A Massively Parallel Method for Fast Computation of Invariant Manifolds

Fausto Vega
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
fvega@andrew.cmu.edu

Martin Lo
Mission Design & Navigation Section
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109
martin.w.lo@jpl.nasa.gov

Zachary Manchester
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
zacm@cmu.edu

Jon Sims
Mission Design & Navigation Section
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109
jon.a.sims@jpl.nasa.gov

***Abstract*—The computation of invariant manifolds in the Circular Restricted 3-Body Problem (CR3BP) is a key step in space mission design as these manifolds provide low-energy trajectories for various mission applications, including the design of gravity-assist flybys and low-energy capture and escape around bodies in the solar system. The conventional way of computing invariant manifolds involves determining the stable and unstable directions across the entire orbit through an eigendecomposition of the monodromy matrix of the periodic orbit. These directions are then propagated along the entire orbit with the state transition matrix. This paper presents a simpler method for computing invariant manifolds without relying on the monodromy matrix or its eigendecomposition, so that invariant manifolds of unstable objects besides periodic orbits can be computed. Our new algorithm is also efficiently parallelizable on modern graphics processing units, leading to speed-ups of several orders of magnitude compared to the conventional method executed on a conventional CPU.**

TABLE OF CONTENTS

1. INTRODUCTION

Invariant manifolds of unstable orbits are an important tool in dynamical systems theory, and they are the key to understanding transport within the solar system. These manifolds are tube-like structures that can be computed along a periodic orbit to provide-low energy pathways that lead to the periodic orbit, as well as escape from the periodic orbit with minimal fuel [1]. One example of using manifolds for mission design is the Genesis mission, which utilized stable manifolds to construct a transfer trajectory and the unstable manifold to design a free-return trajectory [2].

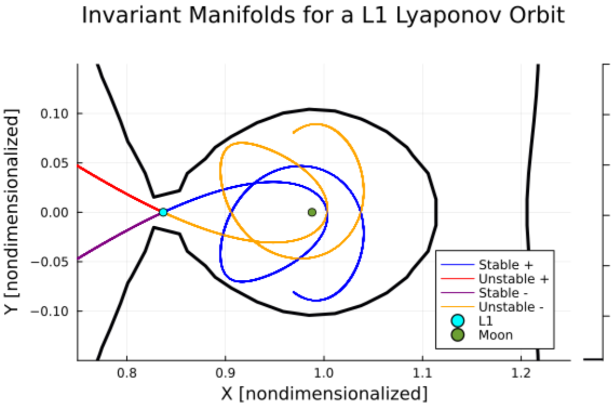

Figure 1: Stable an unstable manifolds around an L1 Lyapunov orbit in the Earth-Moon system. The zero-velocity curve is outlined in black.

We present a new method for computing invariant manifolds of unstable orbits. Instead of using the unstable eigenvectors of the monodromy matrix, as in the standard method [3] [4], we only rely on integrating the ordinary differential equations (ODE's) of the circular restricted three-body problem (CR3BP) forward in time to generate the unstable manifolds and backward in time for the stable manifolds. The initial conditions for the simulations are generated by a low-energy maneuver in a user-defined direction, which guarantees that all the manifolds along the entire orbit escape away from or toward the orbit.

Another method for calculating invariant manifolds is the parameterization method used by Haro [5] and Kumar [6] to calculate both the quasiperiodic invariant tori and the invariant manifolds (whiskers) of the tori. Kumar first solves for the center, stable, and unstable directions (bundles), and then uses a recursive parameterization method to obtain the approximation of the invariant manifolds. Our method simplifies the approach by eliminating the need to compute the stable and unstable directions, since an arbitrary unit

979-8-3503-0462-6/24/$31.00 ©2024 IEEE

direction is sufficient to obtain an approximation of the invariant manifold. This simplicity allows our algorithm to be parallelized across multiple GPU cores, which provides a significant speed up.

For concreteness, we describe this work in the context of periodic orbits of the CR3BP. However, our method is applicable to general dynamical systems. Our mathematical proof, described later, is limited to dynamical systems with monodromy matrices with the same structure as that of the CR3BP. Our main contribution is a fast and simple way of computing invariant manifolds that is parallelizable, along with a set of examples and comparisons to the conventional method of computing invariant manifolds in the Earth-Moon system. Our method achieves a computational speed-up of three orders of magnitude compared to the conventional method thanks to its parallelizability on modern GPU hardware.

