,k}$ is assigned a placeholder value: $\boldsymbol{s}_{i,k} = [0, \pm 10, 0, 0, 0, 0, 0]^T$ [10]. The positive sign is used for trajectories stopping prior to reaching an apoapsis, while the negative sign is used for a trajectory that terminates prior to reaching a periapsis. To mitigate the potential for ill-conditioning between the components of the feature vector, a normalization scheme is also employed such that each parameter is normalized to possess a value within the range $[-1, 1]$ across the entire data set.

C. Measures of Similarity

In this analysis, two distinct distance metrics are leveraged to assess similarity: the Euclidean l^2-norm and a modified Hausdorff distance. The Euclidean l^2-norm enables definition of a distance metric, $d_2(\cdot,\cdot)$, via an isochronous comparison between two trajectories that are represented as a sequence of states. This quantity is calculated as follows:

$$d_2(\boldsymbol{t}_i,\, \boldsymbol{t}_j) = ||\boldsymbol{t}_i - \boldsymbol{t}_j||_2 = \sqrt{\sum_{k=1}^{N_a} \left(\boldsymbol{s}_{i,k} - \boldsymbol{s}_{j,k}\right)^T \left(\boldsymbol{s}_{i,k} - \boldsymbol{s}_{j,k}\right)} \tag{5}$$

For a time-independent and geometry-based measure of similarity between two trajectories, a modified Hausdorff distance, $d_{mHD}(\cdot,\, \cdot)$, is useful. In this paper, this quantity is defined as:

$$d_{mHD}(\boldsymbol{t}_i,\, \boldsymbol{t}_j) = \min\left(d_{mHD,1}(\boldsymbol{t}_i,\, \boldsymbol{t}_j),\, d_{mHD,1}(\boldsymbol{t}_j,\, \boldsymbol{t}_i)\right) \tag{6}$$

where:

$$d_{mHD,1}(\boldsymbol{t}_i,\, \boldsymbol{t}_j) = \sum_{k=1}^{N_a} \left(\min_{l=1,\ldots N_a} ||\boldsymbol{s}_{i,k} - \boldsymbol{s}_{j,l}||_2\right) / N_a \tag{7}$$

Due to its form, the modified Hausdorff distance requires more computational time during computations or clustering than the l^2-norm [20]. An approach to mitigate the impact of using the Hasudorff distance on the computational time via prepartitioning the dataset and merging clusters across partitions is discussed in later sections.

D. Cluster Representative Definition

Following application of the HDBSCAN algorithm, the map crossings associated with each cluster are summarized by a single representative member and associated trajectory. Following the approach of Bosanac, the medoid of a cluster is used to define a representative data point [10]. A medoid, sometimes referred to as clustroid, is the element of a cluster that is most similar to the other members of the same cluster [21]. Mathematically, for cluster $C_j = \{\boldsymbol{t}_1^{(j)}, \boldsymbol{t}_2^{(j)}, \ldots, \boldsymbol{t}_{M_j}^{(j)}\}$, with cluster cardinality $|C_j| = M_j$, the medoid of the j-th cluster is defined as:

$$\boldsymbol{t}_{med}^{(j)} = \operatorname{argmin}_{\boldsymbol{t}_k^{(j)} \in C_j} \sum_{i=1,\, i \neq k}^{M_j} d\left(\boldsymbol{t}_i^{(j)}, \boldsymbol{t}_k^{(j)}\right) \tag{8}$$

where $d(\cdot, \cdot)$ corresponds to the selected distance metric. To ensure that identification of the medoid is performed in a computationally-efficient manner and is not significantly influenced by data at the boundary of the cluster, the concept of soft-clustering is employed [22]. Specifically, only members of a cluster that possess a probability of 1 for belonging to a cluster are used to calculate the medoid.

E. Cluster validation

Cluster validation is leveraged in this analysis both to select the parameters governing the clustering algorithm and to merge clusters of similar solutions across partitions of the dataset. For clustering algorithms, validation usually falls under three different types: external, internal or relative. Of most interest in this analysis, relative validation uses the internal results of the clustering algorithm for comparison across various values of the input parameters. For a Poincaré map capturing spatial trajectories in the CR3BP or solutions in a nonautonomous dynamical system, there is limited a priori knowledge of an appropriate division of the dataset formed by the map crossings. Thus, following the work of Bosanac, the relative Density-Based Clustering Validation ($DBCV$) index introduced by Moulavi et al. is used in this work to validate clustering results – both to select input parameters and to identify when clusters of similar solutions should be merged across partitions of the dataset [10, 12]. The $DBCV$ index is built upon the definition of the *all-points*-core distance; for the generic datapoint $\boldsymbol{t}_i$ in cluster C_j , the *all-points*-core distance is defined as:

$$d_{acore} = \left(\frac{\sum_{k=2}^{|C_j|} \left(\frac{1}{KNN(\boldsymbol{t}_i, k)} \right)^b}{|C_j| - 1} \right)^{-1/b} \tag{9}$$

where b represents an integer value, often equal to the dimension of the feature vector. For each member of a cluster, the *all-points* Mutual Reachability Distance (aMRD) is then defined as $d_{areach}(\boldsymbol{t}_i, \boldsymbol{t}_j) = \max\left\{d_{acore}(\boldsymbol{t}_i), d_{acore}(\boldsymbol{t}_j), d(\boldsymbol{t}_i, \boldsymbol{t}_j)\right\}$. Then, Moulavi et al. define two more quantities: the density sparsness of a cluster, $DSC(C_i)$ which uses the maximum internal weight of the MST based on the aMRD of cluster C_i and the density separation of a cluster pair, $DSPC(C_i, C_j)$, defined to reflect the minimum weight of the MST based on the aMRD of both clusters, C_i and C_j. The $DSC(C_i)$ index is essentially a measure of the internal density compactness of a cluster, whereas the $DSPC(C_i, C_j)$ index is an indication of the density separation between two clusters. The validity index of a cluster, $V_C(C_i)$, then incorporates both DSC and $DSPC$ to represent the quality of the obtained clustering result: a good cluster is compact with a large separation from the other clusters. Thus, a positive value of $V_C(C_i)$, indicates a compact cluster while a negative value of $V_C(C_i)$ corresponds to a loose cluster. When a dataset is grouped into l clusters, the $DBCV$ index is computed as:

$$DBCV = \sum_{i=1}^{l} \frac{|C_i|}{|T|} V_C(C_i) \tag{10}$$

From this definition, $-1 < DBCV < 1$, with positive values indicating a good clustering result. Since the number of noise points contributes to the cardinality of the dataset, i.e., $|T|$, a clustering result with a large percentage of noise points possesses a lower absolute value of the $DBCV$ index [12]. An approximation of this $DBCV$ index is leveraged in this analysis both for parameter selection and to improve computational speed during prepartioning and cluster merging; this approximation is obtained via the *hdbscan* library in Python and uses the MRD rather than the aMRD in computation. Although it is merely an approximation of the true index, the approximation, $DBCV_a$, represents a valid alternative for relative comparison between HDBSCAN runs on the same dataset [19].

VI. Application of Clustering to Poincaré Maps in the Spatial CR3BP

In this section, HDBSCAN is employed to cluster the crossings of a Poincaré map associated with spatial trajectories in the autonomous Sun–Earth CR3BP. A technique inspired by tomography is presented to reduce the computational effort associated with clustering a large dataset. This procedure is then explored in the context of the Lunar IceCube mission, with a focus on analyzing the solution space for the phasing and energy adjustment segment of the trajectory.

