F. Topputo

Fast Numerical Approximation of Invariant Manifolds in the Circular Restricted Three-Body Problem

Communications in Nonlinear Science and Numerical Simulation, Vol. 32, 2016, p. 89-98
doi:10.1016/j.cnsns.2015.08.004

Fast Numerical Approximation of Invariant Manifolds in the Circular Restricted Three-Body Problem

F. Topputo

Department of Aerospace Science and Technology, Politecnico di Milano,
Via La Masa 34, 20156 Milano, Italy

Abstract

In this paper a two-step approach to approximate the invariant manifolds in the circular restricted three-body problem is presented. The method consists in a two-dimensional interpolation, followed by a nonlinear correction. A two-dimensional cubic convolution interpolation is implemented to reduce the computational effort. A nonlinear correction is applied to enforce the energy level of the approximated state. The manifolds are parameterized by using two scalars. Results show efficiency and moderate accuracy. The present method fits the needs of trajectory optimization algorithms, where a great number of manifold insertion points has to be evaluated online.

Keywords: Circular restricted three-body problem, Invariant manifolds approximation, Cubic convolution interpolation

1. Introduction

In astrodynamics, space trajectory design in n-body problems is gaining increasingly importance [1, 2, 3, 4]. This is because n-body dynamics can be exploited to achieve unique solutions where conic approximations fail. This is the case, for instance, of the periodic orbits about the Lagrange equilibrium points of the restricted three-body problem [5, 6, 7, 8, 9]. As these points are at rest with respect to a pair of primaries, these regions represent a strategic outpost for the implementation of innovative space missions [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. The focus is then on realizing solutions to transfer the spacecraft from an Earth parking orbit to a Lagrange point orbit. In the case of collinear points, this is done by placing the spacecraft on the stable manifold associated to the final orbit [20, 21, 22]. With this approach the natural free transport is exploited, and the final orbit is reached at zero cost. In this perspective it is of paramount importance to handle a global representation of the invariant manifolds in the restricted three-body problem.

Invariant manifolds can be represented with numerical or semi-analytical means. The latter method is based on some sort of expansion and manipulation of the flow around the equilibrium region [23, 24]. As semi-analytical methods entail only local representations of

Email address: francesco.topputo@polimi.it (F. Topputo)

the manifolds, numerical methods are used to design transfer trajectories, albeit increasing the computational burden. In fully numerical methods a point of the final orbit is perturbed in the local direction of the manifold and then integrated [25, 26]. To construct the whole manifold, this process is repeated for all the points in which the orbit is discretized.

In trajectory optimization, the optimal manifold insertion point is searched for. This constraint is enforced in the form of a boundary condition, and involves the online evaluation of many different points on the target manifold [27]. In the Earth–Moon system, where the invariant manifolds do not approach the Earth, a low-thrust transfer is used to target a point on the stable manifold [28, 29, 30, 31]. Due to the high number of functions evaluations of typical trajectory optimization algorithms, it is desirable to handle an efficient representation of the stable manifold. Numerical integration is therefore not suitable for this purpose, and alternative methods to synthetically represent the manifolds are needed [32]. For given optimization variables (two scalars in [28, 29, 30]), this method has to allow evaluating the corresponding point on the stable manifold, without performing numerical integrations. This must be done with emphasis on computational efficiency, at the expense of losing accuracy of the representation, which can be later recovered in successive trajectory refinements, performed in more accurate models.

Inspired by the methodology in [33], this paper presents a method to approximate the invariant manifolds obtained by numerical integration in the in the circular restricted three-body problem. In this approach, each component of the states lying on the manifold is seen as a two-dimensional surface. Once two variables are given, these components are evaluated by performing a two-dimensional interpolation. A cubic convolution method [34] is implemented to avoid computing the interpolation coefficients once a grid of sample data is given. The interpolated state is then corrected by enforcing the manifold energy level, which is known a priori. This is done by finding the zero of an algebraic nonlinear function. The sequence "interpolation and correction" has been implemented successively, and results are shown in terms of accuracy and efficiency.

The remained of the paper is organized as follows. In Section 2 the dynamics and the main properties of the circular restricted three-body problem are recalled. In Section 3 the standard numerical method used to computed the invariant manifolds is described. In Section 4 the developed approach is presented and results are shown in Section 5. Concluding remarks are given in Section 6.

2. The Circular Restricted Three-Body Problem

In the circular restricted three-body problem (CRTBP) the motion of a massless particle, P, is studied in the gravitational field generated by two primaries, P_1, P_2 of masses m_1, m_2, respectively. The primaries revolve with constant distance and angular velocity (ℓ and n, respectively) in circular orbits about their common center of mass, due to their mutual gravitational interaction. The equations that describe the motion of P are written in a barycentric, rotating coordinate frame, where P_1, P_2 are at rest on the x-axis. Let $\boldsymbol{\rho}_1 = (-\ell m_2/(m_1+m_2), 0, 0)$ and $\boldsymbol{\rho}_2 = (\ell m_1/(m_1+m_2), 0, 0)$ denote the position vectors of P_1 and P_2, respectively. The position vector of P, $\boldsymbol{r}$, is subject to [35]

$$\frac{\mathrm{d}^2\boldsymbol{r}}{\mathrm{d}t^2} + 2\boldsymbol{\omega} \times \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}) = -\,Gm_1\,\frac{\boldsymbol{r}-\boldsymbol{\rho}_1}{\|\boldsymbol{r}-\boldsymbol{\rho}_1\|} - Gm_2\,\frac{\boldsymbol{r}-\boldsymbol{\rho}_2}{\|\boldsymbol{r}-\boldsymbol{\rho}_2\|}, \tag{1}$$

