(Preprint) AAS 25-646

ADAPTIVE DATA-DRIVEN SUMMARY OF NATURAL MOTION IN CISLUNAR SPACE

Miguel Rebelo[*], Natasha Bosanac[†]

This paper introduces an adaptive clustering framework for summarizing planar trajectories by their geometry in the Earth-Moon circular restricted three-body problem. First, a coarse grid of initial states is defined using the hypersurface formed by maxima in the curvature along a trajectory. Their trajectories are generated and clustered by geometry. New initial conditions are then sampled to balance 1) global exploration, using the curvature of the hypersurface, and 2) local exploitation, using the change in the total absolute curvature along trajectories and null cluster assignments. This process is repeated to adaptively generate a data-driven summary of a complex solution space.

INTRODUCTION

Planning, operating, and protecting a diverse array of spacecraft in cislunar space will benefit from a comprehensive understanding of possible motions. However, in a multi-body gravitational environment, an analytical solution does not exist and the solution space is chaotic, diverse, and high-dimensional. These characteristics can challenge manual analysis and design tasks, or demand sufficient a priori expertise to bound the analysis. These challenges are exacerbated when using higher-fidelity dynamical models or considering motion across a large array of energies with diverse control profiles. To address these challenges, data-driven approaches have the capacity to automatically summarize the solution space in support of trajectory design and prediction.

Data mining methods, in particular, have found success across multiple disciplines that encounter complex and high-dimensional datasets. One popular tool is clustering, an unsupervised technique used to partition a dataset into smaller groupings based on their similarity.[1] For example, in air traffic management, Gallego et al.[2] used density-based clustering to extract primary air traffic flows from a large dataset whereas in biology, Paccanaro et al.[3] devised a spectral clustering scheme to identify proteins which share a common evolutionary origin. Other notable applications of clustering for analysis and knowledge discovery have occurred in fields such as astronomy, medicine and human movement analysis.[4–6] In astrodynamics, clustering has been used by Nakhjiri and Villac to detect regions of stability near distant retrograde orbits on a Poincaré map[7] as well as by Villac, Anderson and Pini.[8] Smith and Bosanac[9] as well as Gillespie, Miceli, and Bosanac[10] have also used clustering to extract motion primitive sets as summaries of periodic orbits families, their hyperbolic invariant manifolds, and thrust-enabled approaches and departures.

[*]Graduate Research Assistant, Colorado Center for Astrodynamics Research, Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, Boulder, CO, 80303.
[†]Associate Professor, Colorado Center for Astrodynamics Research, Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, Boulder, CO, 80303.

Recently, clustering has been used to automatically summarize the prominent geometries of a diverse array of trajectories across a multi-body system. Early proofs of concepts were developed by Bosanac[11] followed by Bonasera and Bosanac[12] to summarize key geometries of trajectories that begin from perigees in the Sun-Earth circular restricted three body problem (CR3BP). Since then, Bosanac has developed an improved framework that has been applied to natural and low-thrust trajectories in the Earth-Moon CR3BP[13,14] and an ephemeris model of cislunar space.[15] This implementation consists of three major steps: 1) constructing a training dataset by sparsely sampling trajectories across the solution space using differential geometry and a predefined grid of initial position vectors and energy levels, 2) describing the diverse array of continuous trajectories with a finite-dimensional feature vector that consistently captures their geometry, and 3) using distributed clustering to group trajectories by their geometry. Representative members of the resulting clusters then supply a summary of a diverse and complex solution space. This paper builds upon this clustering framework by improving the generation of the training dataset and reducing the dependence on predefined parameters for discretizing the solution space.

This paper introduces an adaptive and automated scheme for sampling the trajectories that form the training dataset for the clustering-based summarization framework. First, a coarse grid of initial state vectors is sampled from the hypersurface formed by maxima in the curvature along a trajectory. Then, the trajectories associated with these initial conditions are generated, characterized, and clustered by their geometry using the framework developed by Bosanac.[15] This information is used to automatically refine the grid of initial state vectors in a manner that balances global exploration with local exploitation. To support global coverage of the solution space, concepts from shape interrogation are used to place more samples in regions of high curvature along the hypersurface that is used to define the initial conditions. In addition, to encourage local exploitation, samples are added 1) between neighboring trajectories that possess a significant change in the total absolute curvature along their paths and 2) near trajectories that are not assigned to a cluster. This process is repeated to produce a training dataset that better captures the diverse array of trajectory geometries across a chaotic multi-body system and, as a result, supports a more accurate clustering-based summarization of the prominent geometries. This original contribution is demonstrated by adaptively sampling and summarizing planar trajectories across a single energy level in the Earth-Moon CR3BP.

BACKGROUND

Circular Restricted Three-Body Problem

The dynamics of a spacecraft in cislunar space are approximated by the CR3BP, with the Earth serving as the primary body and the Moon as the secondary. These two primaries are modeled using spherically symmetric gravity fields and are assumed to travel on circular orbits about their barycenter.[16] The spacecraft is the third body, with a negligible mass when compared to the primaries.[16]

States are typically expressed in nondimensional form in the Earth-Moon rotating frame. The characteristic quantities used to normalize length, mass, and time quantities are defined as follows: the characteristic length $l^* = 384,400$ km is selected as the average distance between the primaries; the characteristic mass is the total mass of the system, $m^* = 6.045626 \times 10^{24}$ kg; and the characteristic time $t^* = 3.751903 \times 10^5$ sec, sets the mean motion of the primaries to unity.[17,18] In addition, the Earth-Moon rotating frame is centered at the system barycenter with the axes $\hat{x}\hat{y}\hat{z}$ defined as follows: $\hat{x}$ points towards P_2; $\hat{z}$ is aligned with the primaries' orbital angular momenta; and $\hat{y}$ completes the orthogonal, right-hand triad.[16] The nondimensional state vector is then defined in the rotating frame as $\boldsymbol{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]$.

Using these definitions, the equations of motion for the spacecraft in the CR3BP are written in the rotating frame. These second-order differential equations are equal to[16]

$$\ddot{x} = 2\dot{y} + U_x \qquad \ddot{y} = -2\dot{x} + U_y \qquad \ddot{z} = U_z \tag{1}$$

Where $U(x, y, z)$ is the pseudo-potential function, equal to

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

and U_i is its partial derivative with respect to the ith position coordinate. In addition, $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$ are the distances from the spacecraft to the Earth and Moon, respectively, and $\mu \approx 0.012151$ is the mass ratio. The CR3BP possesses one integral of motion,[16] the energy-like Jacobi Constant $C_J = 2U - v^2$, where $v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$ is the speed of the spacecraft. Furthermore, there are five equilibrium points, L_1 through L_5.[16]

Fast Lyapunov Indicators

In dynamical systems theory, chaos indicators are typically used to distinguish between regular and chaotic motion.[19] Moreover, they provide a quantitative measure of how rapidly nearby trajectories diverge, which can be interpreted as the strength of chaos in the system near a given initial condition.[19] In the astrodynamics community Froeschlé, Gonczi and Lega[20] used a special class of these indicators, the Fast Lyapunov Indicator (FLI), to study the stability of asteroids' orbits for finite time horizons. Later, Nakhjiri and Villac demonstrated that the FLI is also useful for refining a grid of initial conditions to identify trajectories that are bounded near a distant retrograde orbit.[7] For a trajectory generated from an initial state vector $\boldsymbol{x}_0$ at t_0 and perturbed by a small vector $\delta\boldsymbol{x}_0(t)$, the FLI is calculated over a propagation time τ as[21,22]

$$\text{FLI}(\boldsymbol{x}_0, \delta\boldsymbol{x}_0; \tau) = \max_{t_0 \le t \le \tau} \log_{10} \frac{||\delta\boldsymbol{x}(t)||}{||\delta\boldsymbol{x}_0||}$$

where $\delta\boldsymbol{x}(t)$ is the perturbation from the reference trajectory associated with $\boldsymbol{x}_0$ at time t. This quantity can be computed from the variational equations derived from Equations (1), together with the corresponding state transition matrix, $\boldsymbol{\Phi}(t, t_0)$, as $\delta\boldsymbol{x}(t) = \boldsymbol{\Phi}(t, t_0)\,\delta\boldsymbol{x}_0$. The FLI definition is often expanded to capture the maximum possible growth of a generic initial perturbation to the initial condition as[21,22]

$$\text{FLI}_T(\boldsymbol{x}_0, \{\delta\boldsymbol{x}_0^i\}; \tau) = \sup_i \ \text{FLI}(\boldsymbol{x}_0, \delta\boldsymbol{x}_0^i; \tau) \tag{3}$$

where $\{\delta\boldsymbol{x}_0^i\}$ is a set of initial perturbation vectors, selected in this paper as unit vector perturbations in each dimension of the phase space.[22]

Differential Geometry for Curves

Differential geometry is used to mathematically describe and computationally discretize curved trajectories in a geometrically aware manner. Consider a spatial trajectory (i.e., a curve $\mathcal{C}$) that is generated over a time interval $t \in [t_0, t_f]$ and described by its position $\boldsymbol{r}(t) = [x(t), y(t), z(t)]^{\mathrm{T}}$, velocity $\dot{\boldsymbol{r}}(t) = [\dot{x}(t), \dot{y}(t), \dot{z}(t)]^{\mathrm{T}}$ and acceleration $\ddot{\boldsymbol{r}}(t) = [\ddot{x}(t), \ddot{y}(t), \ddot{z}(t)]^{\mathrm{T}}$ vectors. The arclength of this curve, s, is the distance traveled along its path, equal to[23]

$$s = \int_{s_0}^{s_f} ds = \int_{t_0}^{t_f} \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}\, dt \tag{4}$$

The instantaneous unsigned curvature, $\kappa_{\mathcal{C}}(t)$, measures the rate of rotation of the velocity vector with respect to the arclength, i.e., the extent to which the trajectory deviates from a straight line, at a time t. This scalar curvature is calculated as[24]

$$\kappa_{\mathcal{C}}(t) = \frac{||\dot{\boldsymbol{r}}(t) \times \ddot{\boldsymbol{r}}(t)||}{||\dot{\boldsymbol{r}}(t)||^3} \tag{5}$$

and possesses a singularity when the speed is zero; the subscript '$\mathcal{C}$' emphasizes that this quantity is calculated along a curve. The total curvature of the trajectory is calculated using the integral of the curvature along its arclength as[25]

$$\kappa_{\mathcal{C},tot}(t_0, t_f) = \int_{s_0}^{s_f} \kappa_{\mathcal{C}}(s) ds = \int_{t_0}^{t_f} \kappa_{\mathcal{C}}(t) \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} dt \tag{6}$$

and captures the angle traced out by the path within its evolving osculating plane.

