An Improved Numerical Method for Three-dimensional Hyperbolic Lagrangian Coherent Structures using Differential Algebra

Jack Tyler[1] and Dr. Alexander Wittig[2]

[1)]*PhD Candidate, Astronautics Research Group, University of Southampton*

[2)]*Assistant Professor, Astronautics Research Group, University of Southampton*

(*Electronic mail: jack.tyler@soton.ac.uk)

(Dated: 14 April 2022)

In dynamical systems, it is advantageous to be able to identify separate regions of qualitatively different flow. Lagrangian Coherent Structures have been introduced to obtain separatrices between regions of qualitatively different flow in three-dimensional dynamical systems with arbitrary time-dependence. However, the numerical method to compute them requires obtaining derivatives associated with the system, often performed through the approximation of divided differences, which leads to significant numerical error and numerical noise. In this paper, we introduce a novel method for the numerical calculation of hyperbolic Lagrangian Coherent Structures using Differential Algebra called DA-LCS. As a form of automatic forward differentiation, it allows direct computation of the Taylor expansion of the flow, its derivatives, and its eigenvectors, with all derivatives obtained algebraically and to machine precision. It does so without *a priori* information about the system, such as variational equations or explicit derivatives. We demonstrate that this provides significant improvements in the accuracy of the Lagrangian Coherent Structures identified compared to finite-differencing methods in a series of test cases drawn from the literature. We also show how DA-LCS uncovers additional dynamical behaviour in a real-world example drawn from astrodynamics.

Lagrangian Coherent Structures are a generalisation of the concept of the invariant manifold to time-dependent systems, and allow the profiling of three-dimensional dynamical systems with arbitrary time-dependence. However, the numerical method to compute these requires the gradient of the leading eigenvector of the strain tensor for the flow, which is typically achieved using the approximation of divided differences. The grid size for this process must be chosen without *a priori* knowledge to effectively capture the desired dynamical behaviour, and is susceptible to significant numerical noise and numerical error. In this paper, we present DA-LCS, an improved numerical method for determining hyperbolic Lagrangian Coherent Structures in three-dimensional systems. DA-LCS uses Differential Algebra, a set of operations which allows the storage and manipulation of polynomials in a computer environment, to directly construct a Taylor expansion of the flow. Through a novel application of power law iteration using a matrix of DA objects, we are also able to obtain polynomial expansions of the leading eigenvector of the strain tensor for the flow, providing access to derivatives of the leading eigenvector to machine precision. We show that this yields more accurate Lagrangian Coherent Structures based solely on the dynamics of the system. DA-LCS is first shown to reproduce 'toy' problems commonly found in the literature to higher accuracy. Then, we exhibit the performance of DA-LCS on a test case from astrodynamics, in which DA-LCS is able to generate usable insight where divided differences fails to produce any.

I. INTRODUCTION

In dynamical systems, it is often useful to identify surfaces which promote or inhibit transport, as these separate regions of qualitatively different flow. For time-independent systems, one often determines the geometric location of the invariant manifolds, which partition phase space and are found by studying the system's behaviour over infinite time scales[1]. However, in time-dependent flows, the theory of invariant manifolds does not apply and there are no stable or unstable hyperbolic invariant manifolds. Instead, the behaviour of these systems is typically studied over fixed time-scales chosen to match some practical period of interest[2,3].

To overcome this problem, several methods for identifying barriers to transport in temporally aperiodic systems have been suggested. For example, one may study a number of heuristic flow

diagnostics[4], but many of these methods are only effective for simple flows and are dependent on the reference frame[2]. Being heuristic, they also often lack a proper theoretical foundation as to exactly what they are indicating.

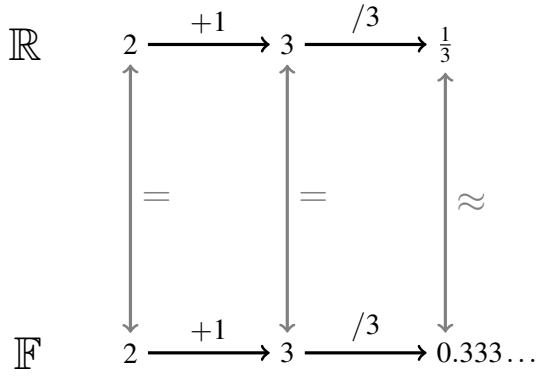
Lagrangian Coherent Structures (LCS) – a generalisation of the concept of invariant manifolds to unsteady flows – have emerged in the literature to solve this problem[5]. LCS are locally the most repulsive or attractive surfaces in a given flow, and dominate the nearby behaviour of the system. Several definitions of LCS have arisen in the literature (for a review, see Hadjighasem *et al.*[6]). A LCS was originally defined specifically for two-dimensional flows with respect to the Finite-Time Lyapunov Exponent (FTLE)[7–9], which measures the separation of particles which begin infinitesimally close. Under certain conditions, one can rigorously relate ridges of the FTLE field to repelling and attractive transport barriers[10–12].

A global, objective approach to the construction of Lagrangian Coherent Structures was presented in Blazevski and Haller[13]. The authors provide both the theoretical underpinning and a practical algorithm to directly construct transport barriers as parameterised surfaces by growing material surfaces which impose locally extreme deformation on nearby sets of initial conditions. These surfaces are shown to be necessarily orthogonal to certain eigendirections of the Cauchy-Green strain tensor of the flow, $C_{t_0}^T$, which quantifies the deformation of a flow. This approach is valid for three-dimensional flows with general time-dependence and over arbitrarily-chosen time periods of observation.

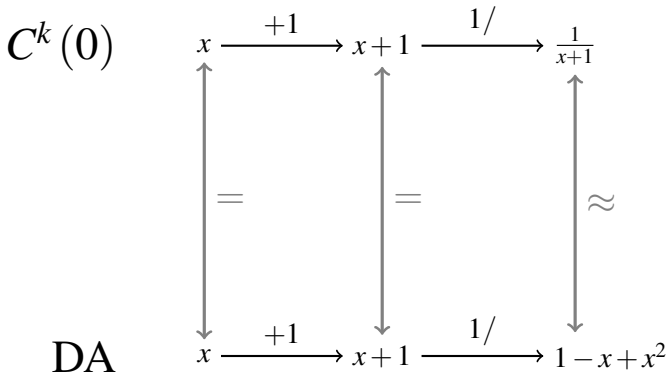
However, there are several computational complexities associated with computing LCS using this global, objective approach[11], such as the need to account for degenerate points and oriental discontinuities in the eigenvector field of $C_{t_0}^T$. More importantly, the eigenvectors of $C_{t_0}^T$ must be computed precisely, yet are very sensitive to numerical errors. These errors are particularly troublesome near regions of intense attraction or repulsion, since large errors in $C_{t_0}^T$ can quickly accumulate, yet these are also the exact regions where one would expect a LCS. The approximation of the derivatives of a flow using finite differencing is often used[2,14–16], but this method is particularly sensitive to the grid-size chosen, which must be carefully selected to account for flow behaviour over different spatial scales, and is generally difficult to determine *a priori*. Other such methods for approximating derivatives exist, such as the use of variational equations, where one manually derives and implements a set of adjoint differential equations that are propagated along with a reference trajectory[17]. While this approach yields derivatives as accurate as the propagation along the reference trajectory, it requires the derivation, implementation and integration of

n^2 additional equations for the first derivatives of an n dimensional flow, and another $n^2(n+1)/2$ equations for the second flow derivatives.

Separately, Differential Algebra (DA) was originally introduced to compute high-order transfer maps for particle accelerator systems[18]. This approach constructs a Taylor series representation of an arbitrary map in a dynamical system, and has since seen widespread use in the study of non-linearities[19–21], the management of uncertainties[22–25], and as a tool for automatic differentiation[26,27] in a wide variety of fields. Crucially, the derivatives found using DA are accurate to machine precision, unlike other numerical methods such as divided differences, and there is no need to derive or implement any additional equations beyond the system itself.



(a) Evaluation of $(2+1)/3$ in the field of real numbers $\mathbb{R}$ (top) and in the floating-point approximation to $\mathbb{R}$, $\mathbb{F}$ (bottom). Each operation in $\mathbb{R}$ has a corresponding operation in $\mathbb{F}$.



(b) Evaluation of $1/(x+1)$ in the $k-$times differentiable functions C^k (top) and truncated polynomials of order 2 represented by DA (bottom). Each operation in $C^k(0)$ has a corresponding operation in DA, approximating the resulting function in $C^k(0)$ by its Taylor expansion around 0.

FIG. 1: Comparison between the field of real numbers $\mathbb{R}$ and function space C^k, and their respective computer representations. The subfigures are taken from Wittig[28].

In this paper we introduce DA-LCS, which uses DA to improve the numerical method presented in Blazevski and Haller[13] for determining hyperbolic LCS. First, in Section III A we briefly recapitulate how polynomial expansions of arbitrary flows of an ordinary differential equation (ODE) can be calculated, with applications to obtaining flow derivatives of arbitrary systems to machine precision. Next, in Section III B we introduce a novel use of DA to construct algebraic expansions

of the leading eigenvector of a matrix of polynomials. Both of these techniques are then combined to form the DA-LCS algorithm for computing LCS in three-dimensional flows. In Section V, we demonstrate that this method works well in reproducing results for commonly-used 'toy' problems from the literature. Lastly, in Section VI we present the application of DA-LCS to a more complex system from astrodynamics where the traditional method fails to produce usable results.

II. MATHEMATICAL BACKGROUND AND NOTATION

We study the behaviour of a dynamical system

$$\dot{\boldsymbol{x}} = f\left(x,\, t\right), \boldsymbol{x} \in D \subset \mathbb{R}^n, t \in [t_0, t_0 + T] \tag{1}$$

where f is a smooth vector field considered over some time T starting at time t_0. Denoting a trajectory of the dynamical system starting at position $\boldsymbol{x}_0$ at time t_0 as $\boldsymbol{x}\left(t_0, \boldsymbol{x}_0; T\right)$, the flow map of Equation 1 is given by

$$\boldsymbol{F}_{t_0}^{T} : \begin{cases} D \to D \\ \boldsymbol{x}_0 \mapsto \boldsymbol{x}\left(t_0, \boldsymbol{x}_0; T\right) \end{cases} \tag{2}$$

which is assumed to be at least k-times continuously differentiable. The Jacobian of this flow map, $\nabla F_{t_0}^{T}$, defines the right Cauchy-Green Strain Tensor (CGST) $C_{t_0}^{T}$, which describes the local deformation of the flow at the end of a given trajectory.

$$C_{t_0}^{T} = \left(\nabla \boldsymbol{F}_{t_0}^{T}\right)^{\top} \left(\nabla \boldsymbol{F}_{t_0}^{T}\right) \tag{3}$$

with $^{\top}$ denoting the matrix transpose. $C_{t_0}^{T}$ is positive-definite and symmetric, with real eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and associated real eigenvectors $\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots, \boldsymbol{\zeta}_n$.

The dominant eigenvalue λ_n can be used to calculate the finite-time Lyapunov exponent (FTLE), a measure of maximum separation of two particles advected forward under Equation 1 that start out infinitesimally close to each other:

$$\sigma_{t_0}^{T} = \frac{1}{2} \frac{\log \lambda_n}{T}. \tag{4}$$

Many previous studies have leveraged the FTLE field as a heuristic indication of high regions of separation in the flow. While the FTLE has been shown to be insufficient to indicate LCS alone[11], the FTLE is a commonly-used metric and is thus used in this paper to preliminarily highlight system behaviour and identify possible locations of high stretch or strain.

III. DIFFERENTIAL ALGEBRA

In the following, we give a very brief introduction to Differential Algebra. For a comprehensive treatment, the reader is referred to the literature[29].

Differential Algebra can be used as a tool to compute the derivatives of functions within a computer environment[29,30]. Similar to how computers represent the field of real numbers as floating-point numbers, DA allows the representation and manipulation of functions in a computer[31].

Consider two real numbers a and $b \in \mathbb{R}$. The approximation to a and b in a computational environment is their floating-point representation $\bar{a}, \bar{b} \in \mathbb{F}$, which essentially stores a set number of digits of its binary expansion. Any operation defined in $\mathbb{R}$, $\square$, has a corresponding operation in $\mathbb{F}$, $\boxtimes$, defined such that the result is another floating-point approximation of the operation on the real numbers a and b, i.e. $\bar{a} \times \bar{b}$ commutes with the floating-point representation of $a \times b$, $\overline{a \times b}$.