where G is the universal gravitational constant and $\boldsymbol{\omega} = (0, 0, n)$. It is convenient to rewrite Eqs. (1) in scaled units where ℓ is the length unit, m_1+m_2 is the mass unit, and the time unit is such that the orbital period of P_1, P_2 equals to 2π. In this way both G and n become unities as well. The system can be described by one single parameter, the mass ratio $\mu = m_2/(m_1 + m_2)$. Thus, P_1, of normalized mass $1 - \mu$, is located at $(-\mu, 0, 0)$, whereas P_2, of normalized mass μ, is located at $(1 - \mu, 0, 0)$. The distances of P from P_1 and P_2 are $r_1 = [(x + \mu)^2 + y^2 + z^2]^{1/2}$ and $r_2 = [(x + \mu - 1)^2 + y^2 + z^2]^{1/2}$, respectively. In a first-order form, the equations of motion are

$$\begin{aligned} \dot{x} &= v_x, & \dot{v}_x &= -(1-\mu)(x+\mu)/r_1^3 - \mu(x+\mu-1)/r_2^3 + x + 2v_y, \\ \dot{y} &= v_y, & \dot{v}_y &= -(1-\mu)y/r_1^3 - \mu y/r_2^3 + y - 2v_x, \\ \dot{z} &= v_z, & \dot{v}_z &= -(1-\mu)z/r_1^3 - \mu z/r_2^3, \end{aligned} \tag{2}$$

or simply $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \mu)$, where $\boldsymbol{x} = (x, y, z, v_x, v_y, v_z)$ is the state and $\boldsymbol{f}$ is the six-dimensional vector valued function given by the right-hand sides of (2).

The CRTBP has five equilibrium points, $\boldsymbol{x}_k$, where $\boldsymbol{f}(\boldsymbol{x}_k, \mu) = 0$, $k = 1, \ldots, 5$. These are known as the Lagrange points, and are labeled L_k. The collinear points (L_1, L_2, L_3) are located on the x-axis, whereas the triangular points (L_4, L_5) lie at the vertices of two equilateral triangles with common base $\overline{P_1P_2}$. All of the equilibria exist in the $z = 0$ plane. Linearly, the collinear points behave like the product saddle × center × center. The two centers are associated to the in-plane and out-of-plane motions, and therefore there exist families of planar and vertical periodic Lyapunov orbits [36, 37] as well as Lissajous orbits [38]. When the in-plane and out-of-plane frequencies match, a special family of three-dimensional periodic halo orbits arise [39]. Due to the presence of the saddle part, there are two-dimensional stable and unstable manifolds that emanate from these periodic orbits [40, 41, 20].

The CRTBP admits an integral of motion, the Jacobi integral,

$$J(\boldsymbol{x}, \mu) = \frac{2(1-\mu)}{r_1} + \frac{2\mu}{r_2} + (x^2 + y^2) - (v_x^2 + v_y^2 + v_z^2) + \mu(1-\mu), \tag{3}$$

which, for a given energy level C, defines a five-dimensional manifold

$$\mathcal{J}(C, \mu) = \{\boldsymbol{x} \in \mathbb{R}^6 | J(\boldsymbol{x}, \mu) = C\}. \tag{4}$$

Since $\mathcal{J}(C, \mu)$ is tangent to $\boldsymbol{f}(\boldsymbol{x}, \mu)$, and therefore to the solutions of the CRTBP, the unit vector $\boldsymbol{n}(\boldsymbol{x}) = J_{\boldsymbol{x}}(\boldsymbol{x},\mu)/\|J_{\boldsymbol{x}}(\boldsymbol{x},\mu)\|$, with $J_{\boldsymbol{x}}$ being the gradient of J, i.e.,

$$\begin{aligned}
&\partial J/\partial x = -2(1-\mu)(x+\mu)/r_1^3 - 2\mu(x+\mu-1)/r_2^3 + 2x, && \partial J/\partial v_x = -2v_x,\\
&\partial J/\partial y = -2(1-\mu)y/r_1^3 - 2\mu y/r_2^3 + 2y, && \partial J/\partial v_y = -2v_y,\\
&\partial J/\partial z = -2(1-\mu)z/r_1^3 - 2\mu z/r_2^3, && \partial J/\partial v_z = -2v_z,
\end{aligned} \tag{5}$$

satisfies $\boldsymbol{f}(\boldsymbol{x},\mu)\cdot\boldsymbol{n}(\boldsymbol{x}) = 0$.

Let $\boldsymbol{\varphi}(\boldsymbol{x}_i,t)$ be the flow of (2); i.e.,

$$\boldsymbol{\varphi}(\boldsymbol{x}_i,t) = \boldsymbol{x}_i + \int_0^t \boldsymbol{f}(\boldsymbol{x}(\tau),\mu)\,\mathrm{d}\tau. \tag{6}$$

Note that the CRTBP is autonomous, and thus the initial time in (6) can be set to 0 without loss of generality; the sign of t determines then the direction of integration (forward or backward). The State Transition Matrix (STM) of (2), that is $\Phi(\boldsymbol{x},t) = \mathrm{d}\boldsymbol{\varphi}(\boldsymbol{x},t)/\mathrm{d}\boldsymbol{x}$, is subject to the variational equation

$$\dot{\Phi}(\boldsymbol{x},t) = D_{\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{\varphi}(\boldsymbol{x}(t),t),\mu)\,\Phi(\boldsymbol{x},t), \qquad \Phi(\boldsymbol{x},0) = I_{6\times6}, \tag{7}$$

where $D_{\boldsymbol{x}}\boldsymbol{f}$ is the Jacobian of the CRTBP vector field (2), namely

$$D_{\boldsymbol{x}}\boldsymbol{f} = \left[\begin{array}{ccc|ccc}
\multicolumn{3}{c|}{0_{3\times3}} & \multicolumn{3}{c}{I_{3\times3}} \\ \hline
f_{4,x} & f_{4,y} & f_{4,z} & 0 & 2 & 0 \\
f_{5,x} & f_{5,y} & f_{5,z} & -2 & 0 & 0 \\
f_{6,x} & f_{6,y} & f_{6,z} & 0 & 0 & 0
\end{array}\right], \tag{8}$$

