Design Transfers to Libration Point Orbits through Graphics Processing Unit Parallel Computing

Hao Peng[1] and Xiaoli Bai[2]
Rutgers, The State University of New Jersey, NJ, 08854

Josep J. Masdemont[3]
Polytechnic University of Catalonia, Barcelona, Spain

Gerard Gómez[4]
University of Barcelona, Barcelona, Spain

Shijie Xu[5]
Beihang University, Beijing, China

Design of the transfer from lunar orbits to the Sun-Earth libration point region by direct searching in the high-fidelity ephemeris model is an accurate but time-consuming practice. We present a computationally efficient methodology that takes advantage of the Patched Elliptic Restricted Three-Body Problem (ERTBP) model, the power of Graphics Processing Unit (GPU) parallel computing, and the programing platform of the Matlab. Taking the Chang'E-2 extension mission as an instance, the proposed implementation obtains almost identical results with that in the ephemeris model and shows significant speedup. Moreover, the methodology can be carried out on inexpensive hardware platforms. Numerical results demonstrate that significant speedups can be achieved using the GPU parallel computing in Matlab with a little learning cost.

---

[1] Postdoctoral Associate, Department of Mechanical & Aerospace Engineering, AstroH.Peng@gmail.com
[2] Assistant Professor, Department of Mechanical & Aerospace Engineering, xiaoli.bai@rutgers.edu (corresponding author)
[3] Professor, Department of Mathematics, josep.masdemont@upc.edu
[4] Professor, Department of Applied Mathematics and Analysis, gerard@maia.ub.es
[5] Professor, School of Astronautics, starsjxu@163.com

I. Introduction

Libration point orbits in the Circular Restricted Three-Body Problem (CRTBP) have been studied for solar system and astronomical studies due to their special location and environment. In the past three decades, many spacecrafts, such as ISEE-3, Soho, Genesis, Herschel, and Gaia among others, have reached the Sun-Earth L_1 and L_2 points for space explorations or astronomical observations [1]. Periodic orbits, quasi-periodic orbits, and their invariant manifolds in three-body systems are well-known dynamical structures and have been employed as tools to design various types of trajectories, such as Sun-Earth libration point missions [2, 3], low-energy captures into Mars halo orbit [4, 5], and transfers between collinear libration points [6–8]. However, when a spacecraft moves in the space where all the gravitational influences of the Sun, Earth, and Moon are too large to be ignored, the complexity of the trajectory design problem increases. During preliminary studies, the standard way to deal with this restricted four-body system, where a spacecraft is considered as the forth body with infinitesimal mass, is to decouple the system into two overlapping restricted three-body problems, for example the Sun-Earth CRTBP and the Earth-Moon CRTBP. The spacecraft can be shifted from one system to another by implementing maneuvers at the intersection points of two manifolds coming from different three-body regimes [9, 10]. Usually a refinement of the patched transfer in a high-fidelity ephemeris model is necessary after these preliminary designs. Nevertheless, sometimes the refinement of is not straightforward and will severely distort the patched trajectory.

The Moon can be used as an outpost for future deep space exploration, under which circumstance the transfers between lunar orbits and Libration Point Orbits (LPOs) will serve as the basis of the transportation. In this study, the transfer from a lunar orbit to a Sun-Earth LPO around L_2 is investigated. The background of such a transfer is the extension mission of the Chinese lunar probe Chang'E-2 (CE-2) [11–13]. In this paper, a purely numerical approach is utilized to construct natural transfers from a lunar orbit to the Sun-Earth LPO neighborhood, in the complete DE405 ephemeris model. The numerical approach utilizes a two-step strategy to extend the duration of a spacecraft spent in the LPO neighborhood. When the duration is long enough, the spacecraft is actually arriving on an LPO, because all motions except the LPO will leave the LPO neighborhood rapidly due to the instability of the CRTBP. Compared to analytical or (semi-analytical)

approaches, a numerical strategy is straightforward, efficient, and capable of introducing various mission constraints. After obtaining large amount of transfers, their features such as the departure velocity and the distribution of all transfers are analyzed. These features are the key points that should captured by any successful approximate model for this problem carried out either on the Central Processing Unit (CPU) or the Graphic Processing Unit (GPU) platform.

Direct trajectory searching in the ephemeris model is an accurate approach of preliminary mission design for deep space missions. However, this is usually a time-consuming practice. To speed up the searching, CPU parallel computing can be brought in to increase the searching speed, which is a common solution to speed up simulations or computations. On the other hand, the state-of-the-art GPU parallel computing has shown great potential in handling massive computations in many areas in recent years.

There have been some applications of the GPU parallel computing for trajectory design problems. A naturally parallelizable global point mascon model for the geopotential is established through effective parallel computing on the GPU by Russell and Arora [14]. Bai et al. develop the Modified Chebyshev-Picard Iteration (MCPI) method to propagate ordinary differential equations and to solve initial and boundary value problems [15–17], which is well suited for GPU implementations using the Compute Unified Device Architecture (CUDA) [18]. Nakhjiri and Villac further improve the MCPI method to be able to calculate state transition tensors only with differentiation of polynomial, by developing particular algorithms appropriate for the GPU micro-architecture [19]. Wagner et al. have examined four Lambert solution methods on the GPU [20], by directly modifying Fortran routines to fit the requirements of CUDA, which is the GPU parallel programming development environment of NVIDIA. Arora et al. utilize GPU parallel computing techniques to develop a high-fidelity trajectory propagation tool, which is used to simulate the orbit of resident space objects [21]. They developed delicate routines using FORTRAN and CUDA to fully make use of the GPU, including a particular approximated ephemeris model to feed GPU integrators with the third-body perturbation data.

A great barrier that stops most researchers to employ GPU parallel computing is the high learning cost. For instance, one needs to design communications between CPU and GPU, which is

a difficult task as elaborated in Arora's dissertation [22]. Writing programs directly with CUDA is a time-consuming job and requires high-level computer science skills. For researchers, it is much preferred to exploit GPU parallel computing easily and quickly. And it is especially helpful when one wants to do some tests before determining whether to migrate to the CUDA environment or not. In this paper, we demonstrate that it is possible to conveniently take advantage of the powerful GPU parallel computing from a high-level programing approach, such as the Matlab platform, to design complex interplanetary trajectories.

Apart from learning cost, there are additional challenges and constraints to writing CUDA programs, which makes it difficult to solve trajectory design problems on the GPU. For instance, in the numerical searching in the ephemeris model mentioned above, the GPU does not support caching ephemeris data for each thread on the GPU due to its physical structure. The GPU is also not designed to support ephemeris packages like SPICE (or its Matlab interface MICE). Another barrier is that CPU parallel computing is mature and it can stop researchers from migration to the GPU platform. But as will be revealed by our results, the GPU is particularly efficient to handle massive searches of transfers.

In this paper, we have made four contributions to overcome the above mentioned challenges: 1) we establish a new Patched ERTBP model that approximates high-fidelity ephemeris model accurately for deep space trajectory design in the Sun-Earth-Moon system, which is a significant improvement of standard Patched CRTBP model; 2) our experiments demonstrate that eccentricities of the Sun-Earth ($\sim$0.0167) and Earth-Moon ($\sim$0.0554) system should be considered in order to improve the accuracy of preliminary trajectory designs by using ERTBP instead of CRTBP models; 3) our results reveal that eccentricities is not the major cause of the monthly variation of the distribution of transfers from lunar orbits to LPOs around the Sun-Earth L_2; and 4) we also show the potential great speedup that GPU parallel computing can achieve for space mission designs, and presente a convenient approach to incorporate GPU parallel computing with a little learning cost and on inexpensive computation platforms.

The outline of the paper is as follows. In Section II, the two-step strategy searching feasible transfers from the lunar orbit of the CE-2 to Sun-Earth L_2 point neighborhood is introduced in

the ephemeris model. In Section III, the Patched ERTBP model is established to approximate the Sun-Earth-Moon system by decoupling it into two ERTBP models. The model is programed using Matlab and can be launched on both CPU and GPU. In Section IV, the search of transfers is first carried out in the DE405 ephemeris model using CPU parallel computing, and then in the Patched ERTBP model using GPU parallel computing. The comparison between the two results validates that the Patched ERTBP model approximates the ephemeris model well for the trajectory design problem in this paper. A speedup analysis is also presented to demonstrate the high enhancement one achieves from of the GPU parallel computing suing Matlab. In Section V, conclusions about the result and the implementation in this study are given.

II. Methodology for Searching Transfers in Ephemeris Model

In this section, the Sun-Earth LPO region is defined first, and then the two-step strategy searching for transfers from a lunar orbit to Sun-Earth LPOs is presented in two subsections.

In the CRTBP, there are five equilibrium points known as libration points, where three of them, L_1, L_2 and L_3, are collinear and unstable. In this work we are interested in the families of LPO around the Sun-Earth collinear points L_2, including periodic (Lyapunov and Halo) and quasi-periodic (Lissajous and quasi-Halo) orbits [23, 24].

