A Data Mining Approach to Poincaré Maps in Multi-Body Trajectory Design

Natasha Bosanac *
University of Colorado Boulder, Boulder, CO, 80303

I. Introduction

Rapid and informed trajectory design strategies within multi-body systems often benefit from the use of Poincaré maps. Specifically, Poincaré mapping enables visualization of a large set of trajectories, generated in a given dynamical system, via their intersections with a hyperplane [1]. When constructed appropriately, Poincaré maps simplify the representation and analysis of fundamental motions in a chaotic dynamical system. For instance, consider a two-dimensional map that uniquely represents a state along a planar trajectory at a single value of a constant of motion in an autonomous system. Patterns that emerge on this map may reveal the characteristics of the solution space and the existence of fundamental dynamical structures [2–6]. Furthermore, a trajectory designer is often interested in assessing the fundamental geometries exhibited by solutions captured on the map along with their regions of existence. Such insight supports selecting individual arcs to construct an initial guess for an end-to-end trajectory [3, 7–10].

The absence of analytical expressions to extract the fundamental geometries exhibited by solutions in a complex flow regime may impede the human-in-the-loop data analysis tasks required for many trajectory design strategies that leverage Poincaré maps; yet, this problem is not unique to the astrodynamics. In fact, these issues are encountered in a variety of disciplines, from astronomy to medicine to air traffic management. These disciplines regularly produce and process large and complex data sets through, for example, stellar observations or clinical studies of human subjects. There are often challenges in manually grouping the members of such complex data sets when: there are no general analytical expressions for separating the data; the underlying structure of the data set and associated groupings of members are not known a priori; or labelling a sufficiently representative subset of the data would be too challenging or time-consuming [11]. These disciplines have leveraged clustering, a form of unsupervised learning algorithms, to enable the discovery of groupings within large and complex data sets to inform scientific analyses [12–14].

In astrodynamics and applied mathematics, there have been several recent contributions to applying clustering techniques to chaotic dynamical models, including the Circular Restricted Three-Body Problem (CR3BP). For instance, Nakhjiri and Villac use k-means clustering to separate stable motion from chaos on a Fast Lyapunov Indicator (FLI) map in the planar CR3BP, with a focus on the region near distant retrograde orbits [15]. They also leverage this approach to govern automated map generation in this specific region. In addition, Hadjighasem, Karrasch, Teramoto and Haller

*Assistant Professor, Colorado Center for Astrodynamics Research, Ann and H.J. Smead Department of Aerospace Engineering Sciences, UCB 431. Member AIAA.

This version of the paper has been accepted for publication by AIAA's Journal of Guidance, Control, and Dynamics after peer review but does not reflect any changes due to copy-editing and formatting by AIAA. The published manuscript is available online at: https://arc.aiaa.org/doi/10.2514/1.G004857

apply spectral clustering to Lagrangian vortex detection in several generalized flow problems [16]. They demonstrate that this spectral clustering approach effectively groups trajectories based on their geometry, with similarity between two trajectories defined using a weighted sum of the distances between two particles sampled at regular times along the solutions. Another example of the use of clustering in astrodynamics includes the work of Villac, Anderson and Pini [17]. These authors leverage k-means clustering to organize periodic orbits, computed in the vicinity of an irregular body, into sets that are analogous to orbit families. In each of these examples, unsupervised clustering approaches successfully enable further analysis of a complex solution space that cannot be described or partitioned analytically.

This paper presents a strategy that leverages hierarchical and density-based clustering to enable analysis of the information contained on a Poincaré map that captures planar trajectories within the autonomous CR3BP. Specifically, hierarchical and density-based clustering is used to group the crossings on a Poincaré map according to the geometry of the associated trajectories. The first step involves parameterizing the trajectories associated with the map crossings via a summarized description that balances the fidelity level of the geometrical representation with the dimensionality [18]. To achieve this goal, each trajectory is described via normalized time and state information at each apse for up to several revolutions around a primary. Then, two distance metrics are used to measure the similarity between trajectories: the Euclidean distance, supporting an isochronous comparison between two solutions; and a modified Hausdorff distance, enabling a global comparison of the geometry of two trajectories, independent of the initial condition [19]. With these distance metrics and the selected trajectory summarization strategy, clustering is implemented via the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm [20]. This algorithm is well-suited to the data associated with a general Poincaré map due to its ability to accommodate: a number of clusters that is not known a priori; clusters of various shapes; clusters of various densities; and an unknown or nonconstant value of the maximum separation between data within a single cluster in a higher-dimensional space [21]. Each cluster of map crossings, identified via HDBSCAN, is then summarized by a single trajectory or map crossing selected using the medoid of the cluster; the result is a representative reduced data set that enables cluster verification and further analysis of the solution space. As an initial proof of concept, this approach is demonstrated within the context of a periapsis map that captures planar solutions in the Sun-Earth CR3BP at a single energy level.

The motivation behind the procedure presented in this paper is to use this data-driven approach to enable a trajectory designer to rapidly assess the solution space and simultaneously gain insight into the region of existence of arcs with a specific geometry. In fact, crossings on the map associated with an arc of interest may be identified from the representative reduced data set via a global view and then used to isolate the corresponding individual cluster of trajectories for further examination. Note that this paper focuses on a proof of concept of this clustering-based approach through the application to a two-dimensional Poincaré map constructed for planar solutions in an autonomous dynamical model. However, the capability to group the crossings on a map by their associated trajectories in an unsupervised manner may also be beneficial for more complex solution sets or dynamical models, e.g., spatial trajectories in an autonomous

system or in nonautonomous dynamical models. Furthermore, a clustering-based approach to analysis and visualization of the solution space within a multi-body system has the potential to reduce the burden on a human-in-the-loop and the time required for trajectory design activities that leverage Poincaré maps; for instance, during mission concept development, planning extensions or in time-critical scenarios necessitating redesign.

II. Dynamical Model

The CR3BP is leveraged to construct a Poincaré map that captures a sufficiently complex set of trajectories with a wide variety of geometries. This dynamical model describes the motion of an assumed massless particle, P_3, under the point mass gravitational interactions of two massive primaries, P_1 and P_2 [22]. Each of these two primaries, with a mass M_i, where $i = 1, 2$, is assumed to follow circular orbits around their mutual barycenter. In addition, a nondimensionalization scheme is introduced to enable a comparison between systems with similar relative masses and to reduce the potential for ill-conditioning. Length quantities are normalized by the constant distance between the two primaries, while mass parameters are nondimensionalized by the total mass of the system. Then, μ is defined as the ratio of the mass of the smaller primary, P_2, to the total mass of system. Finally, time quantities are normalized to set the mean motion of the primary system equal to unity. Following nondimensionalization, a rotating frame, $\hat{x}\hat{y}\hat{z}$, is introduced to reduce the complexity of visualization and to enable the definition of an autonomous dynamical system. This rotating frame is defined with the $\hat{x}$-axis directed from P_1 to P_2, $\hat{z}$ is parallel to the orbital angular momentum vector of the primaries, and $\hat{y}$ completes the right-handed triad. With these definitions, the state of P_3 is written in nondimensional coordinates relative to the system barycenter and in the rotating frame as $\boldsymbol{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$. Then, the nondimensional equations of motion for P_3 in the CR3BP, expressed in the rotating frame, are written as:

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

where $U = (1/2)(x^2 + y^2) + (1 - \mu)/d + \mu/r$ is a pseudo-potential function, while the distances between P_3 and the primaries, P_1 and P_2, are, respectively, $d = \sqrt{(x + \mu)^2 + y^2 + z^2}$ and $r = \sqrt{(x - 1 + \mu)^2 + y^2 + z^2}$ [22]. This autonomous dynamical model admits a constant of motion, commonly labeled the Jacobi constant and equal to $C_J = 2U - \dot{x}^2 - \dot{y}^2 - \dot{z}^2$ [22]. At a single value of the Jacobi constant, the solution space is composed of trajectories that exhibit a large variety of characteristics. However, simultaneous visualization and analysis of a large set of these solutions in configuration or phase space may be challenging and time-consuming for the human-in-the-loop performing data analysis.

III. Poincaré Mapping

Poincaré mapping techniques enable a discrete-time representation of a continuous-time flow, reducing the complexity and dimensionality of visualizing solutions within a chaotic dynamical system. To construct these maps, a hyperplane or surface of section is first defined transverse to the solutions of interest [23]. There are numerous options for defining a useful hyperplane in the CR3BP including, but not limited to: a physically interpretable plane expressed in configuration space variables; a stroboscopic map that captures the flow at constant time intervals; or a known event that occurs along a trajectory, such as the locally minimum distance from a reference location (e.g., periapsis) [1, 24, 25]. Given an appropriately selected hyperplane, initial conditions are seeded within a desired region of the phase space. Each initial condition is propagated forward or backward in time until its i-th intersection with the hyperplane in a desired direction. This process is repeated for a desired number of successive intersections with the hyperplane in a specified direction. Each crossing is then captured and represented on a lower-dimensional, one-sided map. Patterns formed in this map – or even the lack thereof – enable the detection of various types of fundamental motions and distinguish order from chaos within the flow [4].