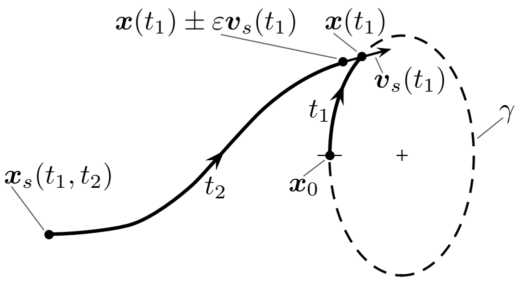
with

$$\begin{aligned}
&f_{4,x} = 1-(1-\mu)/r_1^3 - \mu/r_2^3 + 3(1-\mu)(x+\mu)^2/r_1^5 + 3\mu(x+\mu-1)^2/r_2^5,\\
&f_{5,y} = 1-(1-\mu)/r_1^3 - \mu/r_2^3 + 3(1-\mu)y^2/r_1^5 + 3\mu y^2/r_2^5,\\
&f_{6,z} = -(1-\mu)/r_1^3 - \mu/r_2^3 + 3(1-\mu)z^2/r_1^5 + 3\mu z^2/r_2^5,\\
&f_{4,y} = f_{5,x} = 3y(1-\mu)(x+\mu)/r_1^5 + 3y\mu(x+1-\mu)/r_2^5,\\
&f_{4,z} = f_{6,x} = 3z(1-\mu)(x+\mu)/r_1^5 + 3z\mu(x+1-\mu)/r_2^5,\\
&f_{5,z} = f_{6,y} = 3yz(1-\mu)/r_1^5 + 3yz\mu/r_2^5.
\end{aligned} \tag{9}$$

Eq. (7) is equivalent to 36 first-order differential equations for the elements of $\Phi(\boldsymbol{x},t)$; it requires the terms in (9) to be evaluated along the solution $\boldsymbol{\varphi}(\boldsymbol{x}(t),t)$, and therefore (2) and (7) are integrated simultaneously, thus yielding a system of 42 first-order equations.

3. Computation of Invariant Manifolds

For our analysis, it is convenient to recall the methodology used to numerically compute the invariant manifolds in the CRTBP. This approach relies on finding a linear approximation of the manifold in the neighborhood of an orbit [42, 43]. This is then globalized



$\boldsymbol{x}(t_1) \pm \varepsilon \boldsymbol{v}_s(t_1)$
$\boldsymbol{x}(t_1)$
$\boldsymbol{v}_s(t_1)$
γ
t_1
$\boldsymbol{x}_s(t_1, t_2)$
t_2
$\boldsymbol{x}_0$

Figure 1: Method used to compute the invariant manifolds.

through numerical integration. Let $\boldsymbol{\gamma}$ be a generic periodic orbit in the CRTBP,

$$\boldsymbol{\gamma} = \{\boldsymbol{\varphi}(\boldsymbol{x}_0, t) | t \in \mathbb{R}\}, \qquad \boldsymbol{\varphi}(\boldsymbol{x}_0, T_1) = \boldsymbol{x}_0, \tag{10}$$

where T_1 is the period, $\boldsymbol{x}_0$ is a known initial state, and $J(\boldsymbol{x}_0, \mu) = C$. With reference to Figure 1, any point of $\boldsymbol{\gamma}$ can be computed as $\boldsymbol{x}(t_1) = \boldsymbol{\varphi}(\boldsymbol{x}_0, t_1)$, $t_1 \in [0, T_1]$. Let also $W^s(\boldsymbol{\gamma})$, $W^u(\boldsymbol{\gamma})$ denote the two-dimensional stable, unstable manifolds of $\boldsymbol{\gamma}$. Roughly speaking, the idea is to compute the one-dimensional manifolds of $\boldsymbol{x}(t_1)$, and to obtain $W^s(\boldsymbol{\gamma})$ and $W^u(\boldsymbol{\gamma})$ by varying t_1. At $\boldsymbol{x}(t_1)$, the invariant manifolds are locally spanned by the stable and unstable eigenvectors of $M(t_1)$, the monodromy matrix associated to $\boldsymbol{x}(t_1)$. This is none other than the STM evaluated over one period of the orbit; i.e., $M(t_1) = \Phi(\boldsymbol{x}(t_1), T_1)$.

When $\boldsymbol{\gamma}$ admits stable and unstable manifolds, $M(t_1)$ has four real eigenvalues ($\lambda_s(t_1) < 1$, $\lambda_u(t_1) > 1$, $\lambda_{3,4} = 1$) and two complex-conjugate eigenvalues ($\lambda_5 = \bar{\lambda}_6$, $|\lambda_5| = 1$). The pair $\lambda_s(t_1), \lambda_u(t_1)$ defines stable and an unstable eigenvalues $\boldsymbol{v}_s(t_1), \boldsymbol{v}_u(t_1)$, respectively. The initial conditions used to globalize the manifolds are $\boldsymbol{x}(t_1) \pm \varepsilon \boldsymbol{v}_s(t_1)$, for the stable, and $\boldsymbol{x}(t_1) \pm \varepsilon \boldsymbol{v}_u(t_1)$, for the unstable manifold. The small displacement ε perturbs $\boldsymbol{x}(t_1)$ in the stable, unstable direction, whereas the ± discriminates which of the two branches of the manifold has to be generated. As for ε, it should be small enough to preserve the local validity of the linear approximation, but also large enough to prevent from long integration times needed to globalize the manifold. The value $\varepsilon = 10^{-6}$ has been used in this work consistently with the arguments in [20]. Without any loss of generality, the computation of the stable manifold is treated in the following. Applying the same concept to the unstable manifolds is straightforward.