Similarly, now consider two functions, c and d, which are sufficiently smooth, $k-$differentiable functions of n variables: $c, d : \mathbb{R}^n \to \mathbb{R}$. In the DA framework, a computer operates on the multivariate Taylor expansion of c and d, $[c]$ and $[d]$, with corresponding operations to those defined in the real function space, such that the operation of $[c] \cdot [d]$ commutes with the DA representation of the product $[c \cdot d]$.

An example to demonstrate how real numbers are approximated in a computer environment is provided in Figure 1a for the evaluation of the expression $1/(x+1)$ for $x = 2$ in $\mathbb{F}$ and $\mathbb{R}$. In Figure 1a, we begin with $x = 2$, perform the operation $+1$ to obtain three, and then perform the operation $1/$ to compute the final expression. In $\mathbb{R}$, we obtain the solution $1/3$, and in $\mathbb{F}$ we obtain the solution $0.333\ldots$ up to the limit of precision of the type. The final result of the evaluation in floating-point arithmetic is an approximation of the real computation.

In Figure 1b, we evaluate the same expression comparing the evaluation in the space $C^k(0)$ of real functions, and the DA arithmetic of one variable and expansion order 2. We begin with the function $c(x) = x$, and perform the operation $+1$ followed by the operation $1/$, yielding $1/(x+1)$ in the real function space, and $1 - x + x^2$ in the DA arithmetic. The result of the DA arithmetic is the Taylor expansion of $1/(x+1)$ which represents the function exactly at $x = 0$, and approximates the function locally near $x = 0$ with an error of $\mathscr{O}\left(x^3\right)$.

Under the term Differential Algebra we subsume a full set of elementary operations to efficiently operate on these multivariate expansions, including operations for common intrinsic functions such as division, square roots, trigonometric functions, and exponentials, as well as oper-

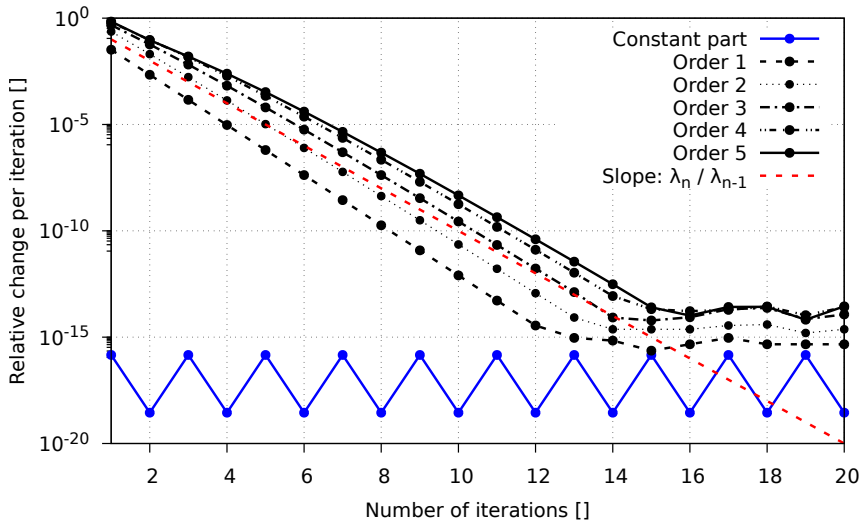


FIG. 2: Relative error across all polynomial orders in successive applications of $\left[C_{t_0}^T\right]$ to an initial guess containing only the floating-point dominant eigenvector at the expansion point as the constant part. Higher expansion orders (black) can all be seen converging at around the expected convergence rate λ_n/λ_{n-1} (dashed red) towards the floating-point floor.

ations for differentiation and integration. Some traditional floating-point algorithms have also been made available in DA[23]. An important application of DA widely used in both the literature and this paper is the high-order expansion of the solution of an ODE as a function of the initial conditions[23,25], which is discussed in more detail in Subsection III A. In this paper, we use the Differential Algebra Computational Engine[26] (DACE) to operate on polynomial expansions ('DA objects' or 'DAs').

A. Flow expansions to arbitrary order using Differential Algebra

A key advantage in using DA is the ability to compute the derivatives of a flow completely algebraically and to machine precision, without the need to derive, implement and compute the variational equations for the system, or approximate derivatives using divided differences.

To demonstrate this, consider as an example the initial value problem (IVP) given by

$$\begin{cases} \dot{\boldsymbol{x}} = f(\boldsymbol{x},\, t) \\ \boldsymbol{x}(t_i) = \boldsymbol{x}_i. \end{cases} \tag{5}$$

This type of ODE is commonly solved numerically using Runge-Kutta schemes. As these schemes

are merely a sequence of operations, they can also be evaluated in DA.

As an example, suppose we solve this IVP using the forward Euler scheme, the simplest of the Runge-Kutta family. A single step in this scheme is given explicitly by the expression

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \Delta t f\left(\boldsymbol{x}_{i-1}\right). \tag{6}$$

To change this into a DA method, we take an initial floating-point representation of the initial condition $\boldsymbol{x}(t_0)$, and substitute the initial value with the DA identity, $[\boldsymbol{x}(t_0)] = \boldsymbol{x}(t_0) + \boldsymbol{\delta x}$, in which $\boldsymbol{x}(t_0)$ is the reference point for the expansion. Evaluating the Euler scheme and the right hand side f with DA operations, we obtain the time-step

$$[\boldsymbol{x}_1] = [\boldsymbol{x}_0] + \Delta t f\left([\boldsymbol{x}_0]\right). \tag{7}$$

Repeating this time step until the final step $\left[\boldsymbol{x}_f\right]$, the final output of the Euler scheme will be a k-th order expansion of the flow $[F(t_0, [\boldsymbol{x}_0]; t)]$ at the final time T with respect to the initial condition.

The forward Euler method is of course merely for demonstration and not used in practice. Following the same procedure, however, most explicit ODE integration schemes can be readily modified to support DA. In this work, we rely on the templated Boost C++ library's 7/8 Dormand-Prince numerical integration scheme to perform numerical integration.

Since Boost is templated, its numerical integrators can be adapted to support DA with only few changes. The first is the replacement of floating-point operations with their corresponding DA operations; this is straightforward since Boost supports operator overloading. The main change is how norms are calculated for error estimation in the integrator when using DA. Evaluating the usual L_2 norm of a vector $|\boldsymbol{x}| = \sqrt{\sum_{i=0}^{n} x_i^2}$ in DA yields another DA object representing a polynomial. As there is no ordering on the space of polynomials, this cannot be directly compared to some tolerance. Instead, we have to define the norm of a DA object which maps it into the non-negative real numbers. In this application, the norm of a DA object is taken to be the largest absolute value of any coefficient of the expansion in any order. Considering all orders in the norm allows the usual step-size control algorithms of embedded Runge-Kutta methods to control the error in all orders of the expansion, rather than just the constant part.

B. Polynomial expansions of leading eigenvectors of $C_{t_0}^T$ to arbitrary order

From the DA expansion of the final condition with respect to the initial condition obtained previously, DA makes it particularly easy to assemble an expansion of $C_{t_0}^T$. Since derivatives of

polynomials are straight forward to compute, DA provides built-in operators for differentiation in the i-th independent variable ∂_i. This means we can directly evaluate the Jacobian as

$$\left[\nabla \boldsymbol{F}_{t_0}^T\right]_{ij} = \partial_j \left[\boldsymbol{x}\right]_{t_0,\,i}^T \tag{8}$$

from which a polynomial expansion of $C_{t_0}^T$ can be assembled yielding

$$\left[C_{t_0}^T\right] = \left[\nabla \boldsymbol{F}_{t_0}^T\right]^\top \left[\nabla \boldsymbol{F}_{t_0}^T\right]. \tag{9}$$

Note that the constant part of $\left[C_{t_0}^T\right]$ is the CGST at the expansion point accurate to machine precision, that is it is the same as would be approximated with via divided differences. The remaining higher order terms represent an expansion of the CGST in the neighbourhood around the expansion point.

To compute the LCS, the derivatives of the leading eigenvector of the Cauchy-Green strain tensor with respect to position are required. While divided differences can in principle again be used to obtain these derivatives, the method is susceptible to numerical noise and it is difficult to determine the most appropriate grid sizes to use. Moreover, eigenvectors are only defined up to a sign, and thus care must be taken when taking the derivatives that nearby eigenvectors have 'smooth' changes in orientation.

Instead, we use a novel application of DA to obtain an expansion of the leading eigenvector of a matrix of DAs, which then can once again be differentiated directly in DA. To do this, we first have to obtain the DA representation of the leading eigenvector. We simply use power (von Mises) iteration[32] performed in DA, which in its floating-point variant is a well-established algorithm[33].

Power iteration performs the repeated evaluation of an arbitrary starting vector b through a matrix A to obtain an approximation to its dominant unit eigenvector through the recurrence relation

$$\boldsymbol{b}_{m+1} = \frac{A\boldsymbol{b}_m}{||A\boldsymbol{b}_m||} \tag{10}$$

where $|\cdot|$ represents a vector norm, here taken to be the L_2 norm, the vector $\boldsymbol{b}_0$ is an arbitrary initial vector, and m is the number of iterations. The vector $\boldsymbol{b}$ will converge provided that the starting vector $\boldsymbol{b}_0$ has a nonzero component in the direction of the dominant eigenvector, and A has a unique largest eigenvalue by absolute value. The theoretical convergence rate of the method between successive iterations is $|\lambda_n/\lambda_{n-1}|$. Practically, the recurrence relation is iterated until the stopping condition $||\boldsymbol{b}_{m+1} - \boldsymbol{b}_m|| \leq \delta$ is first valid, where $\delta > 0$ is a pre-set tolerance and the norm is again taken to be an L_2 norm.

Converting this algorithm to DA, let A now be a DA matrix with DA objects in each entry, $[A]$. Iterating it on a DA vector $[\boldsymbol{b}_0]$ for a sufficiently large number of iterations m yields a DA vector $[\boldsymbol{b}]$ corresponding to the dominant eigenvector of $[A]$ with a polynomial expansion in each entry, that is

$$[\zeta_n]_{m+1} = \frac{[A]\,[\boldsymbol{b}_0]_m}{||[A]\,[\boldsymbol{b}_0]_m||} \tag{11}$$

where ζ_n is the dominant eigenvector. Note that here the norm in the denominator is simply a DA evaluation of the L_2 (Euclidean) norm $|\boldsymbol{x}| = \sqrt{\sum_{i=0}^{n} x_i^2}$. We generalise the stopping condition from floating-point computation such that we iterate until there is no change in any order in any entry of $\left([\boldsymbol{b}]_{m+1} - [\boldsymbol{b}]_m\right)$ above a pre-set tolerance $\delta > 0$. We set δ to be 10^{-12} in this paper.

To speed up convergence, and because eigenvector solvers for floating-point computations are readily available and highly efficient, we set the initial guess for $[\boldsymbol{b}_0]$ to have a constant part equal to the dominant eigenvector of the constant part of $[A]$, since we know by construction that this will be the constant part of $[\zeta_n]$.

An example of the convergence of this method is illustrated in Figure 2, which shows the maximum relative change of coefficients in $[\boldsymbol{b}]$ separated by their expansion order over repeated application of $\left[C_{t_0}^T\right]$ to the initial guess of the dominant eigenvector of a trajectory in the periodic ABC flow (Section V B). The theoretically expected rate of convergence λ_n/λ_{n-1} can clearly be seen in the plot as a dashed red line. While all orders converge at approximately the expected rate, the floating-point portion of the expression converges instantly as it was already set to the double-precision representation of the leading eigenvector.

Once the eigenvector $[\zeta_n]$ is expanded to at least first order, the curl $\nabla \times \zeta_n$ of the eigenvector field, which is used in the calculation of the helicity during LCS construction, can be computed by simply applying the DA derivative operator again.

In order to obtain the value of $\nabla \times \zeta_n$ at the expansion point, the flow map $\boldsymbol{F}_{t_0}^T$ must be computed at least to order 2. This is because one derivative is taken in the construction of $C_{t_0}^T$ (Subsection III A), and another is then taken in $\nabla \times \zeta_n$, both of which reduce the order of the expansion by one.