This paper proceeds as follows: We review background on the CR3BP in Section 2. Next, we describe our algorithm in Section 3 and provide the intuition behind its functionality. Examples and computational benchmarks are detailed in Section 4. Finally, Section 5 summarizes our results and suggests directions for future work.

2. BACKGROUND

Trajectories in early space missions were dominated by the Two-Body Problem (TBP) of the spacecraft around a central body like the Earth, Moon, or Sun. The TBP is completely integrable, and its orbits are conic sections described by (1) where p is the semi-lattice rectum, a is the semi-major axis, e is the eccentricity, and ν is the true anomaly. These orbits are shown in Fig. 2.

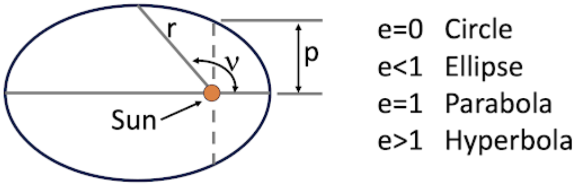

Figure 2: Conic orbits around the Sun

$$r = \frac{p}{1 + ecos\nu} \qquad (1)$$
$$p = a(1 - e^2)$$

Transport in the Two-Body Problem is completely solved by Lamberts Theorem:

Theorem 1 (Lambert's Theorem) Given any two points A and B in space around a central body and a time of travel, T, there is at least one conic arc connecting A and B. (see Easton, Anderson, and Lo [7] for an elementary solution).

Transport in the Circular Restricted Three-Body Problem

Adding a third body changes the problem of transport completely. There is no equivalent to Lambert's Theorem for the Three-Body Problem. To solve this problem, following Poincaré's lead, we use the CR3BP. In this model, the two primary masses such as the Earth and the Moon are assumed to move in circular orbits about their barycenter. The third body, a spacecraft or a small body like an asteroid or comet, is assumed to have infinitesimal mass. The hierarchical masses of the primaries, $m_1 > m_2$, are normalized to unity:

$$\begin{aligned} m_1 + m_2 &= 1, \\ \mu &= m_2. \end{aligned} \qquad (2)$$

In (2), μ is the characteristic mass parameter for the CR3BP. The distance between m_1 and m_2, and the rotation speed of m_1 and m_2 about their barycenter are similarly normalized to one so the non-dimensional period is 2π. To further reduce the dynamics, we use a rotating frame about the barycenter so that the two masses are fixed on the x-axis in Cartesian x, y, z coordinates with m_1 located at $[-1+\mu, 0, 0]$ and m_2 located at $[\mu, 0, 0]$. Fig. 3 gives a diagram of the rotating frame of the Earth-Moon system [8].

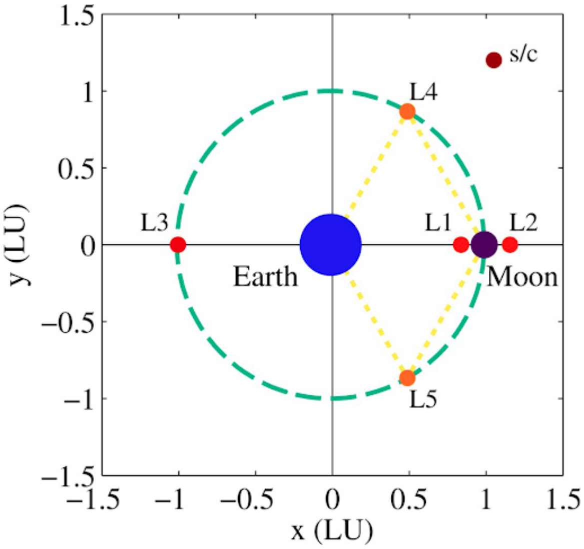

Figure 3: The CR3BP in rotating coordinates with the five equilibrium points (Lagrange points), L1–L5.