Let $\boldsymbol{x}_s \in W^s(\boldsymbol{\gamma})$ be a generic point on the stable manifold. This is computed through

$$\boldsymbol{x}_s(t_1, t_2) = \boldsymbol{\varphi}(\boldsymbol{x}(t_1) \pm \varepsilon\, \boldsymbol{v}_s(t_1), -t_2), \tag{11}$$

where t_2 is the integration time, $t_2 > 0$, $t_2 \in [0, T_2]$; T_2 is an upper bound depending to what extent $W^s(\boldsymbol{\gamma})$ has to be globalized (see Figure 1). In summary, computing $\boldsymbol{x}_s(t_1, t_2)$ requires: 1) integrating (2) within $[0, t_1]$ to get $\boldsymbol{x}(t_1)$; 2) integrating the coupled system (2) and (7) from $\boldsymbol{x}(t_1)$ for one orbit revolution to get $M(t_1)$; 3) computing the stable eigenvector $\boldsymbol{v}_s(t_1)$; 4) performing the backward integration (11) to get $\boldsymbol{x}_s(t_1, t_2)$. Varying t_1, t_2 yields $W^s(\boldsymbol{\gamma})$.

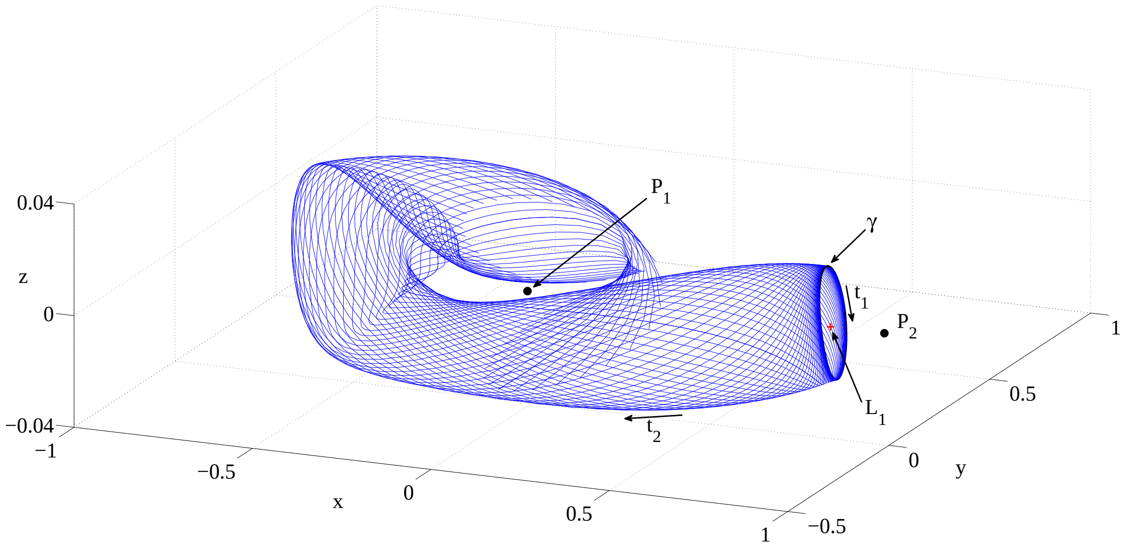


Figure 2: Stable manifold of a periodic halo orbit in the Earth-Moon model projected in the physical space.

To make this procedure more efficient, the monodromy matrix associated to $\boldsymbol{x}_0$, $M_0 = \Phi(\boldsymbol{x}_0, T_1)$, is computed once. Its stable eigenvector, $\boldsymbol{v}_{s,0}$, is pushed forward to get $\boldsymbol{v}_s(t_1)$ through

$$\boldsymbol{v}_s(t_1) = \Phi(\boldsymbol{x}_0, t_1)\boldsymbol{v}_{s,0}, \tag{12}$$

where $\Phi(\boldsymbol{x}_0, t_1)$ is achieved through integration of (2) and (7). This avoids performing step 2) above, which is the one requiring the most effort, at the additional cost of integrating the variational equation in step 1). Moreover, M_0 can be computed through $\Phi(\boldsymbol{x}_0, T_1/2)$ by exploiting the symmetry of the RTBP [44]. This limits the accumulation of numerical integration errors, as the variational equation is integrated for half of the orbital period. Figure 2 shows an example of stable manifold computed with the method described above.

4. Approximation of Invariant Manifolds

Let $\boldsymbol{x}_s$ in (11) be $\boldsymbol{x}_s = (x_s, y_s, z_s, v_{x_s}, v_{y_s}, v_{z_s})$. By virtue of (11), the components of $\boldsymbol{x}_s$ depend on a parameter *along the orbit*, t_1, and a parameter *along the flow*, t_2 [20]. (Other parameterizations can be used to express $\boldsymbol{x}_s$; e.g., ε in (11) can be let to vary, so replacing t_1.) It is convenient to think at each component of $\boldsymbol{x}_s$ as of a two-dimensional surface over the (t_1, t_2) plane. This can be viewed in Figure 3, which reports the components of the stable manifold in Figure 2.

As the goal of this work is to implement an approximation that allows a fast evaluation of $\boldsymbol{x}_s(t_1, t_2)$, while still ensuring reasonable accuracy, a numerical interpolation can be used. In particular, the algorithms developed for the two-dimensional interpolation of surfaces can be applied [45]. This approach requires a grid of samples over which the interpolation is constructed. The idea is therefore to discretize (t_1, t_2) and to use the algorithm in Section 3 to compute such grid. This is done once. Then, the two-dimensional interpolation is carried out online, where a fast evaluation is needed. The algorithm applied for the