IV. LAGRANGIAN COHERENT STRUCTURES

In this section we briefly review the method for computing LCS in three-dimensional systems given in Blazevski and Haller[13]. For a more in-depth discussion, the reader is directed to the original paper. Once the mathematical formulation is introduced, we show how the method is

computed numerically and outline the changes made from the literature in DA-LCS.

The full, three-dimensional LCS, which is defined as a surface that is locally maximally repulsive or attractive over a given time interval $[t_0, T]$, is constructed from its intersections with a family of hyperplanes $\mathscr{S}$. These intersections are called *reduced strainlines* and *reduced stretchlines*[13].

In the following, we show the mathematical formulation for repulsive LCS, whose structure is derived from the dominant eigenvector $\boldsymbol{\zeta}_n$ and whose intersections with $\mathscr{S}$ are the *reduced strainlines*. A similar procedure applies to $\boldsymbol{\zeta}_1$ (reduced stretchlines) to obtain attracting LCS. Once the mathematical formulation is introduced, we show a practical numerical implementation to compute approximations of the LCS and outline the changes made from the literature in DA-LCS.

At any point $\boldsymbol{s}$ on the hyperplane, we define the reduced strainline through that point as follows: using Equation 1, the point is propagated from time t_0 to time T, and $C^T_{t_0}$ and its eigenvectors are computed. The tangent of the reduced strainline at $\boldsymbol{s}$ is orthogonal to the leading eigenvector $\boldsymbol{\zeta}_n$ of $C^T_{t_0}$ and of course also lies within the hyperplane. This is true for any point on the strainline, allowing their parameterisation to be described by the ODE:

$$\boldsymbol{s}' = \hat{\boldsymbol{n}}_{\mathscr{S}} \times \boldsymbol{\zeta}_n \tag{12}$$

where $\hat{\boldsymbol{n}}_{\mathscr{S}}$ is the unit normal to the surface at $\boldsymbol{s}$. Since the eigenvector is only defined up to the sign, we integrate the strainline in both directions corresponding to $\pm\boldsymbol{\zeta}_n$ to capture the entire strainline structure.

Strainlines which have zero *helicity* $H_{\boldsymbol{\zeta}_n}$

$$H_{\boldsymbol{\zeta}_n} = \langle \nabla \times \boldsymbol{\zeta}_n, \boldsymbol{\zeta}_n \rangle, \tag{13}$$

where $\langle \cdot, \cdot \rangle$ is the inner product, are the intersections of the LCS with the reference hyperplane. The strainlines that form part of the LCS are identified by starting the integration of Equation 12 at initial points with zero helicity.

This analysis is repeated for each of the hyperplanes in $\mathscr{S}$. The strainlines forming part of the LCS on each hyperplane are then interpolated to produce the full 3D structure of the LCS.

To numerically implement the above procedure, we first sample points on each hyperplane in $\mathscr{S}$ on a uniformly-spaced grid and compute the helicity $H_{\boldsymbol{\zeta}_n}$ at each point. The ODE in Equation 12 is then rewritten in discretised form as

$$\boldsymbol{s}'_i = \operatorname{sign}(\boldsymbol{\zeta}_{i,n} \cdot \boldsymbol{\zeta}_{i-1,n})\, \hat{\boldsymbol{n}}_{\mathscr{S}} \times \boldsymbol{\zeta}_{i,n} \tag{14}$$

where s_i is the $i-$th point on the strainline and the term $\zeta_{i,n} \cdot \zeta_{i-1,n}$ is introduced to enforce continuity in the vector field by selecting the direction most closely aligned with the previous tangent vector. The selection of zero-helicity initial grid points is relaxed by allowing points where the helicity H_{ζ_n} is below some tolerance $\alpha > 0$. The numerical integration of the ODE along the strainline continues until the sum of the helicity at each s_i divided by the number of steps performed (i) rises above α.

In previous literature, divided differences was used to numerically approximate $C_{t_0}^T$ and $\nabla \times \zeta_n$ required for this procedure[13,15], which can lead to significant numerical error. Divided differences can either be applied on the same grid on which points are sampled, or on a finer grid used solely for the purpose of approximating the derivatives. In DA-LCS, we instead use the flow expansion described in Section III A to compute $C_{t_0}^T$, and the eigenvector expansion in Section III B to compute $\nabla \times \zeta_n$ to high accuracy and without the need to alter grid sizes through trial-and-error.

The trajectories obtained through either method are segments of strainlines forming the LCS. However, since different initial points can belong to the same strainline, the trajectories often overlap. They must, therefore, be filtered to provide a single, continuous curve. Given a suitable metric d_F of how close two strainline segments are, the shorter of the two strainlines is discarded whenever d_F is below some threshold.

In Blazevski and Haller[13], this metric was the Hausdorff distance, a measure of similarity between two curves. We find that we obtain qualitatively better strainlines when using the Fréchet distance as a metric, which is recognised as a better measure of similarity than the Hausdorff distance in trajectory clustering problems[34,35].

It is defined as follows[34]: given two curves A and B that are continuous mappings from $[0,\,1]$ to $\mathbb{R}^n$, define a re-parameterisation of each curve as an injective function $\Pi : [0,\,1] \mapsto [0,\,1]$, such that $\Pi(0) = 0$ and $\Pi(1) = 1$. The Fréchet distance d_F between A and B is then defined with respect to their respective re-parameterisations Π and Λ such that

$$d_F = \inf_{\Pi,\,\Lambda} \max_{m \in [0,\,1]} \left\{ d_E \left(A\left(\Pi(m)\right),\, B\left(\Lambda(m)\right) \right) \right\} \tag{15}$$

where d_E is the Euclidean distance.

V. ARNOLD-BELTRAMI-CHILDRESS FLOWS

We now apply the standard approach of divided differences and the DA-LCS method to several variations of the Arnold-Beltrami-Childress (ABC) flow, as studied in Blazevski and Haller[13]. For

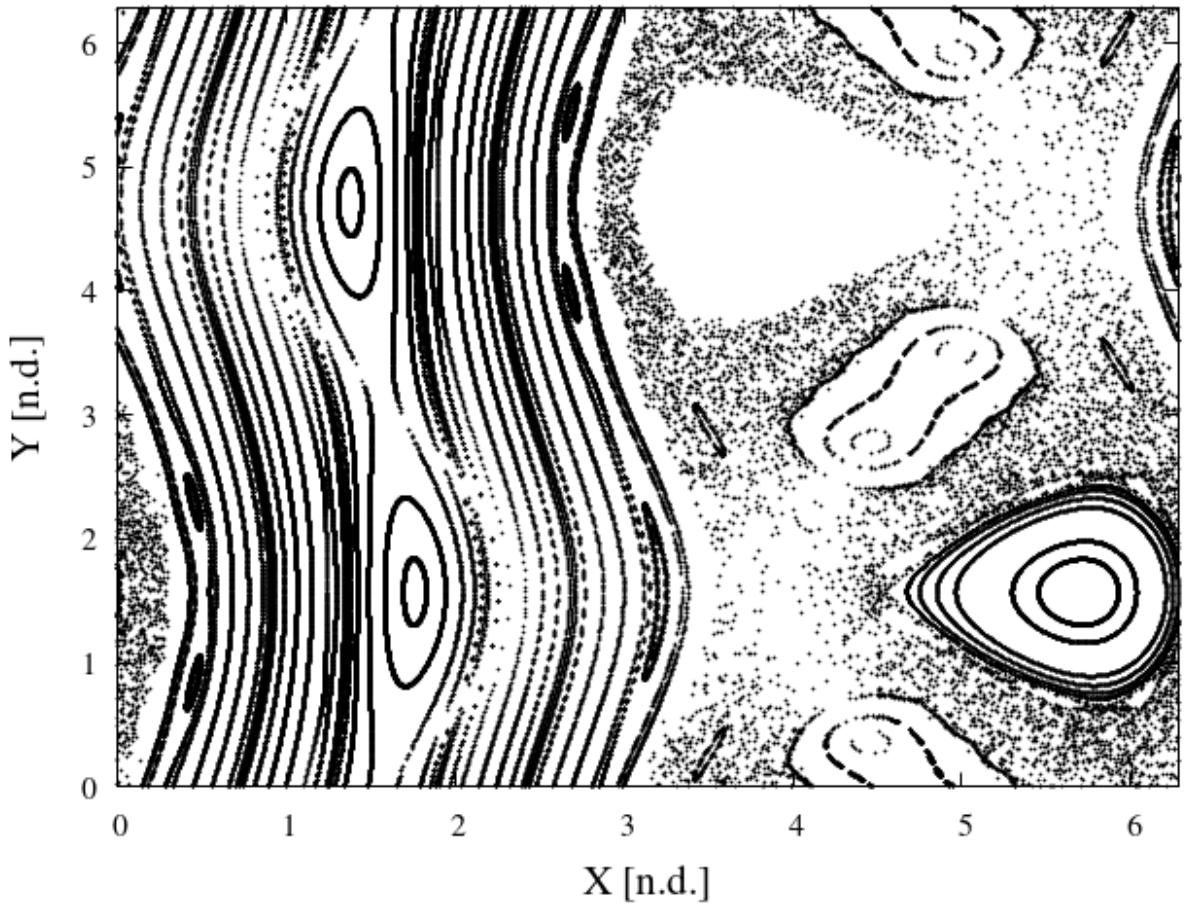
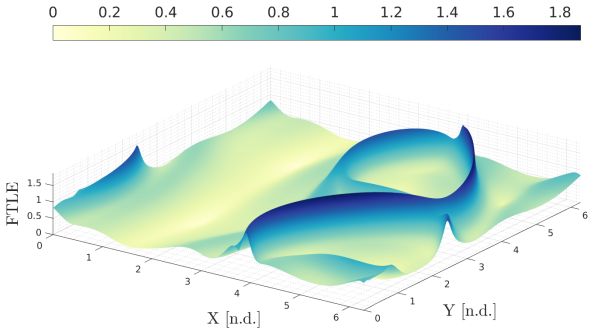


FIG. 3: Poincaré section (return map) for the steady ABC flow on the $z = 0$ plane; generated using a 15×15 grid of initial points with integration time $T = 1500$.

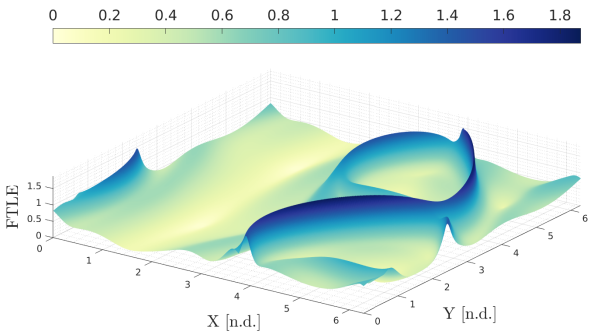
each example, we present the equations of motion, the FTLE field, the helicity field, and the resulting strainlines. The results obtained using divided differences each use the manually determined optimal grid size that produces the qualitatively 'best' results, to allow for a fair comparison. No such adjustments are needed when using DA-LCS.

A. Steady Arnold-Beltrami-Childress flow

We first consider the steady Arnold-Beltrami-Childress flow, as presented in Blazevski and Haller[13]. The ABC flow is an exact solution to Euler's equation, and its equations of motion in

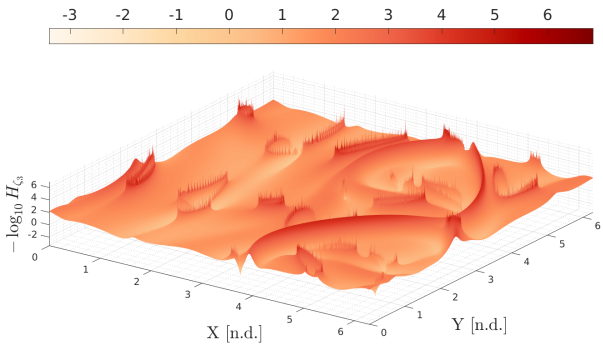


(a) FTLE field obtained using DA-LCS.

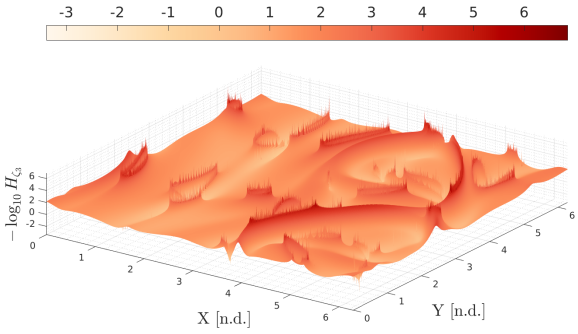


(b) FTLE field obtained using divided differences with auxiliary grid spacing of 0.05 of the nominal grid spacing in all directions.