The equations of motion in six-dimensional phase space, $\mathbb{R}^6$, consisting of position (x, y, z) and velocity (v_x, v_y, v_z), are given by:

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{z} &= v_z \\ \dot{v_x} &= \frac{\partial U}{\partial x} + 2v_y \\ \dot{v_y} &= \frac{\partial U}{\partial y} - 2v_x \\ \dot{v_z} &= \frac{\partial U}{\partial Z} \end{aligned} \qquad (3)$$

where U is the augmented potential expressed in (4).

$$
\begin{aligned}
U &= \frac{1}{2}(x^2+y^2)+\frac{1-\mu}{r_1}+\frac{\mu}{r_2} \\
r_1 &= [(x+\mu)^2+y^2+z^2]^{\frac{1}{2}} \\
r_2 &= [(x-1+\mu)^2+y^2+z^2]^{\frac{1}{2}} \\
m_1 &= 1-\mu \\
m_2 &= \mu
\end{aligned}
\tag{4}
$$

The CR3BP has an integral, the Jacobi Constant (C), consisting of the augmented potential U and kinetic energy:

$$C = 2U - v_x^2 + v_y^2 + v_z^2 \tag{5}$$

The CR3BP has five equilibrium points, called the "Lagrange points," L1–L5. The stability of the Lagrange points depends on the mass parameter μ. For most Solar System bodies, L1–L3 are unstable, while L4 and L5 are locally stable. The unstable equilibria are the most interesting and useful for space missions. They are surrounded by low-energy orbits in chaotic regimes where very small maneuvers using little fuel can either stabilize them or cause large changes in a spacecraft's trajectory. Gravity-assist maneuvers in planetary flybys, spacecraft moon tours, and low-energy captures around moons and asteroids all depend critically on the chaotic dynamics of unstable orbits.

Invariant Manifolds Replace Lambert's Theorem for Transport

Poincaré discovered that unstable periodic orbits are surrounded by asymptotic trajectories that wind around them forming tube-like structures called "invariant manifolds" [4]. These tubes are the transport mechanism in nonlinear dynamics. Given a periodic orbit, $X(t)$, there are two types of invariant manifolds: One set of trajectories, called the "stable manifold" $W_s(X)$, asymptotically winds onto the periodic orbit. Another set of trajectories, called the "unstable manifold" $W_u(X)$, asymptotically escapes the periodic orbit. Fig. 4 shows a Lyapunov orbit around L1 and L2 with its stable and unstable manifolds forming tubes in space. These tubes intersect one another across the solar system to form the "Interplanetary Superhighway" [9].

Computing Invariant Manifolds

We can now describe the standard method for computing invariant manifolds. Given a periodic orbit, $X(t)$ with period T, we linearize the dynamics F described in Eq. (3) and solve the variational equation around $X(t)$:

$$
\begin{aligned}
\dot{X} &= F(X), \\
\delta\dot{X} &= DF(X(t))\delta X = A(t)\delta X.
\end{aligned}
\tag{6}
$$

The term $DF(X(t))$ in Eq. (6) is the jacobian of the dynamics in Eq. (3) with respect to the state. The expression in (6) is a linear system with non-constant coefficients whose solution is the 6×6 fundamental matrix, $\Phi(t)$, also known as the "state-transition matrix:"

$$
\begin{aligned}
\dot{\Phi} &= A(t)\Phi(t), \\
\Phi(0) &= I_6.
\end{aligned}
\tag{7}
$$

The state transition around an entire period T is known as the monodromy matrix M:

$$M = \Phi(T). \tag{8}$$

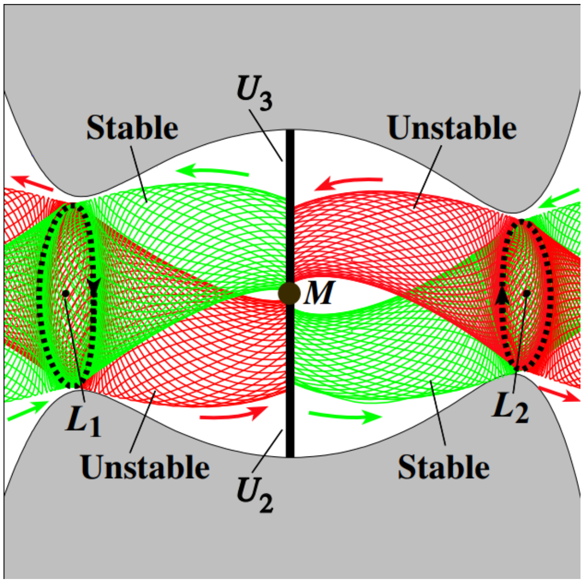

Figure 4: The stable manifold (green) and the unstable manifold (red) of a Lyapunov orbit around L1 for the Jupiter-Moon system [4]