Differential Geometry for Surfaces

The concept of curvature is useful for describing the geometric properties of a surface near a specified point. Consider a general surface in three dimensional space, $\mathcal{S} \subset \mathbb{R}^3$, implicitly described by the equation $f(x_1, x_2, x_3) = 0$, where f is assumed to be differentiable. At any point $\boldsymbol{p} \in \mathcal{S}$, the normal vector $\hat{\boldsymbol{n}}_{\boldsymbol{p}}$ to the surface is calculated using the vector gradient of f, $\boldsymbol{g}_{\boldsymbol{p}} = \boldsymbol{\nabla} f(\boldsymbol{p})$, as[24]

$$\hat{\boldsymbol{n}}_{\boldsymbol{p}} = \frac{\boldsymbol{g}_{\boldsymbol{p}}}{||\boldsymbol{g}_{\boldsymbol{p}}||} \tag{7}$$

This vector locally orients the surface and is perpendicular to the tangent plane $T_{\boldsymbol{p}}$.

The curvature of the surface at a point measures the local deviation from the tangent plane.[24] Consider an arbitrary direction in the tangent plane $\hat{\boldsymbol{u}} \in T_{\boldsymbol{p}}$. A plane M is then formed using the basis vectors $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{n}}_{\boldsymbol{p}}$. As depicted in black in Figure 1, the intersection of M, colored blue, with the surface $\mathcal{S}$, colored gray, is a curve. The normal curvature is defined as the component of the curvature of this curve along $\hat{\boldsymbol{n}}$ and denoted as $\mathcal{K}_{\mathcal{S}}$, where the subscript '$\mathcal{S}$' emphasizes that this quantity corresponds to a surface. This quantity captures the surface curvature in the $\hat{\boldsymbol{u}}$ direction.

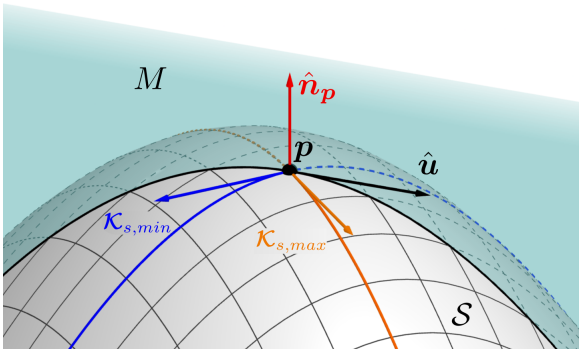

Figure 1. Conceptual definition of variables used to describe the curvature of a surface $\mathcal{S}$ near a point p.

The shape operator mathematically encodes the evolution of the normal vector and, therefore, the tangent plane when stepping away from a specified point along a specified direction.[24] This shape operator $S_{\boldsymbol{p}}$ is calculated using the directional derivative of $\hat{\boldsymbol{n}}$ along $\hat{\boldsymbol{u}}$ as[24]

$$S_{\boldsymbol{p}}(\hat{\boldsymbol{u}}) = -\boldsymbol{\nabla}_{\boldsymbol{u}}\hat{\boldsymbol{n}} = -\boldsymbol{\nabla}\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{u}} = -\left(\boldsymbol{\nabla}\hat{\boldsymbol{n}}\right)^{\mathrm{T}} \hat{\boldsymbol{u}} \tag{8}$$

The gradient of the normal vector is calculated as[26]

$$\begin{aligned} \boldsymbol{\nabla}\hat{\boldsymbol{n}} &= \boldsymbol{\nabla}\left(\frac{\boldsymbol{g_p}}{||\boldsymbol{g_p}||}\right) = \left(\frac{\boldsymbol{\nabla}\boldsymbol{g_p}}{||\boldsymbol{g_p}||} - \frac{\boldsymbol{\nabla}||\boldsymbol{g_p}||\ \boldsymbol{g_p}^{\mathrm{T}}}{||\boldsymbol{g_p}||^2}\right) = \frac{1}{||\boldsymbol{g_p}||}\left(\boldsymbol{H_p} - \frac{\boldsymbol{\nabla}(\boldsymbol{g_p}^{\mathrm{T}}\boldsymbol{g_p})^{1/2}\ \boldsymbol{g_p}^{\mathrm{T}}}{||\boldsymbol{g_p}||}\right) \\ &= \frac{1}{||\boldsymbol{g_p}||}\left(\boldsymbol{H_p} - \frac{\boldsymbol{H_p}\boldsymbol{g_p}\boldsymbol{g_p}^{\mathrm{T}}}{||\boldsymbol{g_p}||^2}\right) = \frac{1}{||\boldsymbol{g_p}||}\boldsymbol{H_p}\left(\boldsymbol{I} - \frac{\boldsymbol{g_p}\boldsymbol{g_p}^{\mathrm{T}}}{||\boldsymbol{g_p}||^2}\right) = \frac{1}{||\boldsymbol{g_p}||}\boldsymbol{H_p}\boldsymbol{P_p} \end{aligned} \tag{9}$$

where $\boldsymbol{H_p} = \boldsymbol{\nabla}^2 f$ is the Hessian of f calculated at $\boldsymbol{p}$ and $\boldsymbol{P_p} = I_{3\times 3} - \hat{\boldsymbol{n}}_{\boldsymbol{p}}\hat{\boldsymbol{n}}_{\boldsymbol{p}}^T$ is a projection operator which projects vectors onto the tangent plane at $\boldsymbol{p}$.[26] Because both $\boldsymbol{H_p}$ and $\boldsymbol{P_p}$ are symmetric matrices, the shape operator simplifies to

$$S_{\boldsymbol{p}}(\hat{\boldsymbol{u}}) = -\frac{1}{||\boldsymbol{g_p}||}\boldsymbol{P_p}\boldsymbol{H_p}\hat{\boldsymbol{u}} \tag{10}$$

Then, the normal curvature of a curve passing through $\boldsymbol{p}$ in the direction $\hat{\boldsymbol{u}}$ is derived as[24]

$$0 = \boldsymbol{\nabla}_{\boldsymbol{u}}(\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{u}}) = (\boldsymbol{\nabla}_{\boldsymbol{u}}\hat{\boldsymbol{u}}) \cdot \hat{\boldsymbol{n}} + (\boldsymbol{\nabla}_{\boldsymbol{u}}\hat{\boldsymbol{n}}) \cdot \hat{\boldsymbol{u}} = \mathcal{K}_{\mathcal{S}}\hat{\boldsymbol{n}} \cdot \hat{\boldsymbol{n}} - S_{\boldsymbol{p}}(\hat{\boldsymbol{u}}) \cdot \hat{\boldsymbol{u}} \Rightarrow \mathcal{K}_{\mathcal{S}} = S_{\boldsymbol{p}}(\hat{\boldsymbol{u}}) \cdot \hat{\boldsymbol{u}} \tag{11}$$

Accordingly, $(\boldsymbol{\nabla}_{\boldsymbol{u}}\hat{\boldsymbol{u}}) = \mathcal{K}_{\mathcal{S}}\hat{\boldsymbol{n}}$.[24] Stepping in different directions $\hat{\boldsymbol{u}}$ away from a selected point may result in distinct values of the normal curvatures.

To summarize the normal curvatures of a surface at point $\boldsymbol{p}$, two principal curvatures are defined. Specifically, $\boldsymbol{\mathcal{K}}_{\mathcal{S},max}$ and $\boldsymbol{\mathcal{K}}_{\mathcal{S},min}$ are defined as the vectors associated with the maximum and minimum values of which $\mathcal{K}_{\mathcal{S}}$, respectively.[24] These directions are depicted conceptually in Figure 1 using orange and blue arrows, respectively. From Equation (10), these principal curvatures occur when $S_{\boldsymbol{p}}(\hat{\boldsymbol{u}})$ and $\hat{\boldsymbol{u}}$ are parallel, i.e., when there exists a constant value of α such that

$$S_{\boldsymbol{p}}(\hat{\boldsymbol{u}}) = \alpha\hat{\boldsymbol{u}} \Longleftrightarrow -\frac{1}{||\boldsymbol{g_p}||}\boldsymbol{P_p}\boldsymbol{H_p}\hat{\boldsymbol{u}} = \alpha\hat{\boldsymbol{u}} \tag{12}$$

Because $\hat{\boldsymbol{u}}$ is a unit vector, $\alpha = \mathcal{K}_{\mathcal{S}}$. Therefore, the principal directions and curvatures correspond to the eigenvectors and eigenvalues of the shape operator.[24]

The principal curvature directions form an orthonormal basis for the tangent plane $T_{\boldsymbol{p}}$. Accordingly, the principal curvatures supply a local description of the surface shape in the local neighborhood of $\boldsymbol{p}$. Consider the orthonormal local reference frame at $\boldsymbol{p}$ defined as $\{\boldsymbol{\mathcal{K}}_{min}, \boldsymbol{\mathcal{K}}_{max}, \hat{\boldsymbol{n}}_p\}$ and corresponding coordinates $\{\chi, \eta, \zeta\}$. From the Implicit Function Theorem, $\mathcal{S}$ is locally described near a sufficiently small neighborhood around $\boldsymbol{p}$ in terms of a height function $\zeta = h(\chi, \eta)$:[27,28]

$$\zeta = h(\chi, \eta) = \mathcal{K}_{min}\chi^2 + \mathcal{K}_{max}\eta^2 \tag{13}$$

This expression supplies a local quadratic approximation of the surface at $\boldsymbol{p}$.