A. Algorithm overview

To efficiently cluster the crossings on a higher-dimensional Poincaré map by the geometries of the associated spatial trajectories, a partition-based approach that is inspired by tomography is presented. First, several distinct sets of initial conditions are defined and the trajectories associated with each of these sets form a partition of the dataset. Clustering is performed separately on each of these partitions. Then, the clusters associated with each partition are compared: if two clusters intersect in phase space, they are merged. This second step corresponds to the connection of cluster across partitions of the dataset. The technical approach for implementing this clustering procedure is summarized as follows:

1) Definition of several sets of initial conditions: several sets of initial conditions are defined in the available phase space as a means to guide the partitioning process. One approach to partitioning the initial conditions is to sample only states that occur at the intersection of the surface of section for defining the map and a set of additional hyperplanes. For spatial trajectories in the CR3BP, these additional hyperplanes may be defined as a set of mutually orthogonal hyperplanes passing through the location of the smaller primary, such as $z = 0$, $y = 0$ and $x = 1 - \mu$. Alternative examples include a set of parallel but distinct hyperplanes.
2) Seeding initial conditions in each partition: the initial conditions are seeded directly from the intersection of the surface of section used to define the map and each additional hyperplane, while also subject to any additional constraints. When studying perigees along trajectories that remain in the Earth vicinity in the spatial CR3BP, these initial conditions are seeded between the L_1 and L_2 gateways using the procedure outlined in Section IV and defined to correspond to prograde perigees at a fixed value of the Jacobi constant with $\dot{z} = 0$.
3) Generation of the partitioned dataset: for the i-th set of initial conditions, the associated trajectories are propagated in the Sun–Earth CR3BP until satisfying any of the following termination conditions: completing a total of N_a apses with respect to the Earth, i.e., completing $N_a/2$ subsequent perigees; passing within a distance of 10^{-5} from the Earth; or passing through either the L_1 or L_2 gateways. Each trajectory generated from the i-th set and performing at least two apses is stored in the i-th partition of the dataset, $[T_i]$. Each crossing of the two-sided apse map is described via a feature vector that reflects a sequence of perigees and apogees, as outlined in Section V.B.
4) Clustering on each partition of the dataset: the input parameters m_{pts} and m_{clSize} are selected to balance maximizing the $DBCV$ index, lowering the subset of the dataset identified as noise and avoiding either an excessively large or negligibly small number of clusters. Using a consistent set of input parameters, clustering is performed independently on each of the partitions of the dataset.
5) Merge clusters across partitioned datasets: the clusters associated with each partition of the dataset are used to identify a minimal set of unique clusters. Merging of two clusters of similar solutions is performed by locating intersections in the phase space. If a cluster from one partition does not intersect the cluster within another partition, it is considered a standalone cluster. Finally, noise points are merged into a single noise set.
6) Analysis of the results: representative solutions of the clusters associated with the complete dataset are generated. The medoid of a cluster is identified either directly from the data points in that cluster or, if the cluster contains more than 10^4 members, via subsampling of the cluster [10]. These cluster representatives are analyzed along with a visualization of the map crossings with each cluster colored uniquely.

This procedure is demonstrated in the context of a prograde periapsis map for spatial motion in the Sun–Earth CR3BP at a Jacobi constant of $JC = 3.00088$ with $\dot{z} = 0$. The results of the clustering procedure are validated through a comparison with a computationally-intensive clustering of the full dataset associated with the Poincaré map in Section VI.B. Then, this approach is applied to the phasing and energy adjustment segment of the Lunar IceCube trajectory in Section VI.C.

B. Technical Approach

To demonstrate the outlined approach for reducing the complexity of clustering the map crossings associated with spatial trajectories in the CR3BP, consider a prograde periapsis map in the Sun–Earth CR3BP at a Jacobi constant of $JC = 3.00088$. Three partitions of the dataset are generated via the intersections of the surface of section used to define the map with three additional hyperplanes; these hyperplanes are summarized in Table 1 along with the properties of each partition, the parameters input to HDBSCAN and the properties of the clusters generated for each partition. The first dataset corresponds to periapses that remain in the plane of the primaries with $\dot{z} = 0$. The second dataset includes initial periapses in the $y = 0$ plane with $\dot{z} = 0$ while the third dataset reflects initial conditions located with $x = 1 - \mu$ and $\dot{z} = 0$. Then, trajectories in each partition of the dataset are generated by integrating the associated initial conditions forward in time for up to six subsequent apses with respect to the Earth. These trajectories are stored in the partition $[T_i]$ using the feature vector definition that reflects a time-ordered sequences of perigees and apogees. Next, HDBSCAN is used along with the input parameters listed in Table 1 to perform clustering on each of the individual partitions of

Dataset No.	Constraints on Initial Perigees	$[m_{pts}, m_{clSize}]$	$\lvert T \rvert$	$N_{clusters}$	$DBCV_a$	Noise level %
1	$z = 0$	[50, 100]	31544	14	0.20177	6.21
2	$y = 0$	[50, 100]	26108	25	0.10868	3.99
3	$x = 1 - \mu$	[50, 100]	18639	8	0.18575	0.22

Table 1 Clustering parameters and results for each partition of the dataset used to generate a prograde periapsis map.

dataset. Then, the merging procedure, outlined in Section VI.A, is employed to merge connected clusters that exist across multiple partitions of the datasets. Following merging, the final clustering result is displayed in Fig. 2 with each individual cluster colored uniquely. The clusters in this figure reveal the region of existence of solutions with similar geometries as well as the variety of distinctly different geometries exhibited by trajectories across the entire data set. Of course, this representation is limited to the selected subsets of perigees used to define the initial condition sets. Furthermore, the recovered clusters are dependent upon the specific parameters input to HDBSCAN.

The fidelity of the prograde periapsis map constructed in the Sun–Earth CR3BP at a Jacobi constant of $JC = 3.00088$ with $\dot{z} = 0$ is increased by introducing several parallel hyperplanes, defined in configuration space, to create a larger number of partitions of the dataset to be used in the clustering process. In particular, additional sets of initial conditions are constrained to possess individual values of the y-coordinate in the range $y \in \{-4, -3.5, -3, \ldots, 4\} \times 10^{-3}$. Each of these additional datasets, reflecting the geometry of the trajectories associated with each set of initial conditions, is clustered individually. Then, clusters that exist across multiple partitions are merged following the procedure outlined in Section VI.A. The resulting increased fidelity map is displayed in Fig. 3 with a subset of map crossings displayed and colored according to their cluster assignment; only selected clusters are displayed to ensure clear visualization. Cluster representatives for the clusters that are labeled in Fig. 3 are displayed in Fig. 4 in the Sun–Earth rotating frame. In this figure, green circle locate the initial conditions, the Earth is identified by a gray circle and red diamonds correspond to the equilibrium points. The transparent blue surface corresponds to the ZVS. These representatives each exhibit

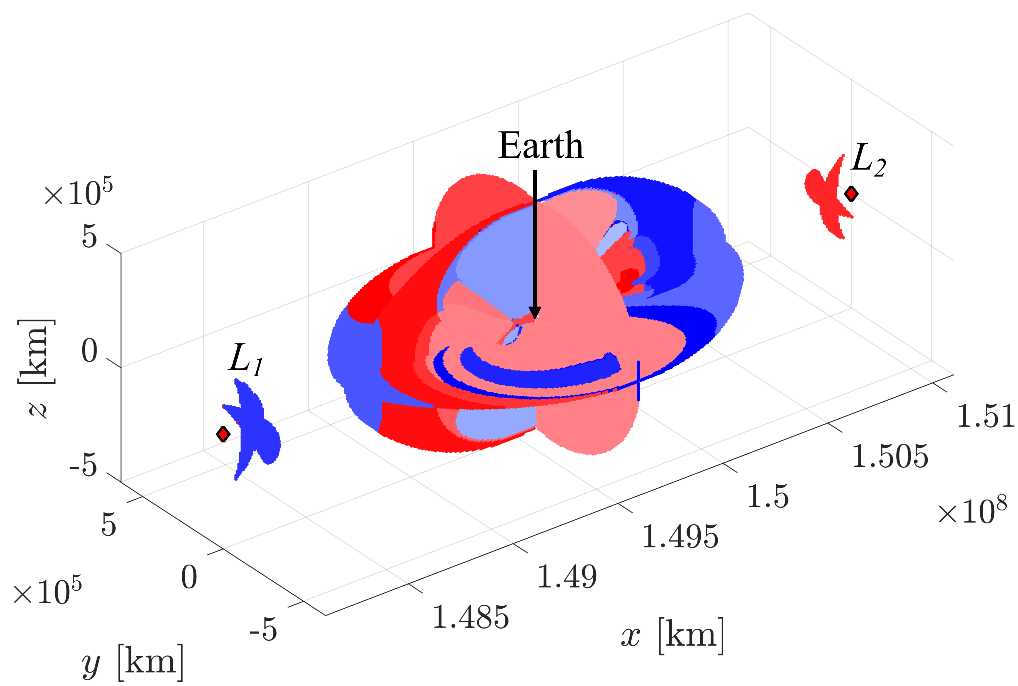

Fig. 2 Poincaré map reflecting prograde periapses in the Sun–Earth CR3BP at a Jacobi constant $JC = 3.00088$ and $\dot{z} = 0$ following the partitioning and cluster merging procedure.