For the CR3BP, M has the following eigenvalues:

$$
\begin{aligned}
\lambda_1 &= \frac{1}{\lambda_2} > 1 \\
\lambda_3 &= \lambda_4 = 1 \\
\lambda_5 = \lambda_6^*, &\quad |\lambda_5| = |\lambda_6| = 1
\end{aligned}
\tag{9}
$$

where $*$ is the complex conjugate. The associated unit eigenvectors are:

- ν_1 = unstable eigenvector, grows uncontrollably
- ν_2 = stable eigenvector, contracts towards the periodic orbit
- ν_3 = tangent to the periodic orbit
- ν_4 = in the direction of the associated family of periodic orbits
- ν_5, ν_6 = center manifold of bounded motion around $X(t)$

To compute the invariant manifold at a point $X(t_0)$ of the periodic orbit, we use ν_1 to compute the unstable manifold and ν_2 to calculate the stable manifold:

$$
\begin{aligned}
&X(t_0) \pm \varepsilon\nu_1 \\
&X(t_0) \pm \varepsilon\nu_2 \\
\varepsilon &\approx [1 \times 10^{-5}, 1 \times 10^{-4}]
\end{aligned}
\tag{10}
$$

The constant ε is a scaling factor applied to the eigenvector when perturbing the periodic orbit state and this perturbation is typically around 50 to 100 km away from $X(t_0)$ in dimensional units. The $\pm$ gives the two components of the trajectories of the invariant manifold at $X(t_0)$. To globalize the trajectory on the manifold, we integrate (3) with the initial conditions (10):

$$
\begin{aligned}
W_u(X(t_0),\tau) &= \int F(X(t_0)+\varepsilon\nu_1), \quad t \in [0,\tau], \\
W_s(X(t_0),\tau) &= \int F(X(t_0)+\varepsilon\nu_2), \quad t \in [-\tau,0].
\end{aligned}
\tag{11}
$$

For clarity, we note that at time $t = 0$, $W_u(X(t_0), 0) = W_u(X(t_0), \tau) = X_0$. Therefore, the time τ denotes the time one is on W_u or W_s starting at $X(t_0)$ at time $t = 0$. For the manifold trajectories at other points on the periodic orbit, such as $X(t_1)$, one can use the state transition matrix, $\Phi(t_0, t_1)$ to move the eigenvectors from t_0 to t_1 and then apply (10) and (11).

3. A Fast Parallel Algorithm For Computing Invariant Manifolds

The fact that invariant manifolds are asymptotic to the periodic orbits means that they are a theoretical concept that cannot be computed exactly. One can only approximate them numerically. Theoretically, the unstable manifold takes an infinitely long time to escape from the periodic orbit, and the stable manifold takes an infinitely long time to approach, but can never reach the periodic orbit. However, nearby the invariant manifolds are trajectories that closely shadow the manifolds with very similar behavior. Using these shadow trajectories, we can either escape the periodic orbit or approach it in finite time. This is the reason there needs to be an offset ε in Eq. (10) (determined heuristically through trial and error) in order to propagate and globalize the invariant manifolds. For the purpose of mission design, these shadow trajectories provide sufficient accuracy for further refinement using high-fidelity models and trajectory-optimization tools.

The classical procedure for computing invariant manifolds described in Section 2 [3] [10] is sufficiently complex to prevent wide usage of this technique in practical mission design and navigation, especially in operations when high accuracy is required. Since the manifolds are defined only for periodic orbits, in high-fidelity models, there are no exact periodic orbits. Often, one is interested in an unstable trajectory segment that is not associated with any periodic or quasi-periodic orbit. A particular disappointment is the lack of application for satellite-tour design, where the transfers between resonant orbits around a single moon or multiple moons are known to follow the invariant manifolds of periodic resonant orbits. However, the moon tours require analysis of huge numbers (millions) of resonant orbits. Computing invariant manifolds in this situation is not practical, and the complexity of the invariant manifold algorithm makes it cumbersome to implement in parallel on modern GPU computing hardware.

Instead of using the eigenvectors of the monodromy matrix, we simply add a unit velocity perturbation to the state vectors along the orbit to produce initial conditions for integration. Due to the local exponential blow-up along unstable directions of the monodromy matrix, the component of this unit perturbation along the unstable eigenvector quickly dominates the solution, and we do not need to compute the monodromy matrix or its eigendecomposition directly. This approach is also easily parallelized, allowing the algorithm to be run across thousands of cores on a GPU, which enables very fast computation of manifolds.