To demonstrate the map construction process, consider planar motion near the Earth vicinity in the Sun-Earth CR3BP. At a single value of the Jacobi constant, trajectories may potentially exhibit a large variety of geometries with behaviors including: captured motion near the Earth vicinity; impacting the Earth; or passing through either the L_1 or L_2 gateways, over various timescales, to visit other regions of the Sun-Earth system [3, 6, 8]. This diverse solution space in a chaotic flow regime benefits from the use of Poincaré maps for visualization and analysis. For this example, consider a map capturing perigees that occur along planar solutions at a single value of the Jacobi constant in the Earth vicinity. First, a hyperplane is defined such that:

$$(x - 1 + \mu)\dot{x} + y\dot{y} + z\dot{z} = 0 \quad \text{and} \quad (x - 1 + \mu)\ddot{x} + y\ddot{y} + z\ddot{z} + \dot{x}^2 + \dot{y}^2 + \dot{z}^2 > 0 \tag{2}$$

thereby capturing the perigees occurring along a trajectory [26, 27]. At a single value of the Jacobi constant, feasible initial conditions are seeded directly from this hyperplane, using states of the form $\boldsymbol{x}_C = [x, y, 0, \dot{x}, \dot{y}, 0]^T$ for N_x equally-spaced values of the x-coordinate between Sun-Earth L_1 and L_2 and N_y equally-spaced values of the y-coordinate in the range $y = [y_{min}, y_{max}]$. At a perigee location along a planar trajectory, the velocity and position vectors, relative to the Earth, are perpendicular. Thus, a unit vector aligned with the in-plane velocity vector is identified directly from this orthogonality condition [27]. A consistent direction of motion, defined in the rotating frame, is selected for the initial conditions: either (1) prograde, with P_3 instantaneously possessing an angular momentum vector relative to the Earth that is aligned with the orbital angular momentum of the primaries; or (2) retrograde, with P_3 traveling in a clockwise direction around the Earth at that instant of time. This direction of motion is used to select the correct direction of the velocity unit vector for each initial condition. Then, the Jacobi constant relationship is rearranged to

provide the following expression for the velocity magnitude, v, as a function of the pseudo-potential, which depends only on the position coordinates and $C_{J,d}$, the desired Jacobi constant: $v = \sqrt{2U - C_{J,d}}$. If this speed corresponds to a real number at a given position relative to the Earth, the value of v is used to scale the in-plane velocity unit vector and directly recover the velocity components, $\dot{x}$ and $\dot{y}$. This process is implemented for each apse location and only states that simultaneously correspond to the desired Jacobi constant and satisfy the periapsis constraint in Eq. (2) are retained as initial conditions. Each of these initial conditions is then propagated forward in time until the associated trajectory either: completes N_{ret} positive intersections of the map, i.e., subsequent perigees; passes within a distance of 10^{-6} nondimensional units to the Earth, located below the Earth's surface; or escapes from the vicinity of the Earth as defined by the trajectories passing through the L_1 or L_2 gateways. The collection of crossings of the hyperplane that occur along each trajectory are then displayed on a two-dimensional plot reflecting the x- and y-coordinates in the rotating frame to visualize the behavior of solutions in the Earth vicinity.

Following the outlined procedure for implementation, a periapsis map is constructed for planar motion in the Sun-Earth system, described by a mass ratio of $\mu = 3.00348 \times 10^{-6}$, at a Jacobi constant of $C_J = 3.00088$. At this value of the Jacobi constant, both the L_1 and L_2 gateways are open. Thus, the solution space is composed of trajectories that exhibit a wide variety of geometries, offering a sufficiently complex data set for testing the presented clustering approach. For this map, prograde initial conditions are seeded as perigees at 200 equally-spaced x-coordinates between L_1 and L_2 and 200 equally-spaced y-coordinates in the range $y = [-0.01, 0.01]$. Only a subset of these initial conditions produce viable state vectors that satisfy the perigee condition and correspond to a real-valued velocity magnitude; these feasible initial conditions are propagated either forward or backward in time in the CR3BP with up to 20 returns to the map recorded. The resulting maps in each case are depicted in Fig. 1 with each map crossing displayed via a black point for: a) trajectories integrated forward in time and b) trajectories integrated backward in time. In these figures, the x- and y-coordinates of the state at each perigee along each trajectory are represented on the horizontal and vertical axes of the figure, respectively. Each of the L_1 and L_2 equilibrium points is located by a red diamond while the Earth is identified, not to scale, at the center of the figure via a purple circle. In addition, the gray shaded regions represent 'forbidden regions' bound by the well-known zero velocity curves [28]. Since each map in Fig. 1 captures periapses along planar solutions at a single value of the Jacobi constant, each crossing of the map uniquely defines the entire state at perigee; yet, these maps reveal a complex solution space that is still challenging to analyze.

The patterns that form on the Poincaré maps in Fig. 1 are governed by the stable and unstable manifolds associated with L_1 and L_2 Lyapunov orbits [8, 27]. First, the periapsis maps in Fig. 1 admit regions with distinctly different densities in the map crossings; the boundaries of these regions tend to correspond to the first few revolutions of the manifolds associated with the L_1 and L_2 Lyapunov orbits at the same Jacobi constant of $C_J = 3.00088$. In Fig. 1a), where the trajectories are integrated forward in time, the low-density regions are governed by the unstable manifolds, while the stable manifolds bound the low-density regions in the map in Fig. 1b), where solutions are propagated

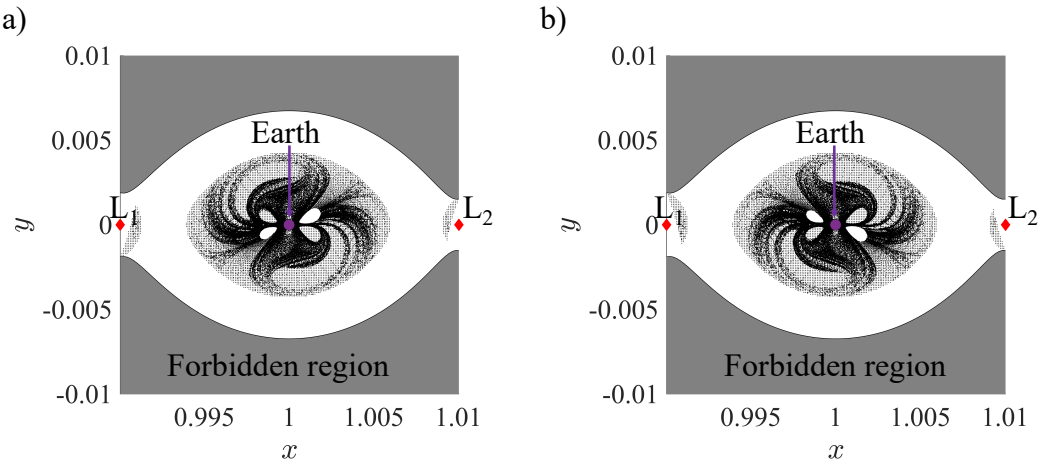

Fig. 1 Periapsis map in the Sun-Earth system for $C_J = 3.00088$, constructed by integrating trajectories a) forward in time and b) backward in time.

backward in time. To visualize this correlation, up to ten intersections of the unstable and stable manifold structures with the perigee surface of section are displayed in blue in Figs. 2a) and b), respectively, with the first three intersections for each manifold labeled [27]. In Fig. 2a), the i-th intersection of the unstable manifold associated with the L_j Lyapunov orbit, for $j = 1, 2$, is labeled using the notation $U_{j,i}$; similarly, in Fig. 2b), the associated crossing of the stable manifold is labeled as $S_{j,i}$. To possess periapses within a low-density region at this energy level, trajectories originally pass through the L_1 or L_2 gateways prior to encircling the Earth in forward or backward time; thus, low-density regions correspond to solutions that have completed fewer revolutions around the Earth before or after passage through either gateway. Such insight reveals the history and fate of the trajectories associated with the map crossings. Furthermore, the stable and unstable manifolds offer a high-level verification of the results of the presented clustering approach to grouping the crossings on a Poincaré map. However, the trajectories within each manifold crossing may exhibit additional differences in geometry.

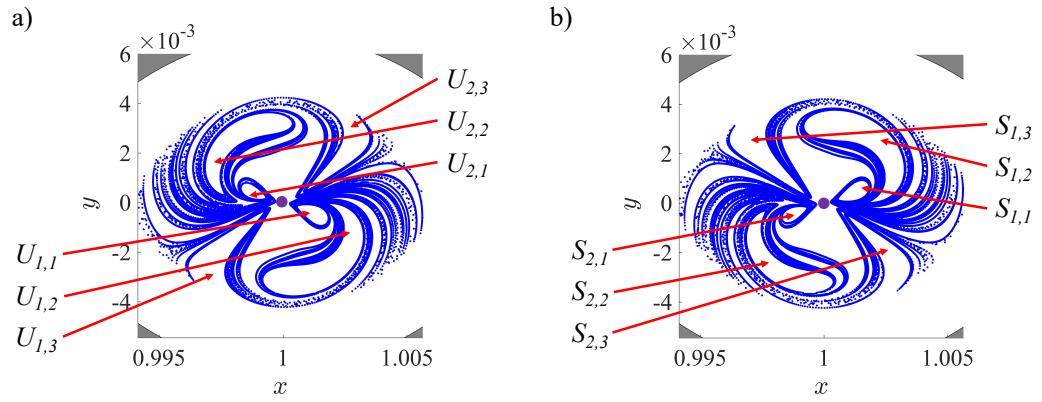

Fig. 2 Periapsis map capturing a) unstable and b) stable manifolds associated with the L_1 and L_2 Lyapunov orbits at $C_J = 3.00088$ in the Sun-Earth CR3BP.