Triangulation and Adaptive Mesh Refinement

Surface discretization has been used to describe and render surfaces computationally across various disciplines, from computational fluid dynamics (CFD) to computer graphics. In these domains, triangulations focus on approximating a curved surface by a set of triangles $\mathcal{T}$ that together define a mesh. These triangulations are typically used due to their simple and robust capability to describe complex surfaces embedded in three-dimensional space. Furthermore, structured meshes, such as those generated by triangulation, have a regular, predictable arrangement of elements that simplifies the computational implementation of algorithms designed to operate over a surface.[29]

Adaptive mesh refinement involves adapting a mesh to local properties of the surface, typically via subdivision. For example, in CFD, it is common to subdivide the mesh near areas with high flow complexity,[30] whereas in computer graphics, regions of the mesh at locations of higher curvature are subdivided to better capture the local surface characteristics.[29] In these examples, adaptive mesh refinement process may result in higher numerical simulation accuracy, greater space and memory efficiency, or improved surface representation fidelity.

Although a variety of techniques for mesh subdivision exist, one foundational approach is the red-green refinement technique.[31] Red-green refinement then consists of the following steps:

1. Select the initial triangles to be refined.
2. Divide all edges of the selected triangles in half.
3. For any triangle that has a divided edge, subdivide the longest edge. Repeat until no additional edges need to be divided.
4. Insert new points at the midpoints of all divided edges.
5. Construct the new triangles. If a triangle has its three sides divided, new triangles are formed by joining the side midpoints. If two sides are divided, the midpoint of the longest edge is joined with the opposing corner and with the other midpoint. If only the longest edge is divided, its midpoint is joined with the opposing corner.

This refinement technique is conforming, meaning that the mesh is topologically consistent everywhere, allowing the refinement algorithm to be used recursively at any region of the mesh.[31]

Clustering

This paper leverages two density-based clustering algorithms to summarize the geometries of trajectories in the CR3BP: Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[32] and HDBSCAN.[33] Both algorithms act on a dataset with N members, described by m-dimensional feature vectors, to produce a set of 1) clusters $\mathcal{C}$ as members that form high-density groupings and 2) null assignments $\mathcal{N}$ that exist in low-density regions within the feature vector space. DBSCAN is accessed via MATLAB whereas HDBSCAN is accessed using the *hdbscan* Python library.[34]

DBSCAN Without a-priori knowledge of the number of clusters, DBSCAN can identify arbitrarily-shaped clusters and label outliers as noise. The grouping relies on identifying density-connected neighborhoods in the m-dimensional dataset. The algorithm relies on two user-defined parameters, the neighborhood radius ϵ and a number of points m_{pts}.[32] Clusters are defined as sets of core points that exist within each other's m_{pts}-neighborhoods with radii less than ϵ.[32] Border point exist in the m_{pts}-neighborhoods of core points but their own m_{pts}-neighborhood are larger than ϵ;[32] accordingly, they are added to the clusters of those core points. The remaining points which do not satisfy these conditions are labeled as noise and correspond to null assignments.[32]

HDBSCAN Expanding on DBSCAN,[32] HDBSCAN[33] eliminates the need for a constant value of ϵ. First, the radius of the m_{pts}-neighborhood of point i is defined as its core distance d_i^{core}. The mutual reachability distance, $d_{i,j}^{MRD}$, is then calculated between any two data points i and j as

$$d_{i,j}^{MRD} = \max(d_i^{core}, d_j^{core}, d_{i,j})$$

where $d_{i,j}$ is the Euclidean distance calculated between the feature vectors of points i and j. This mutual reachability distance further separates points in low-density regions. HDBSCAN then constructs a minimum spanning tree (MST) between all the points in the dataset using the mutual reachability distance to the define the edge weights. By progressively removing edges in decreasing order of their edge weights, a hierarchical tree is built using the resulting connected components present in the modified MST. Finally, clusters which contain more than $m_{min,clust}$ points and exhibit the greatest stability, i.e., persisting the longest across the range of mutual reachability distances, are selected as the final output. Points that are not assigned as clusters are designated as noise.[33] More recently, the algorithm was refined to merge any two points within a distance of ϵ_{merge}.[35]

Clustering-Based Framework for Summarizing Trajectories

Recently, clustering has been used to automatically summarize the prominent geometries of a diverse array of trajectories across a multi-body system. The most recent implementation of this framework has been developed by Bosanac using concepts from differential geometry and distributed clustering.[15] A brief overview of this procedure is presented in this section.

An important definition in the clustering-based framework previously developed by Bosanac[36] is the description of a continuous trajectory via a finite-dimensional feature vector. Given an initial state vector, a trajectory is propagated for up to 21 days. This continuous solution is then sampled in a geometry-based manner at states that are evenly distributed in the total absolute curvature along each half-revolution, n, in the Earth-Moon rotating frame.[14] A trajectory completing up to $n = \lceil \kappa_{\mathcal{C},tot}(t_0, t_f)/\pi \rceil$ half-revolutions within the evolving osculating plane is first sampled at the initial condition. Then, it is evenly sampled every π/N_a in the total absolute curvature for the first $n-1$ half-revolutions. The last half-revolution is sampled by N_a equally spaced states in the remaining total absolute curvature. Accordingly, longer trajectories are sampled by more states and these states tend to be located closer together in regions of high curvature. Following Bosanac and Joyner,[14] we take $N_a = 3$ to accurately describe each trajectory while reducing the number of total samples. These sampled states are then used to construct two feature vectors: one position-based feature vector, $\boldsymbol{f}_p$, composed of the position components of each state and one shape-based feature vector, $\boldsymbol{f}_s$, that consists of the velocity unit vectors of each sampled state:[14]

$$\boldsymbol{f}_p^{\mathrm{T}} = [\boldsymbol{r}_1^{\mathrm{T}}, \boldsymbol{r}_2^{\mathrm{T}}, \ldots, \boldsymbol{r}_m^{\mathrm{T}}] \qquad \boldsymbol{f}_s^{\mathrm{T}} = [\hat{\boldsymbol{v}}_1^{\mathrm{T}}, \hat{\boldsymbol{v}}_2^{\mathrm{T}}, \ldots, \hat{\boldsymbol{v}}_m^{\mathrm{T}}]$$

To select the initial state vectors that are used to generate the set of trajectories to be summarized, a static grid definition and differential geometry are employed. First, the initial position vectors are defined using a static, uniform grid. Prespecified values of the Jacobi constant are then used to calculate the associated speed at each initial condition. For each initial position vector and associated speed, the velocity direction is selected to produce a maximum in the instantaneous curvature along the trajectory, $\kappa_{\mathcal{C}}$, at the initial condition.[14] Similar to how apses are used traditionally, maxima in the curvature offer a geometrically meaningful definition for automatically and robustly sampling initial conditions in a multi-body system.[37] For planar trajectories, up to four curvature maxima exist at any position vector. After combining these initial velocities with their associated

position vector, the associated trajectories are generated, summarized, and stored in the training dataset. However, for spatial trajectories, these velocity vectors exist in continuous families that are discretely approximated.[37] Thus, the associated trajectories are generated and grouped in a local clustering step that applies HDBSCAN to their shape-based feature vectors. Representative members of these local clusters supply the trajectories that are stored in the training dataset.[14]

The training dataset is then distributed across multiple, smaller partitions of trajectories that are sampled using the same number of states. The trajectories within each partition are coarsely clustered via HDBSCAN using the shape-based feature vectors evaluated over their entire duration.[15] These coarse clusters are then processed by a cluster refinement technique inspired by a convoy detection scheme.[38] The ith states along all the trajectories within a coarse group are then independently clustered in each of the position- and shape-based three-dimensional feature vector spaces using DBSCAN with adaptive heuristics for selecting ϵ.[15] This process is repeated for all N_s subsequent sampled states to produce $2N_s$ clustering results. Any trajectories that are consistently grouped together correspond to a refined cluster. The result of this process is a set of local clusters and noise for each partition; each local cluster is described by a representative member, equal to the medoid in the position-based feature vector space.[11] Following Bosanac,[15] we take $m_{min,clust} = 5$ and $m_{pts} = 4$ for either clustering step and $\epsilon_{merge} = 2\sin(10°/2)$ for HDBSCAN.

Consistent with distributed clustering, local clusters are aggregated across partitions if they contain trajectories with a similar geometry.[14] Candidate neighboring pairs are first coarsely identified using nearest neighbors in the position- and shape-based feature vector spaces. Then, the trajectories within these candidate neighboring pairs are input to the cluster refinement process. If any members are grouped together, the associated clusters are merged.[15] The connected sets of local clusters then form the global clusters that summarize the entire training dataset.

TECHNICAL APPROACH

This paper focuses on expanding the existing clustering framework to adaptively summarize the prominent geometries of trajectories originating within a specified region of the phase space in the CR3BP. There are three main steps in this process, as depicted in the flowchart in Figure 2. The input to this framework is a user-defined, initial coarse grid of position vectors. These position vectors are used to generate the initial conditions of trajectories in the training dataset; these initial conditions follow a hypersurface of curvature maxima. At each iteration of the framework, this grid is automatically refined in a manner that balances 1) global exploration of the array of prominent geometries across the solution space, at a desired resolution; and 2) local exploitation to ensure sufficient sampling to improve the quality of the clustering results. The trajectories generated at each iteration of the framework are then clustered using the procedure developed by Bosanac[15] and described in the Background. To support a proof of concept, this paper focuses on summarizing planar motion at a fixed energy level in the Earth-Moon CR3BP.