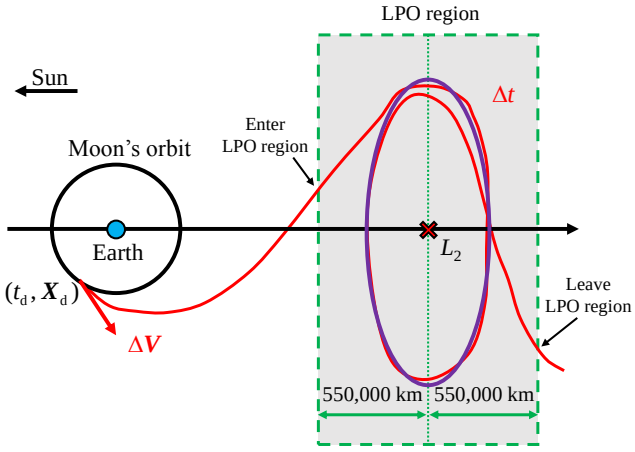

Fig. 1 Illustration of the LPO region about the Sun-Earth L_2 point.

Previous researches [10] indicate that there exists zero-cost transfers between the Sun-Earth and Earth-Moon libration point regions. A spacecraft orbiting the Moon can reach a Sun-Earth

LPO asymptotically without other maneuvers but only an escaping maneuver at the lunar orbit [1]. Based on these results, we first define the LPO region in the Sun-Earth system, which is a cuboid neighborhood centered at the L_2 point in the Sun-Earth rotating frame. Due to the fact that the zero velocity curve limits the size of an LPO [24], only the x component is needed to define the LPO region. Figure 1 illustrates the projection of the defined LPO region on the ecliptic plane, represented by the gray box region, centered at the L_2 point. The width of the region along x-axis should be designed to contain all LPOs of interest, so the upper and lower boundaries for the x component of the region are 5.5×10^5 km away from L_2 point.

Figure 1 also shows a transfer trajectory (red curve), which asymptotically approaches an LPO (purple circle) along its stable manifold, but will finally leave the LPO region. This is because any small deviations will increase exponentially due to the instability of the LPO region. At the departure epoch t_d, the tangential departure maneuver ΔV is performed at the lunar orbit to transfer the spacecraft to a Sun-Earth LPO. After performing the departure maneuver, the initial inertial state $\boldsymbol{X}_\mathrm{d} = [X, Y, Z, V_x, V_y, V_z]^\mathrm{T} \in \mathbb{R}^6$ of the spacecraft is propagated forward in time in the DE405 ephemeris model, until it enters and leaves the LPO or reaches a maximum propagation time. The time duration Δt that the spacecraft remains in the LPO region plays a key role in the following two-step strategy to construct transfers. We remark that the following numerical method is first introduced in a previous study [25], which focuses on more general analysis of the transfer and the correction maneuver.

A. Grid Searching for Potential Transfer

The first step is a grid searching of initial departure conditions with the goal to locate all potential transfers, which lead to trajectories that enter and remain in the LOP neighborhood for a certain time Δt_min.

To compute the transfers, the set of variables $(t_\mathrm{d}, \Delta V)$ which defines an initial condition is discretized into two-dimensional grids on the parameter plane. Each grid of initial condition is propagated up to its first intersection with the LPO region at a certain epoch t_in, and then we start to accumulate the time duration Δt that a spacecraft remains in the LPO region. At each step, the

trajectory is transformed into a synodic frame using standard methods [24] to examine its position with respect to the LPO region. The propagation is stopped when the trajectory leaves the LPO region at a certain epoch t_{out}.

Due to the fact that the escaping rate in the LPO region is exponential, a trajectory that only temporarily passes by the LPO region will leave rapidly, but a trajectory that tends to arrive on an LPO will stay much longer. So, in order to reject those transient trajectories, a minimum duration $\Delta t_{\min}$ is specified as 92 days, which is approximately a half period of a typical Sun-Earth LPO. The period of an LPO is typically around a half of the period of the primaries, so it is around 183 days for a Sun-Earth LPO. During the search, one grid is considered to lead to a potential transfer if its initial condition leads to a trajectory whose Δt is larger than $\Delta t_{\min}$, and then the grid is proceeded for refinement in the next step. Otherwise, the grid is rejected.

B. Refine Potential Transfers with Bisection Method

In the second step, a bisection strategy is used to extend the duration Δt of a potential transfer to a desired duration Δt_{des} specified as 450 days, by refining the initial condition of the transfer. This 450-day requirement corresponds to 2.5 revolutions of a typical Sun-Earth LPO around the L_2 point. The number is determined by a trail-and-error process to ensure both a long enough duration Δt and an acceptable computation cost. If the refinement of a potential transfer succeeds, the refined ΔV is recorded as giving a feasible transfer.

The refinement algorithm is illustrated in Fig. 2. Starting from a potential transfer, we slightly vary the initial condition of the corresponding grid value of ΔV to get two nearby values $\Delta V - \delta V$ and $\Delta V + \delta V$, between which we assume there should be a feasible transfer satisfying $\Delta t \geq \Delta t_{\text{des}}$. This transfer is detected by a bisection iteration summarized below (as shown in Fig. 2):

1. Set the desired duration time $\Delta t_{\text{des}} \geq \Delta t_{\min}$; set the initial maneuver $\Delta \boldsymbol{V}^{(0)}$ as that of the potential transfer; set the initial deviation $\delta V^{(0)} > 0$; set the grid number $N \in \mathbb{N}^+$ and maximum iteration number $i_{\max} > 0$.

2. Assume at iteration i, propagating the trajectory with $\Delta V_k^{(i)}$ gives the maximum $\Delta t_k^{(i)}$.

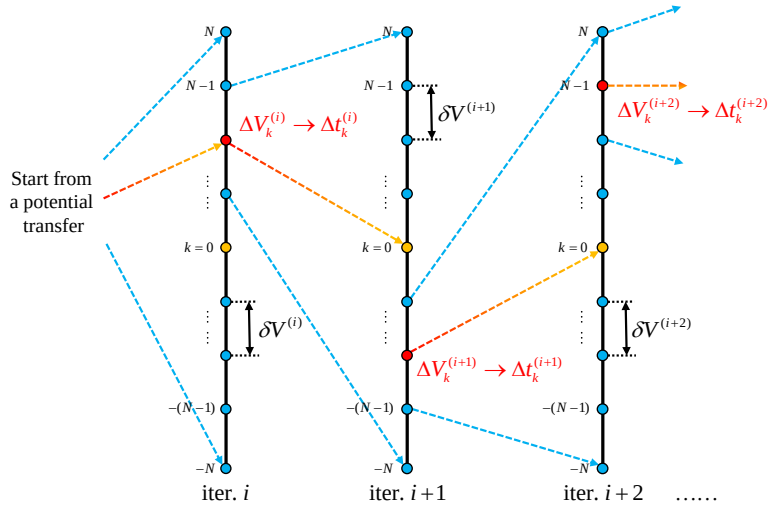

Fig. 2 Illustration of the bisection refinement procedure for a potential transfer.

3. At iteration $i+1$, propagate the $2N+1$ initial conditions with $\Delta V_k^{(i+1)} = \Delta V_k^{(i)} + \frac{k-N}{N} \cdot \delta V_k^{(i)}$ $(k = 0, 1, ..., 2N)$, and recored all durations $\Delta t_k^{(i+1)}$ within the LPO region.

4. Update Δt by the largest $\Delta t_k^{(i+1)}$.

5. If $\Delta t \geq \Delta t_{\text{des}}$, then stop the iteration, and return current Δt and the corresponding $\Delta V_k^{(i+1)}$ as the feasible transfer.

6. If $i \geq i_{\max}$, then stop and report the potential transfer as not feasible.

7. Let $\delta V^{(i+2)} = \delta V^{(i+1)}/N$, go to step 3 and move on to next iteration.

With the above procedure, we can determine transfers that remain inside the LPO region for at least 450 days. Additionally, small corrections of the order of 0.1 m/s or even smaller are needed in the refinement. Most of potential transfers that fail this refinement procedure are in fact transient trajectories passing through the LPO region.

This two-step searching procedure would require a great amount of computation time. However, the above searching algorithm is naturally parallelizable. In next section, we exploit a new approach that utilizes GPU parallel computing.

III. GPU Parallel Searching in Patched ERTBP Model

To design transfers discussed in the last section, at least the Sun-Earth-Moon system should be considered, leading to a four-body problem together with the spacecraft. We choose to decouple this four-body system into the Sun-Earth ERTBP and the Earth-Moon ERTBP model in this paper. In this section, first the ERTBP model is briefly reviewed, then the Patched ERTBP model is established, and at last the GPU implementation of the Patched ERTBP model is presented.

A. Elliptic Restricted Three-Body Problem

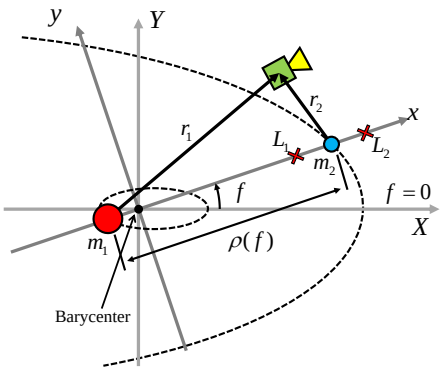

Fig. 3 Pulsating synodic frame of the ERTBP.