IV. Trajectory Summarization

In the context of this analysis, the objective of trajectory summarization is to construct a compressed description, $\boldsymbol{T}_i$, of the trajectory associated with the i-th map crossing. Each crossing on a discrete-time map is associated with a continuous, nonlinear trajectory. In the absence of a closed form solution to a chaotic dynamical system, a trajectory could be represented to a high fidelity via discretization into a large time sequence of state vectors. Such an approach would certainly capture the geometry of the solution; however, the resulting data set would require prohibitively large data storage resources when applied to a large number of trajectories. Alternatively, reducing the dimension of the description reduces the storage requirements for the entire data set, while also mitigating the influence of the well-known curse of dimensionality in clustering [11]. The challenge, then, is to ensure that the trajectory summary is of a sufficient fidelity to enable differentiation between solutions of distinct geometries. Thus, a suitable trajectory summarization strategy must balance these goals to describe each trajectory by a vector of reasonable dimension that also reflects the solution geometry; this problem is encountered in moving object database applications and several solutions exist [29].

To construct a low-dimensional, yet representative, description of a trajectory in the CR3BP, a curve-based representation is employed. In particular, a trajectory is sampled at each apse relative to the Earth when propagated for a finite number of subsequent crossings of the map. As a preliminary approximation of the entire nonlinear trajectory, reduction to a sequence of apses or turning points offers a low-dimensional summarization that captures the general shape of the solution. In contrast to a line-based approximation, this curve-based approach enables straightforward construction of a compressed description that is equal in length for solutions of similar geometry, without the definition of specific tolerances or parameters governing the approximation [29]. Given the geometry of the solutions in the CR3BP, subsampling a continuous trajectory for a finite time interval at its apses will produce only a small number of states in a small computational time; for N_{peri} subsequent returns to the periapsis map, a total of up to $2N_{peri} + 1$ apses occur. Of course, increasing the number of returns to the map will reveal further differences between trajectories and increase the number of distinct geometries exhibited by the solutions associated with the map crossings. However, for this preliminary proof of concept, N_{peri} is selected as a small integer that sufficiently differentiates the geometry of the solutions in the planar CR3BP at a single value of the Jacobi constant.

A compressed description vector, $\boldsymbol{T}_i$, for the i-th trajectory is formed using information about the initial condition and up to $2N_{peri}$ subsequent apses. Recall that, in this analysis, each trajectory is integrated until meeting one of the following termination conditions: completing up to a finite number of subsequent returns to the hyperplane (i.e., perigee), passage through the L_1 or L_2 gateways, or passing within a distance of 10^{-6} nondimensional units to the Earth. The complete compressed description vector for the i-th trajectory is then formed as $\boldsymbol{T}_i = \left[\boldsymbol{R}_{i,1}, ..., \boldsymbol{R}_{i,2N_{peri}+1}\right]$ where $\boldsymbol{R}_{i,j}$ is a row vector that reflects the time, state and direction information associated with the j-th apse along the i-th trajectory, if it occurs. Specifically, the following quantities are included in the definition of $\boldsymbol{R}_{i,j}$ for a planar trajectory and scaled as follows:

1) $\tau_{i,j}$: the time at the j-th apse along the i-th trajectory is measured from zero at the initial condition and normalized by the total integration time along the solution.
2) $(\tilde{x}_{i,j}, \tilde{y}_{i,j})$: the position coordinates of an apse, defined in the rotating frame of the CR3BP and measured relative to the Earth, are normalized by the maximum magnitudes of the x- and y-coordinates across the initial conditions.
3) $\tilde{v}_{i,j}$: the speed, defined in the rotating frame of the CR3BP, is normalized by the maximum value of the speed across the set of initial conditions.
4) $\text{sign}(\hat{h}_{i,j} \cdot \hat{z})$: the sign of the z-component of the orbital angular momentum unit vector, calculated in the rotating frame relative to the Earth; this quantity equals either 1 or -1 for apses occurring along a trajectory.

Using these parameters, the five-dimensional vector $\boldsymbol{R}_{i,j}$ is defined as:

$$\boldsymbol{R}_{i,j} = \left[\tau_{i,j}, \quad \tilde{x}_{i,j}, \quad \tilde{y}_{i,j}, \quad \tilde{v}_{i,j}, \quad \text{sign}(\hat{h}_{i,j} \cdot \hat{z})\right] \tag{3}$$

and the components tend to have a maximum order of magnitude of 10^0. If, however, the solution passes through either of the L_1 or L_2 gateways or passes within a small distance to the center of the Earth after the k-th apse, $\boldsymbol{R}_{i,j}$ is assigned a placeholder value, $\boldsymbol{R}_{i,j} = [0, \pm 10, 0, 0, 0]$, for $j > k$. Here, a positive sign is used if the trajectory has terminated prior to reaching apoapsis while a negative sign indicates that the trajectory has terminated before reaching periapsis. The selected placeholder vector for $\boldsymbol{R}_{i,j}$ is designed to introduce a significantly large separation between members of the data set that reflect trajectories completing a distinctly different number of apses prior to termination. Furthermore, the sign of the nonzero element of $\boldsymbol{R}_{i,j}$ introduces further separation between trajectories of distinct geometries, i.e., those that terminate either before periapsis or apoapsis. Although these placeholder values increase the amount of data that is stored, they also produce compressed descriptions with a consistent length across all trajectories. In fact, the compressed description vector, $\boldsymbol{T}_i$, possesses a length of $(10N_{peri} + 5)$. The vectors, $\boldsymbol{T}_i$, are then combined to form the data set, [**S**], that is input to the clustering algorithm.

V. Data Analysis via Clustering

Clustering is a valuable tool for performing an unsupervised grouping of data with similar properties – and separation of dissimilar data [30]. Building upon prior demonstrations of a clustering approach to scientific data analysis in a variety of disciplines, this paper focuses on applying clustering to the data associated with a two-dimensional Poincaré map, with the goal of enabling analysis during the trajectory design process. There are a large variety of clustering algorithms that have been developed, including: partitioning methods, e.g., k-means, k-medoids; hierarchical methods, e.g., Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH), Clustering Using REpresentatives (CURE); and density-based methods, e.g., Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Ordering Points To Identify the Clustering Structure (OPTICS) [30]. Furthermore, hard clustering algorithms assume that each

member of a data set either belongs to a cluster or it does not, while soft clustering assesses the likelihood of each member belonging to a given cluster [11]. From the wide variety of available clustering techniques, the approach and algorithm used in a specific data analysis application must be selected based on the properties of the data set and the availability of information about the data or desired groupings prior to clustering.

The HDBSCAN algorithm, developed by Campello, Moulavi and Sander, is employed in this analysis [20]. This algorithm leverages a hierarchical and density-based approach to constructing clusters, each corresponding to data points that are densely located within the same neighborhood of a higher-dimensional space. These clusters are assigned hierarchically to capture the most significant clusters that consist of a sufficient number of data points; data that are not assigned to clusters are considered noise. HDBSCAN is specifically leveraged in this paper due to its capability to accommodate the properties of the data generated in the CR3BP, including: a number of clusters that is not necessarily known a priori; clusters that may exhibit a variety of shapes and densities across the data set; and an unknown or nonconstant value of the maximum separation between data points within a single cluster in a higher-dimensional space [21]. To provide a high-level background on this clustering technique, this section offers a brief conceptual overview of the HDBSCAN algorithm as outlined by Campello, Moulavi and Sander and Campello, Moulavi, Zimek and Sander [20, 21]. Then, a density-based definition of cluster validity developed by Moulavi, Jaskowiak, Campello, Zimek and Sander is summarized [31]. Next, two useful distance metrics – the Euclidean distance and a modified Hausdorff distance – are discussed as a means for defining a similarity measure during the clustering process. The concept of a medoid is then presented to enable extraction of a cluster representative from clusters of potentially irregular shapes.

A. Overview of the HDBSCAN Algorithm

To assess similarity between the members of a data set and locate regions of higher density, HDBSCAN relies on a quantity labeled the mutual reachability distance. To define this quantity, consider a data set, $[\mathbf{S}]$, consisting of N vectors. Each component, $\boldsymbol{s}_i$, of this data set is an M-dimensional vector that reflects the properties of the associated data. Then, the core distance of the i-th data point is defined as the distance between the point itself and its $N_{min,core}$-th nearest neighbor in the M-dimensional space. The quantity $N_{min,core}$ is a tunable parameter that defines the number of points required for a data point to be considered a *core* point, i.e., there is a sufficient number of points in its vicinity. Furthermore, the distance is calculated via a selected distance metric, e.g., a Euclidean distance, infinity-norm, l_1-norm, Hausdorff distance, etc. Then, a mutual reachability distance between the i-th data point and the j-th data point is defined as the maximum value of: (1) the core distance of the i-th data point; (2) the core distance of the j-th data point; and (3) the distance between the i-th and j-th data points. The mutual reachability distance for each of the N data points is then used to construct a mutual reachability graph with each of the N data points serving as vertices and the edges weighted by the calculated mutual reachability distance [20, 21]. This graph reflects the similarity between neighboring data points to support cluster identification within the M-dimensional space.