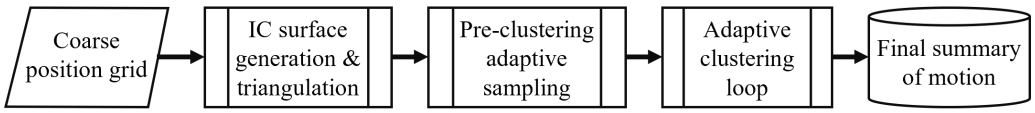

Figure 2. Condensed flowchart of the new adaptive clustering framework.

Defining Initial Conditions

Following the implementation by Bosanac,[15] the initial conditions are defined to produce maxima in the curvature along a trajectory. However, this paper presents an analytical approach to generating these initial state vectors for planar motion. While avoiding numerical inaccuracies of a root-finding scheme, this analytical method also reduces the required computational time.

The maximum curvature condition, evaluated in the rotating frame for planar trajectories, is expressed in terms of the spacecraft state vector. To simplify the derivation, the velocity vector $\mathbf{v} = [\dot{x}, \dot{y}]$ is expressed in polar coordinates (v, θ) where v is the speed of the spacecraft and θ is the angle between the velocity vector and $\hat{x}$. The relationship between these coordinates is

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \end{cases} \implies \begin{cases} v = \sqrt{\dot{x}^2 + \dot{y}^2} \\ \theta = \arctan\left(\frac{\dot{y}}{\dot{x}}\right) \end{cases}$$

The differential equations governing the spacecraft are then rewritten in terms of v and θ as

$$\begin{aligned} \dot{x} &= v\cos\theta & \dot{v} &= \frac{\dot{x}\ddot{x} + \dot{y}\ddot{y}}{v} = \frac{\dot{x}U_x + \dot{y}U_y}{v} = U_x\cos\theta + U_y\sin\theta \\ \dot{y} &= v\sin\theta & \dot{\theta} &= \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{v^2} = \frac{U_y\cos\theta - U_x\sin\theta}{v} - 2 \end{aligned} \tag{14}$$

The second and third time derivatives of v and θ are calculated as

$$\ddot{v} = v\left[\dot{\theta}(\dot{\theta}+2) + U_{xx}\cos^2\theta + U_{yy}\sin^2\theta + U_{xy}\sin 2\theta\right] \tag{15}$$

$$\ddot{\theta} = U_{xy}\cos 2\theta + \frac{1}{2}(U_{yy} - U_{xx})\sin 2\theta - \frac{2\dot{v}\left(\dot{\theta}+1\right)}{v} \tag{16}$$

$$\begin{aligned} \dddot{\theta} = {} & \dot{\theta}\left[(U_{yy}-U_{xx})\cos(2\theta) - 2U_{xy}\sin(2\theta)\right] - \frac{2v\left(\ddot{v}\dot{\theta}+\dot{v}\ddot{\theta}+\ddot{v}\right) - \dot{v}^2\left(\dot{\theta}+1\right)}{v^2} + \\ & \frac{v}{4}\Big[(3U_{xyx} - U_{yyy})\cos(3\theta) + (3U_{xyy} - U_{xxx})\sin(3\theta) + (U_{xyx}+U_{yyy})\cos(\theta) \\ & - (U_{xxx}+U_{xyy})\sin(\theta)\Big] \end{aligned} \tag{17}$$

where $U_{ij} = \partial^2 U/\partial x_i \partial x_j$ and $U_{ijk} = \partial^3 U/\partial x_i\,\partial x_j\,\partial x_k$ for $i, j, k \in \{x, y\}$. Because these derivatives are symmetric, there are only 10 independent values of U_{ijk}.

The curvature of a trajectory, $\kappa_{\mathcal{C}}$, is calculated in terms of the coordinates (x, y, v, θ). In polar velocity coordinates, $\dot{\boldsymbol{v}} = \dot{v}\hat{e}_v + v\dot{\theta}\hat{e}_\theta$. Thus, Equation (5) is rewritten as

$$\kappa_{\mathcal{C}} = \frac{||\boldsymbol{v}\times\dot{\boldsymbol{v}}||}{||\boldsymbol{v}||^3} = \frac{|\dot{\theta}|}{v} = \frac{\dot{\theta}}{v}\cdot\text{sign}(\dot{\theta}) \tag{18}$$

where the explicit time dependence has been omitted to simplify notation. Due to the change in the coordinates that describe the velocity vector, this expression for the curvature along a trajectory is more straightforward and offers an intuitive interpretation: higher curvature occurs when traveling at low speed with rapid directional changes. The time derivatives of $\kappa_{\mathcal{C}}$ are then expressed as

$$\dot{\kappa}_{\mathcal{C}} = \frac{\ddot{\theta}v - \dot{\theta}\dot{v}}{v^2}\cdot\text{sign}(\dot{\theta}) \tag{19}$$

$$\ddot{\kappa}_{\mathcal{C}} = \frac{v\left(\dddot{\theta}\, v - \dot{\theta}\ddot{v}\right) - 2\dot{v}\left(\ddot{\theta}v - \dot{\theta}\dot{v}\right)}{v^3} \cdot \text{sign}(\dot{\theta}) = \frac{\dddot{\theta}\, v - \dot{\theta}\ddot{v}}{v^2} \cdot \text{sign}(\dot{\theta}) \quad \text{at curvature extrema} \tag{20}$$

Each expression relies on the position vector and time derivatives of the velocity vector coordinates.

Extrema in the curvature are calculated as stationary points in the curvature along the trajectory. The condition $\dot{\kappa}_{\mathcal{C}} = 0$ is rewritten as the following trigonometric function by substituting Equations (14) and (16) into Equation (19), and performing some algebraic manipulation:

$$A\cos 2\theta + B\sin 2\theta + C\cos\theta + D\sin\theta = 0 \tag{21}$$

where

$$\begin{aligned} A &= U_{xy} - \frac{3U_xU_y}{v^2} & C &= \frac{4U_x}{v} \\ B &= \frac{1}{2}\left(U_{yy} - U_{xx} + \frac{3(U_x^2 - U_y^2)}{v^2}\right) & D &= \frac{4U_y}{v} \end{aligned}$$

For a given value of the Jacobi constant and mass ratio, the coefficients A, B, C, D are a function of only the position because $v = \sqrt{2U(x,y) - C_J}$. By using Euler's identity and defining $Z = e^{\mathrm{i}\theta} = \cos\theta + \mathrm{i}\sin\theta$, Equation (21) is rewritten as

$$(A - \mathrm{i}B)Z^4 + (C - \mathrm{i}D)Z^3 + (C + iD)Z + (A + iB) = 0 \tag{22}$$

This expression is a quartic equation of the form

$$aZ^4 + bZ^3 + cZ^2 + dZ + e = 0 \tag{23}$$

where $a = (A - \mathrm{i}B)$, $b = (C - \mathrm{i}D)$, $c = 0$, $d = (C + \mathrm{i}D)$, and $e = (A + \mathrm{i}B)$. From the Fundamental Theorem of Algebra, Equation (23) has exactly 4 complex roots: $Z_1^*, Z_2^*, Z_3^*, Z_4^* \in \mathbb{C}$. One of the methods to obtaining the solutions to Equation (23) was first obtained by Ferrari,[39] as

$$\begin{aligned} Z_{1,2}^* &= -\frac{b}{4a} - S \pm \frac{1}{2}\sqrt{-4S^2 - 2p + \frac{q}{S}} \\ Z_{3,4}^* &= -\frac{b}{4a} + S \pm \frac{1}{2}\sqrt{-4S^2 - 2p - \frac{q}{S}} \end{aligned} \tag{24}$$

where

$$\begin{aligned} p &= \frac{8ac - 3b^2}{8a^2} & q &= \frac{b^3 - 4abc + 8a^2d}{8a^3} \\ \Delta_0 &= c^2 - 3bd + 12ae & \Delta_1 &= 2c^3 - 9bcd + 27b^2e + 27ad^2 - 72ace \\ Q &= \sqrt[3]{\frac{\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}} & S &= \frac{1}{2}\sqrt{-\frac{2}{3}p + \frac{1}{3a}\left(Q + \frac{\Delta_0}{Q}\right)} \end{aligned}$$

According to the Abel-Ruffini theorem, a quartic polynomial is the highest order polynomial with roots that can be expressed in terms of radicals.

From the Fundamental Theorem of Algebra, there are at most four curvature extrema per position vector for each value of C_J and μ in the planar CR3BP. To justify this result, the Ferrari method[39] is used to obtain the solutions $Z^* \in \mathbb{C}$ of Equation (22). These solutions correspond to the desired velocity directions $\theta^* = -\mathrm{i}\ln Z^*$, where only real valued θ^* are considered, i.e., those for which $|Z^*| = 1$. Finally, only values of θ^* that yield $\ddot{\kappa}_{\mathcal{C}} < 0$ are retained to produce four or fewer maxima in the curvature at any given position vector for a single Jacobi constant and mass ratio.

Initial Condition Surface Triangulation

For fixed values of C_J and μ, $\dot{\kappa}_{\mathcal{C}}$ is only a function of the position coordinates x, y and the velocity direction θ. As such, the equation $\dot{\kappa}_{\mathcal{C}}(x, y, \theta) = 0$ implicitly defines a surface embedded in $\mathbb{R}^3$. Thus, differential geometry of surfaces is leveraged to analyze this complicated surface of initial conditions. Accordingly, the triangulation procedure presented in this paper to approximate the surface of initial conditions is performed through a sequence of steps.