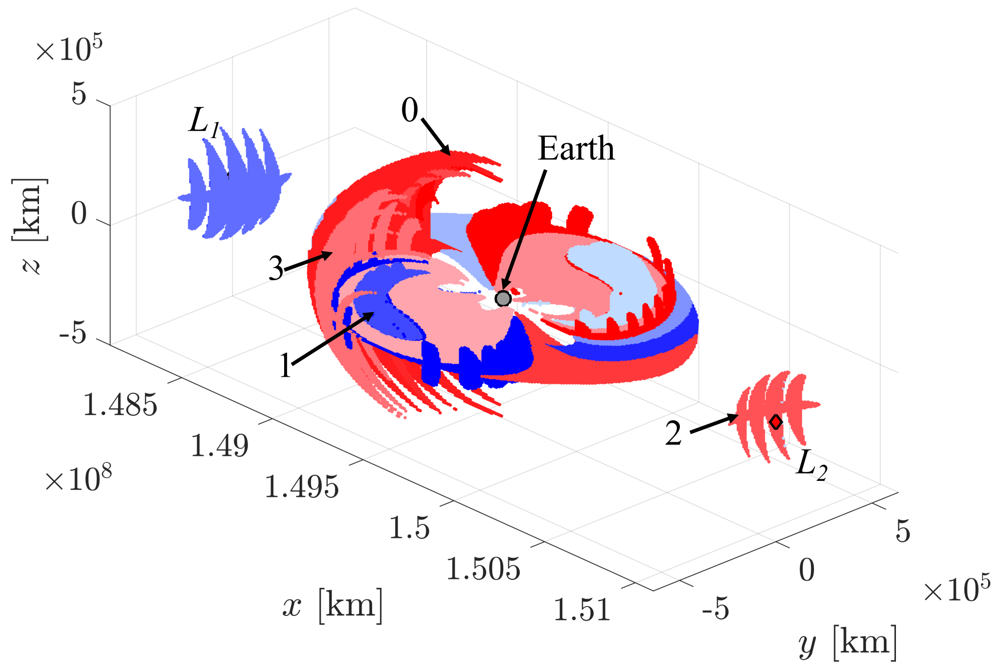

Fig. 3 Poincaré map reflecting prograde periapses in the Sun–Earth CR3BP at a Jacobi constant $JC = 3.00088$ and $\dot{z} = 0$ following the partitioning and cluster merging procedure, constructed using a large number of partitions. Four selected clusters are identified.

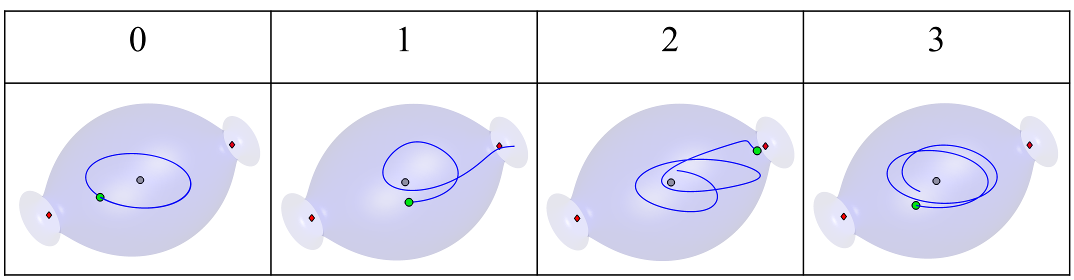

Fig. 4 Representatives of the clusters labelled on the map in Fig. 3, plotted in the Sun–Earth rotating frame.

fundamentally different geometries, illustrating the capability for the clustering approach to group the crossings on a map via the geometry of the associated trajectories.

To validate the results of clustering a large dataset via the described prepartitioning and merging process, a comparison to simultaneous clustering of the entire large data associated with all initial conditions is performed. The full initial condition set is defined by seeding 301 $x-$, 301 $y-$ and 301 $z-$coordinates in the neighborhood of the Earth and between the L_1 and L_2 gateways, with $\dot{z} = 0$ and the prograde periapsis condition. Then, a complete dataset of spatial trajectories is generated and clustered via HDBSCAN. A dataset of $|T| = 542446$ trajectories is produced and is organized into 13 clusters with a noise level of 0.2481% by setting $[m_{pts}, m_{clSize}] = [200, 500]$. The clustering result is displayed in Fig. 5, with only a subset of the initial conditions plotted for clarity and colored by their cluster assignment. The clusters identified in Fig. 3 are also labeled in Fig. 5. The overall structure of the clusters displayed in Fig. 5 is consistent with the results in Figs. 2 and 3, aside from the color differences due to the use of different coloring schemes. Furthermore, comparison of these two figures reveals that both approaches recover clusters with periapses that encompass similar regions of the configuration space. However, the two approaches may not result in exactly the same amount of total clusters, since the clustering process is sensitive to the input parameters selection and the properties of the dataset.

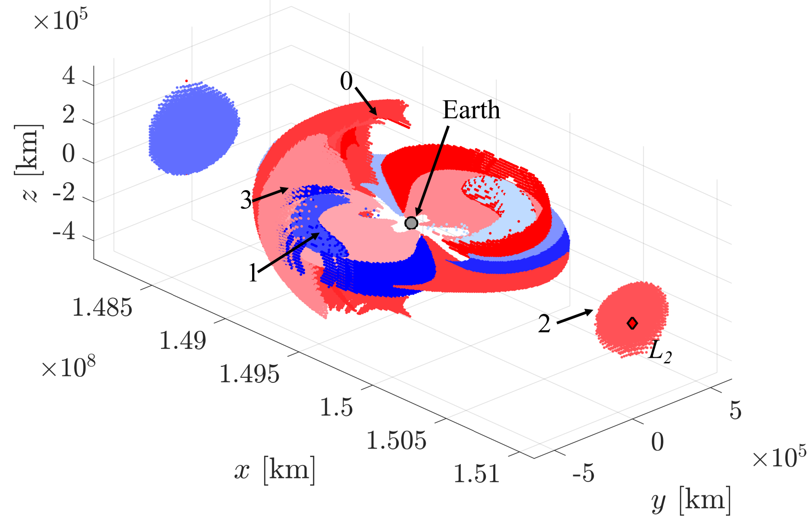

Fig. 5 Poincaré map reflecting prograde periapses in the Sun–Earth system at a Jacobi constant $JC = 3.00088$ and $\dot{z} = 0$ constructed by clustering the full dataset in a single step. The same four clusters identified in Fig. 3 are labeled.

C. Case Study: Lunar IceCube Phasing and Energy Adjustment Segment

The presented approach to efficiently clustering higher-dimensional map crossings via prepartitioning is applied to the phasing and energy adjustment segment of the Lunar IceCube trajectory to assess the region of existence of arcs used in the trajectory construction process. Consider a sample trajectory for Lunar IceCube developed and provided by David Folta at the NASA Goddard Space Flight Center (Private communication, David Folta, 2018). This trajectory is propagated in a high fidelity model of the Sun, Earth and Moon gravitational environment and plotted in the Sun–Earth rotating frame in Fig. 6 via a) a projection onto the (x, y)-plane and b) a projection onto the (x, z)-plane; the locations of apogees that occur along this solution are depicted as green dots and labeled. In this figure, blue arcs indicate natural

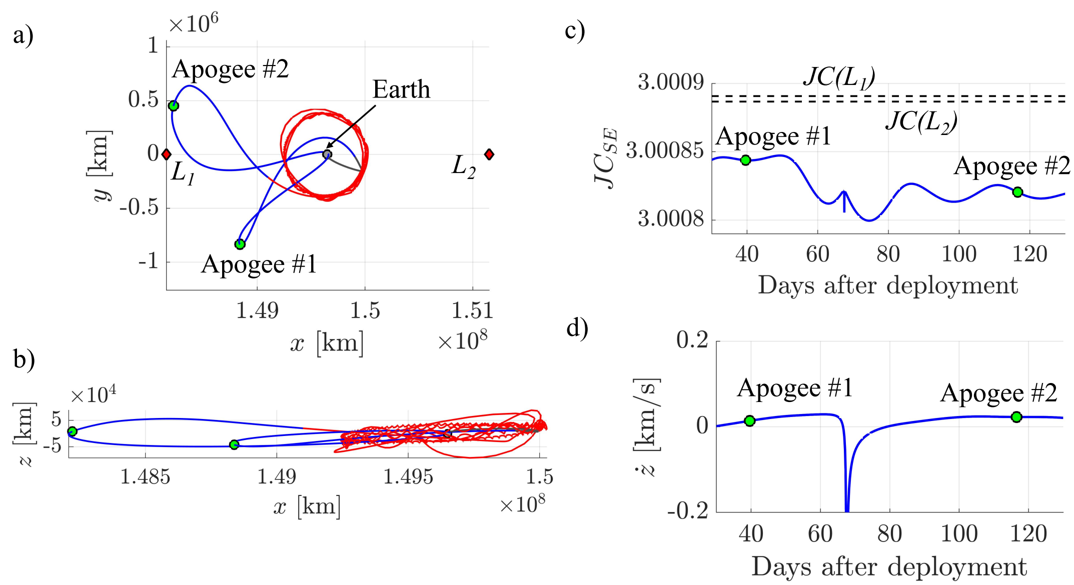

Fig. 6 Lunar IceCube trajectory, propagated in an ephemeris model: a) projected onto the (x, y)-plane and b) projected onto the (x, z)-plane. Time history of the c) Sun–Earth Jacobi constant and d) $\dot{z}$ component of velocity.