Using the mutual reachability distance as a foundation, HDBSCAN produces a hierarchy of all possible clusters that is simplified to remove noise. First, a minimum spanning tree is constructed for the mutual reachability graph, retaining only the edges that produce the minimum total weight as defined by the mutual reachability distance and introducing self edges weighted by the core distance of each data point. This minimum spanning tree is condensed to produce a dendrogram that reflects all possible clusters. Then, the dendrogram is traversed to locate true splits that correspond to new groupings of data with a number of data points that is above the threshold $N_{min,cluster}$. Specifically, HDBSCAN identifies stable or significant clusters as groups of data that persist over a large range of minimum threshold values in the mutual reachability distance; points that are not assigned to a cluster are labeled as noise points.

Following the general procedure for the HDBSCAN algorithm, several input and output parameters exist. Input parameters that are provided by the user are the data set, [**S**], the specific distance metric used to assess similarity, as well as the quantities $N_{min,core}$ and $N_{min,cluster}$. The data set possesses a dimension of $N \times M$: the value of N must be selected to sufficiently capture the information represented by the data set while reducing the computational time, complexity and data storage requirements during data generation and clustering; the M components of each data point must be defined to sufficiently capture the properties of the data while avoiding deterioration in the significance of each component due to the curse of dimensionality [11]. While, Campello, Moulavi and Sander suggest initially selecting $N_{min,core} = N_{min,cluster}$ to reduce the number of parameters governing the performance of HDBSCAN, these quantities are selected individually in this analysis. The outputs from HDBSCAN that are of most interest in this clustering procedure are typically the number of clusters, $N_{cluster}$, and the labels, l_i, identifying the cluster that the i-th member of the data set is assigned to: $0 \leq l_i < N_{cluster}$ if the data point belongs to a cluster or $l_i = -1$ for a noise point.

The HDBSCAN algorithm is accessed in this investigation via the *hdbscan* Clustering Library developed by McInnes, Healy and Astels in Python to offer a fast and efficient implementation [32]. As discussed by Campello, Moulavi and Sander, the computational complexity of the HDBSCAN algorithm is $\sim O(MN^2)$ in time and $\sim O(MN)$ in memory storage when the algorithm is provided an $N \times M$ dimensional data set [21]. However, depending on the user-defined properties and methods used at each step of the clustering algorithm, the computational complexity may be reduced even further. For the Python implementation used in this paper, the computational time complexity approaches $\sim O(NlogN)$ under certain circumstances; the details of the algorithmic implementation are described in McInnes and Healy [33].

B. Cluster Validation

An important, yet challenging, step in applying clustering algorithms to data analysis tasks is validating the cluster results. In data mining, there are three general approaches to cluster validation: (1) internal validation where validation criteria are constructed to directly leverage the data set; (2) external validation to compare the clustering results to a priori knowledge of the structure of the solution space; and (3) relative validation which requires a comparison of multiple clustering results to determine the better solution [31]. Validation via external methods is particularly

challenging when there is little to no a priori insight into the underlying structure of the data set. For the Poincaré mapping data set of interest in this paper, a limited external validation is achieved via comparison to the stable and unstable manifold structures. In addition, Moulavi, Jaskowiak, Campello, Zimek and Sander have developed an internal cluster validity measure labeled the Density Based Clustering Validation (DBCV) index [31]. This scalar quantity is designed to indicate the quality of a clustering result, with the capability to accommodate the potentially irregular-shaped groupings recovered by density-based clustering methods. Specifically, this quantity leverages two definitions: the density sparseness of the i-th cluster, DSC_i, which reflects the lowest density region within the cluster; and the density separation between the i-th and j-th clusters, $DSPC_{i,j}$, which captures the minimum reachability distance between the members of two distinct clusters. These two definitions are used to define a validity index, $V_{C,i}$, for the i-th cluster as:

$$V_{C,i} = \frac{\min_{j\in\{0,\ldots,N_{cluster}-1\},i\neq j}(DSPC_{i,j}) - DSC_i}{\max\left[\min_{j\in\{0,\ldots,N_{cluster}-1\},i\neq j}(DSPC_{i,j}), DSC_i\right]} \tag{4}$$

$V_{C,i}$ may produce either positive or negative scalar values, with a positive index indicating that the i-th cluster possesses a higher density than its density separation from the members of other clusters [31]. Then, for a data set of N members, the DBCV index is defined as the weighted average of this validity index across all $N_{cluster}$ clusters:

$$DBCV = \sum_{i=0}^{N_{cluster}-1}\left[\frac{N_i}{N}V_{C,i}\right] \tag{5}$$

where N_i is the number of members of cluster i. The DBCV index corresponds to values between -1 and +1 with higher values of the DBCV index indicating a better clustering result [31]. In this paper, these scalar quantities are used in two instances: to support selection of the parameters governing the HDBSCAN algorithm [14] and to reduce the computational complexity of employing similarity measures that are expensive to evaluate.

C. Measures of Similarity

Similarity between two trajectories, each defined via a sequence of apses, is assessed using either an isochronous or a normal correspondence. An isochronous correspondence compares two trajectories via time-ordered sequences that depend on the location of the initial state along a trajectory [34]. Alternatively, a normal correspondence leverages a comparison between two solutions via their closest points along the entire trajectories [34]. When applying Poincaré mapping to the analysis of a chaotic multi-body system for a variety of applications, either of these forms of comparison between two solutions may supply valuable insight. If the trajectory design application of interest requires an analysis of the fundamental geometries within the solution space, with a dependence on the initial condition, a distance metric based on an isochronous correspondence may supply a suitable measure of similarity between two trajectories: for instance, during initial guess construction of a trajectory with a constrained itinerary or initial state along a segment.

However, in a scenario where the trajectory designer is simply partitioning the complex solution space into clusters to identify fundamental motions, an isochronous correspondence between two trajectories may not be appropriate. In fact, clustering based on the similarity between trajectories via a time-dependent comparison may distribute a single type of fundamental motion (e.g., arcs that lie along a family of quasi-periodic orbits within a region of bounded motion) across multiple clusters, each corresponding to the trajectory starting at different locations along the solution. In this scenario, a distance metric that is formulated using a normal correspondence would supply a more suitable measure of similarity. Thus, two distance metrics are used in this paper to define the similarity between two solutions: the Euclidean distance, for an isochronous comparison between two trajectories, and a modified Hausdorff distance, to compare two entire solutions irrespective of the initial condition [19]. In this section, each of these distance metrics is defined and the implications for the performance and output of the clustering process is discussed.

The Euclidean distance measures the l_2 norm of the difference between two vectors, offering a straightforward and time-dependent comparison between two solutions. In the context of the trajectory summarization approach leveraged in this paper, the Euclidean distance, d_e, measured between trajectory i and trajectory j is calculated as:

$$d_e(\boldsymbol{T}_i, \boldsymbol{T}_j) = \sqrt{\sum_{k=1}^{2N_{peri}+1} (\boldsymbol{R}_{i,k} - \boldsymbol{R}_{j,k}) \cdot (\boldsymbol{R}_{i,k} - \boldsymbol{R}_{j,k})} \tag{6}$$

where the k-th apse along trajectory i is directly compared to the k-th apse along trajectory j for $k = [1, 2N_{peri} + 1]$. This distance quantity requires a relatively low computational effort to evaluate. Furthermore, the Python-based *hdbscan* clustering library leveraged in this analysis relies on the use of the triangle inequality to accelerate the performance of the HDBSCAN clustering algorithm [32]. Since the Euclidean distance satisfies this condition, this implementation of the HDBSCAN clustering algorithm can rapidly identify clusters within a large data set.

A modified Hausdorff distance enables a time-independent comparison between two trajectories and is often used in a variety of technical disciplines for shape matching [19]. Evaluation of this distance metric involves pairing each member of set i to the closest member of set j. In this paper, a modified and undirected Hausdorff distance, d_h, is calculated as:

$$d_h(\boldsymbol{T}_i, \boldsymbol{T}_j) = d_{h,d}(\boldsymbol{T}_i, \boldsymbol{T}_j) + d_{h,d}(\boldsymbol{T}_j, \boldsymbol{T}_i) \tag{7}$$

where

$$d_{h,d}(\boldsymbol{T}_i, \boldsymbol{T}_j) = \max_{k=[1,2N_{peri}+1]} \min_{l=[1,2N_{peri}+1]} \|\boldsymbol{R}_{i,k} - \boldsymbol{R}_{j,l}\| \tag{8}$$

While this distance metric enables a comparison that does not depend on the ordering of the apses along each trajectory, it is computationally expensive to incorporate into the clustering of an $N \times M$-dimensional data set when N is large.

To use the modified Hausdorff distance as a measure of similarity between two trajectories, a faster clustering procedure is developed by first prepartitioning the data set and then leveraging the cluster validity index, evaluated using

a modified Hausdorff distance, to merge similar clusters. Rapidly clustering the solution space using the Euclidean distance as a similarity measure enables an initial partitioning of the solution space into groups of trajectories with a similar geometry and similar initial conditions, with each cluster labeled Ei for $i = [0, ..., N_{cluster} - 1]$. Cluster Ei and cluster Ej are then examined to determine if they are composed of trajectories that complete the same number of apses prior to satisfying any of the termination conditions. To straightforwardly perform this comparison, recall the definition of the compressed description vector, $\boldsymbol{T}$: if a trajectory completes less than $2N_{peri} + 1$ apses prior to satisfying one of the termination conditions, a placeholder value for $\boldsymbol{R}_{i,j}$ is used instead of a description of the state at an apse. This placeholder value possesses only one nonzero element, ± 10, that is generally an order of magnitude higher than the components of $\boldsymbol{R}_{i,j}$ describing a true apse. Generally, if the Euclidean norm of the difference between the compressed description vectors for a trajectory from each cluster is less than ten, the two trajectories complete the same number of apses prior to satisfying any of the termination conditions. In this case, members of both clusters are compared to determine whether the clusters should be merged.