First, an initial triangulation, $\mathcal{T}_{pos}$, is defined over an initial, coarse, and uniform grid of position vectors using MATLAB's built-in triangulation functions. This procedure is depicted in Figure 3a) for $C_J = 3.14$, where the triangulation is shown in light blue, the initial positions are marked with blue points, the zero-velocity curves are plotted in gray, the Moon is represented by the gray circle, and the L_1 and L_2 Lagrange points are indicated by red diamonds. This step identifies neighbors of each state vector only in position. Next, the speed at a desired value of C_J is calculated at each position vector. Then, as depicted in Figure 3b), the up to four values of θ that define the velocity directions of curvature maxima are calculated for each position using the methodology described in the previous subsection. The next step is to extend the triangulation across the full state description.

Consider an initial condition $\text{IC}^* = [x^*, y^*, \theta^*]^{\mathrm{T}}$ at a point $\boldsymbol{p}$ on the surface of initial conditions. In $\mathcal{T}_{pos}$, IC^* can be connected to up to six nearby position vectors in the configuration space. Because each position vector can produce up to four states that are maxima in curvature, up to

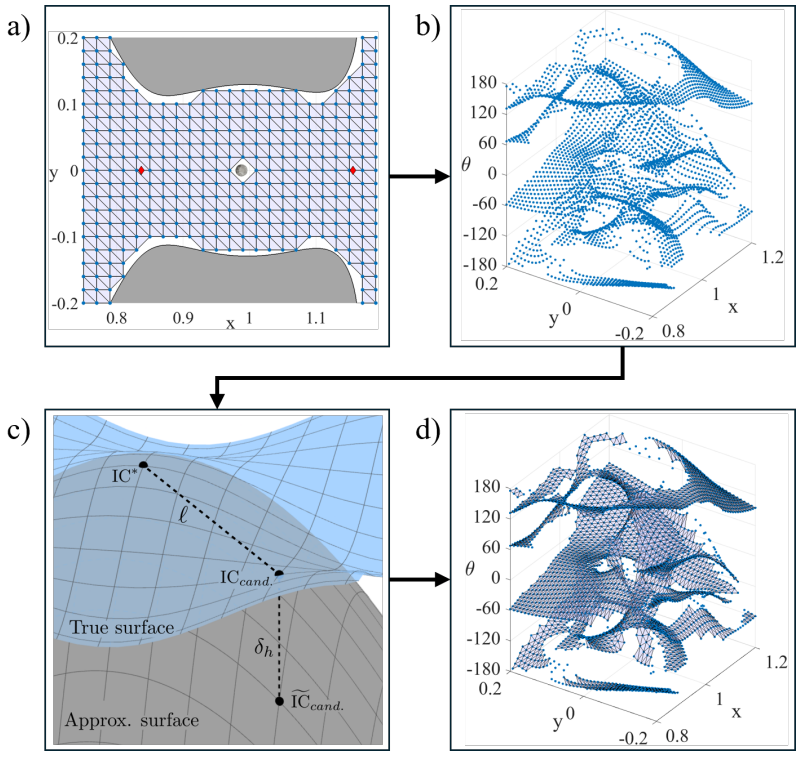

Figure 3. Flowchart depicting the initial condition surface triangulation scheme: a) position triangulation, b) initial condition generation, c) local quadratic surface approximation and d) surface triangulation.

$24 + 3 = 27$ candidate states could be connected to IC*; these states are denoted as IC$_{cand.}$. However, not all these candidates exist nearby IC* on the surface associated with curvature maxima.

To identify neighboring states along the surface, this paper leverages curvature-based criteria.[28] Specifically, candidates that are sufficiently close to IC* are identified using a local quadratic approximation of the local surface curvature using Equation (13). First, the normal and the shape operator at IC*, are calculated using Equations (7) and (10). When applied to the planar CR3BP, the gradient and the Hessian are rewritten as

$$\mathbf{g}_{\boldsymbol{p}} = \boldsymbol{\nabla}\dot{\kappa}_{\mathcal{C}}\Big|_{\text{IC}^*} = \begin{bmatrix} \frac{\partial \dot{\kappa}_{\mathcal{C}}}{\partial x} & \frac{\partial \dot{\kappa}_{\mathcal{C}}}{\partial y} & \frac{\partial \dot{\kappa}_{\mathcal{C}}}{\partial \theta} \end{bmatrix}^{\mathrm{T}}_{\text{IC}^*} \qquad \boldsymbol{H}_{\boldsymbol{p}} = \boldsymbol{H}(\dot{\kappa}_{\mathcal{C}})\Big|_{\text{IC}^*} = \begin{bmatrix} \frac{\partial^2 \dot{\kappa}}{\partial x^2} & \frac{\partial^2 \dot{\kappa}}{\partial x \partial y} & \frac{\partial^2 \dot{\kappa}}{\partial x \partial \theta} \\ \frac{\partial^2 \dot{\kappa}}{\partial y \partial x} & \frac{\partial^2 \dot{\kappa}}{\partial y^2} & \frac{\partial^2 \dot{\kappa}}{\partial y \partial \theta} \\ \frac{\partial^2 \dot{\kappa}}{\partial \theta \partial x} & \frac{\partial^2 \dot{\kappa}}{\partial \theta \partial y} & \frac{\partial^2 \dot{\kappa}}{\partial \theta^2} \end{bmatrix}_{\text{IC}^*}$$

The eigenvalues and eigenvectors of the shape operator are then calculated and sorted to supply a local basis $\boldsymbol{B}_{local} = [\boldsymbol{\mathcal{K}}_{min}, \boldsymbol{\mathcal{K}}_{max}, \hat{\boldsymbol{n}}_p]$, and the principal curvatures at IC*. These basis vectors are used to rewrite IC$_{cand.} = [x, y, \theta]^{\mathrm{T}}$ in the local IC* frame via the following transformation:

$$^{\boldsymbol{B}_{local}}\text{IC}_{cand.} = \begin{bmatrix} \chi \\ \eta \\ \zeta \end{bmatrix} = \left(\text{IC}_{cand.} - \text{IC}^*\right)^{\mathrm{T}} \boldsymbol{B}_{local}$$

Note that IC* is at the origin of this new frame. Then, as depicted in Figure 3c), a local quadratic approximation to the surface is constructed at IC* with Equation (13), and the local coordinates $\{\chi, \eta\}$ of the candidates are used to produce the approximated height values, $\tilde{\zeta}$. When compared to the true height value ζ, the local height error is defined as $\delta_h = |\zeta - \tilde{\zeta}|$. The candidate states IC$_{cand.}$ that are considered triangulation neighbors of IC* must satisfy the following two conditions: 1) δ_h is below a user defined threshold Δh_{max}, and 2) their separation distance $\ell = ||\text{IC}^* - \text{IC}_{cand.}||$ is below a user defined threshold $\Delta \ell_{max}$.

The triangulation $\mathcal{T}_{pos}$ is then extended to the full phase space using this information from the neighboring states along the surface of initial conditions. First, each state IC* defines a node in the triangulation. Then, edges are added between the node associated with IC* and the nodes of candidate states IC$_{cand.}$ that satisfy the neighborhood criteria. Repeating the process for every initial condition in the coarse grid produces a triangulation $\mathcal{T}$ for the surface, as depicted in Figure 3d). Here, the initial conditions are represented by blue dots, connected by black edges that define the triangulation, represented as light blue surface patches. Figure 3d) highlights two current limitations of the algorithm: 1) some initial conditions are not included in the triangulation, and (2) there are no edge connections between points near $\theta = -180°$and $\theta = 180°$, which correspond to similar velocity directions. Addressing these limitations is an avenue of ongoing work.

Adaptive Mesh Refinement

Given an initial triangulation of the surface, the mesh of initial conditions is refined. Similar to the exploration-exploitation tradeoff in machine learning,[40] the goal of mesh refinement is to both improve the resolution of poorly sampled regions (global exploration) and reduce the error in the clustering result (local exploitation). In this paper, each of these goals in the mesh refinement process is defined using information from differential geometry and the clustering results.

In the global exploration phase, the goal is to capture the overall shape of the surface of initial conditions and, therefore, increase the likelihood that the array of prominent trajectories geometries

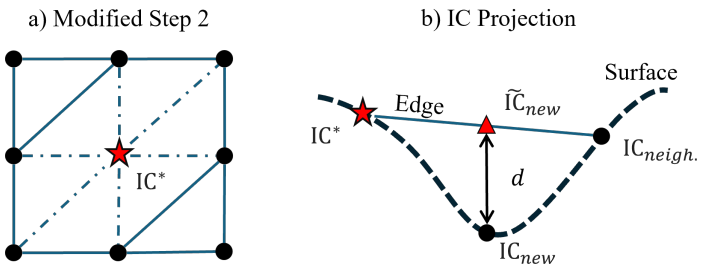

Figure 4. Modified step 2 of red-green mesh refinement and projection onto the surface of initial conditions.

appears within the training dataset with sufficient density. To achieve this goal, the root-mean-squared curvature value, $\mathcal{K}_{rms} = \sqrt{\mathcal{K}^2_{min.} + \mathcal{K}^2_{max.}}$, is calculated at each of the initial conditions in the current mesh. At any given point, a high value of $\mathcal{K}_{rms}$ indicates that the surface is more curved and, therefore, more complex. Thus, every initial condition for which $\mathcal{K}_{rms}$ is larger than a user-defined value $\Delta\mathcal{K}_{rms,max}$, is flagged for refinement by adding initial conditions in its vicinity.