In the ERTBP, as shown in Fig. 3, the two massive primaries m_1 and m_2 ($m_1 > m_2$) revolve round their common barycenter on two elliptic Keplerian orbits respectively, with the same eccentricities and periods. The third body, the spacecraft, has ignorable mass compared with the two primaries. So the spacecraft is assumed to move in the mutual gravitational field of the two primaries, but does not affect the motion of the primaries. The distance ρ between m_1 and m_2 is varying with respect to the time t (or the true anomaly f) as $\rho(f) = \frac{a(1-e^2)}{1+e\cos f}$, where a is the semi-major axis and e is the eccentricity. As shown in Fig. 3, a nonuniformly rotating and periodically pulsating frame (x, y, z) located on the barycenter is used to simplify the equations of motion. The x-axis is always pointing from m_1 to m_2, therefore rotating nonuniformly. In Fig. 3, the synodic frame (x, y, z) is related to the inertial frame (X, Y, Z) by the true anomaly f measured anticlockwisely from the inertial X-axis, which is always connecting the focuses of the two primaries. The z-axis (or Z-axis) is pointing outside of the paper and omitted for clarity, and the y-axis (or Y-axis) completes a right-handed frame.

The instantaneous distance $\rho(f)$ between two primaries, the total mass $m_1 + m_2$, and the reciprocal of the mean motion $\bar{n}$ are chosen as the distance, mass, and time units respectively. Then, the independent variable is transformed from the time t to the true anomaly f by the chain rule $\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\mathrm{d}f}{\mathrm{d}t} \cdot \frac{\mathrm{d}}{\mathrm{d}f}$ where $\frac{\mathrm{d}f}{\mathrm{d}t} = (1 + e\cos f)^2/\sqrt{(1-e^2)^3}$ is the angular velocity. In this way, the two primaries are fixed on the x-axis at $x_1 = -\mu$ and $x_2 = 1 - \mu$, where $\mu = \frac{m_2}{m_2+m_2}$ is the normalized mass of m_2. At last, the equations of motion of the spacecraft can be expressed as [23]

$$\begin{cases} x'' - 2y' = \dfrac{\partial \omega}{\partial x}, \\ y'' + 2x' = \dfrac{\partial \omega}{\partial y}, \\ z'' + \ z \ = \dfrac{\partial \omega}{\partial z}, \end{cases} \tag{1}$$

where the primes indicate that the derivatives are taken with respect to the true anomaly f, and

$$\omega(x, y, z, f) = \frac{1}{1 + e\cos f}\left[\frac{1}{2}(x^2 + y^2 + z^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{1}{2}\mu(1-\mu)\right], \tag{2}$$

with $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$ are the instantaneously normalized distances to the two primaries respectively.

The ERTBP is a nonautonomous system explicitly depending on the time t, due to the dependency of ω on f and thus on t as shown in Eq. (2). Simply substituting $e = 0$ into Eqs. (1) and (2) will lead to the equations of motion in the CRTBP. The ERTBP does not have any first integral and is nonconservative [26, 27], therefore it is much more complicated than the CRTBP.

In this paper, we decouple the Sun-Earth-Moon system into the Sun-Earth ERTBP with $\mu_1 \approx 3.040 \times 10^{-6}$, $e_1 \approx 0.0167$, and the Earth-Moon ERTBP with $\mu_2 \approx 0.0122$, $e_2 \approx 0.0554$. Hereinafter, the subscript i of a symbol ξ_i indicates that it is defined for the Sun-Earth ERTBP when $i = 1$, and the Earth-Moon ERTBP when $i = 2$. The obliquity of the Ecliptic and Moon's orbit has been ignored in the model. Although more parameters can be introduced to model the angle, but as will be demonstrated in Section IV, the result has already approximated the result in the ephemeris model very well.

B. Patched ERTBP Model

Although the CRTBP model is usually used as the basic three-body model when decoupling the Sun-Earth-Moon system into two simpler CRTBP models [28, 29], the ERTBP model has also

been used [27]. By considering the eccentricity, the model becomes more complicated but whereas more realistic for designing transfers.

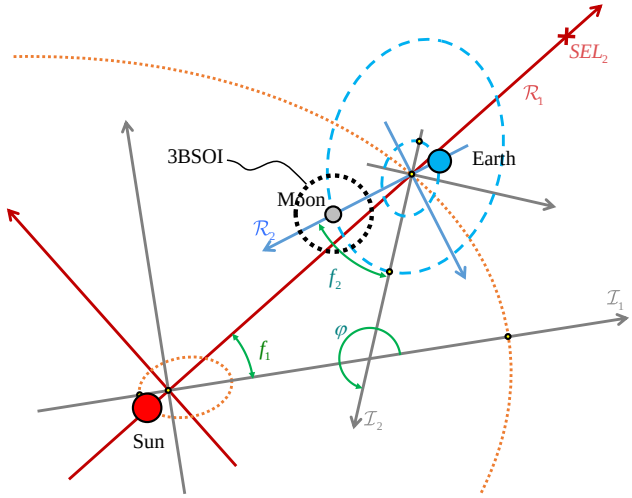

Fig. 4 Illustration of the Patched Sun-Earth ERTBP and Earth-Moon ERTBP model.

Figure 4 shows the geometry of the two ERTBP models, where the z-axis and Z-axis of both ERTBP models are always pointing outside of the paper and thus omitted for clarity. The synodic frames $\mathcal{R}_i$ and inertial frames $\mathcal{I}_i$ $(i = 1, 2)$ are defined in each ERTBP model as described in the previous section. The synodic frame $\mathcal{R}_1$ (red) represents the Sun-Earth ERTBP, where the Sun is the major primary and the Earth-Moon barycenter is the secondary primary. The synodic frame $\mathcal{R}_2$ (blue) represents the Earth-Moon ERTBP, centered at the secondary primary of the Sun-Earth ERTBP. Each pair of $\mathcal{I}_i$ and $\mathcal{R}_i$ is related by the corresponding true anomaly angle f_i. The two inertial frames $\mathcal{I}_1$ and $\mathcal{I}_2$ (gray) are related by the constant angle φ in Fig. 4 between x-axis of the two frames.

To patch the two ERTBP models, we adopt the concept of the Three-Body Sphere-Of-Influence (3BSOI) [27, 30]. In Fig. 4, the dashed black circle centered at the Moon represents this 3BSOI, with a radius of $159,200$ km [30]. The 3BSOI is instantaneously normalized in both ERTBP models, so it appears pulsating periodically during propagation. When a spacecraft is located inside of the 3BSOI, its motion is propagated in the Earth-Moon ERTBP, ignoring the solar perturbation; when the spacecraft gets out of the 3BSOI, the motion is propagated in the Sun-Earth ERTBP, ignoring the irregular motion of the Moon.

Denote the state of a spacecraft in the Sun-Earth ERTBP rotating frame $\mathcal{R}_1$ by $\boldsymbol{x}_1 = [x, y, z, x', y', z']^{\mathrm{T}} \in \mathbb{R}^6$, and in the Earth-Moon ERTBP by $\boldsymbol{x}_2 \in \mathbb{R}^2$. The conversion between the two ERTBP models takes several steps. First, we translate $\boldsymbol{x}_1$ to an intermediate frame located at the Earth-Moon barycenter but parallel with $\mathcal{R}_1$, and clockwisely rotate this frame with an angle of f_1 so that it is parallel with the inertial frame $\mathcal{I}_1$. Denote the new state in this temporary inertial frame by $\boldsymbol{X}_1$, and we have the expression,

$$\boldsymbol{X}_1 = \begin{bmatrix} \boldsymbol{R}_z^{-1}(f_1) \cdot \rho_1(f_1) & \boldsymbol{0}_3 \\ \dfrac{\mathrm{d}\left[\boldsymbol{R}_z^{-1}(f_1) \cdot \rho_1(f_1)\right]}{\mathrm{d}f_1} \cdot \dfrac{\mathrm{d}f_1}{\mathrm{d}t_1} & \boldsymbol{R}_z^{-1}(f_1) \cdot \rho_1(f_1) \cdot \dfrac{\mathrm{d}f_1}{\mathrm{d}t_1} \end{bmatrix} \cdot (\boldsymbol{x}_1 - [1-\mu_1, 0, 0, 0, 0, 0]^{\mathrm{T}}). \tag{3}$$

Second, we rotate the intermediate frame anticlockwisely by an angle of φ, and get the state $\boldsymbol{X}_2$ in the Earth-Moon ERTBP inertial frame $\mathcal{I}_2$, expressed as

$$\boldsymbol{X}_2 = \begin{bmatrix} \boldsymbol{R}_z(\varphi) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_z(\varphi) \end{bmatrix} \cdot \boldsymbol{X}_1. \tag{4}$$

Finally, we rotate the frame anticlockwisely by an angle of f_2 to get the state of the spacecraft $\boldsymbol{x}_2$ in the Earth-Moon ERTBP rotating frame $\mathcal{R}_2$, expresses as

$$\boldsymbol{x}_2 = \begin{bmatrix} \boldsymbol{R}_z(f_2)/r_{12.2}(f_2) & \boldsymbol{0} \\ \dfrac{\mathrm{d}\left[\boldsymbol{R}_z(f_2)/r_{12.2}(f_2)\right]}{\mathrm{d}t_2} \cdot \dfrac{\mathrm{d}t_2}{\mathrm{d}f_2} & \boldsymbol{R}_z(f_2)/r_{12.2}(f_2) \cdot \dfrac{\mathrm{d}t_2}{\mathrm{d}f_2} \end{bmatrix} \cdot \boldsymbol{X}_2. \tag{5}$$

In above equations ρ_i $(i = 1, 2)$ stands for the distance between two primaries in two ERTBP models, and $\boldsymbol{R}_z(\nu)$ is the rotational matrix of a frame with respect to its z-axis, defined as

$$\boldsymbol{R}_z(\nu) = \begin{bmatrix} \cos\nu & \sin\nu & 0 \\ -\sin\nu & \cos\nu & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6}$$

The Patched ERTBP model is initialized as below. Given an initial epoch $\hat{t}_0$, where the hat over t means dimensional variable, we extract from the ephemeris data the geometric configuration of the Sun-Earth-Moon system. Then, the initial true anomalies $f_{i.0}$ $(i = 1, 2)$ of the two ERTBP models can be obtained from the geometric configuration. Additionally, the constant frame angle φ at $\hat{t}_0$ can be obtained by calculating the angle between the eccentricity vector $\boldsymbol{e}_1$ of Sun-Earth system and $\boldsymbol{e}_2$ of the Earth-Moon system.