The cluster validity index, calculated using a subset of members from each clusters, is used in the merging step. While assessment of this cluster validity index requires evaluation of the modified Hausdorff distance, $N_{min,core}$ and $N_{min,cluster}$ do not need to be reselected. If either cluster Ei or cluster Ej possesses fewer than N_{cutoff} members, then all members of the corresponding cluster are used in this step. If, however, either cluster possesses more than N_{cutoff} members, then the associated cluster is subsampled to reduce the computational time: every $N_{step,k}$ member of the k-th cluster is used in this cluster comparison step, where $N_{step,k} = \text{floor}(N_k / N_{cutoff})$. In this subsampling step, the order of each member of the k-th cluster is consistent with their relative order in the original dataset. The value of N_{cutoff} used to define the subsampling process is selected to be larger than the value of $N_{min,cluster}$ and may be adjusted iteratively.

If the selected subset of the solutions in the two clusters are considered similar, as defined by the modified Hausdorff distance, the two clusters are merged. Specifically, clusters Ei and Ej are merged if the values $V_{C,i}$ and $V_{C,j}$ are both negative when evaluated using the modified Hausdorff distance, i.e., trajectories across both clusters are geometrically similar. This approach is verified to achieve a similar result in merging related clusters as reapplying the HDBSCAN algorithm to each pair of clusters – but with a lower computational complexity and without reselecting $N_{min,core}$ and $N_{min,cluster}$. This process is repeated for all unique pairs of clusters identified from the clustering of the original data set via the Euclidean distance as a similarity measure to produce a new set of clusters for the entire data set. The presented approach leverages both the speed of clustering via the Euclidean distance to prepartition the solution space and the cluster validity index to improve the efficiency of implementing a comparison scheme that employs the modified Hausdorff distance. As a result, this approach mitigates the challenges of using a similarity measure that is otherwise computationally expensive to directly apply to large data sets.

D. Extracting a Cluster Representative

To improve visualization and analysis of the clustering results, it is useful to identify a cluster representative, i.e., a solution that reflects the members associated with each cluster. Although there are a variety of possible definitions, the medoid of the cluster is leveraged in this paper as a representative solution. A medoid, also sometimes labeled a clustroid, is defined as the member of a cluster that is most similar to the other members of the cluster [35]. Mathematically, the trajectory associated with the medoid of the i-th cluster of N_i members is summarized as $\mathbf{T}_{m,i}$ and is computed using a distance metric, d, as:

$$\mathbf{T}_{m,i} = \text{argmin}_{T_{k,i} \in \{T_{1,i},\ldots,T_{N_i,i}\}} \left\{ \sum_{j=1}^{N_i} d(\mathbf{T}_{j,i}, \mathbf{T}_{k,i}) \right\} \tag{9}$$

Analysis of this expression reveals that the medoid of the cluster is sampled directly from the members of the cluster. Thus, the medoid is particularly useful for extracting a representative solution that corresponds to an actual trajectory in the dynamical model of interest; this result is especially significant when the cluster possesses an irregular or convex shape within a higher-dimensional space.

VI. Procedure for Clustering Map Data

Clustering the crossings on a Poincaré map via the geometry of their trajectories involves generating, processing and clustering the data and then examining the output. The general procedure used in this paper is summarized as follows:

1) *Define the map parameters* including the CR3BP system, desired value of C_J to constrain trajectories, and direction of motion for initial conditions.
2) *Seed initial conditions* as perigees occurring along planar trajectories between L_1 and L_2 at the desired value of the Jacobi constant and in the desired direction, while also possessing a distance of at least 10^{-6} nondimensional units from the smaller primary.
3) *Generate and summarize trajectories to produce the data to input to the clustering procedure.* For each initial condition, integrate forward in time until either: completing N_{peri} subsequent periapses, passing within a radius of 10^{-6} nondimensional units from the primary, or passing through either the L_1 or L_2 gateways. Summarize the i-th trajectory by constructing the $\boldsymbol{R}_{i,j}$ vectors for the j-th apse occurring prior to any of the termination conditions. Combine the $\boldsymbol{R}_{i,j}$ vectors to produce the compressed description vector, $\boldsymbol{T}_i$, for the i-th trajectory. Repeat this summarization procedure for all initial conditions and combine to form the data set, [**S**].
4) *Define the parameters governing the clustering algorithm,* including the: distance metric used to assess similarity, $N_{min,core}$, and $N_{min,cluster}$. The latter two parameters are selected via analysis of the DBCV index, the resulting number of clusters and the fraction of noise points.
5) *Run the clustering procedure.* The HDBSCAN algorithm is applied to the data set, [**S**]. If the similarity measure is defined using the Euclidean distance metric, this clustering is performed in a single step using the selected

values of $N_{min,core}$, and $N_{min,cluster}$. If, however, the similarity measure is defined using the modified Hausdorff distance, then the clustering is performed in two steps. First, the complete data set is partitioned using clustering performed with the Euclidean distance as a similarity measure. Then, a subset of members from each pair of clusters that are described by the same number of apses along the associated trajectories are compared using the cluster validity index evaluated with the modified Hausdorff distance as a similarity measure. If the cluster validity indices for both clusters are negative, then all members of the original pair of clusters are merged.

6) *Examine the output of the clustering procedure.* Any of the following output may be examined to assess the validity of the clustering results: the number of clusters, number of noise points, visualization of the cluster assignments, representative solutions from each cluster and dendrogram summarizing the clustering hierarchy.
7) *Repeat Steps 4-6 if needed.* Following analysis of the output from the HDBSCAN algorithm, the resulting clusters must be assessed by a human-in-the-loop. If needed, the parameters governing the algorithm may be adjusted and the clustering procedure repeated.

While there may be several options for formulating a data-driven framework to analyzing a periapsis map in the autonomous CR3BP, this approach is used as a proof of concept for planar solutions at a single value of C_J.

VII. Results and Analysis

The developed clustering approach is used to group Poincaré map crossings in the Sun-Earth CR3BP at a single value of the Jacobi constant based on the geometry of their associated planar trajectories. First, the data set $[\mathbf{S}]$ is constructed to consist of the map crossings associated with prograde perigees near the Earth at a Jacobi constant of $C_J = 3.00088$. Up to 400 equally-spaced values of the x-coordinate between L_1 and L_2 as well as up to 400 equally-spaced values of the y-coordinate between -0.01 and 0.01 nondimensional units are used to seed the set of initial perigees. From all possible combinations of x and y within these ranges that produce feasible perigees at $C_J = 3.00088$ located beyond a distance of 10^{-6} nondimensional units from the Earth, 31,367 initial perigees are identified. Then, the associated trajectories are integrated for up to $N_{peri} = 3$ subsequent periapses to produce a data set of dimension $31,367 \times 35$ via the definitions in Eq. (3); these trajectories are generated in 10.5 seconds using a C++ implementation within MATLAB® on a computer with a 3.6 GHz Intel Core i7 processor. This data set is input to the HDBSCAN clustering algorithm, accessed via the *hdbscan* library in Python [32].

Cluster validation techniques are employed to select the parameters $N_{min,core}$ and $N_{min,cluster}$ governing the HDBSCAN clustering algorithm and given the properties of the defined data set. Specifically, each of these parameters is varied within a suitable range of values: for $N_{min,core}$, used to calculate the mutual reachability distance, values between 1 and 200 with a step size of 2 are examined; for $N_{min,cluster}$, which corresponds to the minimum feasible cluster size and is dependent upon the sparseness of the data set, values between 10 and 200 with a step size of 5 are employed. For all possible combinations of $N_{min,core}$ and $N_{min,cluster}$ selected from these values, the HDBSCAN

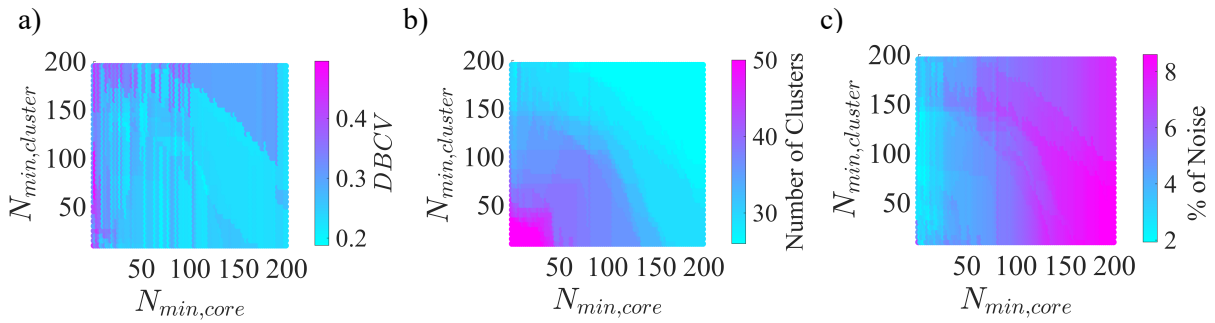

Fig. 3 Selecting $N_{min,core}$ and $N_{min,cluster}$ via the: a) DBCV index, b) recovered number of clusters and c) percentage of the data set labeled as noise.