FIG. 4: Finite-time Lyapunov field for the steady ABC flow from $t = 0$ to $T = 3$ using DA-LCS and divided differences. The fields strongly agree, suggesting that computing C_0^3 using divided differences is not a major source of error in this example.



(a) Helicity field obtained using DA-LCS.



(b) Helicity field obtained using divided differences with auxiliary grid spacing of 0.05 of the nominal grid spacing in both x and y. The same grid is used in computing both C_0^3 and $\nabla \times \zeta_n$.

FIG. 5: $-\log H_{\zeta_3}$ for the steady ABC flow from $t_0 = 0$ to $T = 3$ using DA-LCS and divided differences. Again both strongly agree, showing that DA-LCS is working. The DA-LCS structure is a little smoother along the main ridge on the right compared to divided differences, making integrating the strainline along the ridge more robust.

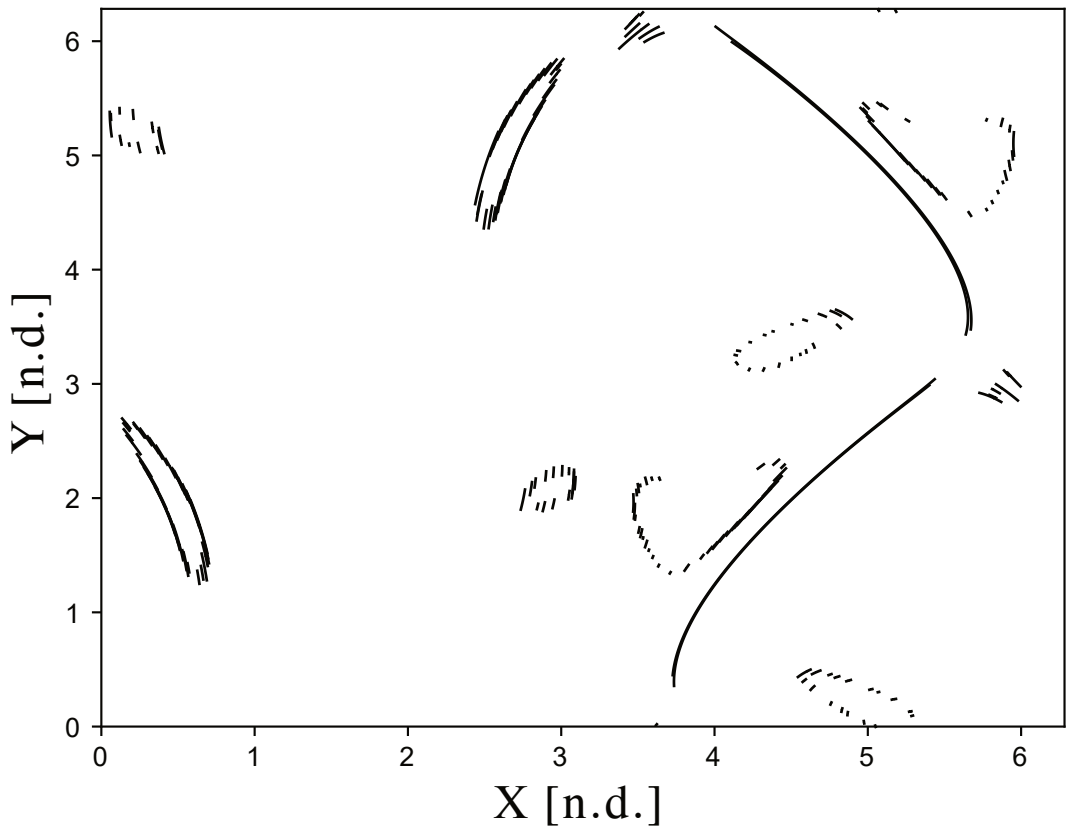


FIG. 6: Final, filtered strainlines for the steady ABC flow on the $z = 0$ plane computed using DA-LCS. The structure is formed of approximately 240 strainline segments.

Cartesian coordinates are

$$\dot{x} = A\sin z + C\cos y \tag{16}$$

$$\dot{y} = B\sin x + A\cos z \tag{17}$$

$$\dot{z} = C\sin y + B\cos x \tag{18}$$

and parameter values $A = \sqrt{3}, B = \sqrt{2}, C = 1.0$ as in Blazevski and Haller[13]. To illustrate the behaviour of this system, the Poincaré section in the x-y plane is shown in Figure 3, computed from a regular 15×15 grid of initial points and an integration time of $T = 1500$.

For the LCS computation, matching previous literature the set of reference planes are taken to be

$$\mathscr{S} = \{(x,y,z) \in [0,2\pi]^3 \ : \ z \in \{0, 0.005, 0.01 \ldots, 0.1\}\},$$

that is the *x*-*y* plane evenly spaced along the z axis. However, within each plane we alter the grid size used. Blazevski and Haller[13] use a 500×500 grid on which to compute the underlying helicity field, and then sample seed points for the ODE in Equation 12 on a reduced grid of 600×10. To simplify analysis and ensure we capture all of the flow's behaviour we perform all stages of the analysis on a 1000×1000 grid defined for each hyperplane in $\mathscr{S}$.

The system defined by Equations 16-18 is integrated forward for 3 non-dimensional time units using the DA-compatible numerical integrator introduced previously, with an integration tolerance of 10^{-13}. A helicity tolerance of $\alpha = 10^{-4}$ is applied to determine seed points and terminate the numerical integration. A minimum distance of $d_F = 0.04$ is used in the strainline segment filtering. Both of these parameters are chosen from visual examination of the helicity field and resulting strainline structure for all of the examples in this paper.

The FTLE fields on the $z = 0$ plane for this flow, computed using DA-LCS and divided differences, are shown in Figures 4a and 4b, respectively. The two FTLE fields are very similar, which suggests that the computation of C_0^3 and its dominant eigenvalue agrees across the two methods.

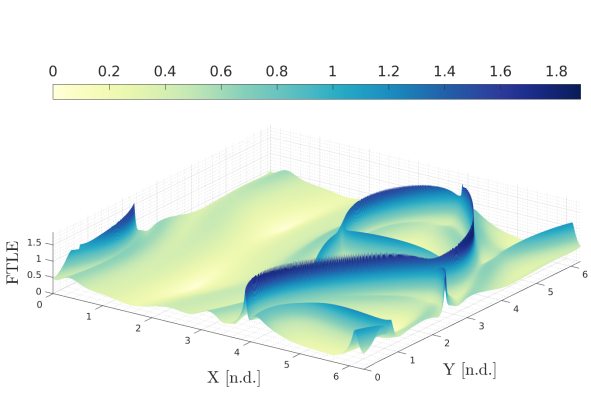
In the DA-LCS and divided difference helicity fields on the $z = 0$ plane, shown in Figures 5a and 5b respectively, some first differences can be seen. While the two methods qualitatively agree on the structure of the field, the DA-LCS method produces smoother peaks and ridges in the field for the primary features in the flow. This is particularly visible on the main ridge in the bottom right corner around $X = 4$ and $Y = 1$. The smoother ridges provide a more well-defined path for the ODE in Equation 12 to track.

The resulting strainlines on the $z = 0$ plane for this flow are shown in Figure 6, and follow the expected structure from the helicity field presented in Figure 5a. The strainline integration is approximately 50% quicker using DA-LCS than using divided differences, since it requires only one DA integration of second order per point, while divided differences requires 42 floating-point integrations. This corresponds to determining $\nabla \times \zeta_n$ using six adjacent grid points plus the eigenvector itself, each of which requires a further six integrations for the respective $C_{t_0}^T$ approximation. The overhead associated with computing the second order flow expansion in DA in this case are lower than performing 42 integrations in double-precision.

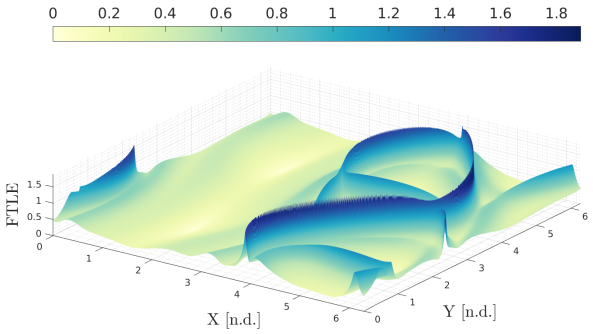
We note the existence of several 'loops' in the helicity field, particularly in the left-hand side of the field. The strainline segments at these points grow transverse to the ridges at certain points, and do not track along the ridge as would be expected. This behaviour is also present when computing LCS with divided differences. These small strainline segments are not present in Blazevski and

Haller[13] due to being missed by the largely reduced 600×10 grid resolution used there. This explains their omission from the literature, and we do not investigate this issue further here, although we note the existence of similar structure in Palmerius, Cooper, and Ynnerman[36].

The total strainline structure in Figure 6 for this test case is formed of approximately 240 individual strainline segments. We remark that the distribution of the number of strainlines with respect to their length is largely bimodal. The 'loops' discussed previously contain lots of short segments, while the main wishbone-like structures are formed from only several long strainlines. This distribution of the number of strainline segments with respect to their length is similar across all test cases studied here that are variations of the ABC flow.

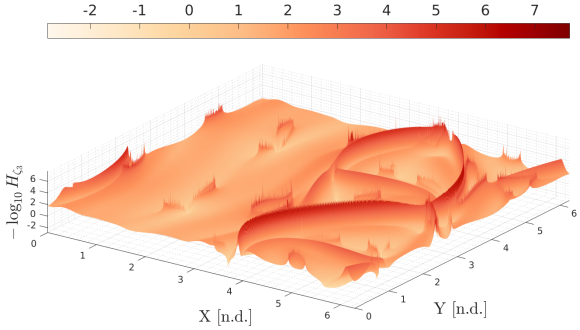


(a) Computed using DA-LCS.

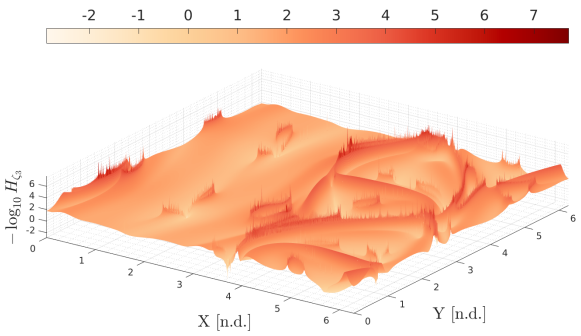


(b) FTLE field obtained using divided differences with auxiliary grid spacing of 0.05 of the nominal grid spacing in all directions.

FIG. 7: Finite-time Lyapunov exponent field for the periodic ABC flow from $t_0 = 0$ to $T = 4.0$, obtained using DA-LCS and divided differences. Again, the FTLE field agrees between divided differences and DA-LCS, suggesting divided differences on the correct auxiliary grid in this case accurately approximates C_0^4.



(a) Computed using DA-LCS.



(b) Helicity field obtained using divided differences with auxiliary grid spacing of 0.05 of the nominal grid spacing in all directions. The same grid is used in computing both C_0^4 and $\nabla \times \zeta_n$.

FIG. 8: $-\log H_{\zeta_3}$ for the periodic ABC flow computed using DA-LCS and divided differences from $t_0 = 0$ to $T = 4.0$. Here DA-LCS highlights in particular the main ridge on the right more clearly and smoothly than divided differences.

B. Periodic Arnold-Beltrami-Childress Flow

We now consider a time-periodic version of the Arnold-Beltrami-Childress flow with equations of motion

$$\dot{x} = (A + 0.1\sin t)\sin z + C\cos y \tag{19}$$

$$\dot{y} = B\sin x + (A + 0.1\sin t)\cos z \tag{20}$$

$$\dot{z} = C\sin y + B\cos x. \tag{21}$$

The hyperplanes $\mathscr{S}$ and grids are the same as in the case of the steady ABC flow, but now with integration times $t_0 = 0$ and $T = 4$. A helicity tolerance of $\alpha = 5 \times 10^{-5}$ is used, with a distance threshold $d_F = 0.02$.