Because both ERTBP models are nonautonomous, after initializing the Patched ERTBP model, the timeline of the two ERTBP models should always be synchronized according to the following relationship

$$(\hat{t} - \hat{t}_0) \underset{\hat{t}/\mathrm{TU}_i}{\overset{t_i \cdot \mathrm{TU}_i}{\rightleftharpoons}} t_i \underset{\text{simple convertion}}{\overset{\text{solving Kepler's equation}}{\rightleftharpoons}} f_i, \tag{7}$$

where $\hat{t}$ denotes the dimensional time duration, TU_i denotes the time unit and t_i denotes the normalized time in the ERTBP. As revealed by Eq. (7), the initial value of t_i is $t_{1.0} = t_{2.0} = 0$ at the epoch $\hat{t}_0$.

The Patched ERTBP is a model that only requires initial conditions, and does not depend on external data files like what the ephemeris model does. So the Patched ERTBP model is naturally parallelizable. Although it is possible to parallel this kind of searching on CPUs on powerful and expensive clusters, for researchers without access to clusters, it will be appealing to benefit from GPU parallel computing.

C. GPU implementation of Grid Searching in Patched ERTBP Model

Substituting the dynamic model in Section II by the Patched ERTBP model allows carrying out a similar search for transfers from a lunar polar orbit to the Sun-Earth LPO region around L_2 point. The grid searching methodology only requires a series of propagation in the Patched ERTBP model starting with different initial conditions. Since the model can be implemented on the GPU, the searching methodology is now naturally a parallelizable problem with separated input data.

The remaining task to launch this search algorithm on a GPU card is to find a proper numerical integrator. In fact, most numerical integrator based on serial operations can be directly migrated onto the GPU, for example the popular Dormand-Prince integrators. A serial integrator usually runs slower on a GPU then on a CPU due to the lower frequency of the GPU. But the GPU can be more efficient when there are a large amount of computations, because it can launch thousands of integrators concurrently. In this paper, the numerical propagation algorithm implemented by `ode45` in Matlab is modified to run on a GPU core. This choice of `ode45` provides an easy validation of the computing accuracy of the GPU, by simply comparing the results of the regular `ode45` and the GPU-version `ode45`. The regular `ode45` returns all the propagation history to users but the

GPU-version `ode45` only save a current step when looping the propagation iterations. However, this is adequate for our grid searching in the Patched ERTBP model. The propagation history is not required to calculate the total duration Δt in the LPO region, because it is accumulated at each propagation iteration, as elaborated in Section II.

Although Matlab has already provided a high-level access to exploit GPU parallel computing, the limitations of GPU parallel computing still exist. The official documents of the Matlab has elaborated all the essentials. Here we only review some critical concepts based our programming experiences, which are believed to be helpful to audiences with few GPU experiences. A Matlab function or script that works on the GPU cannot use matrix indexing, cannot dynamically allocate memories for variables, cannot use most high-level built-in functions in Matlab, and cannot use most advanced data structures. A convenient way to write a Matlab GPU program is to assume that it only supports the very basic element-wise program, similar as what the ANSI-C language supports. For instance, the inner product of two vectors $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^n$ should be directly programed as an element-by-element summation $\sum_{i=1}^{n} x_i\, y_i$, rather than using vectorized programming in Matlab. Fortunately, all computations within the Patched ERTBP model can be implemented element-wisely without using unsupported functions.

IV. Results and Discussions

In this section, the search results in the ephemeris model and the Patched ERTBP model are demonstrated and compared in the first two subsections; the effect of the eccentricities of the two ERTBP is analyzed in the third subsection; and at last a speedup analysis of the GPU parallel computing on the transfer design problem is presented.

A. CPU search result in Ephemeris Model

We use the Chang'E-2 (CE-2) as an application example, and the Keplerian elements of the orbit are listed in Table 1. The departure state $\boldsymbol{X}_{\mathrm{d}} \in \mathbb{R}^6$ at an epoch t_{d} before the lunar departure maneuver $\Delta \boldsymbol{V}$ are converted from the orbital elements.

Using the approach developed in Section II, we have investigated the transfers from the lunar polar orbit used by the CE-2 spacecraft to an LPO around the Sun-Earth L_2 point. A Runge-Kutta

Table 1 Lunar orbital elements of CHANG'E-2 at epoch 55656.0 MJD [5].

a(km)	e	$i(^\circ)$	$\omega(^\circ)$	$\Omega(^\circ)$	$M(^\circ)$
1838	0.003	86	323	259	152

7/8-order integrator has been used to propagate the trajectories with a maximum local truncation error of 10^{-13} in the nondimensional unit. Gravitational attractions of the Moon and all major planets in the solar system are included, with their positions provided by the DE405 ephemeris data. The searching program is written in FORTRAN77 language and carried out on a Linux cluster with Intel Xeon CPUs. The CPU parallel computing is manually carried out by dividing the whole searching to multiple threads.

The grid size of the departure epoch t_{d} is set as 10 minutes, and the grid size of ΔV is set as 0.0004 km/s. The departure state $\boldsymbol{X}_{\mathrm{d}}$ is converted from the CE-2's orbital elements in Table 1 to Cartesian coordinates in the inertial frame. Since the CE-2 departed from the lunar orbit on June 9th, 2011 [13], we explore the departing epochs t_{d} within the whole year of 2011, i.e., between 2011/01/01 and 2012/01/01. The magnitude ΔV of the departure maneuver is chosen in the range of [0.640, 0.720] km/s. This interval is designed to be large enough to fill up the gap in the Sun-Earth CRTBP model between the Jacobian constant J_C of the CE-2 orbit and the J_C of a potential LPO. In the parameter space $(t_{\mathrm{d}}, \Delta V)$, a complete grid searching for feasible transfers that stay in the LPO region for more than 450 days is performed. The successful grid points are recorded as initial conditions for feasible transfers, including the modified departure maneuver ΔV, flight time before entering LPO region t_{FT}, and other useful information for program diagnostics.

The searching procedure has detected a huge number of transfer candidates fulfilling the above requirements. We will consider the feasible transfers with $t_{\mathrm{FT}} \leq 43$ days in following studies, which are displayed in Fig. 5. The horizontal axis represents the departure epoch t_{d} in year 2011, and the vertical axis represents the magnitude of the departure maneuver ΔV. Dots of $(t_{\mathrm{d}}, \Delta V)$ in each subplot are colored according to their flight time t_{FT}, as shown by the colorbar.

In Fig. 5, there are two different families of transfers with $t_{\mathrm{FT}} \leq 43$ days: *Family I* in the lower part of the plot, and *Family II* with parabolic boundaries in the upper part. The pattern of both families is repeated with a periodicity about 28 days. Figure 6 shows a time interval of initial

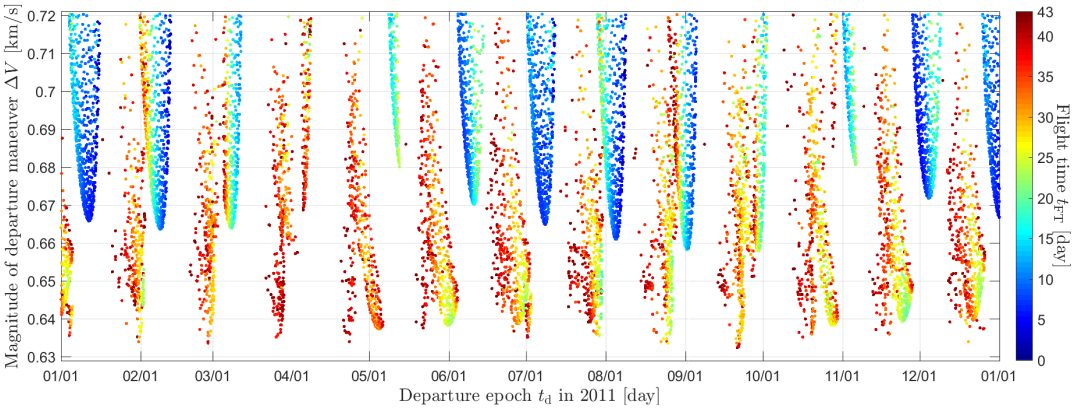

Fig. 5 Distribution of feasible transfers in the year of 2011.