Algorithm Insight

The eigenvectors of M from (8) span the six-dimensional phase space. Hence any vector is a linear combination of these six eigenvectors.

Let Υ be an arbitrary random unit vector in the phase space.

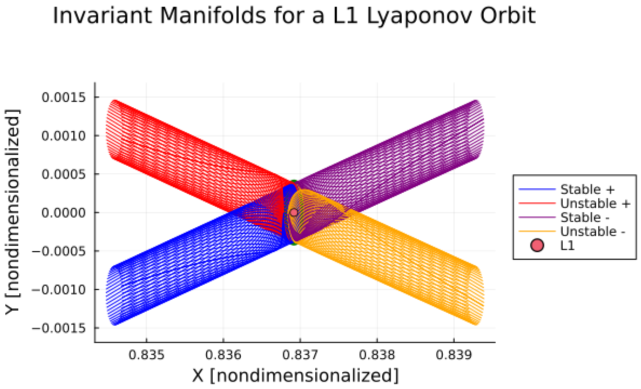

Figure 5: Invariant manifolds of an L1 Lyapunov orbit in the Earth-Moon system computed with the monodromy matrix method

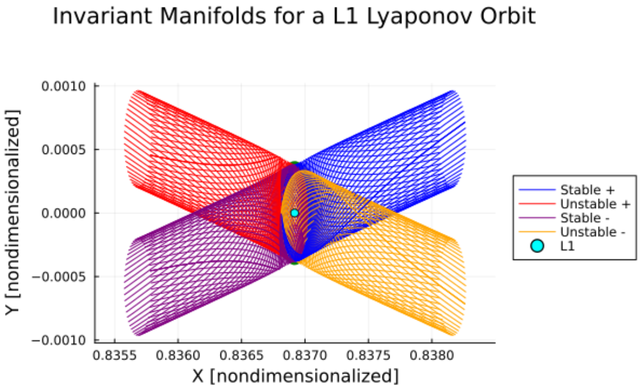

Figure 6: Invariant manifolds of an L1 Lyapunov orbit in the Earth-Moon system computed with our parallel method

Represent Υ with the eigenvector basis:

$$\begin{aligned} \Upsilon \in \mathbb{R}^6, \quad |\Upsilon| = 1, \\ \Upsilon = \Upsilon_1\nu_1 + \Upsilon_2\nu_2 + \Upsilon_3\nu_3 + \Upsilon_4\nu_4 + \Upsilon_5\nu_5 + \Upsilon_6\nu_6. \end{aligned} \tag{12}$$

Now, we globalize $W_u(X(t_0))$ and $W_s(X(t_0))$ using the following initial condition:

$$\begin{aligned} X(t_0) + \varepsilon\Upsilon = \\ X(t_0) + \varepsilon(\Upsilon_1\nu_1 + \Upsilon_2\nu_2 + \Upsilon_3\nu_3 + \Upsilon_4\nu_4 + \Upsilon_5\nu_5 + \Upsilon_6\nu_6) \\ W_u(X(t_0), \tau) = \int F(X(t_0) + \varepsilon\Upsilon), \quad t \in [0, \tau] \\ W_s(X(t_0), \tau) = \int F(X(t_0) + \varepsilon\Upsilon), \quad t \in [-\tau, 0] \end{aligned} \tag{13}$$

Note that W_u is integrated forward in time, while W_s is integrated backward in time. This is because the eigenvalues $\lambda_1 = \frac{1}{\lambda_2} > 1$. So, when integrated forward, the unstable component dominates; when integrated backward, the stable component dominates as the eigenvalues reverse roles. This behavior is due to the fact that the integration of the $\Upsilon_2\nu_2$ component will shrink towards the periodic orbit, the components $\Upsilon_3\nu_3 + \Upsilon_4\nu_4 + \Upsilon_5\nu_5 + \Upsilon_6\nu_6$ are bounded and slow growing, while the $\Upsilon_1\nu_1$ component grows unbounded and dominates the entire computation.

Algorithm 1 Compute Manifolds in Parallel

```
function COMPUTE MANIFOLDS(X)
    ▷ Loop through discrete timesteps along the orbit
    parfor t in range(T) do
        ▷ Orbit state at time t
        $x_t = X[t]$
        ▷ Perturbed orbit state
        $x_p = x_t \pm \epsilon v$
        ▷ Simulate forward to time $T_s$
        $W_u$ = forwardsimulation$(x_p, T_s)$
        ▷ Simulate backward to time $-T_s$
        $W_s$ = backwardsimulation$(x_p, -T_s)$
    end parfor
end function
```

It is theoretically possible that one might pick a $\Upsilon \perp \nu_1$, in which case there is no expanding direction. However, this is practically impossible because the set of vectors $\perp \nu_1$ has zero measure in our sample space.