In the local exploitation phase, the goal is to reduce clustering error by increasing the resolution in dynamically sensitive regions and/or areas where initial conditions correspond to trajectories that were labeled as noise by the clustering framework. Dynamically sensitive regions are identified based on two criteria: geometric changes, detected heuristically using the total trajectory curvature, and relative trajectory stability, measured via the Fast Lyapunov Indicator defined in Equation (3). These quantities are calculated for all initial conditions by propagating the total curvature and the STM alongside the equations of motion. Note that the total curvature adds one equation to the propagation whereas the STM adds 16 extra equations for planar motion. Then, for any given initial condition IC*, the variance of these quantities is calculated over its neighborhood, defined as the set of initial conditions belonging to every triangle containing IC* in the current surface triangulation. Let $\mathcal{N}(\text{IC}^*)$ denote this set, which includes IC* itself, with a total of N members. Then, for any scalar quantity q associated with each initial condition, its variance over the neighborhood of IC* is

$$\text{Var}_{\mathcal{N}(\text{IC}^*)}(q) = \frac{1}{N-1} \sum_{\text{IC}_i \in \mathcal{N}(\text{IC}^*)} (q(\text{IC}_i) - \overline{q})^2 \qquad \overline{q} = \frac{1}{N} \sum_{\text{IC}_i \in \mathcal{N}(\text{IC}^*)} q(\text{IC}_i)$$

Here, q can be either $\kappa_{\mathcal{C},tot.}$ or FLI_T. Initial conditions for which this variance is greater than user-defined thresholds $\sigma^2_{max}(\kappa_{\mathcal{C},tot.})$ and $\sigma^2_{max}(\text{FLI}_T)$ are flagged for refinement. Finally, initial conditions of trajectories that are labeled as noise during clustering are also flagged.

Given a set of flagged initial conditions, the mesh refinement algorithm is inspired by red-green refinement, with minor modifications. The primary difference is that in step 2, instead of refining entire triangles, the procedure used in this paper only marks for subdivision the edges between flagged initial conditions and their triangulation neighbors, as depicted in Figure 4a). Here, the red star is the flagged initial condition, the solid blue lines represent triangulation edges between other initial conditions, shown as black dots, while the dashed lines indicate edges marked for subdivision.

Whenever an edge is divided during refinement, the new initial condition, IC$_{new}$, must be projected back onto the surface defined by curvature maxima. The position vector at the center of this edge is first calculated between IC* and its neighbor IC$_{neigh.}$ as $\boldsymbol{r}_{new} = (\boldsymbol{r}^* + \boldsymbol{r}_{neigh.})/2$. Because

up to four curvature maxima may exist at this position, the resulting set of actual curvature maxima IC_{new} are compared with the approximated state $\widetilde{\text{IC}}_{new}$ as depicted in Figure 4b). Then, only the initial condition in IC_{new} that lies on the surface but possesses a distance d to $\widetilde{\text{IC}}_{new}$ that is lower than a user defined threshold Δd_{new} is retained as the new sample for this specific edge.

Finally, the user can specify two mesh resolution thresholds. One threshold governs the minimum separation between initial conditions in position, Δr_{min}, whereas the other threshold governs the minimum separation in velocity direction $\Delta\theta_{min}$. During any subsampling step, if the distance between IC^* and $\text{IC}_{neigh.}$ is less than $2\Delta r_{\min}$ in the configuration subspace and less than $2\Delta\theta_{\min}$ in the velocity angle subspace, the edge is not subdivided.

Adaptive Clustering Framework

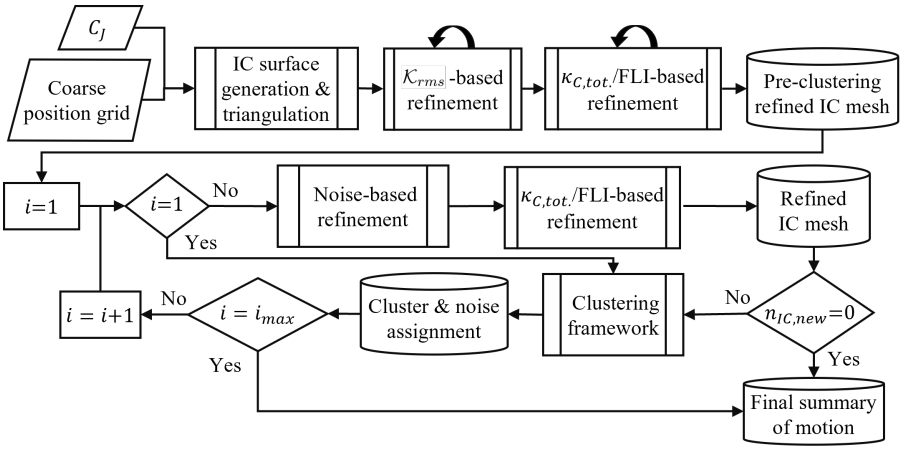

Figure 5. Flowchart describing the full adaptive clustering framework.

The three components of adaptive mesh refinement are leveraged to construct an adaptive clustering framework. Figure 5 presents an overview of this general framework. Given a single value of the Jacobi constant and a coarse, initial position grid, the surface of initial conditions is calculated and triangulated using the methodology described above. Then, in a global exploration phase, the resolution of the surface is improved by performing mesh refinement, driven by the root mean squared surface curvature. This step can be performed iteratively up to a number of $N_{\text{geo.}}$ iterations. After the surface of initial conditions is well-resolved, the local exploitation phase begins. This mesh is first refined using the variance of the total trajectory curvature and FLI within each initial conditions' neighborhood. This refinement is performed for up to iterations, $N_{\text{pre-clust.}}$. The result is a sufficiently dense initial condition mesh $\mathcal{M}_{IC}$ to begin the adaptive clustering loop.

For the first iteration of adaptive clustering, the dataset is summarized using the clustering framework by Bosanac,[15] to extract the first set of cluster and noise assignments. Then, $\mathcal{M}_{IC}$ is refined sequentially by first considering the presence of noise and using the local variance of $\kappa_{\mathcal{C},tot.}$ and FLI_T. The resulting mesh is then passed through the clustering framework again to update the summary of motion, and the process repeats. The adaptive clustering loop terminates when either 1) reaching a maximum number of iterations N_{AC} or 2) no initial conditions added to the mesh, by virtue of either the mesh resolution thresholds or no more flagged initial conditions. At this point, the mesh is considered to have converged and the last obtained set of cluster and noise assignments supplies the final summary of motion.

RESULTS

The presented adaptive clustering framework is used to summarize planar motion in the vicinity of the Moon in the Earth-Moon CR3BP at a Jacobi constant of $C_J = 3.12$. Specifically, the region of interest is defined by: $R = \{(x, y) \mid 0.8 \leq x \leq 1.2,\ -0.2 \leq y \leq 0.2\}$. This section analyzes the results of two main steps in Figure 2: the pre-clustering adaptive sampling and the adaptive clustering loop. For this application, Table 1 lists the selected values of the hyperparameters that govern the adaptive sampling scheme. The coarse position grid is initially defined by sampling the region R with a step of 0.01 in each direction of the configuration space, producing only 4,342 initial state vectors.

Triangulation			**Adaptive Sampling**			**Mesh Resolution**		**Max. iterations**		
Δh_{max}	Δl_{max}	Δd_{max}	$\Delta \mathcal{K}_{rms,max}$	$\sigma^2_{max}(\kappa_{\mathcal{C},tot.})$	$\sigma^2_{max}(\text{FLI}_T)$	Δr_{min}	$\Delta\theta_{min}$	N_{geo}	$N_{pre-clust}$	N_{AC}
0.05	0.5	0.1	80	0.25	0.015	5×10^{-4}	0.5°	6	2	3

Table 1. Selected values of parameters governing the adaptive sampling framework.

Mesh Refinement via Surface Curvature

Once the surface of initial conditions has been generated and triangulated for the first time, the mesh is refined using the surface curvature. This step corresponds to the global exploration phase of the adaptive sampling framework, where the goal is to discretize the initial condition surface efficiently while obtaining a sufficient representation of its geometric features. After the $\mathcal{K}_{rms}$-based refinement, the initial condition mesh is as plotted in Figure 6a), where the color indicates the value of the root mean square curvature at each point.

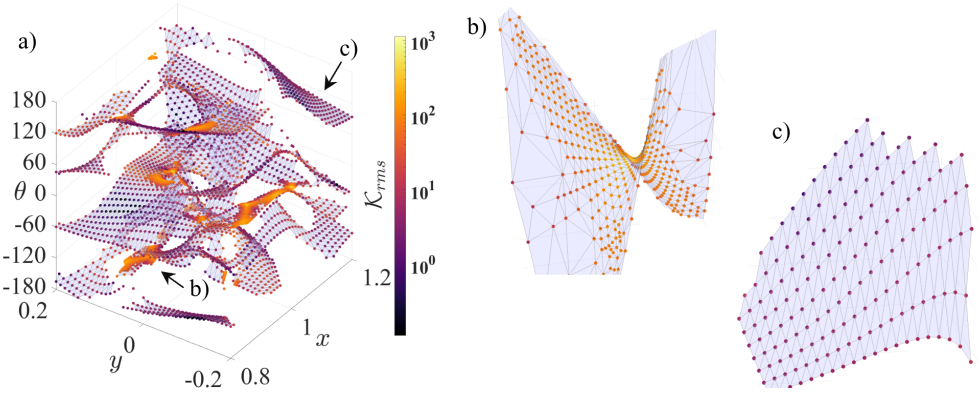

Figure 6. Initial condition mesh after $\mathcal{K}_{rms}$-based refinement and zoom on curved (b)) and flat (c)) regions.

The refinement process at this step successfully adapts to the geometry of the surface of initial conditions. For instance, the indicated region labeled with the text annotation "b)" is visualized in a zoomed-in view in Figure 6b). In this local region, the surface possesses a saddle shape, which would not be visible without refinement. The refinement process successfully captures the geometric features of the surface, strategically adding samples where necessary. However, there are other portions of the surface that are less geometrically complex. For instance, in the local region visualized in Figure 6c), the surface appears to be relatively flat and requires no refinement under the

selected curvature threshold. With the surface geometry better captured through adaptive sampling, there is a higher likelihood that the trajectories generated from these initial conditions better capture the array of prominent geometries.