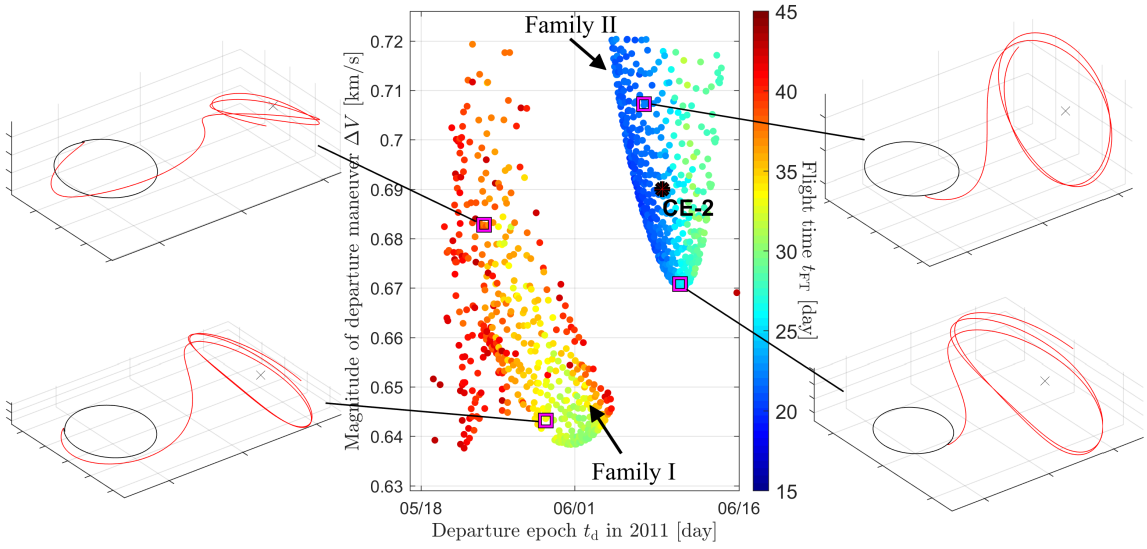

Fig. 6 Examples of transfers in Families I (left) and II (right) colored from 15 to 45 days. The red asterisk stands for the initial condition corresponding to the CE-2 [13]. The black curves displayed along with transfers in surrounding subplots represent the lunar orbit. It also reflects the relative size of the final LPOs.

conditions of feasible transfers around the departing epoch of the CE-2 (June 9th, 2011) with more details. Four transfer examples from each families are demonstrated on two sides of the distribution plot, whose corresponding departure conditions are shown by squares in the plot. The transfers in Family I usually take longer flight times, but do not always require smaller ΔV than the one used by the transfers in Family II. The transfer examples in Family I (the left column in Fig. 6) show that the trajectory will slightly return to the Moon after the initial departure, while the examples

in Family II (the right column) directly enter the LPO region. These different behaviors can explain the lower ΔV cost required by some transfers in Family I, because the Moon provides additional gravitational assistances to them.

The departure condition of the CE-2 is represented by the asterisk in Fig. 6, departing on June 9 with a departure maneuver of approximately 0.690 km/s [13]. Therefore, the CE-2 probe actually exploited a transfer in Family II. As shown in the figure, leaving aside practical operational constraints, a transfer with less ΔV or smaller t_{FT} can be designed. So, in practical mission design, it is always helpful to perform a high-fidelity searching to either discover new opportunities or validate preliminary designs.

We summarize three key features of the variation of all feasible transfers with flight time less than 43 days: 1) there are two families of transfers; 2) the flight time t_{FT} in different families has different distribution; and 3) the two families have a monthly variation within a year. These features should be captured by the approximate models used in designing transfers from lunar orbits to the Sun-Earth LPO region, such as the Patched ERTBP model introduced in Section III.

B. GPU search results in Patched ERTBP model

The computation is performed on two platforms. The first one is a laptop with one NVIDIA GeForce GTX850m GPU card. This GPU has 640 CUDA cores with a graphics clock of up to 902 MHz. This card is mainly designed for visualization but is also capable of GPU parallel computing. The second platform is one node of a cluster with one NVIDIA Tesla K20m GPU card. This GPU has 2496 CUDA cores with a clock of 706 MHz. The search algorithm is first developed on the laptop platform without any settings of GPU, then it is directly sent to the cluster without settings for the change of platforms. This is a particular interest of this paper to show that high performance computation on GPU carried in Matlab requires only a little programing experience. Our simulation results show that both platforms have generated the same result.

Figure 7 shows the search results in the ephemeris model (top plot) and the Patched ERTBP model (bottom plot). Each dot represents a feasible transfer departing on t_{d} with a tangential maneuver of magnitude ΔV, and the dot is colored by its corresponding flight time t_{FT} which is

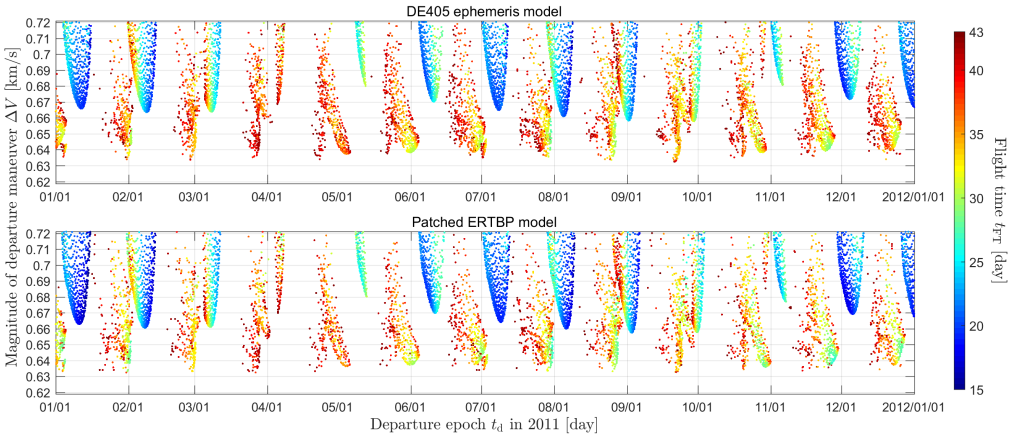

Fig. 7 Distribution of feasible transfers detected in Ephemeris model (top) and Patched ERTBP model (bottom) where $e_1 = 0.0167$ and $e_2 = 0.0554$.

less than 43 days. The feature of the two families are well captured by the Patched ERTBP model, using which the separations of Family I and II are clear and same to the result in the ephemeris model. The distributions of the color that represents t_{FT} for the ephemeris model and the Patched ERTBP are also very close to each other. The approximate model would inevitable bear model errors. Two minor differences we observed are: 1) the red part of Family I is sparser in the Patched ERTBP model than in the ephemeris model, which means transfers of Family I detected in the Patched ERTBP model has relative shorter flight time t_{FT}; and 2) at the beginning part of Family II around 02/01, the patched model shows less red dots than the ephemeris model, which means transfers in this part of Family II detected in the Patched ERTBP model have shorter flight time t_{FT}.

The resulted ΔV in the Patched ERTBP model has been uniformly displaced by +0.0125 km/s along the vertical axis when being demonstrated in Fig. 7, such that the patterns are close to those in the ephemeris model. This constant bias of ΔV in the Patched ERTBP model is calibrated through a trail-and-error process, by directly comparing the results in the ephemeris model and the Patched ERTBP model. The model error leading to this bias of ΔV can be caused by many facts, such as the ignored obliquity of the moon path, the complex lunar motion, and the parameters chosen for the two ERTBP models patched together (like semimajor axes, eccentricities, and periods). However, from a practical point of view, as long as the bias can be calibrated priorly, the search result we

obtained in the Patched ERTBP has guidance significances for trajectory design in the ephemeris model.

C. Effect of Eccentricities in Patched ERTBP model

Next, we demonstrate another benefit of using the Patched ERTBP model. We can control all settings of the model, which is not feasible when working in the ephemeris model. By simply setting the Sun-Earth or the Earth-Moon eccentricity to be zero, i.e. $e_1 = 0$ or $e_2 = 0$ in Eq. (1), we can carry out similar searches in the new Patched ERTBP models on a GPU card. These experiments could all be done very efficiently on a regular laptop. We remark that in the ephemeris model it is not possible to directly study the effect of a parameter on the result in such a way.

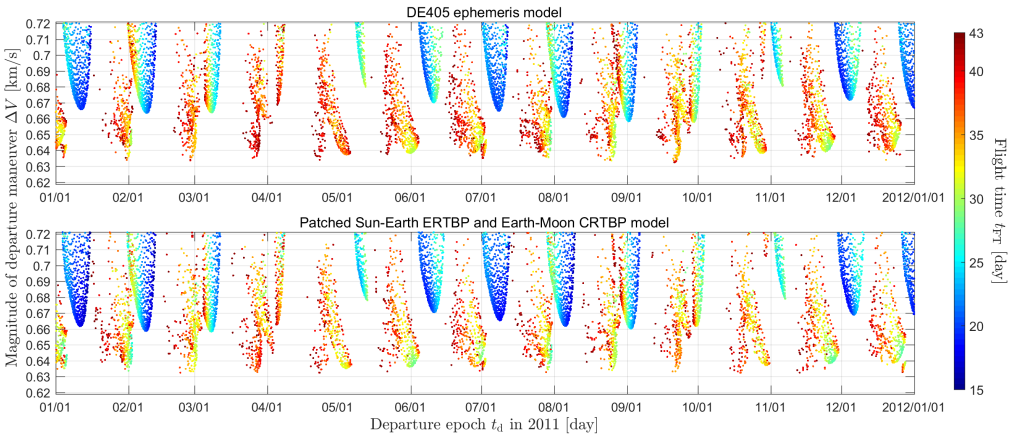

Fig. 8 Distribution of feasible transfers detected in Ephemeris model (top) and Patched Sun-Earth ERTBP and Earth-Moon CRTBP model (bottom) where $e_1 = 0.0167$ and $e_2 = 0$.