algorithm is applied to the constructed data set using the Euclidean distance as a similarity metric. For each clustering result, the DBCV index, defined in Eq. (5), is evaluated along with the number of clusters identified by the HDBSCAN algorithm and the percentage of the data set that is labeled noise. These three properties of the clusters, produced for various values of $N_{min,core}$ and $N_{min,cluster}$, are used simultaneously to select a single combination of the parameters governing the HDBSCAN clustering algorithm; the results are plotted in Fig. 3. In each subfigure, the horizontal axis displays potential values of $N_{min,core}$, while the vertical axis corresponds to $N_{min,cluster}$. In Fig. 3a), the magenta colored region at $N_{min,core} = 3$ and encompassing the range $N_{min,cluster} = [60, 105]$, indicates a higher value of the DBCV index and, therefore, a better clustering result: for these values of the governing parameters, similar trajectories are better grouped, while dissimilar trajectories are better separated. Across this combination of values for $N_{min,core}$ and $N_{min,cluster}$, Fig. 3b) indicates the recovery of between 33 and 38 clusters, while Fig. 3c) indicates that 1.93-2.42% of the data set is labeled as noise after clustering. Although any combination of $N_{min,core}$ and $N_{min,cluster}$ in the identified range could produce a reasonable clustering result that is useful to the human analyst, the following values for the governing parameters within the identified range are selected in this paper: $N_{min,core} = 3$ and $N_{min,cluster} = 100$. This combination of parameters produces 34 clusters and tends to correspond to a group of representative solutions that are distinct, while 2.2% of the data set is labeled as noise, which is close to the minimum value across the entire range of values analyzed for $N_{min,core}$ and $N_{min,cluster}$.

The procedure outlined in this paper is used to cluster the trajectories associated with the constructed prograde periapsis map near the Earth vicinity in the Sun-Earth planar CR3BP using the selected parameters and the Euclidean distance as a similarity metric. With the selected parameterization and trajectory summarization strategy, the outlined clustering procedure uncovers 34 clusters with only 693 of the 31,367 data points labeled as noise in 3.25 seconds on a computer with a 3.6 GHz Intel Core i7 processor. The result of this clustering procedure is displayed in Fig. 4 with the initial conditions associated with each trajectory, integrated for up to three subsequent returns to the map, uniquely colored by their cluster and labeled by their cluster identifier, E0 to E33, located near the cluster medoid. The prefix "E" in the cluster labels indicates that the clustering is performed using the Euclidean distance as a similarity metric. In

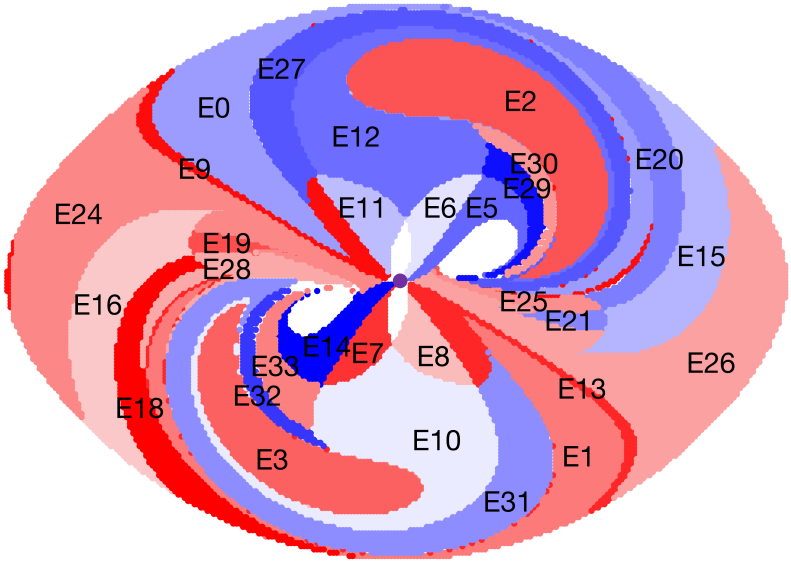

Fig. 4 Zoomed-in view of the periapsis map in the Sun-Earth system at $C_J = 3.00088$, partitioned into 34 clusters using the Euclidean distance as a similarity metric.

addition, noise points are not plotted on this figure. The Earth is depicted as a purple circle. Note that this view is zoomed-in to the main region of the map crossings for visual clarity and does not capture the following four clusters of crossings near the L_1 and L_2 gateways: E4, E17, E22, E23. Analysis of this figure, and comparison to Fig. 2 reveals that, at a minimum, the clustering process can separate the regions bound by the first few crossings of the stable manifolds of the L_1 and L_2 Lyapunov orbits – without a priori knowledge of these manifold structures. Furthermore, additional differentiation between trajectories and their geometry appears: subdivisions occur within these regions as trajectories exhibit distinct characteristics. Note, however, that approximately 693 trajectories are labeled by HDBSCAN as noise and do not appear in Fig. 4. This designation is likely due to the location of these trajectories in sparse regions of the 35-dimensional space associated with the compressed description vector. Further investigation into whether the number of noise points may be reduced further is warranted. Possible solutions may include: a nonuniform discretization of the initial condition set, particularly near the Earth where the majority of the trajectories designated as noise originate; modifications to the compressed description vector; or the use of soft clustering to probabilistically assign each member of the noise set to an existing cluster.

For further insight into the capability of the outlined clustering approach to differentiate trajectories of distinct geometries on the periapsis map, an analysis of the solutions associated with each cluster is useful. To aid visualization and interpretation, one representative solution from each of the clusters plotted in Fig. 4 is extracted via the medoid of the cluster. These solutions are plotted in Fig. 5 along with the associated cluster label and a blue circle indicating the location of the initial condition. All representative solutions in this set initially revolve around the Earth in a prograde manner since the initial conditions are defined as prograde; however, along some solutions, the direction of motion does

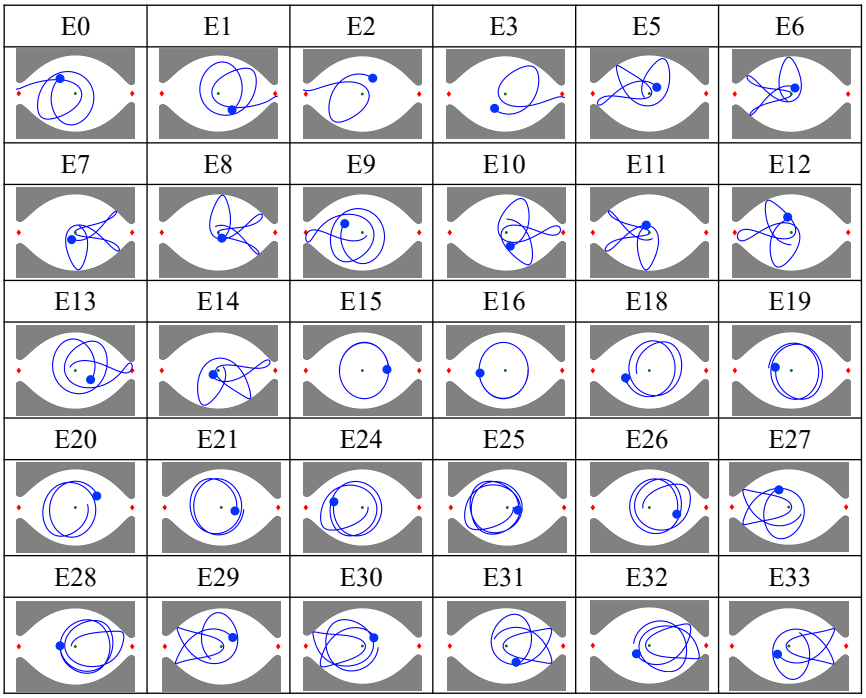

Fig. 5 Representative trajectories from each cluster in Fig. 4 plotted in the rotating frame for up to three revolutions around the Earth.

change temporarily. The L_1 and L_2 equilibrium points are plotted as red diamonds, the Earth is depicted by a small green circle and the forbidden regions are shaded in gray. Analysis of Fig. 4 and Fig. 5 reveals that, in general, solutions of various geometries are separated into distinctly different clusters. For solutions with a similar geometry, such as those in clusters E15 and E16, the clusters are considered separate and distinct based on the initial condition. Of course, since the Euclidean distance is used to determine similarity between two trajectories, such a result is expected. As a supplement to the reduced set of representative solutions in Fig. 5, each cluster is analyzed visually in this preliminary analysis to verify that the solutions across a single cluster exhibit a similar geometry. For instance, consider cluster E15, which is composed of trajectories that encircle the Earth in an entirely prograde manner. This cluster is isolated and the map crossings are plotted in gray in Fig. 6a); selected map crossings are overlaid as uniquely-colored circles. Then, the trajectory associated with each of the selected map crossings is propagated for three returns to the hyperplane and plotted with the same color assignment in Fig. 6b). Analysis of this figure reveals that across cluster E15, the trajectories all exhibit a similar geometry with only a small apsidal rotation between revolutions. In fact, similar solutions, as defined using an isochronous correspondence, are generally grouped within the same cluster. Thus, the representative reduced data set in Fig. 5 offers a straightforward summary of the distinct and finite number of geometries exhibited by solutions that are captured by the Poincaré map depicted in Fig. 4. This insight, derived without the need for explicit analytical expressions or parameterizations of solution geometry, enables a human analyst to rapidly assess and visualize the solution space for use in the trajectory design process.