motion while red arcs correspond to the application of low-thrust. The phasing and energy adjustment segment begins at the first apogee and ends prior to the low-thrust-enabled approach into the Earth–Moon L_2 gateway. Since this segment corresponds to purely ballistic motion, the Sun–Earth CR3BP is useful in predicting the fundamental dynamical structures governing the solution space. In Fig. 6c) the Jacobi constant, JC, in the Sun–Earth system is evaluated along the trajectory and the values of the JC at the L_1 and L_2 equilibrium points are marked as black dashed lines. Since the reference trajectory is generated in a high-fidelity dynamical model, the Jacobi constant fluctuates. However, the value of the Jacobi constant generally indicates that the Sun–Earth L_2 gateway is open, allowing the spacecraft to potentially exit the vicinity of the Earth. The first apogee that occurs along this reference trajectory is used to analyze the geometry of arcs in the phasing and energy adjustment segment for use in the trajectory construction process.

In this case study, a higher-dimensional Poincaré map corresponding to retrograde apoapses producing trajectories that remain near the Earth vicinity for two revolutions around the Earth are analyzed via clustering. The goal of this analysis is to gain insight into the distinct geometries of arcs that exist in the Sun–Earth CR3BP and the associated regions of existence; both are useful insights during the trajectory construction process. To generate the Poincaré map, three hyperplanes are defined to create three sets of initial conditions at the Jacobi constant level associated with the first apogee, i.e., $JC = 3.000843740$. These three initial condition sets are generated as retrograde apoapses in the Sun–Earth CR3BP that also intersect the hyperplanes listed in Table 2; this table also displays properties of the dataset and clustering results. The definition of these hyperplanes corresponds to three mutually orthogonal hyperplanes that pass through the projection of the first apogee onto the plane of the primaries along the reference trajectory depicted in Fig. 6. The constraint of $\dot{z} = 0$ is applied to each initial condition due to the low out-of-plane component of the velocity of Lunar IceCube trajectory throughout this trajectory segment, as depicted in Fig. 6d). The three partitioned datasets are independently clustered and clusters existing across multiple partitions are merged. The final clustering result is depicted in Fig. 7 with each apogee colored according to the cluster assignment. Then, the first apogee that occurs along the Lunar IceCube trajectory is indicated by a dark black dot. A zoomed-in view appears in Fig. 8a). In this plot, three map crossings are highlighted by black dots and the associated trajectories are displayed via a planar projection in the rotating frame in Fig. 8b) with the same color as the associated cluster. Recall that the trajectory of Lunar IceCube is designed to reach the Moon after the second apogee. The pink trajectory, which possesses an initial condition that is located closest to the first apogee along the Lunar IceCube reference trajectory, intersects a circular approximation of the lunar orbit at an angle of approximately 45 degrees. However, the sample arcs from the other two clusters possess distinctly different geometries as they revolve around the Earth in a different direction and intersect an approximation of the lunar orbit after the second apogee at angle of nearly 90 degrees. In addition, these solutions both exhibit a low perigee. Furthermore, the specific geometry of the pink colored cluster where the first apogee of the Lunar IceCube trajectory is located offers useful insight. Solutions with a similar geometry exist throughout the configuration space: in the plane of the primaries, the region of existence of this solution is sensitive to the distance from the Earth; however, out of the plane of the primaries, there are a wide variety of similar solutions if a sufficiently high z-component may be achieved at the first apogee. Although the Lunar IceCube trajectory presented in Fig. 6 is generated in a high-fidelity dynamical model, the clustering result obtained in the low-fidelity Sun–Earth CR3BP still offers a useful prediction of the characteristics of the solution space.

Dataset No.	Constraints on Initial Apogees	$[m_{pts}, m_{clSize}]$	$\lvert T \rvert$	$N_{clusters}$	$DBCV_a$	Noise level %
1	$z = -0.00029$	[50, 100]	73538	17	0.022	0.6432
2	$y = -0.00557$	[50, 100]	24231	5	0.4763	0.0083
3	$x = 0.99458$	[50, 100]	18639	3	0.8944	0

Table 2 Clustering parameters and results for each partition of the dataset.

VII. Application of Clustering to Poincaré Maps in the Low-Thrust-Enabled CR3BP

The presented clustering approach is applied to a Poincaré map analysis of low-thrust-enabled lunar orbit insertion arcs used to design the last segment of the Lunar IceCube trajectory. Recall that for this segment, the goal during trajectory construction is to identify a trajectory that passes through the Earth–Moon L_2 gateway in forward time and, through the application of a low-thrust engine, captures into a highly-inclined and eccentric lunar orbit with perilune located over the equator. As Bosanac and Folta, Bosanac, Cox and Howell note, solutions that achieve this goal may be identified by discretizing feasible lunar orbits, integrating these boundary conditions backwards in time using a

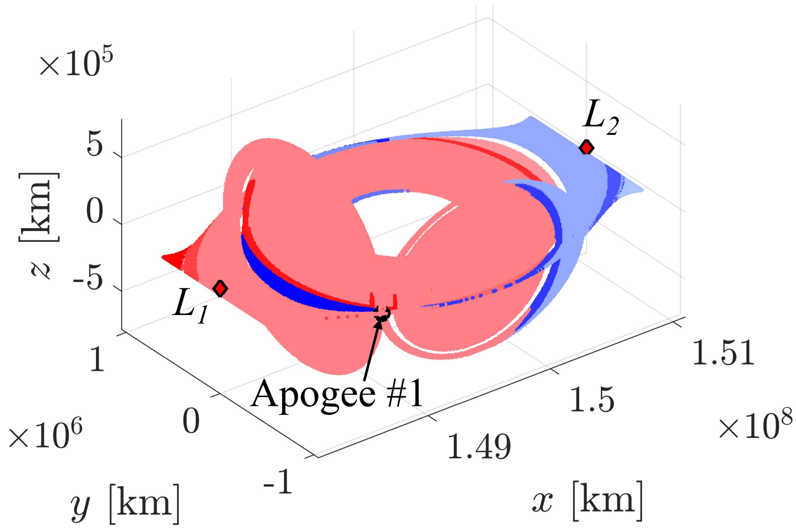

Fig. 7 Retrograde apoapsis map in the Sun–Earth CR3BP at $JC = 3.000843740$ with $\dot{z} = 0$. The first apogee along the reference Lunar IceCube trajectory is located by a black dot.

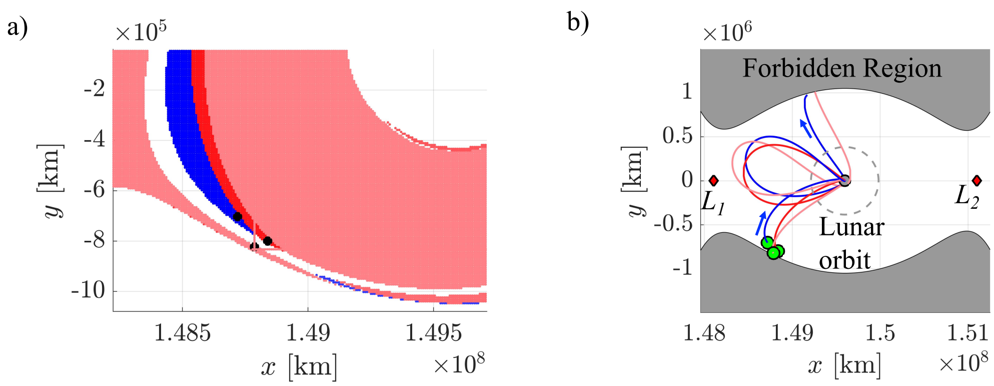

Fig. 8 a) Zoomed-in view of the retrograde apoapsis map in Fig. 7 and b) three sample trajectories propagated in the Sun–Earth CR3BP.

combination of low-thrust-enabled and natural arcs, and analyzing only solutions that pass through the Earth–Moon L_2 gateway [13, 14]. Visualization and analysis may then be performed via Poincaré maps constructed using a hyperplane located at the x-coordinate of the Earth–Moon L_2 equilibrium point. Each crossing on this map is multi-dimensional and may not be represented uniquely on a two-dimensional or three-dimensional projection. Thus, the presented clustering approach is used to group the crossings on this Poincaré map via the geometry of the associated solutions. However, the prepartioning and merging process must be modified to accommodate the added complexity of the solution space.