In Figs 8 and 9, the search results by setting $e_2 = 0$ and $e_1 = e_2 = 0$ in the Patched ERTBP model are demonstrated respectively. The eccentricity of the Earth-Moon ERTBP e_2 is set as zero in Fig. 8, so we are actually using a patched Sun-Earth ERTBP and Earth-Moon CRTBP model. The eccentricities of the Sun-Earth e_1 and the Earth-Moon ERTBP e_2 are both set to be zero in Fig. 9, so we are actually using a Patched CRTBP model. It can be observed that all models generate similar results as shown in Figs 7 to 9, capturing most of the features of the result in the ephemeris model. Therefore we can conclude that the eccentricity is not a factor significant enough to vary the distribution of transfers we investigate in this paper.

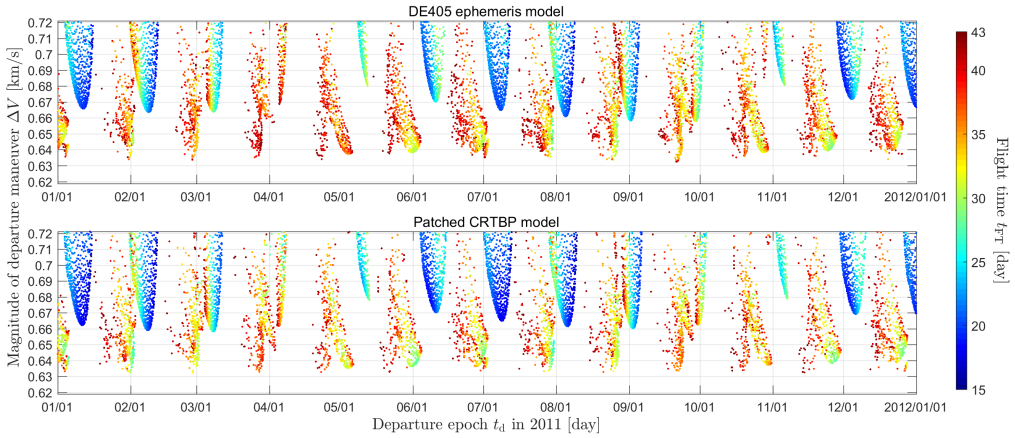

Fig. 9 Distribution of feasible transfers detected in Ephemeris model (top) and Patched CRTBP model (bottom) where $e_1 = e_2 = 0$.

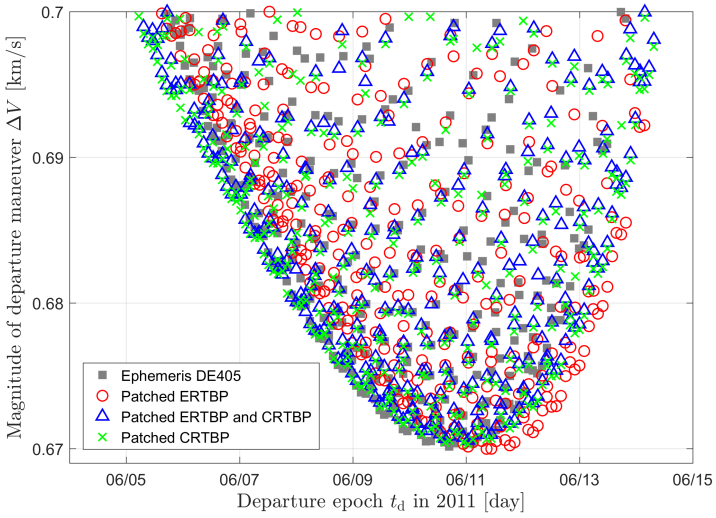

Fig. 10 Transfer opportunities in the ephemeris model and three patched models from June 5 to 15 in 2011.

A more detailed comparison is shown in Fig. 10, which is an enlarged plot concentrating on the Family II in June. The dark square represents results in the Ephemeris model, the red circle represents in the Patched ERTBP, the blue triangle represents in the Patche ERTBP and CRTBP model, and the green cross in the Patched CRTBP. On a larger scale, the results cover the same region with minor deviations. However, a comparison between the Patched ERTBP and CRTBP model (blue triangles) and the Patched CRTBP model (green crosses) reveals that they always appear as pairs with the blue triangle slightly above the green cross. Therefore, introducing the

Sun-Earth eccentricity e_1 will slightly increase the transfer cost but can be ignored in a preliminary study. A comparison between the Patched Patched ERTBP (red circles) and other two analytical models shows clear differences. It means the introducing of the Earth-Moon eccentricity will change the distribution significantly, rather than only a minor displacement.

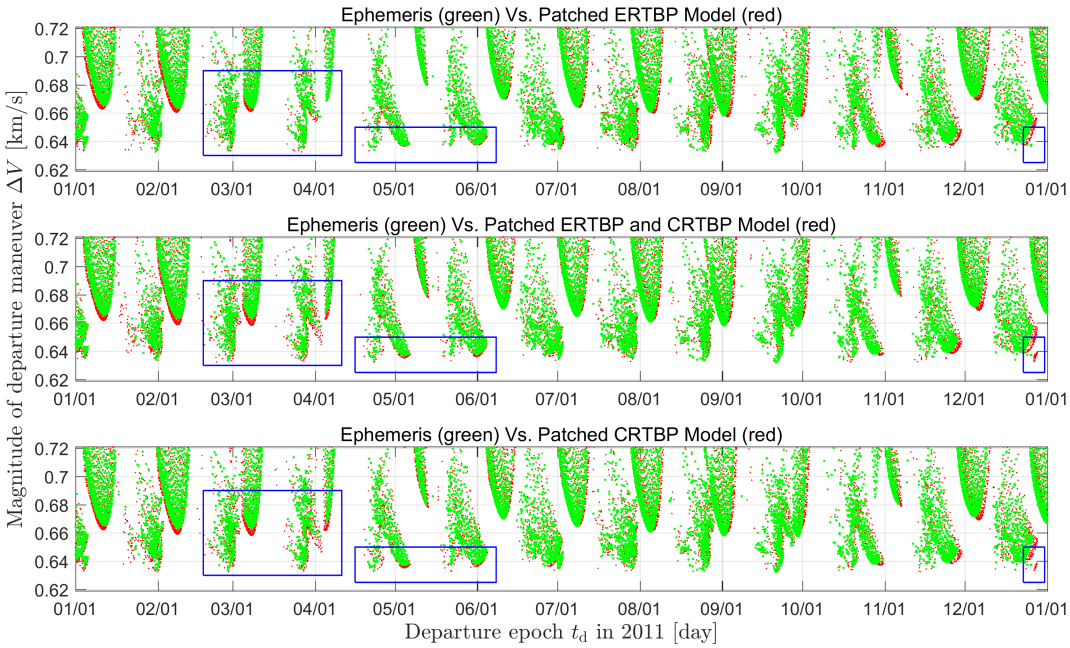

Fig. 11 Deviations of search results in three patched models with respect to that in the ephemeris model.

In Fig. 10, it seems that the Patched ERTBP model is not much improved compared with other two Patched models. However, in Fig. 11, we overlap the result in each three patched models (red dots) by the result in the ephemeris model (green dots), such that the less non-overlapping red dots observed the more accurate the model is. As shown in the figure, the Patched ERTBP model has obvious less non-overlapping regions, especially in the three boxed regions (blue) in each plot. So the Patched ERTBP model captures the distribution of the result in the ephemeris model better. For example, in the rightmost boxed regions, two patched models using CRTBP models both capture non-existing transfers in the ephemeris, which are correctly rejected by the Patched ERTBP model. There are remaining deviations between the ephemeris model and the Patched ERTBP model, which is understandable because the Patched ERTBP model is only expected to be a good approximation, by ignoring many perturbations.

As revealed above, it is beneficial to consider the eccentricities in preliminary studies. Although the theoretical studies and topics in the ERTBP model are much more complicated than that in the CRTBP model [23, 24, 31], from a point view of numerical studies, the ERTBP model introduces almost no more difficulties than the CRTBP model. Therefore, the Patched ERTBP model is a better choice.

D. Speedup Analysis

In this section, we present the performance enhancement we have achieved by using the GPU parallel computing. Different searches of transfers are carried out by using different models. Similar results have been achieved, as discussed in the previous sections. The computation time of above searches are summarized in Table 2. The second column "Platform" gives specific information about the hardware and software used in each case. The fourth column "Grid Num." gives the total number of computations performed.

Table 2 Computation time on different platforms.

Case	Platform	Model	Grid Num.	Computation Time [hrs]
1	laptop.Matlab.GPU[a]	Patched ERTBP	13,192,560	8.36
2	cluster.Matlab.GPU[b]	Patched ERTBP	13,192,560	11.65
3	cluster.Matlab.GPU[b]	Patched CRTBP	13,192,560	11.21
4	cluster.Fortran.CPUx16[c]	Ephemeris DE405	13,300,992[e]	12.5 (12.0∼13.4)[f]
5	cluster.Fortran.CPUx8[d]	Ephemeris DE405	13,300,992[e]	17.1 (16.6∼17.9)[f]

[a] With Intel Core i5-4210M CPU, and NVIDIA GTX 850M GPU 902 MHz.