Parallelization

Since our algorithm only consists of forward and backward simulations from an arbitrary unit-velocity perturbation applied to discrete points along the orbit, we can easily parallelize over all forward simulations on a GPU. Our simulations provide stable and unstable manifolds along the entire orbit in a fast way, which is very valuable for mission design with minimum-energy trajectories. We use JAX [11] to speed up and parallelize our code on a GPU. JAX uses XLA, which is an accelerated linear algebra framework that generates high-performance code to run on accelerators like GPUs. The JAX functions used in this work are $vmap$ for automatic vectorization and $laxscan$ to conduct the forward simulation of the dynamics in an efficient way [11]. Algorithm 1 summarizes the new algorithm, where $parfor$ indicates parallel execution over all iterates. Note that we entirely avoid calculation of the state transition matrix, the monodromy matrix, and its eigendecomposition.

4. NUMERICAL EXPERIMENTS

Table 1: Parameters for normalized coordinates

Parameter	Value
μ	1.215×10^{-2}
L	3.850×10^{5}
V	1.025
T	2.361×10^{6}

We conduct all numerical experiments about the L1 and L2 Lagrange points in the Earth-Moon system. We first analyze the invariant manifolds of a 2D Lyapunov orbit and a 3D halo orbit using our new algorithm and compare the results to the conventional approach for computing invariant manifolds using the monodromy matrix. Then we parallelize the algorithm on a GPU and conduct a speed test on the computation of the unstable manifolds of a Lyaponov orbit around L1. The forward and backward integration allows the stable and unstable directions to grow or contract exponentially and generate the corresponding invariant manifolds.

For this system, the mass parameter and all other unit normalization constants are summarized in Table 1. We compute the Lyapunov and halo orbits for this analysis using a differential-corrector algorithm. Once we obtain these orbits, we sample 50 points along the orbit, add a perturbation to each discrete state along the orbit, and simulate forward and backward. The perturbation used in this analysis was $v = [0, 0, 0, 1, 0, 0]$. We first implement the algorithm in Julia [12], and use RKF78 for the forward and backward simulations using the *DifferentialEquation.jl* Julia package [13]. We then speed up the algorithm by implementing it using the JAX library to parallelize each of the simulations. For the JAX implementation, we use a fixed-step RK8 integrator.

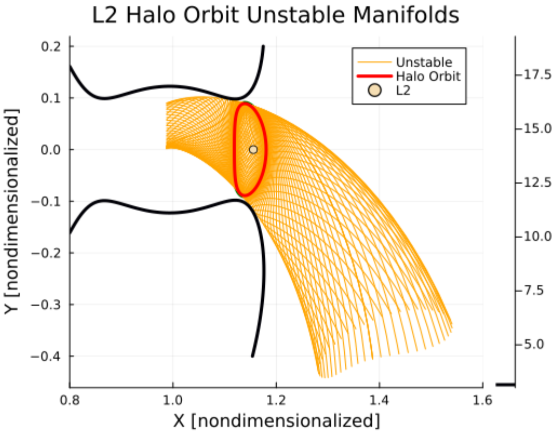

Figure 7: Unstable manifolds of an L2 Halo orbit in the Earth-Moon system computed with the monodromy matrix method

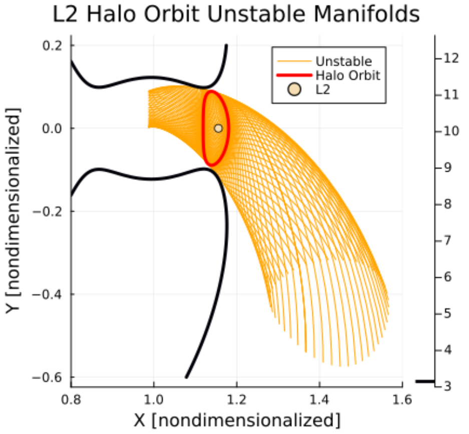

Figure 8: Unstable manifolds of an L2 Halo orbit in the Earth-Moon system computed with our method