Mirroring the analysis in the steady case, the FTLE fields for both DA-LCS and divided differences are shown in Figures 7a and 7b, respectively. Again, there is little qualitative difference between the two fields. The differences in smoothness in the helicity fields are, however, more pronounced between Figures 8a and 8b. The main wishbone-like structure is particularly 'spiky' when using divided differences, which conceptually requires the strainline integration to track a numerically noisy ridge. With DA-LCS, there is a smooth, well-defined ridge of consistently low

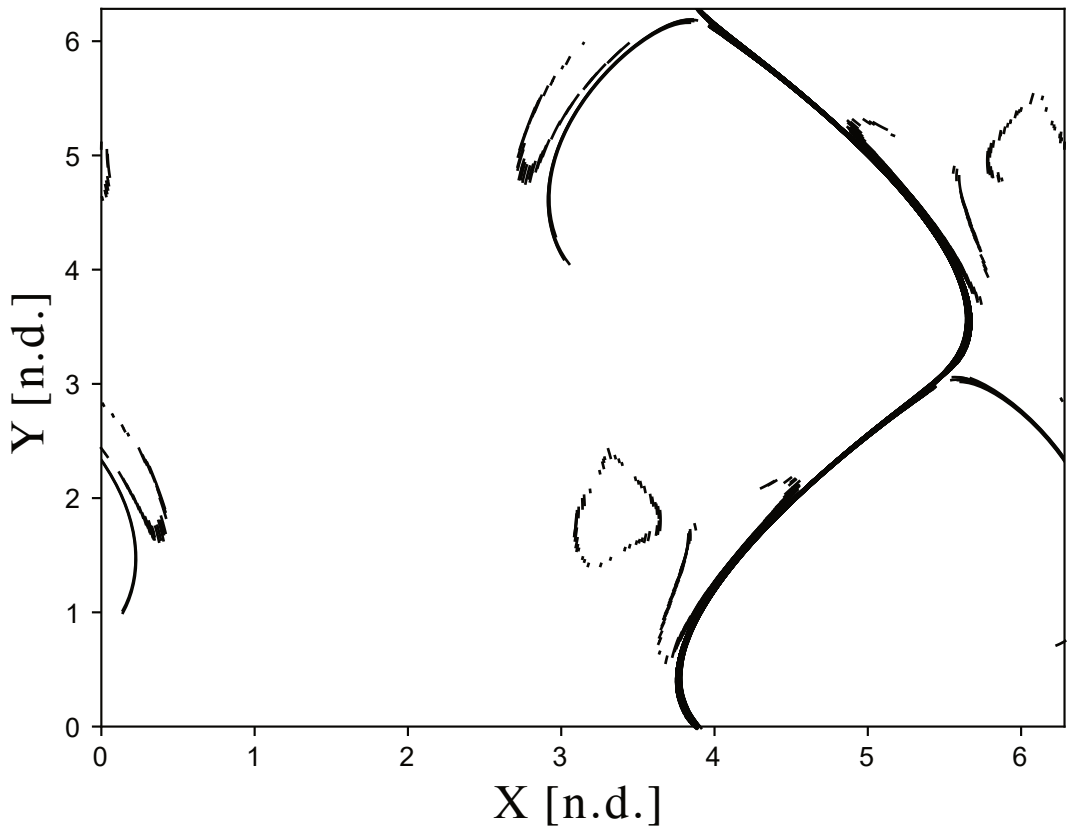
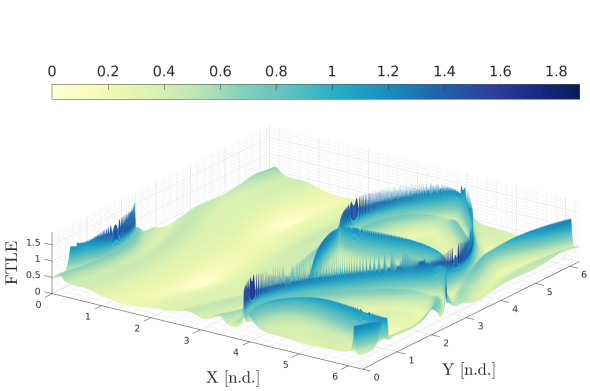


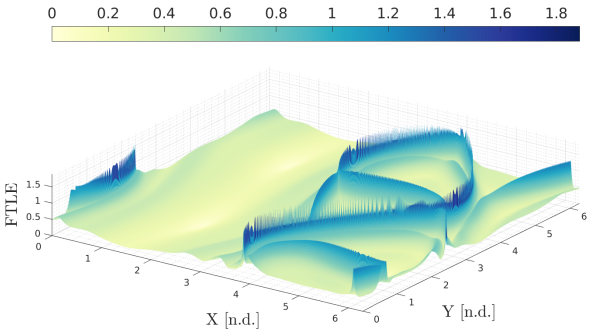
FIG. 9: Final strainlines for the periodic ABC flow on the $z = 0$ plane computed using DA-LCS, after filtering. The strainline structure is composed of approximately 250 strainline segments.

helicity for the algorithm to track with much lower numerical noise; in fact, our helicity threshold is approximately two orders of magnitude lower than used in literature but recovers qualitatively similar structures.

Finally, the strainlines on the $z = 0$ plane for this system computed using DA-LCS are shown in Figure 9. Approximately 250 strainline segments determine the full strainline structure on the $z = 0$ plane for this example. As with the steady ABC flow, the distribution of the number of strainlines with respect to their length is largely bimodal, and the majority of these segments are found in the 'loops' in the strainline structure.



(a) Computed using DA-LCS.



(b) FTLE field obtained using divided differences with auxiliary grid spacing of 0.1 of the nominal grid spacing in all directions.

FIG. 10: Finite-time Lyapunov exponent field for the chaotically-forced ABC flow and an integration time from $t_0 = 0$ to $T = 5.0$. The fields still strongly agree, suggesting that again the computation of C_0^5 on the optimal auxiliary grid is not a major source of error for this example.

C. Chaotically-forced Arnold-Beltrami-Childress flow

Following Blazevski and Haller[13], we now demonstrate that DA-LCS is robust under perturbations from a chaotic forcing function $g(t)$. The motion is forced by a chaotic Duffing oscillator, with equations of motion given by

$$\dot{x} = (A + 0.1\sin t)\sin z + C\cos y \tag{22}$$

$$\dot{y} = B\sin x + (A + 0.1g(t))\cos z \tag{23}$$

$$\dot{z} = C\sin y + B\cos x \tag{24}$$

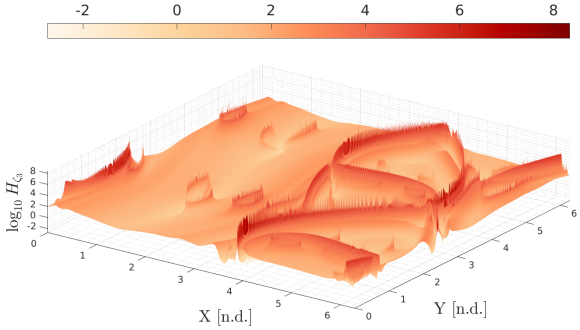
where $g(t)$ is the $x-$coordinate of the solution to the Duffing equation

$$\ddot{x} = -\delta\dot{x} - \beta x - \alpha x^3 + \gamma\cos(\omega t). \tag{25}$$

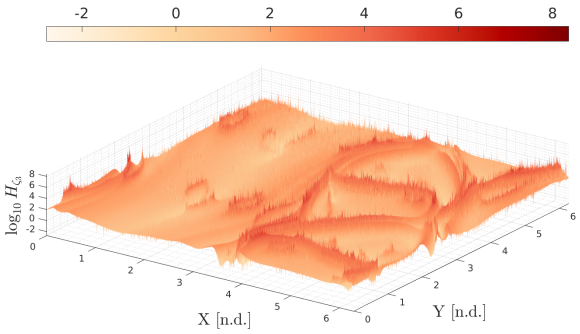
with parameters $\alpha = 1$, $\beta = -1$, $\gamma = 0.3$, $\delta = 0.2$, $\omega = 1$.

The computational grid is again the same as for the previous test cases involving the ABC flow, including the hyperplanes $\mathscr{S} = \{(x,y,z) \in [0,2\pi]^3 \ : \ z = s_1\}, s_1 = 0.0, 0.005, 0.01, \ldots, 0.1$, but a longer integration time of $T = 5$ is used to match the literature. Again, a helicity tolerance of $\alpha = 5 \times 10^{-5}$ is used with a filtering distance of $d_F = 0.05$.

The FTLE fields computed using DA-LCS and divided differences are again shown in Figure 10a and Figure 10b, respectively. The helicity fields are shown in Figures 11a and 11b, respec-



(a) Computed using DA-LCS.



(b) Helicity field obtained using divided differences with auxiliary grid spacing of 0.1 of the nominal grid spacing in all directions. The same grid is used for computing both C_0^5 and $\nabla \times \zeta_n$.

FIG. 11: $-\log H_{\zeta_n}$ for the chaotically-forced ABC flow from $t_0 = 0$ to $T = 5.0$. DA-LCS produces visibly better-defined ridges on which the strainline integration is performed, making integration along the strainline more robust.

tively. The helicity field in particular now exhibits a significant difference compared to the two previous cases. Using DA-LCS, we are able to resolve a relatively smooth ridge, whereas the use of divided differences leads to noticeable numerical noise throughout the field as well as an overall much higher helicity.

The strainlines for this system on the $z = 0$ plane computed using DA-LCS are presented in Figure 12. A total of 160 strainline segments give the full structure on the $z = 0$ plane.

VI. THE ELLIPTIC-RESTRICTED THREE-BODY PROBLEM

We now demonstrate the numerical out-performance of DA-LCS compared to standard approaches on a test problem from astrodynamics. The system presented in this Section is the Elliptic-Restricted Three-body Problem (ER3BP), which studies the motion of a small mass m_3 under the motion of two far larger masses m_1 and m_2 such that $m_1 \geqslant m_2 \gg m_3$. The system is parameterised by the mass parameter $\mu = m_2/(m_1 + m_2)$.

In an inertial coordinate system, m_2 and m_1 orbit their centre of mass on an ellipse of fixed eccentricity e_p, which is the second system parameter. The angle of m_2 with respect to the $+x$-axis of the inertial coordinate system is the true anomaly ν.

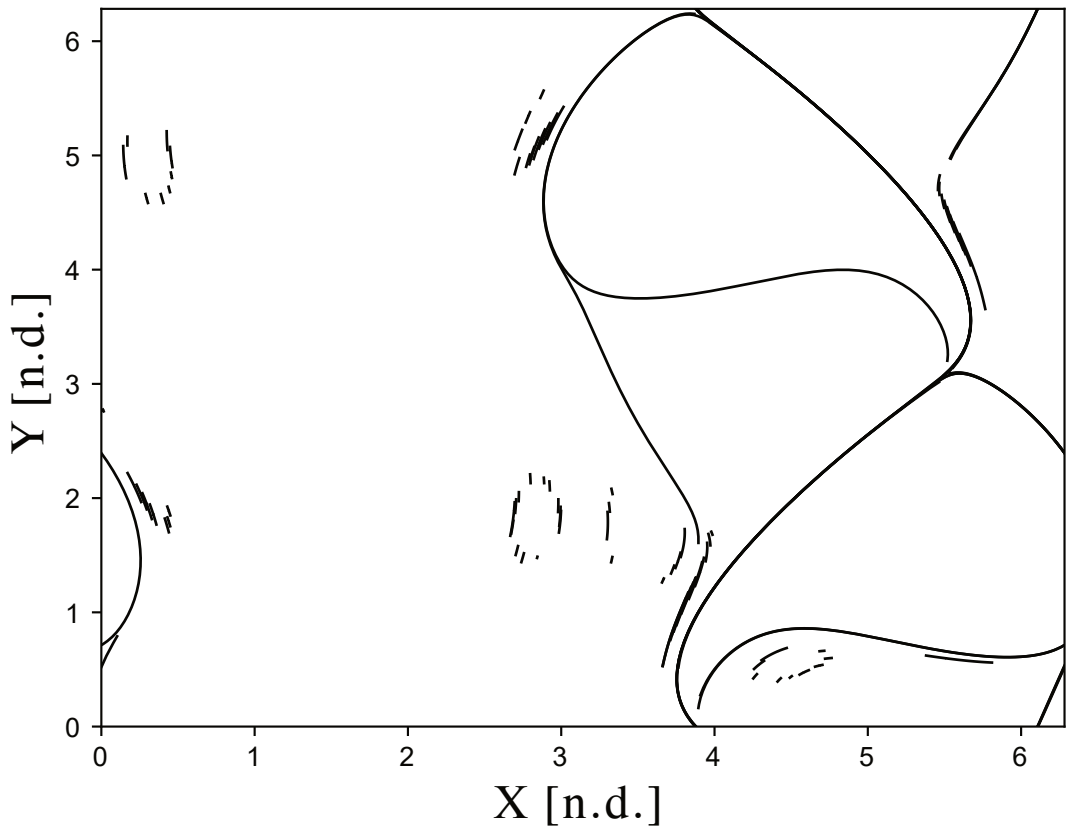


FIG. 12: Final strainline structure on the $z = 0$ plane for the chaotically-forced ABC flow computed using DA-LCS. The structure is formed of 160 individual strainline segments.