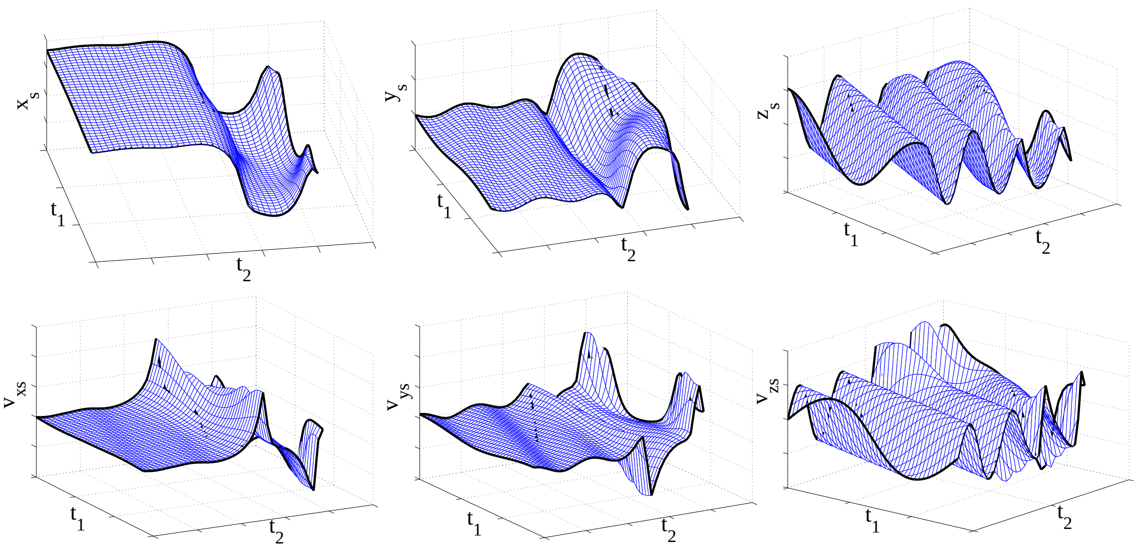


Figure 3: Components of the stable manifold in Figure 2.

two-dimensional interpolation is reported in Section 4.1. As this leads to low-accuracy approximations, a nonlinear correction is needed. The scheme developed to perform such correction is presented in Section 4.2.

4.1. Two-Dimensional Cubic Convolution Interpolation

Let N_1, N_2 be the number of points in which the intervals $[0,\, T_1]$, $[0,\, T_2]$ are discretized, respectively. The two-dimensional time grid

$$t_{1,i} = (i-1)\frac{T_1}{N_1 - 1}, \quad t_{2,j} = (j-1)\frac{T_2}{N_2 - 1}, \qquad i = 1, \ldots, N_1, \quad j = 1, \ldots, N_2, \tag{13}$$

is used to compute a database of sampled data through (11); i.e.,

$$\boldsymbol{x}_{s,ij} = \boldsymbol{\varphi}(\boldsymbol{x}(t_{1,i}) \pm \varepsilon\, \boldsymbol{v}_s(t_{1,i}), -t_{2,j}), \qquad i = 1, \ldots, N_1, \quad j = 1, \ldots, N_2. \tag{14}$$

Without any loss of generality, the approximation algorithm is developed for the component $x_s(t_1, t_2)$ only; the whole state $\boldsymbol{x}_s(t_1, t_2)$ can be interpolated by applying the same algorithm to the other components. Given $x_{s,ij}$ from (14), the focus is to compute $x_s(t_1, t_2)$ for any other pair t_1, t_2 *without* performing numerical integration. For every combination of t_1, t_2 in the rectangular subdivision $[t_{1,i},\, t_{1,i+1}] \times [t_{2,j},\, t_{2,j+1}]$, $i = 1, \ldots, N_1 - 1$, $j = 1, \ldots, N_2 - 1$, a cubic convolution interpolation is implemented through

$$x_s^{\mathrm{ipl}}(t_1, t_2) = \sum_{\ell=-1}^{2} \sum_{m=-1}^{2} c_{i+\ell, j+m}\; u\left(\frac{t_1 - t_{1,i+\ell}}{h_1}\right) u\left(\frac{t_2 - t_{2,j+m}}{h_2}\right), \tag{15}$$

where $c_{i+\ell,j+m}$ are the interpolation coefficients, $h_1 = T_1/(N_1-1)$ and $h_2 = T_2/(N_2-1)$ are the sampling increments, and the function u is the interpolation kernel [34]; the superscript 'ipl' is used to distinguish the interpolated (x_s^{ipl}) from the integrated (x_s) component.

The interpolation kernel in (15) converts discrete data into continuous function. This is done by using piecewise cubic polynomials defined on the unit subintervals between -2 and $+2$. The kernel is symmetric, continuous, and has a continuous first derivative. These conditions can be applied to yield (see [34] for the complete derivation)

$$u(s) = \begin{cases} 3/2|s|^3 - 5/2|s|^2 + 1 & 0 < |s| < 1 \\ -1/2|s|^3 + 5/2|s|^2 - 4|s| + 2 & 1 < |s| < 2 \\ 0 & 2 < |s|. \end{cases} \tag{16}$$

The kernel (16) assumes the values $u(0) = 1$ and $u(n) = 0$ for any nonzero integer n. Thus, for the interpolation (15) to be exact at the interpolation nodes, it is required that $c_{ij} = x_{s,ij}$. This condition has an important computational significance: as the interior interpolation coefficients c_{ij} are simply the sampled data points $x_{s,ij}$, they are not to be computed. This is a great departure from two-dimensional cubic spline, where the interpolation coefficients are computed by solving a block tridiagonal matrix problem. Only the exterior coefficients are computed by imposing the boundary conditions. This feature reduces the computational time required to perform the interpolation, which is desirable when a fast approximation is needed. After some manipulations, the exterior coefficients read [34]

$$\begin{aligned} & c_{0,0} = 3c_{1,0} - 3c_{2,0} + c_{3,0}, \quad c_{N_1+1,N_2+1} = 3c_{N_1,N_2+1} - 3c_{N_1-1,N_2+1} + c_{N_1-2,N_2+1}, \\ & c_{N_1+1,0} = 3c_{N_1,0} - 3c_{N_1-1,0} + c_{N_1-2,0}, \quad c_{0,N_2+1} = 3c_{1,N_2+1} - 3c_{2,N_2+1} + c_{3,N_2+1} \\ & \begin{cases} c_{i,0} = 3x_{s,i1} - 3x_{s,i2} + x_{s,i3} \\ c_{i,N_2+1} = 3x_{s,iN_2} - 3x_{s,i(N_2-1)} + x_{s,i(N_2-2)} \end{cases} \qquad i = 1, \ldots, N_1 \\ & \begin{cases} c_{0,j} = 3x_{s,1j} - 3x_{s,2j} + x_{s,3j} \\ c_{N_1+1,j} = 3x_{s,N_1j} - 3x_{s,(N_1-1)j} + x_{s,(N_1-2)j} \end{cases} \qquad j = 1, \ldots, N_2. \end{aligned} \tag{17}$$