Comparing Mesh Refinement via Total Absolute Curvature and Chaos Indicators

Following $\mathcal{K}_{rms}$-based refinement, the mesh is then passed through two refinement steps using the total trajectory absolute curvature and the chaos indicator FLI_T. However, to investigate the effects of each quantity independently, each refinement is performed with only one parameter active at a time. Figures 7a) and c) (as well as b) and d)) display the mesh before and after refinement using the total absolute curvature (and chaos indicator FLI_T), with the initial conditions colored by their corresponding value of $\kappa_{\mathcal{C},tot.}$(and FLI_T). In this example, the values of $\sigma^2_{max}(\kappa_{\mathcal{C},tot.})$ and $\sigma^2_{max}(\text{FLI}_T)$ are listed in Table 1 and selected to produce approximately the same number of initial conditions after refinement ($\sim 27,700$).

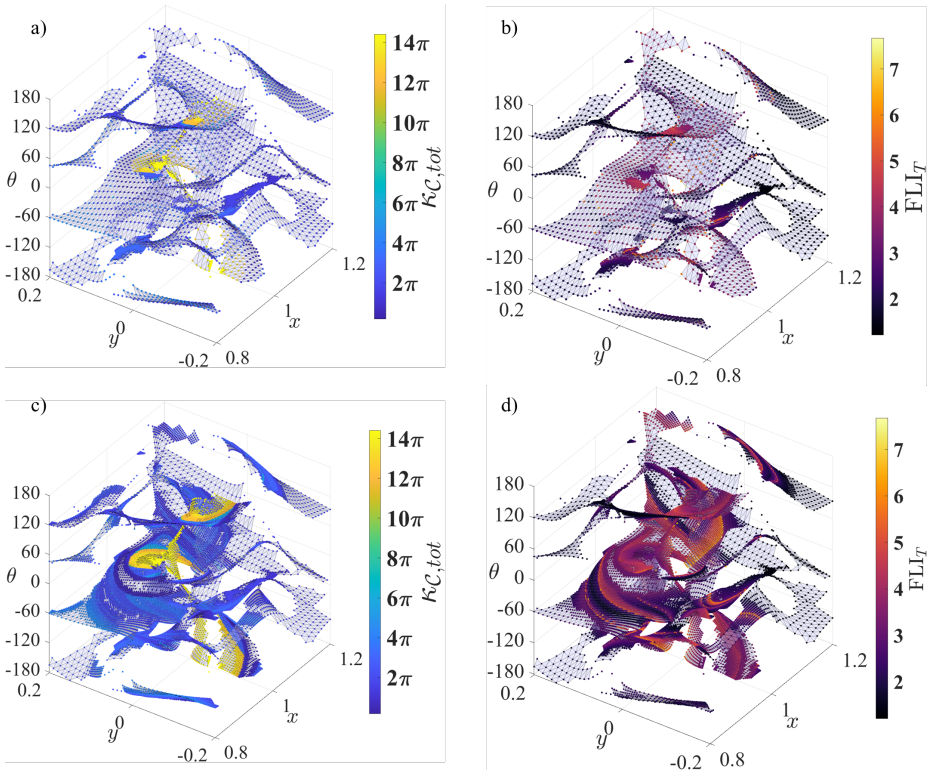

Figure 7. Refined initial condition meshes using either $\kappa_{\mathcal{C},tot.}$, a) and c), or FLI_T , b) and d).

Using the total curvature along a trajectory to refine the mesh effectively resolves regions where the quantity changes rapidly. In Figures 7a) and c), initial conditions near the Moon, colored in yellow near the center of the figure, result in trajectories with high values of $\kappa_{\mathcal{C},tot.}$. These trajectories are quasi-periodic trajectories that perform multiple revolutions around the Moon. Moreover, mesh refinements are observed in regions where the stable manifold of the Lyapunov orbits at this energy level intersects the initial condition surface. Ongoing work aims to investigate this observation.

Using the chaos indicator to refine the mesh effectively resolves regions where the dynamics are more sensitive. For instance, Figure 7d) features ridges in the value of the chaos indicator along

the surface of initial conditions. Similar to a separatrix in phase space diagrams, these ridges form the boundary between regions of distinct dynamical behavior. The adaptive scheme successfully resolved these features, which were not clearly visible in the initial coarse grid.

The meshes refined using either $\kappa_{\mathcal{C},tot.}$ or FLI_T are similar. In each case, the areas where new initial conditions are placed are closely correlated. This result is consistent with the expectation that a trajectory located near another path that is sensitive to small changes in the initial condition, as quantified by FLI_T, will likely exhibit a different shape, leading to a different value of $\kappa_{\mathcal{C},tot.}$. However, $\kappa_{\mathcal{C},tot.}$ offers an interesting benefit for performing refinement as this approach better correlates with the partitioning process used in the clustering framework by Bosanac[15] and requires 15 fewer additional ordinary differential equations to be integrated, speeding up trajectory propagation.

Adaptive Clustering

The grid depicted in Figure 7c) is used to perform three iterations of the adaptive clustering loop depicted in Figure 5. Figure 8 displays two views of the grid: (left) after the first clustering step and (right) after three iterations of the adaptive clustering loop. Cluster labels are indicated with color and noise is displayed in brown. Note that the same color does not correlate to the same exact cluster before and after the adaptive clustering refinement.

Near the L_2 gateway, the initial mesh is quite coarse and produces many unlabeled trajectories. Thus, the adaptive clustering scheme places more initial condition in that area, leading to the successful clustering of the respective trajectories and eliminating noise. The grid in the region $\{(x, y, \theta) \mid 0.8 \leq x \leq 0.85,\ 0.15 \leq y \leq 0.2,\ -180^\circ \leq \theta \leq -160^\circ\}$ is also coarse. However, the sample density there is sufficient for clustering, so no initial conditions are added.

Even after refinement, noise tends to persist near cluster boundaries. Due to the density-based nature of clustering and the lack of clear, separable boundaries between trajectory geometries in a continuous system, the designation of some trajectories as noise cannot be completely eliminated. However, the adaptive clustering scheme successfully reduces the noise in these areas, as depicted in the inset of Figure 8. This noise reduction leads to better defined clusters across the dataset.

Figure 9 summarizes statistics of the dataset throughout the adaptive clustering scheme. The number of initial conditions, noise percentage, and number of local clusters is plotted across iterations of the loop, marked by bars of different colors. The vertical axes are displayed on a logarithmic scale. The bars are grouped together by the number of half-revolutions n, i.e. number of arcs, of the trajectories in the total data set. With the exception of $n = 10$, the noise percentage generally decreases throughout the refinement process. Ongoing work includes investigating the high number of trajectories designated at noise when $n = 10$. For example, for $n = 1$, the number of local clusters slightly decreases, as does the noise percentage, as more initial conditions are added.

Through the adaptive clustering process, the percentage of trajectories designated as noise drops from 11% to 2%. The final number of initial conditions is $187,470$, resulting in 1,205 local clusters which are aggregated to produce 955 global clusters. For comparison, generating a uniform initial position grid with a resolution of $\Delta r_{min} = 5 \times 10^{-4}$, would produce 1,688,528 initial conditions, 9 times larger than the size of the dataset after the adaptive clustering scheme. This increase in the size of the dataset would have significant implications for data storage requirements and the computational time required to propagate and cluster all the trajectories.

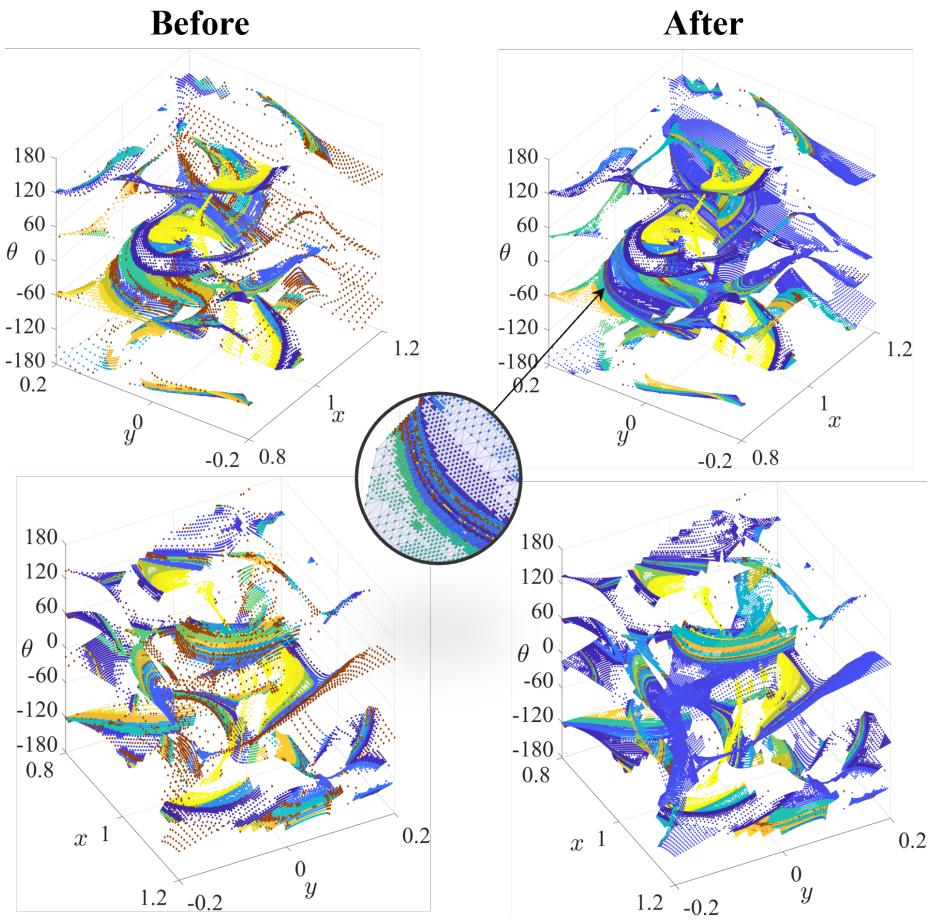

Figure 8. Initial condition mesh before and after the adaptive clustering loop. Inset shows the persistence of noise between cluster boundaries.