2D Lyapunov Orbit Invariant Manifolds

The invariant manifolds for a 2D Lyapunov orbit around L1 in the Earth-Moon system using the CR3BP are shown in Figs. 5 and 6. These manifolds were simulated for $T = 1.583286$ (nondimensionalized), and both methods overall show the same manifold behavior. The Stable + denotes the stable manifold generated when the perturbation was added to the

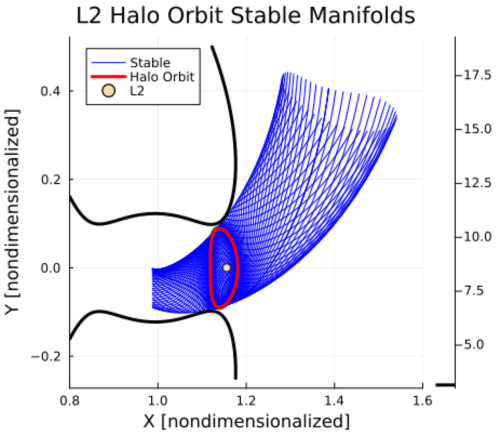

Figure 9: Stable manifolds of an L2 Halo orbit in the Earth-Moon system computed with the monodromy matrix method

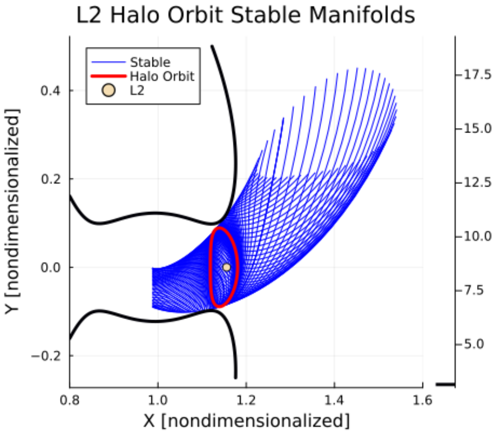

Figure 10: Stable manifolds of an L2 Halo orbit in the Earth-Moon system computed with our method

orbit state and Stable $-$ is generated when the perturbation is subtracted from the orbit state. Note that the $+-$ sides of the manifolds are reversed for Fig. 5 and 6, which may happen as the perturbation is different for both methods, however both still correspond to the stable manifold. Fig. 1 shows the same manifolds integrated using our method for double the time (2T). This shows the structure of the manifolds that can be used for transfer trajectories or exit strategies during missions.

3D Halo Orbit Invariant Manifolds

The manifolds for a halo orbit around the Earth-Moon L2 point using both methods are shown in Figs. 7- 10. Similar to the Lyapunov case, both methods provide the same structure for the stable and unstable manifolds. However, they are not exact, which is expected because the random vector will take time for the stable and unstable direction to expand/contract. Both solutions are still dynamically feasible and approximate the true manifold. The epsilon for this case was set to 1×10^{-5} and the simulation time was T = 5.971226 (nondimensionalized). The black plot in Figs. 7- 10 represents the zero-velocity curve for the specific energy of the orbit.

Timing Results

The algorithm was ported to JAX and parallelized on a server with a 12th Gen Intel Core i9 12900K CPU and an NVIDIA GeForce RTX 3090 GPU. For the timing test, we discretized 10,000 points along a Lyapunov orbit, perturbed these states by an arbitrary vector, and simulated forward by time T to obtain the unstable manifolds in both directions. The arbitrary vector that was added and subtracted was set to $v = \varepsilon[0, 0, 0, 1, 0, 0]$ where $\varepsilon = 1 \times 10^{-4}$. Therefore, there are 20,000 simulation rollouts in total. We time the addition of the perturbation onto the discrete orbit states as well as the forward simulation using the conventional monodromy matrix method on a CPU in Julia, versus our method on a GPU using JAX. The CPU using the monodromy method took 39.050 s while our algorithm solved in 5.050 ms – a speed-up of more than three orders of magnitude.

5. Conclusions

We present a massively parallel method for fast computation of invariant manifolds. Our algorithm outperforms the current method of finding manifolds by several orders of magnitude in execution speed by taking advantage of the thousands of cores available on modern GPU computing hardware. We show that any arbitrary direction can be used to compute these manifolds faster as we avoid the eigendecomposition of the monodromy matrix, which is not trivial to parallelize on a GPU. In these tests, we use the unit velocity direction, however the algorithm generalizes to any velocity vector as the perturbation. Our method also serves as an approximation to the true manifold, which is asymptotic and cannot be computed numerically. However, this approximation provides insight into low-energy trajectories that a spacecraft can follow to enter a periodic orbit to conduct science. Future work will include testing the algorithm with ephemeris data and more complicated scenarios, such as the restricted planar elliptic three body problem. We also plan to run tests on the JPL GPU integrator to compare it with our implementation in JAX.