4.2. Nonlinear Correction

Once the third-order interpolation (15) is applied to all the components, the fully interpolated state $\boldsymbol{x}_s^{\mathrm{ipl}}(t_1, t_2)$ is achieved. The accuracy with which $\boldsymbol{x}_s^{\mathrm{ipl}}(t_1, t_2)$ represents $\boldsymbol{x}_s(t_1, t_2)$ depends on the size of the two-dimensional time grid as well as on the shape of the functions to approximate. As the latter feature cannot be controlled (it is an inherent property of the problem), the idea is using the least number of grid points to shorten the computation of $\boldsymbol{x}_{s,ij}$ in (14). Since this leads to low-accurate interpolations, a correction step is implemented.

Let $C = J(\boldsymbol{x}_s(t_1, t_2), \mu)$ be the manifold energy level, and let $C^{\mathrm{ipl}} = J(\boldsymbol{x}_s^{\mathrm{ipl}}(t_1, t_2), \mu)$ be the interpolated state energy level; let also their difference be $\Delta C = C - C^{\mathrm{ipl}}$. It is likely that $\Delta C \neq 0$. Thus, a new state approximation is defined $\boldsymbol{x}_s^{\mathrm{app}} = \boldsymbol{x}_s^{\mathrm{ipl}} + \Delta\boldsymbol{x}$ such that $J(\boldsymbol{x}_s^{\mathrm{app}}, \mu) = C$. In principle, $\Delta\boldsymbol{x}$ may affect all the components of $\boldsymbol{x}_s^{\mathrm{ipl}}$. To ease this correction and speed up the computation, it is assumed that $\Delta\boldsymbol{x} = \delta\, \boldsymbol{n}(\boldsymbol{x}^{\mathrm{ipl}})$, where $\boldsymbol{n}(\boldsymbol{x}^{\mathrm{ipl}})$ is the local normal to the flow (see Section 2) and δ is an unknown scalar. The problem consists then in finding $\delta \in \mathbb{R}$ such that

$$J(\boldsymbol{x}_s^{\text{ipl}} + \delta\, \boldsymbol{n}(\boldsymbol{x}^{\text{ipl}}), \mu) - C = 0, \tag{18}$$

which is a scalar equation in a scalar unknown. Equation (18) can be solved efficiently by using a Newton method. At the k-th iteration, the solutions is

$$\delta^{(k)} = \delta^{(k-1)} - \frac{J(\boldsymbol{x}_s^{\text{ipl}} + \delta^{(k-1)}\, \boldsymbol{n}(\boldsymbol{x}^{\text{ipl}}), \mu) - C}{\boldsymbol{n}(\boldsymbol{x}^{\text{ipl}}) \cdot \boldsymbol{n}(\boldsymbol{x}^{\text{ipl}})}. \tag{19}$$

Numerical experiments show that the convergence is attained in 4–5 iterations with $\delta^{(0)} = 0$ and termination tolerance $|\delta^{(k)} - \delta^{(k-1)}| < 10^{-14}$.

5. Results

5.1. Assessment of the Method

The developed method is based on the successive implementation of cubic convolution interpolation and nonlinear correction. A way to check the usefulness of this method is to compare it in terms of accuracy and efficiency with the standard method requiring numerical integration (Section 3). The accuracy is assessed by evaluating the error, the efficiency by comparing the computational time.

In order to perform a systematic assessment, a new grid has to be defined. By definition, the interpolation is exact on the *approximation grid* $(t_{1,i}, t_{2,j})$, $i = 1, \ldots, N_1$, $j = 1, \ldots, N_2$, and therefore evaluating the method on this grid is not useful. The approximation is instead tested on the *evaluation grid* $(\tau_{1,k_1}, \tau_{2,k_2})$, with $\tau_{1,k_1} = (t_{1,k_1+1} - t_{1,k_1})/2$, $\tau_{2,k_2} = (t_{2,k_2+1} - t_{2,k_2})/2$, and $k_1 = 1, \ldots, N_1 - 1$, $k_2 = 1, \ldots, N_2 - 1$. In practice, each point of the evaluation grid lies in the center of the rectangle $[t_{1,i},\, t_{1,i+1}] \times [t_{2,j},\, t_{2,j+1}]$, where the interpolation error is expected to be the highest (see Figure 4).

As the aim of the work is to approximate the manifold achieved by numerical integration, the error of the approximation is defined on the evaluation grid as

$$\epsilon_{k_1 k_2} = \|\boldsymbol{x}_s(\tau_{1,k_1}, \tau_{2,k_2}) - \boldsymbol{x}_s^{\text{app}}(\tau_{1,k_1}, \tau_{2,k_2})\| \tag{20}$$

where $\boldsymbol{x}_s$ is computed via numerical integration with the procedure in Section 3, whereas $\boldsymbol{x}_s^{\text{app}}$ is computed with interpolation and correction as illustrated in Section 4. The efficiency is measured through the speed up; i.e., the ratio $\text{CPUt}(\boldsymbol{x}_{\text{s}}^{\text{app}})/\text{CPUt}(\boldsymbol{x}_{\text{s}})$, where $\text{CPUt}(\boldsymbol{x}_{\text{s}}^{\text{app}})$ and $\text{CPUt}(\boldsymbol{x}_{\text{s}})$ are the CPU times needed to compute $\boldsymbol{x}_s^{\text{app}}$ and $\boldsymbol{x}_s$ over the evaluation grid, whose pseudocodes are given in Algorithm 1 and 2, respectively. In the remainder, the CPU time is relative to a sequential implementation of the algorithms in Matlab R2014b (64 bit) and simulations on an platform with 3.2 GHz Intel Core i5 CPU, 8 GB 1600 MHz DDR3 RAM, running Mac OS X 10.10.3. Integrations are performed with a variable-order, multi-step Adams–Bashforth–Moulton scheme with absolute and relative tolerances set to 10^{-14}.