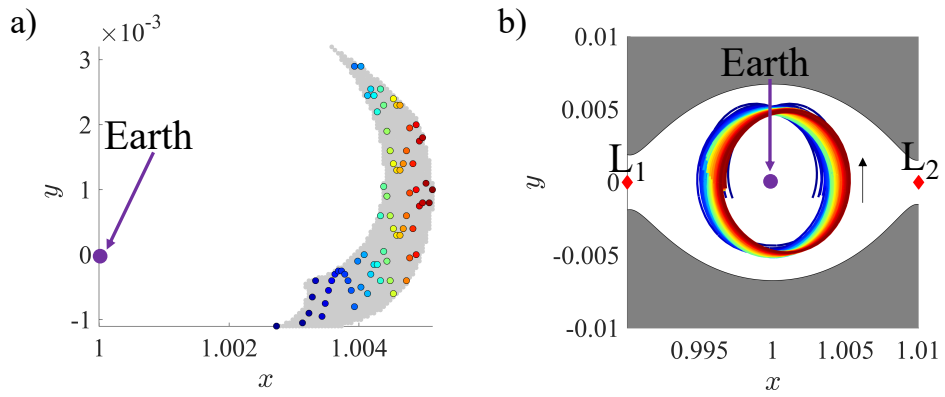

Fig. 6 Cluster E15: a) selected map crossings sampled across the cluster and b) associated trajectories plotted in configuration space.

An alternative approach to grouping the crossings of a Poincaré map by the geometry of the associated trajectories is to use the modified Hausdorff distance to assess similarity via a normal correspondence. Since the modified Hausdorff distance is computationally expensive to evaluate, clustering is performed using prepartioning, as outlined in the algorithmic overview in Section VI. First, rapid clustering is performed using the Euclidean distance as a similarity metric, producing the results displayed in Figs. 4 and 5. Then, pairs of clusters are compared using the cluster validity index evaluated across a subset of 100-300 solutions from both clusters using the Hausdorff distance; this evaluation requires a computational time on the order of seconds for each pair of clusters. If the cluster validity indices are negative for both clusters, then the two clusters are merged. Once this process has been repeated for all possible pairs of clusters of trajectories completing a consistent number of apses prior to termination, the results are displayed in Fig. 7. In this figure, the map crossings are uniquely colored by their cluster and labeled by their cluster identifier. The prefix "H" in the

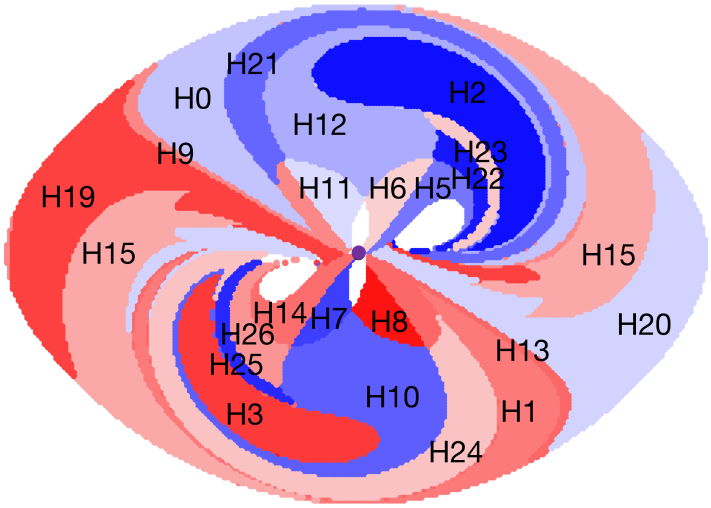

Fig. 7 Zoomed-in view of the periapsis map constructed in the Sun-Earth system at $C_J = 3.00088$, partitioned into 27 clusters using a modified Hausdorff distance as a similarity metric.

cluster labels indicates that the modified Hausdorff distance is used as a similarity metric. The associated representative trajectories for each of the 27 clusters are displayed in Fig. 8 using a configuration that is consistent with Fig. 5, the clusters that are merged are listed in brackets in Fig. 8. Note that the following clusters and their representatives are not displayed as they occur outside of the zoomed-in region and close to the L_1 and L_2 gateways: H4, H16, H17, and H18. Analysis of these figures reveals that several clusters from Fig. 4 are merged to produce the results in Fig. 7. When integrated for three returns to the hyperplane, the trajectories in clusters E15 and E16 tend to correspond to the same prograde motions around the Earth, but with different initial conditions. Clusters E18, E19, E20 and E21 tend to exhibit a similar general geometry over this finite time interval but with a larger apsidal rotation. When integrated for more than three revolutions around the Earth, trajectories in these clusters would eventually appear distinctly different from the solutions in E15 and E16. However, when propagated for only three returns to the periapsis hyperplane, these six clusters – E15, E16, E18, E19, E20 and E21 – appear similar enough to be merged into a single group, H15, when using the modified Hausdorff distance as a similarity metric. Similarly, clusters H19 and H20 are each formed by merging two clusters from Fig. 4. These results of the clustering procedure using the modified Hausdorff distance as a similarity metric, displayed in Fig. 7 and Fig. 8, may be useful to a trajectory designer analyzing the fundamental geometries underlying the solution space or determining the specific accessibility of a solution geometry from various regions within the Earth vicinity. Furthermore, these results suggest that a hierarchical and density-based clustering approach successfully organizes map crossings into clusters based on the geometry of the associated trajectories – regardless of whether the trajectory designer has defined geometry as dependent or independent of the initial condition.

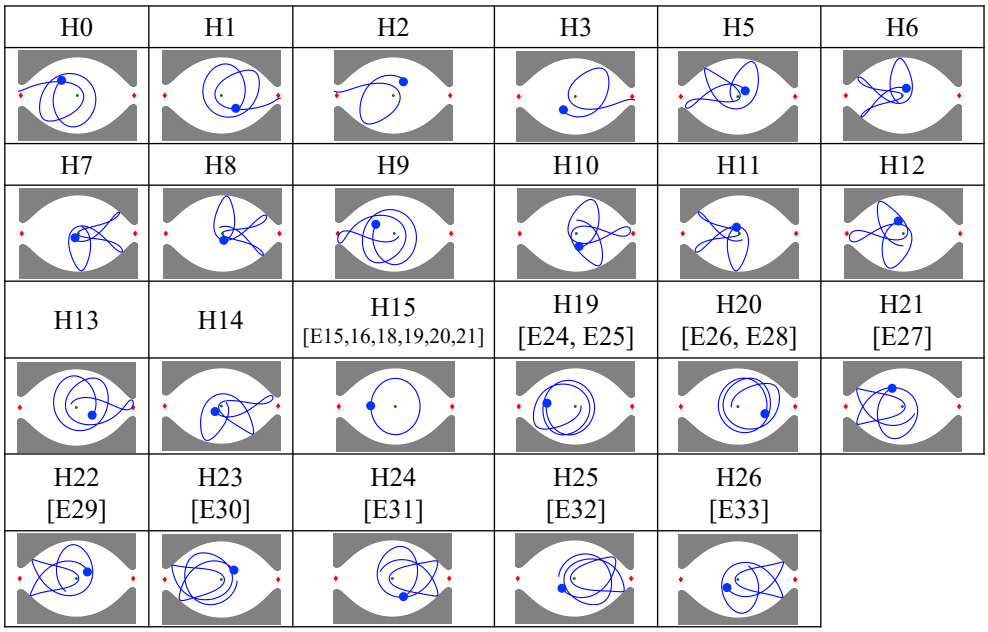

Fig. 8 Representative trajectories from each cluster in Fig. 7 plotted in the rotating frame for up to three revolutions around the Earth.

VIII. Conclusion

A hierarchical and density-based clustering algorithm, HDBSCAN, is employed to implement an unsupervised clustering of map crossings on a general Poincaré map based on the geometry of the associated trajectories. As opposed to other clustering algorithms, this particular method is selected due its ability to accommodate the properties of the data generated via Poincaré mapping: a number of clusters that is not known a priori; clusters of various shapes; clusters of various densities; and an unknown or nonconstant value of the maximum separation between data points within a single cluster in a higher-dimensional space. Furthermore, data is generated by associating each map crossing with its trajectory integrated for multiple revolutions around the Earth and summarizing the solution via the location of each periapsis and apoapsis, as well as the associated epoch and direction of motion. The parameters governing the HDBSCAN algorithm are then selected using cluster validation techniques. Next, solutions are considered similar if the vectors describing the finite set of apses are close, as calculated using either the Euclidean distance or a modified Hausdorff distance to formulate a similarity measure. HDBSCAN, implemented in Python via the *hdbscan* library, is used to cluster the map crossings based on the geometry of the solutions. This clustering algorithm requires a computational time on the order of seconds when applied to the constructed Poincaré map data set and using the Euclidean distance as a similarity metric. Since the modified Hausdorff distance requires a higher computational time to evaluate than the Euclidean distance, clustering via the modified Hausdorff distance is accomplished by prepartitioning the data and then merging clusters of similar solutions. Specifically, the clusters identified with the Euclidean distance as a similarity metric are used to prepartition the data set into a finite number of groups. Then, pairs of these initial clusters, subsampled to produce a sufficiently representative set of members, are compared and merged via a cluster validity index.