Acknowledgments

The authors thank the Jet Propulsion Laboratory Strategic University Research Partnerships (SURP) Program and the JPL Spontaneous R&TD Program for support. Author F. Vega thanks the JPL Mission Design and Navigation Section for hosting his summer internship in 2023. This research was carried out in part of the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

References

[1] E. Perozzi and S. Ferraz-Mello, *Space manifold dynamics*. Springer, 2010.

[2] M. W. Lo, B. G. Williams, W. E. Bollman, D. Han, Y. Hahn, J. L. Bell, E. A. Hirst, R. A. Corwin, P. E. Hong, K. C. Howell *et al.*, “Genesis mission design,” *The Journal of the astronautical sciences*, vol. 49, pp. 169–184, 2001.

[3] B. T. Barden, “Using stable manifolds to generate transfers in the circular restricted problem of three bodies,” *Master degree thesis. West Lafayette: School of Aeronautics and Astronautics, Purdue University*, 1994.

[4] H. Poincaré, *Les méthodes nouvelles de la mécanique céleste*. Gauthier-Villars et fils, imprimeurs-libraires, 1893, vol. 2.

[5] A. Haro and R. de la Llave, "A parameterization method for the computation of invariant tori and their whiskers in quasi-periodic maps: numerical algorithms," *Discrete and Continuous Dynamical Systems Series B*, vol. 6, no. 6, p. 1261, 2006.

[6] B. Kumar, R. L. Anderson, and R. de la Llave, "Rapid and accurate methods for computing whiskered tori and their manifolds in periodically perturbed planar circular restricted 3-body problems," *Celestial Mechanics and Dynamical Astronomy*, vol. 134, no. 1, p. 3, 2022.

[7] R. W. Easton, R. L. Anderson, and M. W. Lo, "Conic transfer arcs for kepler's problem," *American Journal of Physics*, vol. 90, no. 9, pp. 666–671, 2022.

[8] W. S. Koon, M. W. Lo, J. E. Marsden, and S. D. Ross, "Dynamical systems, the three-body problem and space mission design," in *Equadiff 99: (In 2 Volumes)*. World Scientific, 2000, pp. 1167–1181.

[9] M. W. Lo, "The interplanetary superhighway and the origins program," in *Proceedings, IEEE Aerospace Conference*, vol. 7. IEEE, 2002, pp. 7–7.

[10] M. S. Gomez, Llibre, "Dynamics and mission design near libration points."

[11] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/google/jax

[12] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017. [Online]. Available: https://epubs.siam.org/doi/10.1137/141000671

[13] C. Rackauckas and Q. Nie, "DifferentialEquations.jl–a performant and feature-rich ecosystem for solving differential equations in Julia," *Journal of Open Research Software*, vol. 5, no. 1, 2017.

Biography

Fausto Vega *is a PhD student at the Robotics Institute at Carnegie Mellon University (CMU). He received a BS in mechanical engineering from the University of Nevada, Las Vegas and an MS in Robotics at CMU. His research interests include small spacecraft state estimation and optimal control.*

Martin Lo *is a mission designer in the Navigation and Mission Design Section of the Jet Propulsion Laboratory, California Institute of Technology. He has a BS (75) from Caltech and a PhD (80) from Cornell University in mathematics. His specialties are in the three body problem (libration missions using dynamical systems theory) and in constellation design.*

Zac Manchester *is an assistant professor in the Robotics Institute at Carnegie Mellon University and founder of the Robotic Exploration Lab. He received a PhD in aerospace engineering in 2015 and a BS in applied physics in 2009, both from Cornell University. His research interests include control and optimization with applications to aerospace and robotic systems.*

Jon Sims *is a Principal Engineer and Technical Group Supervisor in the Mission Design and Navigation Section at the Jet Propulsion Laboratory. Over the past 25 years at JPL, he has developed new mission design capabilities, led mission proposals, and created mission designs for multiple projects. He received his Ph.D. and M.S. in Aeronautics and Astronautics from Purdue University and his Bachelor's degree from The Ohio State University.*