[b] With Intel Xeon E5-2670 CPU, and NVIDIA Tesla K20m GPU 706 MHz.

[c] With 2 Intel Xeon E5530 CPUs. Each has 4 cores and supports 8 threads.

[d] Thread number limited to the core number 8.

[e] A little bit more grids used, in order to equally divide among threads.

[f] Average time among threads, along with the minimum and maximum in parenthesis.

For Cases 1 to 3, the GPU parallel computing is triggered through Matlab on either a laptop or a cluster, with the same Patched ERTBP model and exactly the same code. Comparing between Case 1 and Case 2 indicates that the GPU card with higher frequency have a better performance for the

search of transfers, as Case 1 requires only 71.8% of the computation time of Case 2. Comparing between Case 2 and Case 3 indicates that although the Patched ERTBP is more complex as it considers the perturbation, it leads to only 3.9% more computation time due to the efficient GPU parallel computing. Comparing between Case 1 and Case 4 shows that we can reach a speedup of approximately 1.483 (use $\frac{13192560}{13300992} \cdot \frac{12.5}{8.36}$ to eliminate the effect of excessive grids) through utilizing the Patched ERTBP model and the GPU parallel computing. Here Case 4 is carried out on the CPU with 16 threads. In the case access to a cluster is not available and the search has to be carried out on the laptop platform of Case 1, the speedup ratio can be much larger because less CPU cores will be available. Case 5 is carried out to examine the effect of the effect of the multi-thread technique of Intel CPU we used. This technique allows one physical core of a CPU to handle two threads, which acts like two physical cores to an operation system. Comparing Case 4 and Case 5 shows that this multi-thread technique can provide a speedup ratio of $17.1/12.5 \approx 1.368$, less than 2, which is the theoretical anticipation when doubling physical CPU cores. It is interesting that the best performance for our study is obtained on the laptop platform, which is the least expensive platform.

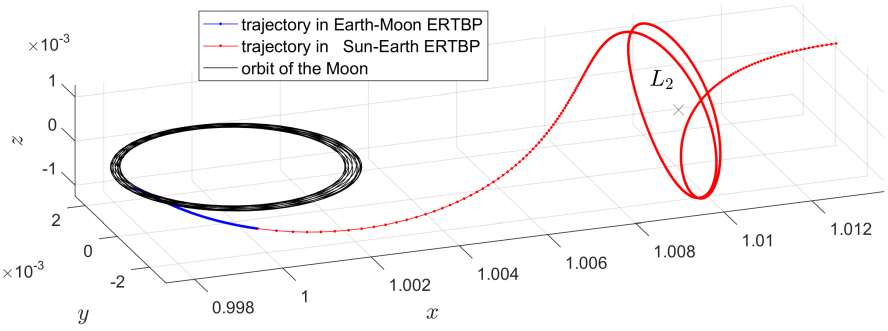

Fig. 12 Transfer example used to analysis GPU parallel computing performance.

Since the above five cases have many differences among each other, including different hardware and programing language, it is difficult to give a quantitative evaluation of the GPU parallel computing strategy we used. So next, a direct comparison of CPU and GPU parallel computing is performed on the laptop platform using Matlab, as adopted in Case 1 in Table 2. Additionally, different initial conditions will give different transfers which require different computation time, because some will be rejected at the first-step searching but some others will be refined at the second-step

bisection method. To eliminate the effect of these variable factors, we choose one feasible transfer in the Patched ERTBP model, and reproduce it multiple times on both CPU and GPU to analyze the speedup. The initial grid of the transfer chosen is $t_{\mathrm{d}} = 2455566.53472$ Julian Day, and $\Delta V = 0.690$ km/s. Starting from this initial condition, the two-step strategy generates a feasible transfer as shown in Fig. 12, where the blue curve represents the transfer in the Earth-Moon ERTBP (within the lunar 3BSOI), the red curve represents the trajectory in the Sun-Earth ERTBP, and the black curve represents the orbit of the Moon.

Table 3 computation time of reproducing the transfer example multiple times on CPU and GPU.

Number of orbit n	total time [s]		average time [s]		speedup
	t_{CPU}	t_{GPU}	t_{CPU}/n	t_{GPU}/n	$t_{\mathrm{CPU}}/t_{\mathrm{GPU}}$
1	0.26	4.40	0.257	4.401	0.06
10	3.81	4.41	0.381	0.441	0.86
20	7.46	4.42	0.373	0.221	1.69
50	15.01	4.70	0.300	0.094	3.19
100	27.68	4.79	0.277	0.048	5.78
200	52.97	6.36	0.265	0.032	8.33
500	128.1	6.33	0.256	0.013	20.24
1,000	257.1	6.33	0.257	0.006	40.62
1,200	307.1	6.34	0.256	0.005	48.48
1,500	382.0	12.66	0.255	0.008	30.17
1,800	456.9	12.63	0.254	0.007	36.18
2,000	506.7	12.68	0.253	0.006	39.96
5,000	1,255.2	25.29	0.251	0.005	49.62
10,000	2,503.1	50.53	0.250	0.005	49.54

We compute the search procedure of this example multiple times serially by CPU, and record the computation time t_{CPU}. Then we launch GPU parallel search with the same number of orbits, and record each computation time t_{GPU}. The results are summarized in Table 3. Comparing t_{CPU} and t_{GPU}, the GPU performs worse than the CPU when $n < 20$, but as n increases the GPU soon wins over the CPU. For $n = 10,000$ computations, the GPU costs only 50 seconds while the

CPU requires 2,503 seconds, which is a reduction of almost 98%. The fourth column of average computation time of CPU t_{CPU}/n are almost the same, which meets the expectation that the total time t_{CPU} should be in proportion to the orbit number n. The fifth column of t_{GPU}/n shows that the average computation time of the GPU decreases rapidly as n grows, instead of being proportion to n. This is the special property of GPU parallel computing due to its micro-architecture. The speedup given by $t_{\mathrm{CPU}}/t_{\mathrm{CPU}}$ is shown in the last column, the GPU parallel computing shows more speedup ratio as n becomes large. For the last two rows, the speedup reaches almost 50, which indicates that a cluster with 50 CPU cores could barely beat a laptop with an NVIDIA GPU card when 10,000 trajectory searches are performed. We remark that in each previous searches there are around 13 million grids being explored, so even greater speedups can be expected.

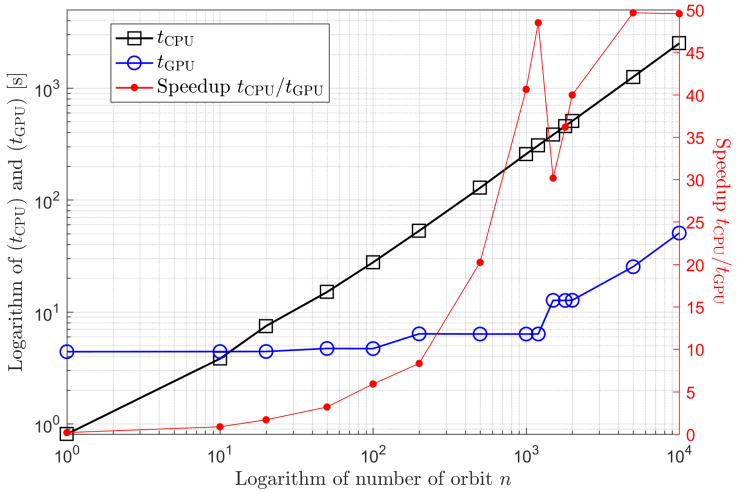

Fig. 13 Curves of computation time and speedup.

Figure 13 shows the logarithmic plot of the computation time t_{CPU} and t_{GPU} on the left axis, and the corresponding speedup $t_{\mathrm{CPU}}/t_{\mathrm{GPU}}$ on the right axis. t_{CPU} is increasing linearly as n grows, but t_{GPU} shows staged growth and increases much slower than t_{CPU}. The drop of speedup after 10^3 is caused by the staged growth of t_{GPU}. Fundamentally, the GPU launches computations concurrently, so the computation time t_{GPU} will increase discretely as n increases. For example, if a GPU can run maximumly $N_{\max}$ computations concurrently, theoretically, when the computation number is less than $N_{\max}$, the computations time will always be the same constant δt_{GPU}, and the computation time for $N_{\max}+1$ to $2\cdot N_{\max}$ computations will all be $2\cdot\delta t_{\mathrm{GPU}}$. This feature of the

GPU parallel computing is determined by its micro-architecture [18].

V. Conclusions

In this paper, efficient numerical searches using GPU technique are presented to design transfers from a lunar orbit to the Sun-Earth Libration Point Orbit (LPO). The numerical search is first conducted with the ephemeris model, which serves as a baseline. Since the ephemeris model cannot be directly implemented on the GPU, a Patched Elliptic Restricted Three-Body Problem (ERTBP) model is established, where the Sun-Earth-Moon system is first decoupled into the Sun-Earth ERTBP and the Earth-Moon ERTBP and then patched together through a lunar three-body sphere of influence (3BSOI) centered at the Moon. Using the Chinese CHANG'E-2 (CE-2) extension mission as the testbed, the search results in the Patched ERTBP model are shown to capture most of the features of the result in the ephemeris model, validating that the Patched ERTBP model approximates the ephemeris model well enough for the design of transfer trajectories in the Sun-Earth-Moon system. Moreover, we also find other alternative transfers for the CE-2 requiring smaller departure maneuvers or less flight time.