The presented approach is applied to a planar periapsis map constructed in the Sun-Earth CR3BP at a single value of the Jacobi constant. The outputs of this clustering approach are: 1) maps with individual clusters indicated by distinct colors and each cluster sufficiently capturing only solutions of similar geometry, either dependent or independent of the initial condition based on the selected similarity measure; and 2) a reduced set of representative solutions summarizing the distinct geometries associated with each cluster. Using a clustering algorithm to identify the fundamental geometries of the associated planar solutions eliminates the need to define analytical expressions for separation or to perform a manual grouping. In fact, this automated and unsupervised approach may enable a trajectory designer to rapidly assess and visualize the underlying solution space for use in more complex tasks such as initial guess construction. Furthermore, the presented approach offers both a proof of concept and a foundation for future applications of clustering to maps with higher-dimensional data and for dynamical models of increased complexity.

Acknowledgements

This work was completed at the University of Colorado Boulder under NASA Grant 80NSSC18K1536. The author thanks each of the anonymous reviewers for their feedback.

References

[1] Paskowitz, M.E.; Scheeres, D.J., "Robust Capture and Transfer Trajectories for Planetary Satellite Orbiters," *Journal of Guidance, Control and Dynamics*, Vol. 29, No. 2, 2006, pp. 342-353.
doi: 10.2514/1.13761

[2] Craig Davis, D.; Phillips, S.M.; McCarthy, B.P., "Trajectory Design for Saturnian Ocean Worlds Orbiters Using Multidimensional Poincaré Maps," *Acta Astronautica*, Vol. 143, 2018, pp. 16-28.
doi: 10.1016/j.actaastro.2017.11.004

[3] Bosanac, N, "Leveraging Natural Dynamical Structures to Explore Multi-Body Systems," PhD Dissertation, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 2016.

[4] Contopoulos, G, *Order and Chaos in Dynamical Astronomy*, Germany: Springer-Verlag, 2002, pp. 126-139, 224-227.

[5] Schlei, W; Howell, K.C.; Tricoche, X.M.; Garth, C, "Enhanced Visualization and Autonomous Extraction of Poincaré Map Topology," *The Journal of the Astronautical Sciences*, Vol. 61, 2014, pp. 170-197.
doi: 10.1007/s40295-015-0042-4

[6] Craig Davis, D., "Multi-Body Trajectory Design Strategies Based on Periapsis Poincaré Maps," PhD Dissertation, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 2011.

[7] Haapala, A.F., "Trajectory Design in the Spatial Circular Restricted Three-Body Problem Exploiting Higher-Dimensional Poincaré Maps," PhD Dissertation, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 2014.

[8] Howell, K.C.; Craig Davis, D.; Haapala, A.F., "Application of Periapse Maps for the Design of Trajectories Near the Smaller Primary in Multi-Body Regimes," *Mathematical Problems in Engineering*, Vol. 2012, 2011.
doi: 10.1155/2012/351759

[9] Bosanac, N.; Cox, A.D.; Howell, K.C.; Folta, D.C., "Trajectory Design for a Cislunar CubeSat Leveraging Dynamical Systems Techniques: the Lunar IceCube Mission," *Acta Astronautica* Vol. 144, 2018, pp 283-296.
doi: 10.1016/j.actaastro.2017.12.025

[10] Parker, J.S.; Anderson, R.L., *Low-Energy Lunar Trajectory Design*, Hoboken, New Jersey: John Wiley & Sons, Inc, 2014, pp 103-106.

[11] Aggarwal, C.C.; Reddy, C.K., *Data Clustering: Algorithms and Applications,* Chapman and Hall CRC, 2018, Ch. 2.1.2, 3, 9.2.

[12] Joncour, I., Duchêne, G., Moraux, E., Motte, F., "Multiplicity and Clustering in Taurus Star Forming Region II. From Ultra-Wide Pairs to Dense NESTs," *Astronomy and Astrophysics*, Vol. 620, 2018, A27.
doi: 10.1051/0004-6361/201833042

[13] McLachlan, G.J., “Cluster Analysis and Related Techniques in Medical Research," *Statistics Methods in Medical Research*, Vol. 1, 1992, pp. 27-48.
doi: 10.1177/096228029200100103

[14] Gallego, C.E.V.; Gómez Comendador, V.F.; Saez Nieto, F.J.; Martinez, M.G., “Discussion of Density-Based Clustering Methods Applied for Automated Identification of Airspace Flows," *37th Digital Avionics Systems Conference*, London, England, 2018.
doi: 10.1109/DASC.2018.8569219

[15] Nakhjiri, N; Villac, B., “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.
doi: 10.1007/s10569-015-9629-0

[16] Hadjighasem, A; Karrasch, D; Teramoto, H; Haller, G, “Spectral-Clustering Approach to Lagrangian Vortex Detection," *Physical Review E*, Vol. 93, 2016, No. 6.
doi: 10.1103/PhysRevE.93.063107

[17] Villac, B.F.; Anderson, R.L.; Pini, A.J., “Computer Aided Ballistic Orbit Classification Around Small Bodies," *The Journal of the Astronautical Sciences*, Vol. 63, No. 3, 2016, pp. 175-205.
doi: 10.1007/s40295-016-0089-x

[18] Zheng, Y, “Trajectory Data Mining: An Overview," *ACM Transactions on Intelligent Systems and Technology*, Vol. 6, No. 3, 2015, Article 29.
doi: 10.1145/2743025

[19] Dubuisson, M.-P.; Jain, A.K., “A Modified Hausdorff Distance for Object Matching," *Proceedings of the 12th International Conference on Pattern Recognition* Jerusalem, Israel, 1994.
doi: 10.1109/ICPR.1994.576361

[20] Campello, R.J.G.B.; Moulavi, D; Sander, J, “Density-Based Clustering Based on Hierarchical Density Estimates," In: Pei J., Tseng V.S., Cao L., Motoda H., Xu G. (eds) *Advances in Knowledge Discovery and Data Mining. PAKDD 2013. Lecture Notes in Computer Science*, Vol. 7819, Springer, Berlin, Heidelberg, 2013.
doi: 10.1007/978-3-642-37456-2_14

[21] Campello, R.J.G.B.; Moulavi, D; Zimek, A; Sander, J, “Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection," *ACM Transactions on Knowledge Discovery from Data* Vol. 10, No. 1, 2015.
doi: 10.1145/2733381

[22] Szebehely, V, *Theory of Orbits: The Restricted Problem of Three Bodies*, London, United Kingdom: Academic Press, 1967, Ch. 1.

[23] Perko, L, *Differential Equations and Dynamical Systems, Second Edition* New York: Springer-Verlag, 1996, pp. 209-217.

[24] Gómez, G; Mondelo, J.M, “The Dynamics Around the Collinear Equilibrium Points of the RTBP," *Physica D: Nonlinear Phenomenon*, Vol. 157, No. 4, 2001, pp. 283-321.
doi: 10.1016/S0167-2789(01)00312-8

[25] Olikara, Z.P, “Computation of Quasi-Periodic Tori and Heteroclinic Connections in Astrodynamics Using Collocation Techniques," Ph.D. Dissertation, Aerospace Engineering Sciences, University of Colorado Boulder, Boulder, Colorado, 2016.

[26] Villac, B.F.; Scheeres, D.J., “On the Concept of Periapsis in Hill’s Problem," *Celestial Mechanics and Dynamical Astronomy* Vol. 90, 2004, pp. 165-178.
doi: 10.1007/s10569-004-0405-9

[27] Haapala, A.F., “Trajectory Design Using Periapse Maps and Invariant Manifolds," MS Thesis, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 2010.

[28] Szebehely, V.G., “Zero Velocity Curves and Orbits in the Restricted Problem of Three Bodies," *The Astronomical Journal*, Vol. 68, No. 3, 1963, pp. 147-151.
doi: 10.1086/108931

[29] Zheng, Y.; Zhou, X., *Computing with Spatial Trajectories*, New York, New York: Springer Science & Business Media, 2011, Ch. 1, 5.

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

[31] Moulavi, D.; Jaskowiak, P.A.; Campello, R.J.G.B.; Zimek, A.; Sander, J, “Density-Based Cluster Validation," *Proceedings of the 2014 SIAM International Conference on Data Mining*, Philadelphia, Pennsylvania, 2014.
doi: 10.1137/1.9781611973440.96

[32] McInnes, L; Healy, J; Astels, S; “hdbscan: Hierarchical Density Based Clustering," *Journal of Open Source Software, The Open Journal*, Vol. 2, No. 11, 2017.
doi: 10.21105/joss.00205

[33] McInnes, L; Healy, J, “Accelerated Hierarchical Density Based Clustering," *2017 IEEE International Conference on Data Mining Workshops*, New Orleans, Louisiana, 2017.
doi: 10.1109/ICDMW.2017.12

[34] Szebehely, V., “Review of Concepts of Stability," *Celestial Mechanics*, Vol. 34, 1984, pp. 49-64.
doi: 10.1007/BF01235791

[35] Cichosz, P., *Data Mining Algorithms: Explained Using R*, West Sussex, United Kingdom: John Wiley and Sons, 2015, pp. 339-341.