A. Algorithm overview

To cluster the crossings on a higher-dimensional Poincaré map via the geometry of the low-thrust-enabled trajectories that intersect a hyperplane at Earth–Moon L_2 in backwards time, the following partition-based approach is employed:

1) Definition of the full set of initial conditions: the constrained initial conditions for generating low-thrust trajectories in backwards time are defined using the approach presented by Bosanac and Folta, Bosanac, Cox and Howell [13, 14]. First, the semi-major axis, inclination, eccentricity and argument of periapsis of target lunar orbits are constrained to the values presented in Section II. Then, the right ascension of the ascending node, Ω, is set to 85 distinct values within the range $[0, 2\pi)$ rad and the true anomaly, f, is set to 85 distinct values within the range $[0, 2\pi)$ rad. These orbital elements are used to define a full set of initial conditions for states in a Moon

inertial frame along a feasible lunar science orbit. Then, the spacecraft wet mass at the end of the lunar approach segment or, equivalently, the initial state for integration backwards in time, is set equal to 13.5 kg [3].

2) Generation of the full data set: the initial conditions or, equivalently, final states at the end of the lunar approach segment, are converted from a Moon inertial frame to the Earth–Moon rotating frame. Then, each state vector is propagated backwards in time using Eq. 2 with the thrust unit vector aligned with the anti-velocity direction; in backwards time, this thrust direction corresponds to an efficient increase in energy to open the L_2 gateway. Following the approach presented by Bosanac and Folta, Bosanac, Cox and Howell, the low-thrust engine is activated until the value of the Jacobi constant is equal to a specified value that opens the L_2 gateway [13, 14]. Suitable values for this stopping condition for propagation in the low-thrust-enabled CR3BP are defined using 15 distinct values of the Jacobi constant between the values of the JC at the Earth–Moon L_2 and L_3 equilibrium points. Once a specified value of the Jacobi constant is reached, the low-thrust engine is deactivated and the trajectory is naturally propagated backwards in time in the Earth–Moon CR3BP. This integration continues until the trajectory pierces a hyperplane defined at the x-coordinate of the Earth–Moon L_2 equilibrium point. Only trajectories that reach this hyperplane within 250 days and without passing within a nondimensional distance of 10^{-5} from the Moon or through the L_1 gateway are stored.
3) Partitioning the dataset: due to the large number of solutions that must be generated to sufficiently reflect the characteristics of the complex set of low-thrust-enabled lunar approach arcs, the full dataset is partitioned. The stored trajectories are divided into three partitions based on the number of apses that occur while the Earth's gravity significantly influences the path of the spacecraft; in general, this is observed to occur after the spacecraft completes 325 apses relative to the Moon in backwards time along the low-thrust segment of the generated trajectories. Thus, the dataset is partitioned based on the number of lunar apses after the spacecraft completes 325 apses in backwards time from the desired lunar science orbit until piercing the specified hyperplane. Three partitions are constructed to correspond to trajectories that complete: 1) less than 21 apses; 2) between 21 and 35 apses; and 3) between 36 and 65 apses.
4) Summarize each trajectory via a feature vector: each of the stored trajectories is summarized via a feature vector reflecting several apolunes that occur immediately after the spacecraft passes through the L_2 gateway in forward time – or, equivalently, immediately before the spacecraft pierces the specified hyperplane in backwards time.
5) Initial clustering on each partitioned dataset: m_{pts} and m_{clSize} are selected for each partition to balance maximizing the $DBCV$ index, lowering the subset of the dataset identified as noise and producing a reasonable number of clusters. Using the selected input parameters and defining the distance metric via the l^2-norm, clustering is performed independently on each of the partitions of the dataset.
6) Merging clusters within each partition: using the l^2-norm to define similarity between solutions may separate geometrically-similar trajectories that complete a slightly different number of apses; such a result is typical for isochronous comparisons. Thus, clusters within each partition are merged in this step if the associated trajectories are geometrically similar. This merging is performed using the cluster validity index, $V_C(C_i)$, evaluated with the modified Hausdorff distance used to defined similarity. Specifically, pairs of clusters within each partition are compared. Then, if $V_C(C_i) < 0$ for both clusters, the clusters are merged within the partition.
7) Merging clusters across partitions: clusters that occur at the boundaries of the partitions are analyzed to determine whether the associated trajectories exist across multiple partitions. This analysis is performed by comparing pairs of clusters at the boundary of neighboring partitions using the cluster validity index, $V_C(C_i)$, evaluated with the modified Hausdorff distance used to define similarity. If $V_C(C_i) < 0$ for both clusters, the clusters are merged across two partitions. This merging process produces a reduced set of clusters that span the full dataset.
8) Analysis of the results: representative solutions of the clusters associated with the complete dataset are generated. The medoid of a cluster is identified either directly from the data points in that cluster or via subsampling of the cluster. These cluster representatives are analyzed along with the clusters.

B. Case Study: Lunar IceCube Lunar Orbit Insertion Segment

The algorithm described in Section VII.A is applied to the analysis of candidate lunar orbit insertion arcs in the final phase of the trajectory for the Lunar IceCube mission. Arcs that insert into a feasible lunar orbit after passing through the Earth–Moon L_2 gateway are generated following the procedure outlined in Step 1 of the presented algorithm. An example of one of these trajectories is displayed in Fig. 9a) in the Earth–Moon rotating frame. In this figure, natural arcs are colored blue while low-thrust-enabled segments are plotted in red; arrows indicate direction of motion in forward time and black circles locate several apolunes. The hyperplane defined at the x-coordinate associated with the

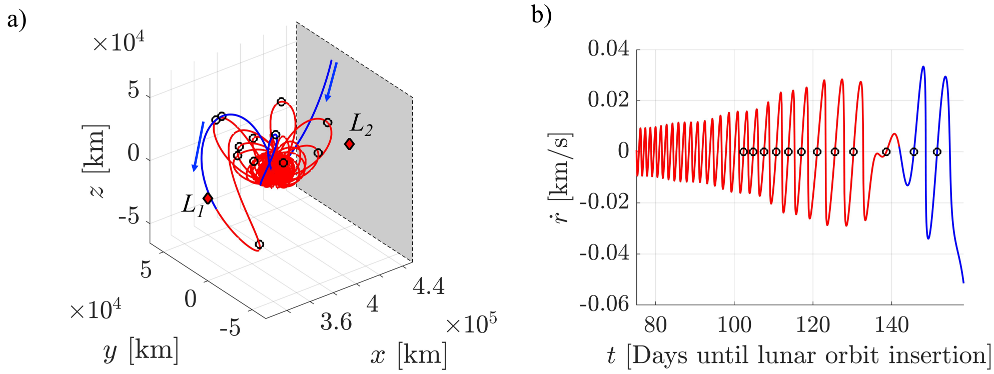

Fig. 9 Example trajectory for lunar orbit approach phase: a) in Earth–Moon rotating frame with the hyperplane in gray and b) associated time history of $\dot{r}$ with respect to the Moon. Apolunes located via black circles.

Earth–Moon L_2 equilibrium point is depicted as a gray plane. Then, Fig. 9b) displays the time history of the radial velocity, $\dot{r}$, relative to the Moon for this same trajectory with a color scheme that is consistent with Fig. 9a). For the first hundred days of propagating this trajectory backwards in time from a feasible lunar science orbit, the spacecraft is predominantly influenced by the gravity of the Moon. However, once the spacecraft is located further from the Moon – occurring after approximately 100 days before reaching the final lunar science orbit for this particular trajectory – the Earth's gravity significantly influences the path of the spacecraft. It is during this phase that the geometry of the lunar approach trajectory is most complex to analyze and translate into target conditions during trajectory design [13, 14]. Similar characteristics are observed across the entire dataset. Thus, the geometry of these trajectories is defined in the feature vector via the states and times at several apolunes that occur immediately after the spacecraft enters the lunar vicinity. These apolunes, displayed as black circles in Fig. 9 occur along both natural and low-thrust-enabled arcs.