For the special case of $e_p = 0$, one recovers an autonomous dynamical system for which fixed points and invariant manifolds exist[17]; for the more general $e_p > 0$, such structures do not exist. LCS have thus been suggested to analyse the behaviour for the cases of $e_p > 0$. In this example, we analyse the interesting dynamical phenomena around m_2. For small differences in initial position and velocity, orbits can vary from being bound entirely around m_2, being only temporarily captured around m_2, or escaping entirely[37]. Profiling these regions is of high importance in the design of space missions[38].

Since the ER3BP lives in a phase space defined in $\mathbb{R}^6$, but the algorithm above functions for a CGST that is 3×3 in dimension and represents a system with three-dimensional dynamics, we embed a three-dimensional submanifold in the six-dimensional phase space on which we compute the LCS. We parameterise the manifold in the three spatial directions to represent position around

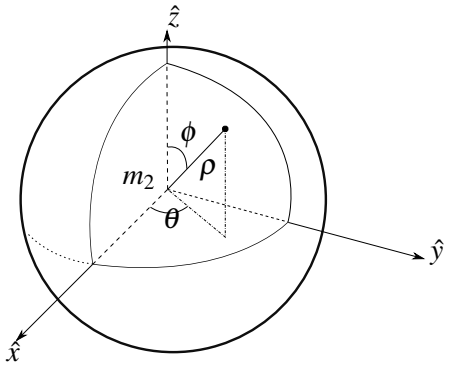


FIG. 13: The parameterisation of the space around m_2 using spherical coordinates relative to the inertial coordinate frame. By careful choice of the ranges of ρ, θ and ϕ, the reference hyperplanes can encapsulate regions of 'interesting' dynamics about m_2.

m_2 using spherical coordinates $\Psi = (\rho, \theta, \phi)$ (Figure 13). We complete the embedding by uniquely associating a velocity $\boldsymbol{v}$ with each point in space to complete the full phase space.

Given the Cartesian position $\boldsymbol{x} = (x, y, z)^\top$ corresponding to Ψ

$$x = \rho \cos\theta \sin\phi \tag{26}$$

$$y = \rho \sin\theta \sin\phi \tag{27}$$

$$z = \rho \cos\phi \tag{28}$$

the velocity at this point $\boldsymbol{v}(\boldsymbol{x})$ is chosen to be

$$\boldsymbol{v}(\boldsymbol{x}) = \sqrt{\mathrm{G}m_2 \frac{(1+e)}{\rho^3}} \left[\begin{pmatrix} x \\ y \\ z \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right], \tag{29}$$

where the problem parameters $\mathrm{G}m_2$ and e are the gravitational parameter of m_2 and an orbital eccentricity, respectively. Conceptually, this fixes the velocity direction tangential to a cylinder around the z-axis, while the magnitude corresponds to a Keplerian orbit of eccentricity e around m_2. Together, this choice of velocity vector reveals the 'dynamically interesting' behaviour introduced previously.

Rather than using the inertial coordinate system about m_2 to propagate the initial condition, it is beneficial to use a rotating-pulsating Cartesian coordinate system centred on the barycentre of m_1 and m_2. In this system, m_1 and m_2 are fixed, and the true anomaly ν replaces time as the

independent variable. The transformation of the initial condition into this coordinate system is shown in Appendix A. In this system the equations of motion are given by

$$x'' = 2y + \frac{\partial \Omega}{\partial x} \tag{30}$$

$$y'' = -2x + \frac{\partial \Omega}{\partial y} \tag{31}$$

$$z'' = \frac{\partial \Omega}{\partial z} \tag{32}$$

where

$$\Omega = \frac{1}{1 + e_p \cos \nu} \left[\left(x^2 + y^2 + z^2 \right) + \frac{\mu}{r_1} + \frac{1-\mu}{r_2} \right] \tag{33}$$

and

$$r_1 = \sqrt{(x-\mu)^2 + y^2 + z^2} \tag{34}$$

$$r_2 = \sqrt{(x+1-\mu)^2 + y^2 + z^2}. \tag{35}$$

After propagation under the equations of motion, the transformation into the rotating coordinate system is inverted, and the final position is projected back into spherical coordinates. Another big advantage of DA-LCS is that, provided the intermediate transformations are coded as DA operations, the derivatives of this process are computed fully automatically and there is no need to derive further equations for the coordinate transformations.

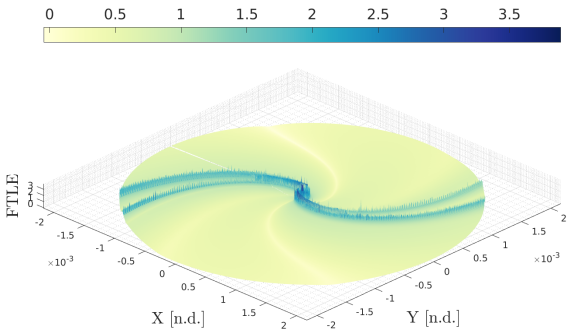
For this example, we choose m_1 to be the Sun and m_2 to be Mars, with the system parameters as given in Table I. The set of reference hyperplanes is defined as

$$\mathscr{S} = \{ \Psi \in [r, r_s] \times [0, 2\pi] \times [5^\circ, 15^\circ, \ldots, 175^\circ] \}.$$

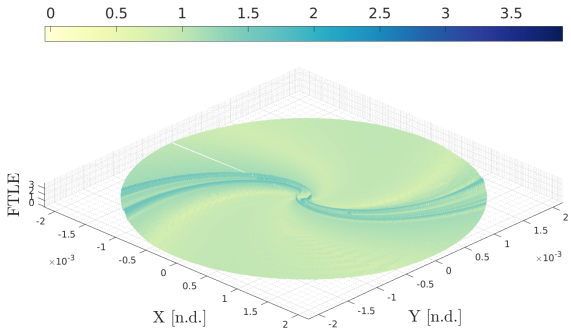
The variables r and r_s here are the radius and the Hill sphere of Mars, respectively; the latter is the maximum distance from Mars at which it still dominates gravitational attraction. Together, the reference planes cover the 'dynamically interesting' region around m_2. The initial integration time is set equal to $t_0 = \nu_0 = 0$ and the final time is $T = \nu = 2\pi$. The helicity tolerance α used is 10^{-5}.

1. Results

The FTLE fields computed using DA-LCS and divided differences on the $\theta = 115^\circ$ plane is presented in Figures 14a and 14b, respectively. The structure found using DA-LCS agrees with

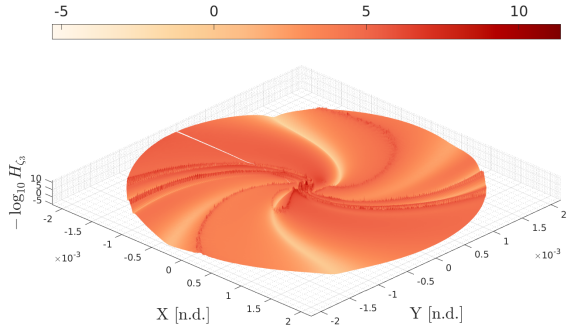


(a) Computed using DA-LCS.

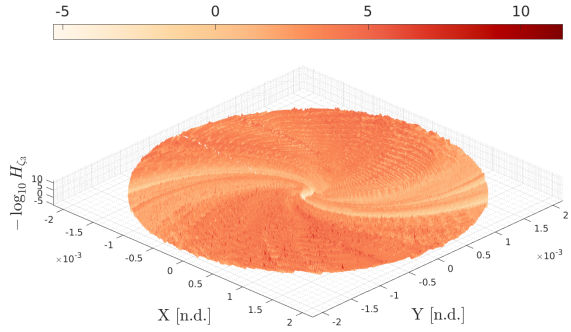


(b) FTLE field obtained using divided differences with auxiliary grid spacing of 0.05 of the nominal grid spacing in r and ϕ and the nominal grid spacing in θ.

FIG. 14: Finite-time Lyapunov exponent field for the Elliptic-Restricted Three-body Problem on the $\theta = 115°$ plane from $t_0 = \nu_0 = 0$ to $T = \nu = 2\pi$. While the structure is qualitatively the same, the ridges in the FTLE field are more well-defined with DA-LCS.



(a) Computed using DA-LCS.



(b) Helicity field obtained using divided differences with auxiliary grid spacing of 0.05 of the nominal grid spacing in r and ϕ and the nominal grid spacing in θ. The same grid is used for computing both $C_0^{2\pi}$ and $\nabla \times \boldsymbol{\zeta_n}$.

FIG. 15: $-\log H_{\zeta_n}$ for the Elliptic-Restricted Three-body Problem on the $\theta = 115°$ plane from $t_0 = \nu_0 = 0$ to $T = \nu = 2\pi$. No defined ridges of low helicity are visible with divided differences, but with DA-LCS we can extract well-defined ridges on which the strainline integration can track.

Parameter	Description	Value
e_p	Eccentricity of the orbit of m_2 about m_1	0.0935
μ	Mass parameter	3.227154×10^{-7}
e	Eccentricity of the orbit of m_3 about m_2	0.9
Gm_2	Standard gravitational parameter of m_2	1.50499×10^{-14}
r	Planetary radius of m_2	1.641×10^{-5}
r_s	Hill sphere of m_2	0.00513

TABLE I: Parameter values used in the ER3BP investigation where m_1 is arbitrarily chosen to be the Sun and m_2 arbitrarily chosen to be Mars. All values are given in non-dimensional units and valid at $\nu = 2n\pi$, $n \in \mathbb{Z}$.

what would be expected from previous literature, with the structures in the two 'arms' being consistent with the transition between orbits that escape and are permanently or temporarily captured about m_2[37]. Similar performance, albeit with poorer definition of the FTLE ridges, can be obtained using divided differences after tuning the grid sizes used to generate the derivatives. We note that the ER3BP does admit variational equations that can be integrated with the equations of motion which may improve the quality of the derivatives used to compute $C^T_{t_0}$.

Importantly, these variational equations cannot be used to compute $\nabla \times \zeta_n$, which must still be approximated using divided differences and appear to produce the majority of the error for this test case. This is to be expected, as the estimation of second derivatives using divided differences is numerically difficult. Figure 15a presents the helicity field on the $\theta = 115^\circ$ plane for the ER3BP computed using DA-LCS, which like the FTLE field highlights the 'arms' as being influential portions of flow. Qualitative inspection of the trajectories in this region reveals the low-helicity portions of the field to separate regions of different dynamical behaviour. However, using divided differences to compute the helicity, given in Figure 15b, produces no meaningful insight into the helicity field even after tuning the grid-sizes used; the numerical noise in the determination of the helicity reveals no distinct ridges along which the numerical integration of the strainline can be performed. The use of DA-LCS, therefore, actually enables the analysis of this system using the algorithm presented in Blazevski and Haller[13] where it was previously too numerically noisy to generate insight. Moreover, this numerical improvement comes completely automatically, without the need to tune grid-sizes and thus functions without any *a priori* knowledge.