Controlled experiments can also be designed with the Patched ERTBP model. By comparing experiments ignoring the eccentricities of the Sun-Earth system or the Earth-Moon system, we discover that considering the eccentricities increases the accuracy of the preliminary trajectory design, whereas only requires only 3.9% more computation time than a Patched CRTBP model. However, the eccentricities will not change the general trend and the monthly variation of feasible transfers within a year.

Another benefit of this approximate model is that it can be conveniently implemented on a GPU card. The speedup analysis demonstrates that great speedups can be achieved from GPU parallel computing for space trajectory design. The greatest enhancement of computation in this research is achieved on an inexpensive laptop. Moreover, compared with other studies utilizing the GPU parallel computing that involve advanced programing skills with the CUDA, this research demonstrates a new approach to use the GPU parallel computing with only a little learning cost by using Matlab.

Acknowledgments

The authors acknowledge the research support from the Air Force Office of Scientific Research (AFOSR) and the Office of Naval Research (ONR). The author J.J.M. thanks MINECO-FEDER for the grant MTM2015-65715-P and the Catalan government for the grant 2014SGR504. The author G.G. thanks the Catalan government for the grant SGR115 and the Spanish one for the grant MTM2013-41168-P. Massive simulations have been run in the cluster EIXAM (at MAT-UPC) and the SOE HPC Cluster (at Rutgers University).

References

[1] Canalias, E., Gómez, G., Marcote, M., and Masdemont, J. J., "Assessment of mission design including utilization of libration points and weak stability boundaries," Tech. Rep. 0, 2004.

[2] Barden, B., Howell, K. C., and Lo, M. W., "Application of dynamical systems theory to trajectory design for a libration point mission," in "Astrodynamics Conference," American Institute of Aeronautics and Astronautics, Reston, Virigina, 1996, pp. 268–281,
doi:10.2514/6.1996-3602.

[3] Gómez, G., Jorba, A., Masdemont, J. J., and Simó, C., "Study of the transfer from the Earth to a halo orbit around the equilibrium point L1," *Celestial Mechanics & Dynamical Astronomy*, Vol. 56, No. 4, 1993, pp. 541–562,
doi:10.1007/BF00696185.

[4] Nakamiya, M., Scheeres, D. J., Yamakawa, H., and Yoshikawa, M., "Analysis of Capture Trajectories into Periodic Orbits About Libration Points," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1344–1351,
doi:10.2514/1.33796.

[5] Wang, Y., Qiao, D., and Cui, P., "Analysis of Two-Impulse Capture Trajectories into Halo Orbits of SunâĂŞMars System," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 985–990,
doi:10.2514/1.62272.

[6] Davis, K. E., Anderson, R. L., Scheeres, D. J., and Born, G. H., "The use of invariant manifolds for transfers between unstable periodic orbits of different energies," *Celestial Mechanics and Dynamical Astronomy*, Vol. 107, No. 4, 2010, pp. 471–485,
doi:10.1007/s10569-010-9285-3.

[7] Gómez, G. and Masdemont, J. J., "Some zero cost transfers between libration point orbits," in

"AAS/AIAA Astrodynamics Specialist Conference," , 2000,
doi:10.1.1.28.2144.

[8] Howell, K. C. and Hiday-Johnston, L. A., "Time-free transfers between libration-point orbits in the elliptic restricted problem," *Acta Astronautica*, Vol. 32, No. 4, 1994, pp. 245–254,
doi:10.1016/0094-5765(94)90077-9.

[9] Koon, W. S., Lo, M. W., Marsden, J. E., and Ross, S. D., "Low energy transfer to the moon," in "Dynamics of Natural and Artificial Celestial Bodies," pp. 63–73, 2001,
doi:10.1007/978-94-017-1327-6{_}8.

[10] Gómez, G., Koon, W. S., Lo, M. W., Marsden, J. E., Masdemont, J. J., and Ross, S. D., "Connecting orbits and invariant manifolds in the spatial restricted three-body problem," *Nonlinearity*, Vol. 17, No. 5, 2004, pp. 1571–1606,
doi:10.1088/0951-7715/17/5/002.

[11] Wu, W., Liu, Y., Liu, L., Zhou, J., Tang, G., and Chen, Y., "Pre-LOI trajectory maneuvers of the CHANGâĂšE-2 libration point mission," *Science China Information Sciences*, Vol. 55, No. 6, 2012, pp. 1249–1258,
doi:10.1007/s11432-012-4585-8.

[12] Qiao, D., Cui, P., Wang, Y., Huang, J., Meng, L., and Jie, D., "Design and analysis of an extended mission of CE-2: From lunar orbit to SunâĂŞEarth L2 region," *Advances in Space Research*, Vol. 54, No. 10, 2014, pp. 2087–2093,
doi:10.1016/j.asr.2013.11.032.

[13] Liu, L., Liu, Y., Cao, J., Hu, S., Tang, G., and Xie, J., "CHANG'E-2 lunar escape maneuvers to the SunâĂŞEarth L2 libration point mission," *Acta Astronautica*, Vol. 93, 2014, pp. 390–399,
doi:10.1016/j.actaastro.2013.07.032.

[14] Russell, R. P. and Arora, N., "Global point mascon models for simple, accurate and parallel geopotential computation," *Advances in the Astronautical Sciences*, Vol. 140, No. 5, 2011, pp. 813–832,
doi:10.2514/1.54533.

[15] Bai, X., *Modified Chebyshev-Picard iteration methods for solution of initial value and doundary value problems*, phd, Texas A&M University, 2010.

[16] Bai, X. and Junkins, J. L., "Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value Problems," *The Journal of the Astronautical Sciences*, Vol. 59, No. 1-2, 2012, pp. 327–351,
doi:10.1007/s40295-013-0021-6.

[17] Bai, X. and Junkins, J. L., "Modified Chebyshev-Picard Iteration Methods for Solution of Boundary

Value Problems," *The Journal of the Astronautical Sciences*, Vol. 58, No. 4, 2011, pp. 615–642, doi:10.1007/BF03321534.

[18] Wilt, N., *The cuda handbook: A comprehensive guide to gpu programming*, Pearson Education, 2013.

[19] Nakhjiri, N. and Villac, B., "Modified Picard Integrator for Spaceflight Mechanics," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 5, 2014, pp. 1625–1637, doi:10.2514/1.G000303.

[20] Wagner, S., Wie, B., and Kaplinger, B., "Computational Solutions to Lambert's Problem on Modern Graphics Processing Units," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 7, 2015, pp. 1–6, doi:10.2514/1.G000840.

[21] Arora, N., Vittaldev, V., and Russell, R. P., "Parallel Computation of Trajectories Using Graphics Processing Units and Interpolated Gravity Models," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 8, 2015, pp. 1345–1355, doi:10.2514/1.G000571.

[22] Arora, N., *High Performance Algorithms To Improve the Runtime Computation of Spacecraft Trajectories*, Doctor of philosophy, Georgia Institute of Technology, 2013.

[23] Szebehely, V. G., *Theory of Orbits - The Restricted Problem of Three Bodies*, Academic Press, New York and London, 1967.

[24] Gómez, G., Llibre, J., Martínez, R., and Simó, C., *Dynamics and Mission Design Near Libration Points - Volume I: Fundamentals: The Case of Collinear Libration Points*, Vol. 4 of *World Scientific Monograph Series in Mathematics: Volume 2*, World Scientific Publishing Company, 2001, doi:10.1142/4402.

[25] Peng, H., Wang, Y., Masdemont, J. J., and Gómez, G., "A General Analysis of Transfers from Lunar Polar Orbits to Sun-Earth Libration Point Orbits (in preparation)," .

[26] Campagnola, S., Lo, M. W., and Newton, P., "Subregions of motion and elliptic halo orbits in the elliptic restricted three-body problem," in "18th AAS/AIAA Spaceflight Mechanics Meeting," Galveston, 2008.

[27] Peng, H. and Xu, S., "Low-energy transfers to a Lunar multi-revolution elliptic halo orbit," *Astrophysics and Space Science*, Vol. 357, No. 1, 2015, pp. 86–, doi:10.1007/s10509-015-2236-4.

[28] Koon, W. S., Lo, M. W., Marsden, J. E., and Ross, S. D., "Shoot the moon," in "Spaceflight mechanics 2000," San Diego, 2000, pp. 1017–1030.

[29] Parker, J. S. and Lo, M. W., "Shoot the Moon 3D," in "AAS/AIAA Astrodynamics Specialist Confer-

ence," Lake Tahoe, California, 2005.

[30] Parker, J. S. and Anderson, R. L., *Low-Energy Lunar Trajectory Design*, No. July in JPL Deep-Space Communications and Navigation Series, Wiley, 1st ed., 2014.

[31] Peng, H. and Xu, S., "Stability of two groups of multi-revolution elliptic halo orbits in the elliptic restricted three-body problem," *Celestial Mechanics & Dynamical Astronomy*, Vol. 123, No. 3, 2015, pp. 279–303,
doi:10.1007/s10569-015-9635-2.