Algorithm 1 Standard computation of invariant manifolds through numerical integration.

```
solve $\left\{\begin{array}{ll} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \mu), & \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \dot{\Phi} = [\partial \boldsymbol{f}/\partial \boldsymbol{x}]\Phi, & \Phi(0) = I \end{array}\right.$ in $[0,\ T_1]$        ▷ Integrate 42 eqs. once
initialize $M_0 = \Phi(\boldsymbol{x}_0, T_1)$
compute $\boldsymbol{v}_{s,0}$ from $M_0$

for $k_1 = 1 \to N_1 - 1$ do
    $\tau_{1,k_1} = (t_{1,k_1+1} - t_{1,k_1})/2$
    for $k_2 = 1 \to N_2 - 1$ do
        $\tau_{2,k_2} = (t_{2,k_2+1} - t_{2,k_2})/2$
        solve $\left\{\begin{array}{ll} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \mu), & \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \dot{\Phi} = [\partial \boldsymbol{f}/\partial \boldsymbol{x}]\Phi, & \Phi(0) = I \end{array}\right.$ in $[0,\ \tau_{1,k_1}]$        ▷ 42 eqs. × $N_1 - 1$
        compute $\boldsymbol{v}_s(\tau_{1,k_1}) = \Phi(\boldsymbol{x}_0, \tau_{1,k_1})\boldsymbol{v}_{s,0}$
        compute $\boldsymbol{x}_s(\tau_{1,k_1}, \tau_{2,k_2}) = \boldsymbol{\varphi}(\boldsymbol{x}(\tau_{1,k_1}) \pm \varepsilon\, \boldsymbol{v}_s(\tau_{1,k_1}), -\tau_{2,k_2})$        ▷ 6 eqs. × $N_2 - 1$
    end for
end for
```

Algorithm 2 Algorithm for approximation of invariant manifolds.

```
solve $\left\{\begin{array}{ll} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \mu), & \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \dot{\Phi} = [\partial \boldsymbol{f}/\partial \boldsymbol{x}]\Phi, & \Phi(0) = I \end{array}\right.$ in $[0,\ T_1]$        ▷ Integrate 42 eqs. once
store $\boldsymbol{x}_i = \boldsymbol{x}(t_i)$, $\Phi_i = \Phi(\boldsymbol{x}_0, t_i)$, $i = 1, \ldots, N_1$
initialize $M_0 = \Phi(\boldsymbol{x}_0, T_1)$
compute $\boldsymbol{v}_{s,0}$ from $M_0$

for $i = 1 \to N_1$ do
    compute $\boldsymbol{v}_{s,i} = \Phi_i \boldsymbol{v}_{s,0}$
    compute $\boldsymbol{x}_s(t_{1,i}, t_{2,j}) = \boldsymbol{\varphi}(\boldsymbol{x}_i \pm \varepsilon \boldsymbol{v}_{s,i}), -t_{2,j})$, $j = 1, \ldots, N_2$        ▷ 6 eqs. × $N_1$
    store $\boldsymbol{x}_{s,ij} = \boldsymbol{x}_s(t_{1,i}, t_{2,j})$, $j = 1, \ldots, N_2$
end for

for $k_1 = 1 \to N_1 - 1$ do
    $\tau_{1,k_1} = (t_{1,k_1+1} - t_{1,k_1})/2$
    for $k_2 = 1 \to N_2 - 1$ do
        $\tau_{2,k_2} = (t_{2,k_2+1} - t_{2,k_2})/2$
        compute $\boldsymbol{x}_s^{\text{ipl}}(\tau_{k_1}, \tau_{k_2})$ from (15)        ▷ Perform interpolation
        computre $\boldsymbol{n}(\boldsymbol{x}_s^{\text{ipl}}(\tau_{k_1}, \tau_{k_2}))$ from (5)
        compute $\delta$ from (19)        ▷ Solve nonlinear eq.
        define $\boldsymbol{x}_s^{\text{app}}(\tau_{k_1}, \tau_{k_2}) = \boldsymbol{x}_s^{\text{ipl}}(\tau_{k_1}, \tau_{k_2}) + \delta\, \boldsymbol{n}(\boldsymbol{x}_s^{\text{ipl}}(\tau_{k_1}, \tau_{k_2}))$
    end for
end for
```

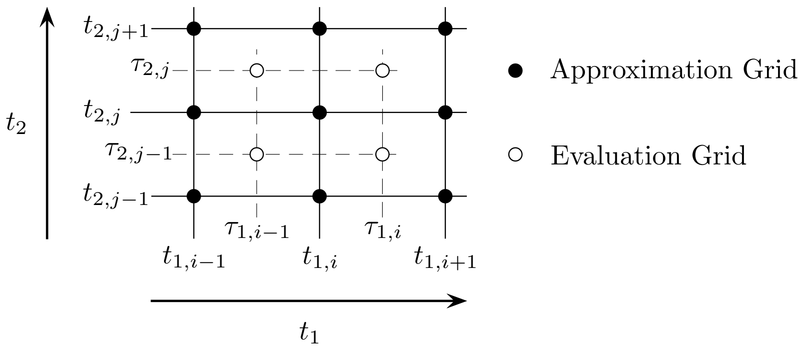


Figure 4: Approximation and evaluation grids.