The full dataset capturing 12,095 low-thrust-enabled lunar orbit insertion arcs is partitioned into three smaller datasets that are clustered individually using the l^2-norm to assess similarity. This partitioning is performed using the number of apses that occur along the trajectory – generated backwards in time from a feasible lunar orbit – after the 325th apse and before piercing the selected hyperplane. The conditions used to perform this partitioning are displayed in the second column of Table 3 while the fourth column reflects the number of trajectories in each partition. The feature vector describing each member of the partitioned dataset is then formed to reflect the apolunes that occur after the spacecraft passes through the Earth–Moon L_2 gateway in forward time and after completing the 325th apse in backward time. Once the feature vectors are evaluated for the members of each partition, the HDBSCAN input parameters, m_{pts} and m_{clSize}, are selected. Due to the wide variety of solutions captured across the three partitions, these quantities must be selected individually for each partition; the values selected based on analysis of the $DBCV$ index, noise level and total number of generated clusters are displayed in the third column of Table 3. Then, clustering is performed on each of the individual partitions using the l^2-norm to assess similarity. The properties of the clustering results for each partition are summarized in the last three columns of Table 3. Of course, these properties are a direct consequence of the selected values of m_{pts} and m_{clSize}. Modifying these values may, for instance, produce similar clusters with a lower fraction of members identified as noise. Furthermore, an alternative discretization of the feasible lunar orbit set may produce a different density of solutions within the higher-dimensional space associated with the feature vectors, thereby influencing the number of clusters and percentage of members identified as noise. Nevertheless, the results of

Dataset No.	Number of Apses, N_a	$[m_{pts}, m_{clSize}]$	$\|T\|$	$N_{clusters}$	$DBCV$	Noise level %
1	$N_a < 21$	[33, 95]	4406	6	0.5703	7.1947
2	$21 \leq N_a < 35$	[38, 190]	6787	10	0.4570	6.2178
3	$36 \leq N_a < 65$	[52, 55]	902	7	0.4069	16.7406

Table 3 Clustering parameters and results for each partition of the dataset composed of low-thrust-enabled lunar insertion arcs.

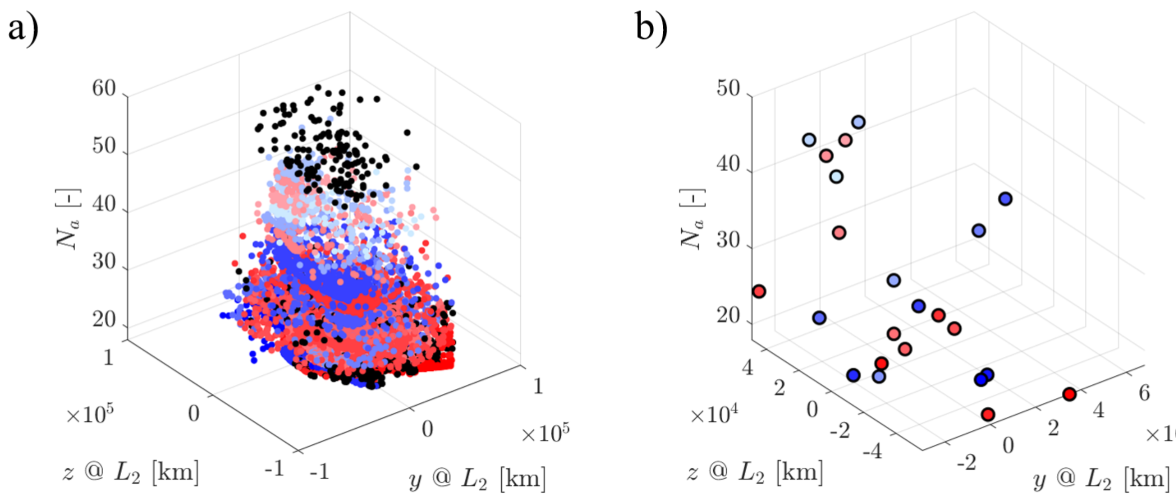

Fig. 10 Aggregate results of clustering within each of the partition using the l^2-norm to assess similarity displaying a) all members of the full dataset and b) only representatives of each cluster.

clustering within each individual partition are displayed together in Fig. 10. In Fig. 10a), each map crossing is colored by the cluster assignment and plotted in three dimensions: the y-coordinate and z-coordinate of the intersection of the trajectory with the hyperplane at Earth–Moon L_2 are represented along with the number of apses, N_a that occur after the spacecraft passes through the Earth–Moon L_2 gateway in forward time and after completing the 325th apse in backward time. In Fig. 10b), only the cluster representatives, i.e., the medoids of each cluster, are plotted in the same three dimensional view to reduce the complexity of visualization and supply insight into the variety of solution geometries captured by the full dataset. However, these clusters only correspond to a grouping performed within each of the three partitions and with similarity assessed via an isochronous correspondence; merging of clusters of geometrically similar solutions within and across each of the partitions is necessary.

Within each partition of the dataset, clusters are merged if they are composed of geometrically similar solutions that differ only in the number of recorded apses. To motivate this first merging step, consider two representative solutions from the first dataset, composed of solutions that admit less than 21 apses after the spacecraft passes through the Earth–Moon L_2 gateway in forward time and after completing the 325th apse in backward time. These two representatives, from clusters 0 and 1, are displayed in Fig. 11 with red arcs indicating that the low-thrust engine is activated and blue arcs corresponding to natural motion. Figs. 11a) and c) display a three-dimensional view of these two trajectories, while Figs. 11b) and d) display the associated projections onto the xy-plane of the Earth–Moon rotating frame. Analysis of these figures reveals that these two solutions are geometrically similar; yet, they differ in the number of apolunes completed after the spacecraft passes through the Earth–Moon L_2 gateway in forward time and after completing the 325th apse in backward time. In Fig. 11b), the first apolune that occurs along the representative trajectory for cluster 0 is highlighted via a red dashed circle. However, the representative trajectory for cluster 1 does not admit an apolune in the same region. Since the feature vector is defined using a time-ordered sequence of apolunes along each trajectory, using the l^2-norm as a similarity measure results in an isochronous correspondence between the two sequences of apolunes. However, using the modified Hausdorff distance to assess similarity between solutions in these two clusters enables a straightforward procedure for merging clusters of solutions that are geometrically similar within each partition. In fact, the cluster validity index is computed using the Hausdorff distance for each possible pair of clusters within each partition; if the cluster validity index, $V_C(C_i)$, is negative for both clusters, they are merged. For clusters 0 and 1, described by the geometrically similar representatives displayed in Fig. 11, the cluster validity indices calculated using the modified Hausdorrf distance are $\boldsymbol{V_C} = [-0.9704, -0.9096]$. Since both of these quantities are negative, these clusters are merged. Once this procedure has been completed for every cluster in the first partition of the dataset, the number of clusters is reduced from six to three. The representative solutions of these three clusters are displayed in Fig. 12 using a color scheme that is consistent with Fig. 11. Visual inspection of these representatives reveals that they are geometrically dissimilar and, therefore, should exist in separate clusters.

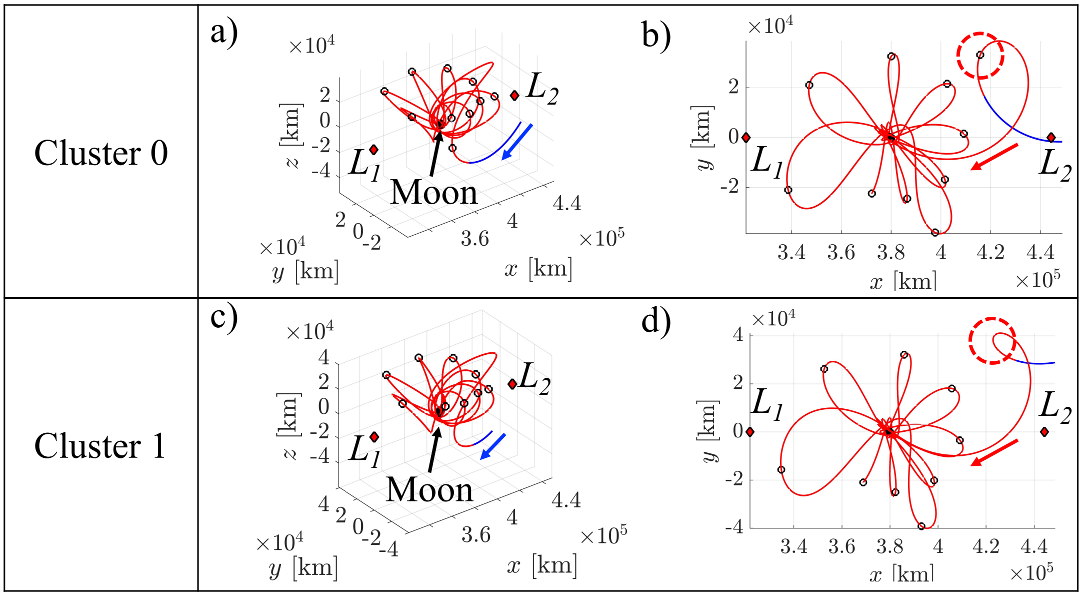

Fig. 11 Representative trajectories for cluster 0 and 1 from the first partition, plotted in the Earth–Moon rotating frame: a) and c) three-dimensional views; b) and d) as planar projections. Apolunes identified via black circles.