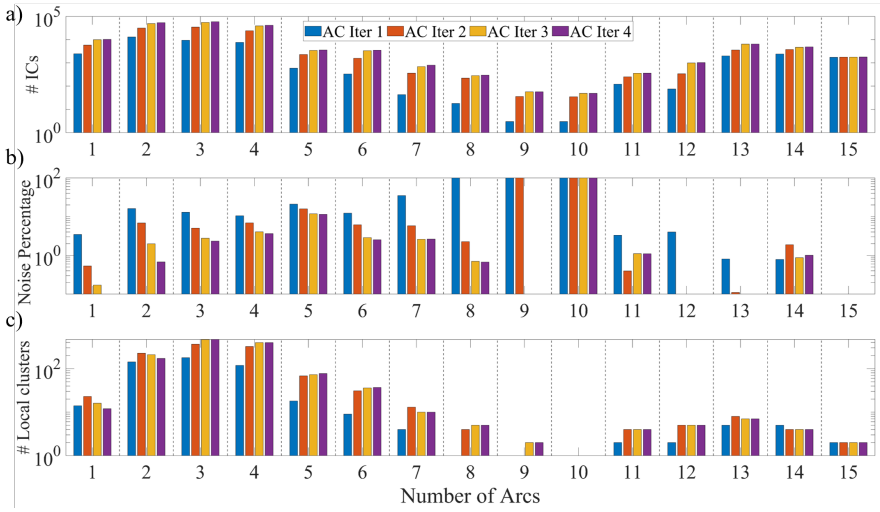

Figure 9. Data set statistics throughout the adaptive clustering loop.

CONCLUSIONS

This paper expands on the framework developed by Bosanac[15] by introducing an adaptive and automated scheme to generate the training dataset for the cluster-based summarization of trajectories in the planar Earth-Moon CR3BP. This paper demonstrates that, for planar motion, the set of curvature maxima belongs to a surface. This surface is discretized based on its geometric features by using differential geometry. The clustering framework is then embedded in a loop where the summary of motion is iteratively improved through the strategic addition of samples near noise points and dynamically sensitive regions, driven by the total absolute curvature of the trajectories. Consequently, the algorithm supplies a heterogeneous, automatic, and adaptive sampling procedure, densely sampling areas with greater dynamical complexity while sampling simpler regions more sparsely. As a result, the approach reduces the reliance on prior expertise about the specific environment and decreases the time required from a human analyst to generate the data set. This, in turn, facilitates the study of a wider range of systems while producing motion summaries that more accurately capture the diverse geometries exhibited by trajectories in a chaotic solution space.

ACKNOWLEDGMENT

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-24-1-0217. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

REFERENCES

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, ch. 10. New York, NY: Elsevier Science Technology, 3rd ed., 2011.

[2] C. E. V. Gallego, V. F. Gómez Comendador, F. J. Saez Nieto, and M. G. Martinez, "Discussion On Density-Based Clustering Methods Applied for Automated Identification of Airspace Flows," *IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 2018, pp. 1–10.

[3] A. Paccanaro, J. A. Casbon, and M. A. S. Saqi, "Spectral clustering of protein sequences," *Nucleic Acids Research*, Vol. 34, No. 5, 2006, pp. 1571–1580.

[4] I. Joncour, G. Duchêne, and E. Moraux, "Multiplicity and clustering in Taurus star-forming region-I. Unexpected ultra-wide pairs of high-order multiplicity in Taurus," *Astronomy & Astrophysics*, Vol. 599, 2017, p. A14.

[5] G. McLachlan, "Cluster analysis and related techniques in medical research," *Statistical Methods in Medical Research*, Vol. 1, No. 1, 1992, pp. 27–48.

[6] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma, "Recommending friends and locations based on individual location history," *ACM Trans. Web*, Vol. 5, No. 1, 2011.

[7] N. Nakhjiri and B. Villac, "Automated stable region generation, detection, and representation for applications to mission design," *Celestial Mechanics and Dynamical Astronomy*, Vol. 123, 2015.

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

[9] T. R. Smith and N. Bosanac, "Constructing motion primitive sets to summarize periodic orbit families and hyperbolic invariant manifolds in a multi-body system," *Celestial Mechanics and Dynamical Astronomy*, Vol. 134, Feb. 2022, p. 7.

[10] C. Gillespie, N. Bosanac, and G. E. Miceli, "Summarizing Natural and Controlled Motion in Cislunar Space with Behavioral Motion Primitives," *AAS/AIAA Space Flight Mechanics Meeting*, Kauai, HI, January 2025.

[11] N. Bosanac, "Data-Mining Approach to Poincaré Maps in Multi-Body Trajectory Design," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 6, 2020, pp. 1190–1200.

[12] S. Bonasera and N. Bosanac, "Applying data mining techniques to higher-dimensional Poincaré maps in the circular restricted three-body problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 133, 12 2021.

[13] N. Bosanac, "Clustering Natural Trajectories in the Earth-Moon Circular Restricted Three-Body Problem," 2025, Under Review.
[14] N. Bosanac and M. Joyner, "Data-Driven Summary of Continuous Thrust Trajectories in a Low-Fidelity Model of Cislunar Space," *AAS/AIAA Astrodynamics Specialist Conference*, Broomfield, CO, August 2024.
[15] N. Bosanac, "Data-Driven Summary of Motion in an Ephemeris Model of Cislunar Space," *AAS/AIAA Space Flight Mechanics Meeting*, Kauai, HI, January 2025.
[16] V. Szebehely, *Theory of Orbits: The Restricted Problem of Three Bodies*. London: Academic Press, 1967.
[17] R. S. Park, W. M. Folkner, J. G. Williams, and D. H. Boggs, "The JPL Planetary and Lunar Ephemerides DE440 and DE441," *The Astronomical Journal*, Vol. 161, No. 3, 2021, p. 105.
[18] J. D. Giorgini and J. S. S. D. Group, "NASA/JPL Horizons On-Line Ephemeris System," `https://ssd.jpl.nasa.gov/horizons/`.
[19] C. Froeschlé and E. Lega, "On the structure of symplectic mappings. The fast Lyapunov indicator: a very sensitive tool," *Celestial Mechanics and Dynamical Astronomy*, Vol. 78, No. 1-4, 2000, pp. 167–195.
[20] C. Froeschlé, E. Lega, and R. Gonczi, "Fast Lyapunov Indicators. Application to Asteroidal Motion," *Celestial Mechanics and Dynamical Astronomy*, Vol. 67, No. 1, 1997, pp. 41–62.
[21] M. Guzzo and E. Lega, "Theory and applications of Fast Lyapunov Indicators for the computation of transit orbits in the three-body problem," 2021.
[22] E. Lega, M. Guzzo, and C. Froeschlé, *Theory and Applications of the Fast Lyapunov Indicator (FLI) Method*, Vol. 915, pp. 35–54. 2016.
[23] K. L. Wardle, *Differential geometry*. Courier Corporation, 2008.
[24] T. Needham, *Visual Differential Geometry and Forms: A Mathematical Drama in Five Acts*. Princeton University Press, 2021.
[25] N. M. Patrikalakis and T. Maekawa, *Shape interrogation for computer aided design and manufacturing*, Vol. 15. Springer, 2002.
[26] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller, "Curvature-based transfer functions for direct volume rendering: Methods and applications," *IEEE Visualization, 2003. VIS 2003.*, IEEE, 2003, pp. 513–520.
[27] B. O'Neill, *Elementary differential geometry*. Elsevier, 2006.
[28] M. Gopi, S. Krishnan, and C. T. Silva, "Surface reconstruction based on lower dimensional localized Delaunay triangulation," *Computer Graphics Forum*, Vol. 19, Wiley Online Library, 2000, pp. 467–478.
[29] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon mesh processing*. CRC press, 2010.
[30] M. J. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics," *Journal of computational Physics*, Vol. 82, No. 1, 1989, pp. 64–84.
[31] R. Bank and A. Sherman, "Some Refinement Algorithms And Data Structures For Regular Local Mesh Refinement," *Applications of Mathematics and Computing to the Physical Sciences*, 1999.
[32] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, AAAI Press, 1996, p. 226–231.
[33] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, 2013, pp. 160–172.
[34] L. McInnes, J. Healy, S. Astels, *et al.*, "hdbscan: Hierarchical density based clustering.," *J. Open Source Softw.*, Vol. 2, No. 11, 2017, p. 205.
[35] C. Malzer and M. Baum, "A hybrid approach to hierarchical density-based cluster selection," *2020 IEEE international conference on multisensor fusion and integration for intelligent systems (MFI)*, 2020.
[36] N. Bosanac, "Data-Driven Summary of Natural Spacecraft Trajectories in the Earth-Moon System," *AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Montana*, 2023.
[37] N. Bosanac, "Curvature Extrema Along Trajectories in the Circular Restricted Three-Body Problem," *AAS/AIAA Astrodynamics Specialist Conference*, Broomfield, CO, August 2024.
[38] Y. Zheng and X. Zhou, *Computing with Spatial Trajectories*, ch. 1,5. Springer Publishing Company, Incorporated, 1st ed., 2011.
[39] G. Cardano, *The Great Art or The Rules of Algebra (Ars Magna)*. Cambridge, MA and London: The MIT Press, 1968. Originally published in 1545 as Ars Magna.
[40] H. Liu, Y.-S. Ong, and J. Cai, "A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design," *Structural and Multidisciplinary Optimization*, Vol. 57, No. 1, 2018, pp. 393–416.