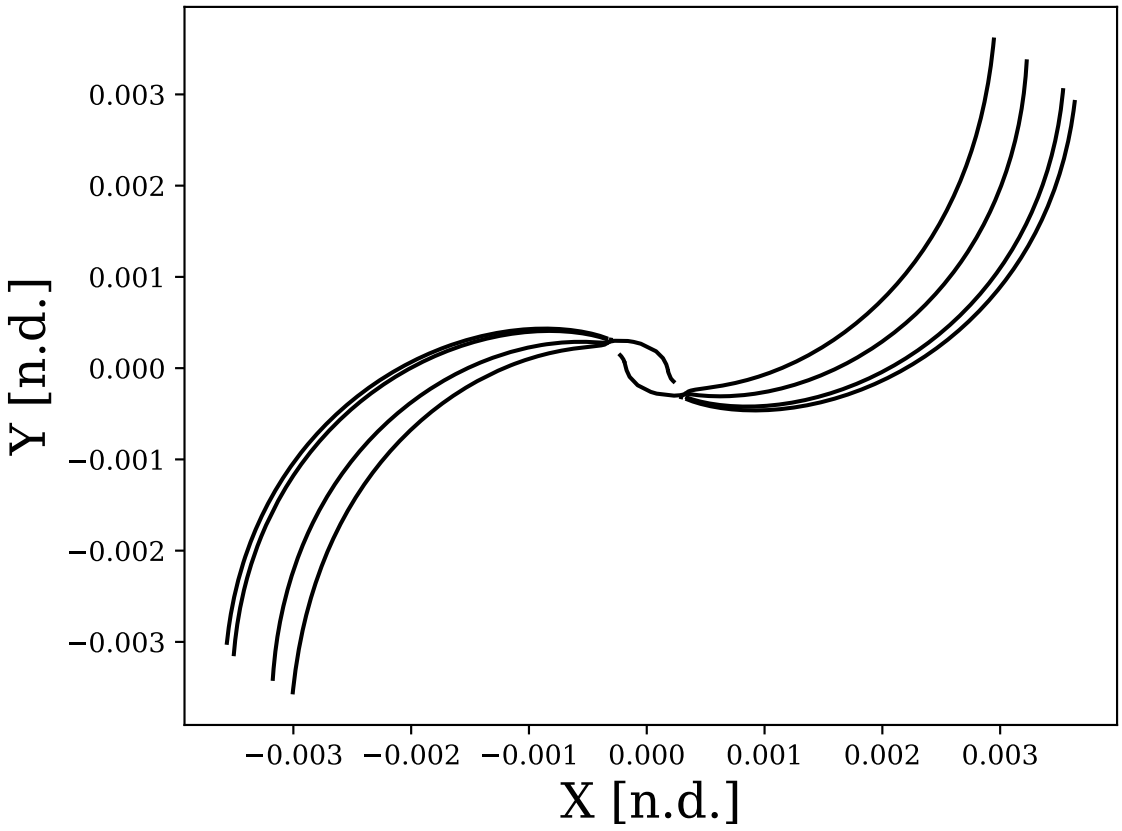
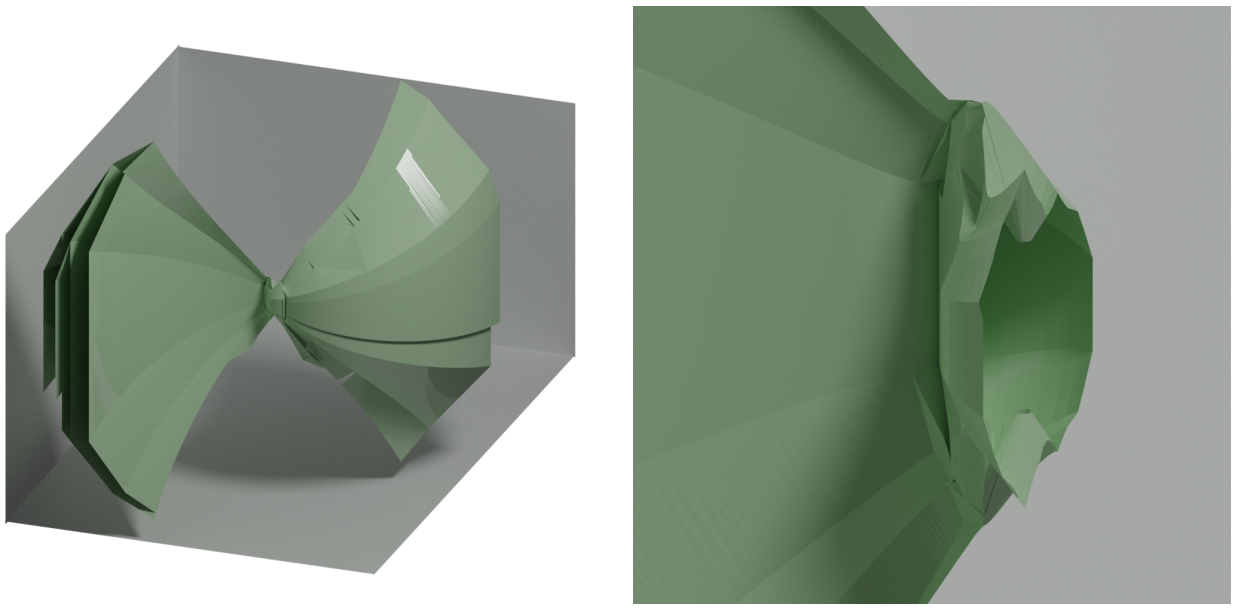


FIG. 16: Strainlines on the $\theta = 115^\circ$ plane for the Elliptic-Restricted Three-Body Problem computed using DA-LCS. We are unable to generate any strainlines when using divided differences, but with DA-LCS we can deduce the structure of the LCS readily and with only 8 strainlines.

The final strainlines for this flow computed using DA-LCS on the $\theta = 115^\circ$ plane are shown in Figure 16, and largely follow from the helicity field given earlier. We were not able to generate any meaningful strainlines using divided differences, a phenomenon that has also been present in several earlier attempts at the problem by other authors[39,40]. A representative rendering of the full 3D LCS for this test case is shown in Figure 17.

VII. CONCLUSION

This paper has introduced DA-LCS, an improved numerical method for determining hyperbolic Lagrangian Coherent Structures in time-dependent dynamical systems. We showed how Differen-



(a) Full 3D structure of the LCS over the entire set of reference planes.

(b) A zoomed-in section of the full LCS highlighting the interior structure.

FIG. 17: A set of representative renders of the 3D LCS for the ER3BP test case. The left figure is the full 3D LCS over all hyperplanes in $\mathscr{S}$. On the right is a zoomed-in portion of the centre of the LCS, with the right half removed to highlight the internal structure.

tial Algebra can be used to directly construct high-order Taylor expansions of the flow, its derivatives and a field of leading eigenvectors of the flow's strain tensor, accurate to machine precision. We have shown that with this information we can construct a highly-accurate LCS based solely on the underlying dynamics of the system, even in highly complex flows. We demonstrated the effectiveness of the method through applications to common variations of the Arnold-Beltrami-Childress flow from the literature, as well as introducing a new and particularly challenging test problem from astrodynamics where the classical methods fail to produce usable results. DA-LCS also constructs the LCS automatically and without any *a priori* information, requiring no additional implementation beyond the dynamics of the system.

ACKNOWLEDGMENTS

The authors acknowledge financial support from the EPSRC Centre for Doctoral Training in Next Generation Computational Modelling grant EP/L015382/1, and the use of the IRIDIS High

Performance Computing Facility and associated support services at the University of Southampton. The authors also thank Davide Lasagna for his helpful suggestions.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in a public University of Southampton repository at `https://doi.org/10.5258/SOTON/D2143`.

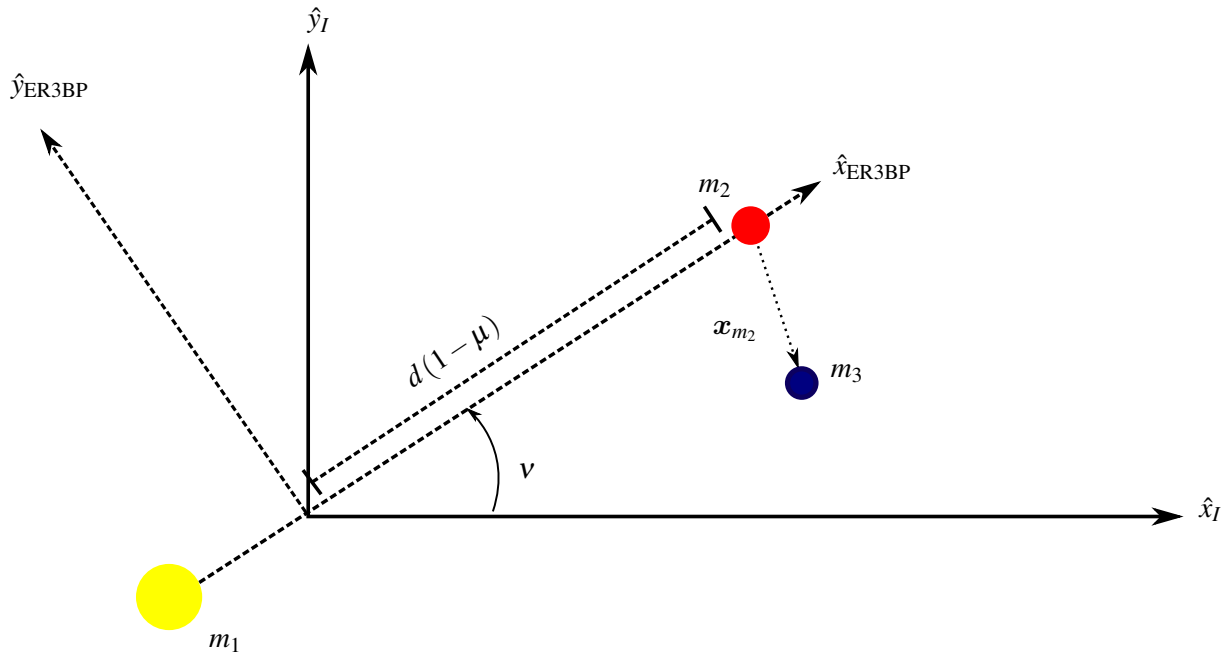


FIG. 18: Schematic of the inertial frame (subscript $_I$) and the rotating-pulsating frame (subscript $_{\text{ER3BP}}$) for use in Appendix A. The transformation between the inertial and rotating-pulsating frame is a composite translation, rotation and normalisation.

Appendix A: Transformation into the rotating-pulsating frame of the Elliptic-Restricted Three-body Problem

As previously introduced, the Elliptic-Restricted Three-body Problem (ER3BP) models the motion of a small object m_3 under the influence of two far larger masses m_1 and m_2, such that $m_1 \geqslant m_2 \gg m_3$. The object m_3 is sufficiently small compared to m_1 and m_2 that it is considered massless. The system is parameterised by the mass parameter $\mu = m_2/(m_1+m_2)$, and in an inertial coordinate system m_1 and m_2 orbit their center of mass on an ellipse with fixed eccentricity e_p.

In Section VI, we chose the parameterisation of the sub-manifold to represent initial position around m_2 in the inertial frame using spherical coordinates, and the embedding to represent the initial velocity of the point in the inertial frame. This was done to simplify the problem set-up and

more easily define the regions of 'interesting' dynamical behaviour. However, in the literature[41] the ER3BP is integrated in a rotating coordinate system where m_1 and m_2 are fixed on the $x-$axis at $(-\mu,0,0)$ and $(1-\mu,0,0)$, respectively, and the distance between them is normalised to unity. In this frame, the independent variable in the motion of m_3 is the true anomaly ν. To simplify the test case, the transformation that follows is valid only for values of ν that are scalar multiples of 2π; for an in-depth derivation of the general case of this transformation, the reader is directed to Szebehely and Jefferys[41].

With reference to Figure 18, the transformation of the position from the m_2-centred inertial frame to the rotating-pulsating frame formed of a translation to move the centre of the system to the centre of mass of m_1 and m_2, a rotation to align the $+x$ axis to the line joining m_1 and m_2, and a scaling to normalise the distance between m_1 and m_2 to unity.