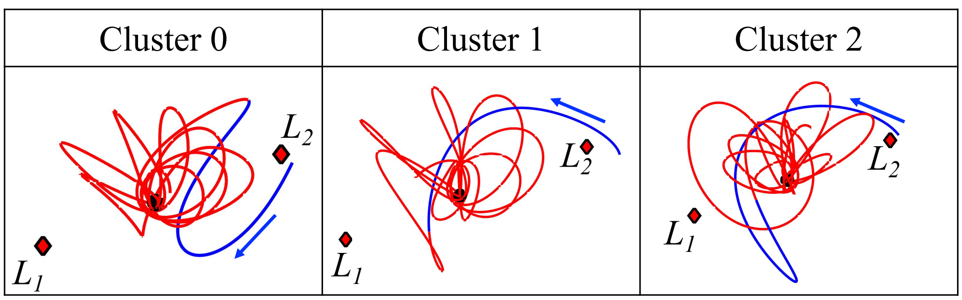

Fig. 12 Representative solutions for clusters within the first partition following the first merging step.

Since solutions with a similar geometry may, potentially, exist across multiple partitions of the dataset, a second cluster merging process is performed across partitions. The general procedure for merging clusters that exist across multiple partitions is similar to the process for merging solutions within a single cluster. The cluster validity index, evaluated using the modified Hausdorff distance, is calculated for each possible pair of clusters that exist at the boundaries of contiguous partitions; if the cluster validity index for both clusters in a pair are negative, the clusters are composed of geometrically similar solutions and are, therefore, merged. To demonstrate this process, consider the number of apolunes that occur along trajectories within each cluster from the first two partitions, as depicted in Fig. 13a). In this figure, the cluster ID is an integer identifying each grouping of trajectories discovered following application of HDBSCAN and the subsequent cluster merging process within each individual partition: nonnegative values indicate a true cluster, represented by filled circles colored uniquely for each cluster, while the black circles at a cluster ID of -1 correspond to members of the dataset that are considered noise. Then, consider a pair of clusters that exist at the boundaries of partitions: cluster 0 from the first dataset, composed of trajectories similar to those displayed in the left column of Fig. 12, and cluster 4 from the second partition. Each of these clusters is highlighted by an orange box in Fig. 13a). Trajectories within these clusters complete between 18 and 23 apses after the spacecraft passes through the Earth–Moon L_2 gateway in forward time and after completing the 325th apse in backward time – indicating that these clusters may, potentially, encompass geometrically similar trajectories. Applying this inter-partition cluster merging procedure to clusters 0 and 4 produces values of the cluster validity index equal to $\boldsymbol{V_C} = [-0.9012, -0.9190]$ when evaluated using the Hausdorff distance. These two negative values indicate that the clusters are composed of solutions that are geometrically similar and should be merged. This assessment is confirmed via visual inspection of the cluster representatives. The representative trajectory for cluster 0 appears in Fig. 12a) while the representative trajectory for cluster 4 is displayed in Fig. 14 in via a) a three-dimensional view and b) as a projection onto the xy-plane of the Earth–Moon rotating frame; these two representative solutions are indeed geometrically similar. A similar merging process is performed for the two clusters highlighted by the blue box in Fig. 13a). Following cluster merging across partitions, the first and second partitions of the dataset admit only nine unique clusters, as displayed in Fig. 13b). A similar procedure is performed for the clusters at the boundary of the second and third partitions. Following this merging procedure, 18 independent clusters are recovered for the entire dataset. The representative trajectories associated with these clusters are displayed in Fig. 15 in the Earth–Moon rotating frame with low-thrust arcs colored red and natural arcs colored blue. Visual inspection of this reduced set of representative trajectories reveals that each solution exhibits a fundamentally distinct geometry, demonstrating the value of a clustering-based approach to exploring and analyzing the complex solution space captured on a higher-dimensional Poincaré map.

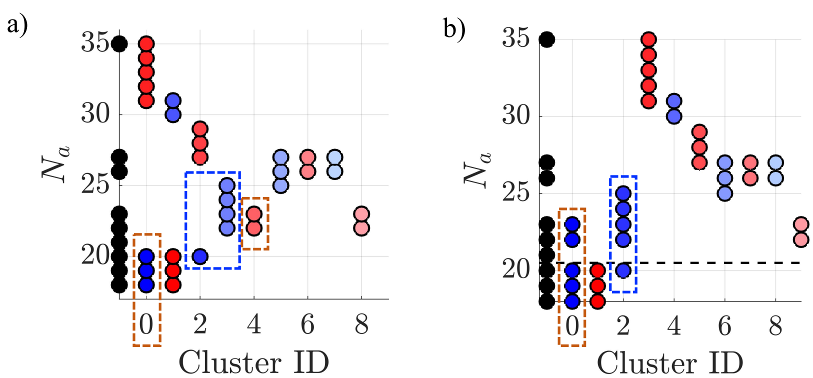

Fig. 13 Number of recorded apolunes, N_a after the spacecraft passes through the Earth–Moon L_2 gateway in forward time and after completing the 325th apse in backward time: a) before and b) after merging clusters across the first and second partitions.

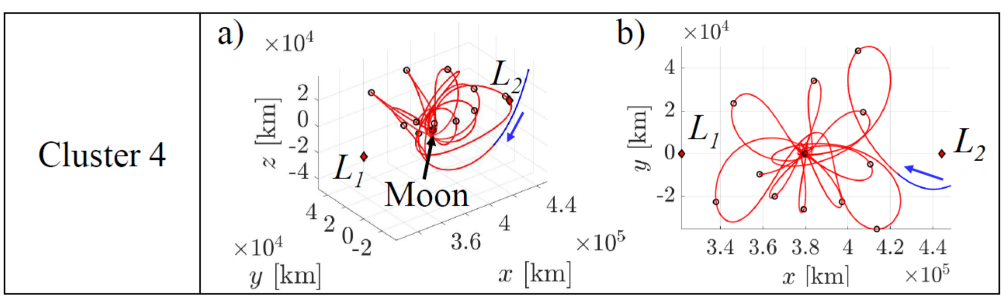

Fig. 14 Representative solution for cluster 4 in the second partition prior to merging across partitions.

VIII. Conclusion

In this paper, the density-based clustering algorithm HDBSCAN is employed to cluster the higher-dimensional crossings on a Poincaré map by the geometry of the associated trajectories. Using clustering to group the trajectories in a complex solution space enables the recovery of representative solutions that provide insight into the geometries of feasible arcs along with their region of existence. The presented clustering approach is demonstrated for Poincaré maps capturing spatial trajectories generated in two dynamical models: the natural CR3BP and a low-thrust-enabled CR3BP. Then, this technique is used to gain further insight into the trajectory design process for the Lunar IceCube mission.

The first example explored in this paper focuses applying clustering to a Poincaré map capturing spatial trajectories in the Sun–Earth CR3BP. First, a strategy inspired by tomography is leveraged to reduce the computational burden of

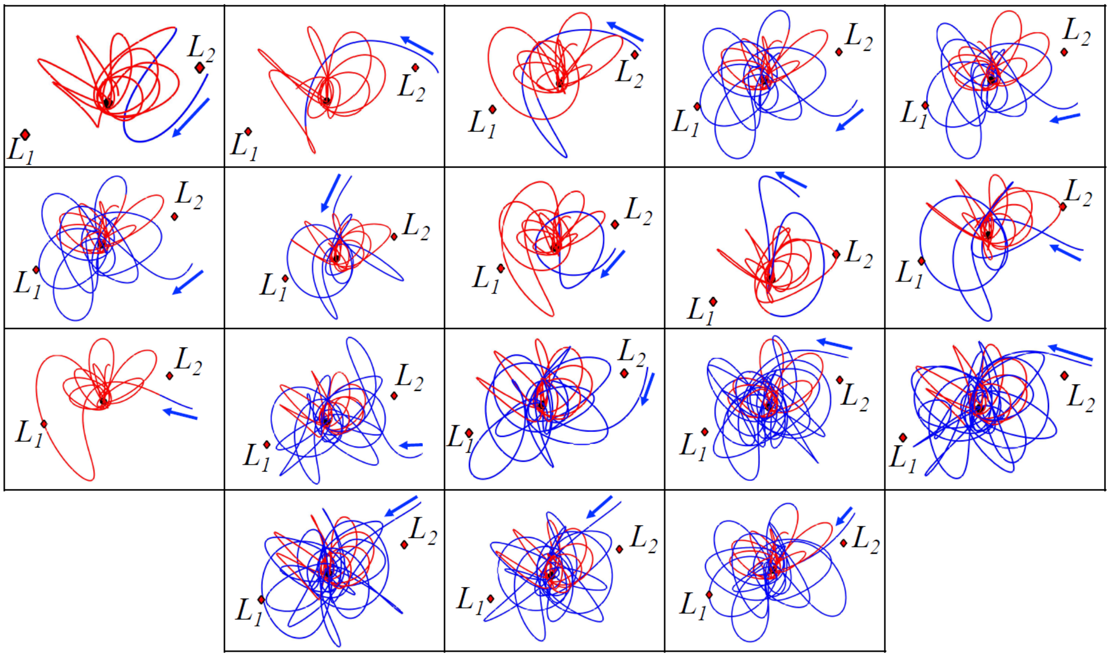

Fig. 15 Representative solutions for all 18 clusters associated with the full lunar orbit insertion dataset after both cluster merging steps. Trajectories are displayed in the Earth–Moon rotating frame with low-thrust arcs (red) and natural arcs (blue); arrows indicate direction of motion in forward time.

data generation and clustering. Specifically, the set of initial conditions used to generate the map data is partitioned using additional hyperplanes. Then, the partitions of the dataset are independently clustered. Next, clusters that exist across multiple partitions via intersections in the phase space are merged. This approach is first applied to a periapsis map constructed at a single value of the Jacobi constant, recovering groupings of solutions with fundamentally different geometries – and producing results that are consistent with the clustering of a single, large dataset. Then, this approach is applied to the analysis of arcs used in trajectory construction for the Lunar IceCube mission, with a focus on the phasing and energy adjustment segment.

The second example explored in this paper focuses on a Poincaré map capturing low-thrust-enabled solutions in the Earth–Moon CR3BP with direct application to the Lunar IceCube mission. In this example, a large set of trajectories that approach a specified lunar orbit are generated in the low-thrust-enabled Earth–Moon CR3BP. Then, to reduce the computational burden during clustering, the dataset is prepartitioned according to the number of lunar apses that occur along each trajectory. Each partition is clustered via HDBSCAN using an isochronous measure of similarity, i.e., the l^2-norm. Then, within each partition, clusters are merged if they are composed of geometrically similar solutions. This merging is performed using the cluster validity index evaluated using a modified Hausdorff distance. Then, clusters are merged across the boundaries of partitions using the same strategy. The result of this approach is a computationally-efficient means for grouping the solutions associated with higher-dimensional crossings on a Poincaré map according to their geometry.

Through these two examples, this paper demonstrates the value of a clustering-based approach to analysis of the complex solution space in nonautonomous and chaotic dynamical models. In fact, clustering enables a straightforward summarization of the solution space via a set of representative solutions along with insight into the region of existence of each arc; insight that is valuable to the human analyst during the trajectory design process. In addition, leveraging a prepartitioning and merging process reduces the computational complexity of clustering the large dataset typically associated with higher-dimensional Poincaré maps.

Acknowledgments

This work was completed at the University of Colorado Boulder under NASA Grant 80NSSC18K1536. The first author is grateful for the support of the Bahls Endowed Fund for travel to this conference.

References

[1] Parker, J. S., and Anderson, R. L., *Low-Energy Lunar Trajectory Design*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2014. https://doi.org/10.1002/9781118855065.

[2] Koon, W. S., Lo, M. W., Marsden, J. E., and Ross, S. D., "Resonance and Capture of Jupiter Comets," *Celestial Mechanics and Dynamical Astronomy*, Vol. 81, No. 27, September 2001, pp. 27 – 38. https://doi.org/10.1023/A:1013398801813.

[3] Bosanac, N., Cox, A., Howell, K., and Folta, D., "Trajectory Design for a Cislunar CubeSat Leveraging Dynamical Systems Techniques: The Lunar IceCube Mission," *Acta Astronautica*, Vol. 144, March 2018, pp. 283–296. https://doi.org/10.1016/j.actaastro.2017.12.025.

[4] Haapala, A., "Trajectory Design in the Spatial Circular Restricted Three-Body Problem Exploiting Higher-Dimensional Poincare Maps," Ph.D. thesis, Purdue University, West Lafayette, IN, 2014.

[5] Gómez, G., Koon, W., Lo, M., Marsden, J., Masdemont, J., and Ross, S., "Connecting Orbits and Invariant Manifolds in the Spatial Restricted Three-Body Problem," *Nonlinearity*, Vol. 17, 2004, pp. 1571–1606.

[6] Hadjighasem, A., Karrasch, D., Teramoto, H., and Haller, G., "Spectral-Clustering Approach to Lagrangian Vortex Detection," *Physical Review E*, Vol. 93, No. 6, 2016.

[7] Nakhjiri, N., and Villac, B. F., "Automated Stable Region Generation, Detection, and Representation for Applications to Mission Design," *Celestial Mechanics and Dynamical Astronomy*, Vol. 123, No. 1, 2015, pp. 63–83.

[8] Villac, B., Anderson, R., and Pini, A., "Computer Aided Ballistic Orbit Classification Around Small Bodies," *The Journal of the Astronautical Sciences*, Vol. 63, No. 3, 2016, pp. 175–205.

[9] Campello, R., Moulavi, D., and Sander, J., *Density-Based Clustering Based on Hierarchical Density Estimates*, Springer, Berlin, Heidelberg, 2013. https://doi.org/10.1145/2733381.

[10] Bosanac, N., "A Data Mining Approach to Using Poincaré Maps in Multi-Body Trajectory Design Strategies," *29th AAS/AIAA Spaceflight Mechanics Meeting*, Ka'anapali, HI, 2019. Updated version currently under journal review.

[11] Smith, T., and Bosanac, N., "Constructing a Set of Motion Primitives in the Circular Restricted Three-Body Problem via Clustering," *AAS/AIAA Astrodynamics Specialist Conference*, Portland, ME, 2019.

[12] Moulavi, D., Jaskowiak, P., Campello, R., Zimek, A., and Sander, J., "Density-Based Cluster Validation," *Proceedings of the 2014 SIAM International Conference on Data Mining*, Philadelphia, PA, 2014. https://doi.org/10.1137/1.9781611973440.96.

[13] Folta, D., Bosanac, N., Cox, A., and Howell., K., "The Lunar IceCube Mission Challenge: Attaining Sience Orbit Parameters from a Constrained Approach Trajectory," *27th AAS/AIAA SpaceFlight Mechanics Meeting*, San Antonio, TX, 2017.

[14] Bosanac, N., "Leveraging Natural Dynamical Structures to Explore Multi-Body Systems," Ph.D. thesis, Purdue University, West Lafayette, IN, 2016.

[15] Szebehely, V., *Theory of Orbits: The Restricted Problem of Three Bodies*, Academic Press, London, UK, 1967.

[16] Verhulst, F., *Nonlinear Differential Equations and Dynamical Systems*, Springer-Verlag, Berlin, Heidelberg, 1996. https://doi.org/10.1007/978-3-642-61453-8.

[17] Villac, B., and Scheeres, D., "On the Concept of Periapsis in Hill's Problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 90, 2004, pp. 165–178. https://doi.org/10.1007/s10569-004-0405-9.

[18] Han, J., and Kamber, M., *Data Mining: Concepts and Techniques, Second Edition*, Ch.7, Proquest EBook Central: Elsevier Science and Technology, Waltham, MA, 2006.

[19] McInnes, L., Healy, J., and Astels, S., "hdbscan: Hierarchical Density Based Clustering," *The Journal of Open Source Software*, Vol. 2, 2017.

[20] Dubuisson, M.-P., and Jain, A., "A Modified Hausdorff Distance for Object Matching," 1994. https://doi.org/10.1109/ICPR.1994.576361.

[21] Cichosz, P., *Data Mining Algorithms: Explained Using R*, John Wiley and Sons, West Sussex, United Kingdom, 2015.

[22] McInnes, L., Healy, J., and Astels, S., "How HDBSCAN Works," , 2016. URL https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html, accessed October 2019.