We perform the translation first. Define the Cartesian position of m_3 about m_2 in the inertial frame as $\boldsymbol{x}_{m_2}$, such that the translated position around the barycentre (centre of mass) of m_1 and m_2, $\boldsymbol{x}_{\mathrm{BC}}$, is

$$\boldsymbol{x}_{\mathrm{BC}} = \boldsymbol{x}_{m_2} + d\,(1-\mu)\begin{pmatrix}\cos\nu\\ \sin\nu\\ 0\end{pmatrix} \tag{A1}$$

where d is the full distance between m_1 and m_2, and $(1-\mu)$ gives the proportion of the distance d between m_2 and the centre of mass. The distance d can be retrieved from the orbit equation (more generally known as the ellipse equation)

$$d\,(\nu) = \frac{a\left(1-e_p^2\right)}{1+e_p\cos\nu} \tag{A2}$$

with a the semi-major axis of m_2 about m_1. For the case of m_1 being the Sun and m_2 being Mars studied in this paper, at scalar multiples of 2π the semi-major axis $a = 1.10314$.

The coordinate axes must now be rotated such that m_1 and m_2 lie on the $+x$-axis. This is a clockwise rotation about $+z$ of an angle ν. We apply the standard Euler rotation matrix to $\boldsymbol{x}_{\mathrm{BC}}$ to find its equivalent state in the rotated coordinate system $\boldsymbol{x}_{\mathrm{rot}}$

$$\boldsymbol{x}_{\mathrm{rot}} = R_z\,(\nu)\,\boldsymbol{x}_{\mathrm{BC}} = \begin{pmatrix}\cos\nu & \sin\nu & 0\\ -\sin\nu & \cos\nu & 0\\ 0 & 0 & 1\end{pmatrix}\boldsymbol{x}_{\mathrm{BC}}. \tag{A3}$$

Finally, the distance between m_1 and m_2 is normalised to 1 by scaling the length unit of the

system by d. This yields the final ER3BP position $\boldsymbol{x}_{\text{ER3BP}}$

$$\boldsymbol{x}_{\text{ER3BP}} = \frac{\boldsymbol{x}_{\text{rot}}}{d}. \tag{A4}$$

The composite transformation can be combined into a single equation for brevity:

$$\boldsymbol{x}_{\text{ER3BP}} = \frac{R_z(\nu)}{d(\nu)} \left(\boldsymbol{x}_{m_2} + d(1-\mu) \begin{pmatrix} \cos\nu \\ \sin\nu \\ 0 \end{pmatrix} \right) \tag{A5}$$

$$= \frac{R_z(\nu)}{d(\nu)} \boldsymbol{x}_{m_2} + (1-\mu) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{A6}$$

The equation above completes the transformation of the position from the inertial coordinate system around m_2 to the rotating coordinate system of the ER3BP. However, integrating the ER3BP equations of motion also requires the initial velocity of m_3 in the rotating coordinate system. Thus, the velocity in the inertial frame about m_2 with respect to time given by the embedding introduced in the main text, $\boldsymbol{v}$, must also be transformed into the ER3BP coordinate frame.

To do this, Equation A6 is differentiated with respect to the true anomaly ν, which is the independent variable in the ER3BP. In the following, $\square'$ denotes derivatives with respect to ν (as in the ER3BP coordinate system), and $\dot{\square}$ denotes derivatives with respect to time (the inertial coordinate system.) Via the chain rule, the derivative of Equation A6 is

$$\boldsymbol{x}'_{\text{ER3BP}} = \frac{R_z(\nu)'}{d(\nu)} \boldsymbol{x}_{m_2} + \frac{R_z(\nu)}{d(\nu)} \boldsymbol{x}'_{m_2} \tag{A7}$$

since the quantity $(1/d(\nu))'$ is zero in the case of ν being a scalar multiple of 2π. The quantity $R_z(\nu)'$ is trivial to infer from its use previously

$$R'_z(\nu) = \begin{pmatrix} -\sin\nu & \cos\nu & 0 \\ -\cos\nu & -\sin\nu & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{A8}$$

The velocity with respect to time in the inertial frame $\boldsymbol{v}$ represents $\dot{\boldsymbol{x}}_{m_2}$. To obtain $\boldsymbol{x}'_{m_2}$, we use

$$\frac{\mathrm{d}\boldsymbol{x}_{m_2}}{\mathrm{d}\nu} = \frac{\mathrm{d}\boldsymbol{x}_{m_2}}{\mathrm{d}t} \frac{\mathrm{d}t}{\mathrm{d}\nu} = \boldsymbol{v}/\dot{\nu} \tag{A9}$$

where $\dot{\nu}$ is given by considering the angular momentum of m_2 about m_1

$$\dot{\nu} = \frac{\mathrm{G}m_1^{\frac{1}{2}}(1+e_p)^2}{a^{\frac{3}{2}}\left(1-e_p^2\right)^{\frac{3}{2}}} \tag{A10}$$

which completes the transformation of a position in the inertial frame about m_2 to the rotating coordinate system of the ER3BP for use in Section VI.

Since we are computing the LCS on a submanifold that represents the spatial dimensions about m_2, the inverse transformation need only consider the position. Equation A6 is inverted to give $\boldsymbol{x}_{m_2}$ and then converted back into spherical coordinates for use in computing the LCS.

REFERENCES

[1]J. D. Meiss, "Symplectic maps, variational principles, and transport," Reviews of Modern Physics **64**, 795–848 (1992).

[2]G. Haller, "Lagrangian coherent structures," Annual Review of Fluid Mechanics , 1–19 (2015).

[3]F. Lekien and S. D. Ross, "The computation of finite-time Lyapunov exponents on unstructured meshes and for non-Euclidean manifolds," Chaos: An Interdisciplinary Journal of Nonlinear Science **20**, 017505 (2010).

[4]G. Boffetta, G. Lacorata, G. Redaelli, and A. Vulpiani, "Detecting barriers to transport: A review of different techniques," Physica D: Nonlinear Phenomena **159**, 58–70 (2001).

[5]G. Haller and G. Yuan, "Lagrangian coherent structures and mixing in two-dimensional turbulence," Physica D: Nonlinear Phenomena **147**, 352–370 (2000).

[6]A. Hadjighasem, M. Farazmand, D. Blazevski, G. Froyland, and G. Haller, "A Critical Comparison of Lagrangian Methods for Coherent Structure Detection," (2017).

[7]S. C. Shadden, F. Lekien, and J. E. Marsden, "Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows," Physica D: Nonlinear Phenomena **212**, 271–304 (2005).

[8]G. Haller and T. Sapsis, "Lagrangian coherent structures and the smallest finite-time Lyapunov exponent," Chaos **21**, 1–7 (2011).

[9]C. Jones and S. Winkler, *Handbook of Dynamical Systems*, Vol. 2 (Elsevier B.V., 2002) pp. 55–92.

[10]G. Haller, "A variational theory of hyperbolic Lagrangian Coherent Structures," Physica D: Nonlinear Phenomena **240**, 574–598 (2011).

[11]M. Farazmand and G. Haller, "Computing Lagrangian coherent structures from their variational theory," Chaos **22** (2012), 10.1063/1.3690153.

[12]M. Farazmand and G. Haller, "Attracting and repelling Lagrangian coherent structures from a single computation," Chaos **23** (2013), 10.1063/1.4800210.

[13]D. Blazevski and G. Haller, "Hyperbolic and elliptic transport barriers in three-dimensional unsteady flows," Physica D: Nonlinear Phenomena **273-274**, 46–62 (2014).

[14]M. Farazmand, D. Blazevski, and G. Haller, "Shearless transport barriers in unsteady two-dimensional flows and maps," Physica D: Nonlinear Phenomena **278-279**, 44–57 (2014).

[15]C. R. Short, D. Blazevski, K. C. Howell, and G. Haller, "Stretching in phase space and appli-

cations in general nonautonomous multi-body problems," Celestial Mechanics and Dynamical Astronomy **122**, 213–238 (2015).

[16]Q. Qingyu, L. Mingpei, and X. Ming, "Lagrangian Coherent Structures in the Planar Parabolic/Hyperbolic Restricted Three-Body Problem," Monthly Notices of the Royal Astronomical Society (2020), 10.1093/mnras/staa199.

[17]W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross, *Dynamical Systems, the Three-body Problem and Space Mission Design* (Marsden Books, 2008).

[18]M. Berz, "The method of power series tracking for the mathematical description of beam dynamics," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **258**, 431–436 (1987).

[19]K. Makino and M. Berz, "Remainder Differential Algebras and their Applications," Computational Differentiation: Techniques, Applications, and Tools , 63–74 (1996).

[20]K. Makino, *Rigorous Analysis of Nonlinear Motion in Particle Accelerators*, Ph.D. thesis, Michigan State University (1998).

[21]G. Di Mauro, M. Schlotterer, S. Theil, and M. Lavagna, "Nonlinear Control for Proximity Operations Based on Differential Algebra," Journal of Guidance, Control, and Dynamics **38**, 2173–2187 (2015).

[22]A. Wittig, P. Di Lizia, R. Armellin, F. Bernelli-Zazzera, K. Makino, and M. Berzş, "An automatic domain splitting technique to propagate uncertainties in highly nonlinear orbital dynamics," Advances in the Astronautical Sciences **152**, 1923–1941 (2014).

[23]A. Wittig, P. Di Lizia, R. Armellin, K. Makino, F. Bernelli-Zazzera, and M. Berz, "Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting," Celestial Mechanics and Dynamical Astronomy **122**, 239–261 (2015).

[24]M. Massari, P. Di Lizia, and M. Rasotto, "Nonlinear Uncertainty Propagation in Astrodynamics Using Differential Algebra and Graphics Processing Units," Journal of Aerospace Information Systems **14**, 493–503 (2017).

[25]R. Armellin, P. Di Lizia, F. Bernelli-Zazzera, and M. Berz, "Asteroid close encounters characterization using differential algebra: The case of Apophis," Celestial Mechanics and Dynamical Astronomy **107**, 451–470 (2010).

[26]M. Massari, P. Di Lizia, F. Cavenago, and A. Wittig, "Differential Algebra software library with automatic code generation for space embedded applications," (2018), 10.2514/6.2018-0398.

[27]A. Haro, "Automatic differentiation methods," (2011).

[28]A. Wittig, *Rigorous High-Precision Enclosures of Fixed Points and their Invariant Manifolds*, Ph.D. thesis, Michigan State University (2012).

[29]M. Berz, *Advances in Imaging and Electron Physics*, Vol. 108 (1999) pp. 1–318.

[30]F. Cavenago, P. Di Lizia, M. Massari, and A. Wittig, "On-board DA-based state estimation algorithm for spacecraft relative navigation," 7th European conference for aeronautics and space sciences (EUCASS) , 1–14 (2017).

[31]A. Wittig, C. Colombo, and R. Armellin, "Long-term density evolution through semi-analytical and differential algebra techniques," Celestial Mechanics and Dynamical Astronomy **128**, 435–452 (2017).

[32]E. V. Haynsworth and A. S. Householder, "The Theory of Matrices in Numerical Analysis." The American Mathematical Monthly **73** (1966), 10.2307/2314680.

[33]R. H. Chan, Y. Qiu, and G. Yin, "Iterative Methods for Eigenvalues/Eigenvectors," in *Encyclopedia of Social Network Analysis and Mining* (2018).

[34]T. Devogele, M. Esnault, L. Etienne, and F. Lardy, "Optimized Discrete Fréchet Distance between trajectories," BigSpatial 2017 - Proceedings of the 6th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data , 11–19 (2017).

[35]A. Driemel, A. Krivosija, and C. Sohler, "Clustering time series under the Fréchet distance," Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms **2**, 766–785 (2016).

[36]K. L. Palmerius, M. Cooper, and A. Ynnerman, "Flow field visualization using vector field perpendicular surfaces," Proceedings - SCCG 2009: 25th Spring Conference on Computer Graphics , 27–34 (2009).

[37]Z. F. Luo and F. Topputo, "Analysis of ballistic capture in Sun-planet models," Advances in Space Research (2015), 10.1016/j.asr.2015.05.042.

[38]E. A. Belbruno, "Lunar capture orbits, a method of constructing earth moon trajectories and the lunar cas mission," 19th International Electric Propulsion Conference, 1987 (1987), 10.2514/6.1987-1054.

[39]X. Ros Roca, *Computation of Lagrangian Coherent Structures with Application to Weak Stability Boundaries*, Master's thesis, Politecnico di Milano (2015).

[40]A. S. Parkash, *Application of Lagrangian Coherent Structures to the computation and understanding of ballistic capture trajectories*, Master's thesis, Delft University of Technology (2019).

[41]V. Szebehely and W. H. Jefferys, "Theory of Orbits: The Restricted Problem of Three Bodies," American Journal of Physics **36**, 375–375 (1968).