5.2. Test Cases

The developed algorithm is used to approximate the stable manifold in Figures 2. The problem parameters are $\mu = 0.012150$, $C = 3.182454$, $T_1 = 2.746083$, $T_2 = 12.566370$. A first approximation grid is constructed with $N_1 = 100$ and $N_2 = 200$, which yield step sizes of 2.77×10^{-2} and 6.31×10^{-2} (see Eq. (15)). A random point is sampled with $t_1 = 1$ and $t_2 = 5$, and the corresponding states $\boldsymbol{x}_s$, $\boldsymbol{x}_s^{\text{ipl}}$, and $\boldsymbol{x}_s^{\text{app}}$ are computed. In Table 1 these states are reported along with their Jacobi energies, C, and errors, ε. It can be seen that performing interpolation only produces $\Delta C = -3.10605315 \times 10^{-5}$ and $\epsilon = 2.51694055 \times 10^{-5}$, while the sequence interpolation and nonlinear correction yields $\Delta C = -8.88178419 \times 10^{-16}$ and $\epsilon = 2.38885372 \times 10^{-5}$. Although the difference in Jacobi energy decreases dramatically when performing the nonlinear correction, the accuracy in the estimation of the state improves moderately enough to deem valuable the sequential procedure developed.

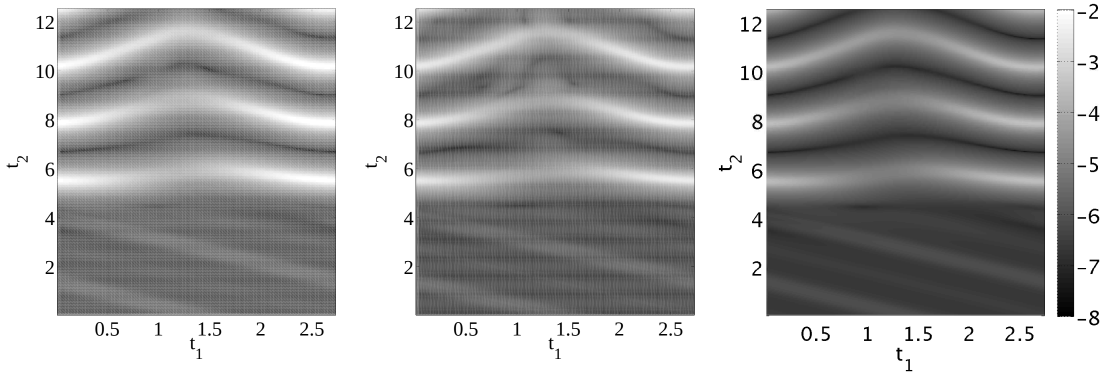
Table 1: Sample state of the stable manifold in Figure 2 with $t_1 = 1$, $t_2 = 5$.

	$\boldsymbol{x}_s$	$\boldsymbol{x}_s^{\text{ipl}}$	$\boldsymbol{x}_s^{\text{app}}$
x	0.583606315548440	0.583599597171183	0.583606656441017
y	-0.196069410503332	-0.196067727193217	-0.196070212242085
z	0.018609750034304	0.018609759931961	0.018610071098876
v_x	0.483332979420175	0.483345360093013	0.483347315799745
v_y	0.420658175717234	0.420637397923229	0.420639099901687
v_z	0.027414285066469	0.027413556391291	0.027413667311724
C	3.182454737262995	3.182485797794570	3.182454737262996
ϵ	—	$2.51694055 \times 10^{-5}$	$2.38885372 \times 10^{-5}$

With the same settings, three cases with different grid points (and therefore different step sizes) have been systematically analyzed over their approximation grids: case a) with $N_1 = 100$, $N_2 = 200$; case b) with $N_1 = 100$, $N_2 = 300$; case c) with $N_1 = 200$, $N_2 = 300$. Table 2 reports the outcome of this assessment in terms of maximum, mean, and minimum values of the error and speed up (SU); i.e., the ratio of the computational time needed to execute Algorithm 2 over that needed to carry out Algorithm 1. The error pattern

Table 2: Summary of the test cases.

	N_1	N_2	h_1	h_2	$\max \epsilon$	$\text{mean}\, \epsilon$	$\min \epsilon$	SU
a)	100	200	2.77×10^{-2}	6.31×10^{-2}	1.47×10^{-2}	3.10×10^{-4}	9.13×10^{-8}	52.9
b)	100	300	2.77×10^{-2}	4.20×10^{-2}	4.82×10^{-3}	7.27×10^{-5}	5.53×10^{-8}	61.6
c)	200	300	1.38×10^{-2}	4.20×10^{-2}	4.60×10^{-3}	6.43×10^{-5}	8.69×10^{-9}	61.3



(a) Case a)

(b) Case b)

(c) Case c)

Figure 5: $\log_{10} \epsilon$ over the evaluation grid for the three cases considered.

($\log_{10} \epsilon$) for the three cases is shown in Figure 5 with greyscale code (see the bar on the right of Figure 5(c)). The error profile over the evaluation grid for case b) is shown in Figure 6.

From inspection of Figure 5, it can be seen that the error is generally acceptable, except for three distinct regions (white wave-shaped strips in Figure 5) showing moderate errors. These areas correspond to the regions where the stable manifold experiences close Earth passage. This difference in approximation accuracy is inherent in a strategy implementing uniform time grids, as the step size does not adapt to capture rapid state variations. However, the error decreases when the number of grid points increases, or equivalently when the sampling increments shrink. On the other hand, the considerable speed up (see Table 2) demonstrates the numerical efficiency of the approximation. Overall, the developed methodology well fits trajectory optimization frameworks, where a fast evaluation of the invariant manifolds, with moderate accuracy, is desirable.

5.3. Runtime Analysis

By inspection of Algorithms 1 and 2, we can infer the rate of growth of the computational time for increasing N_1, N_2. It should be said that in practical applications only some portions of Algorithms 1 and 2 are executed repeatedly (lines 8–10 of Algorithm 1 and lines 14–17 of Algorithm 2), while others are evaluated just once (the computation of the monodromy matrix in lines 1–3 of both algorithms). The construction of the database of points required by Algorithm 2 is accounted for in the analysis below. Thus, the runtime analysis involves lines 4–12 of Algorithm 1 and lines 5–19 of Algorithm 2.

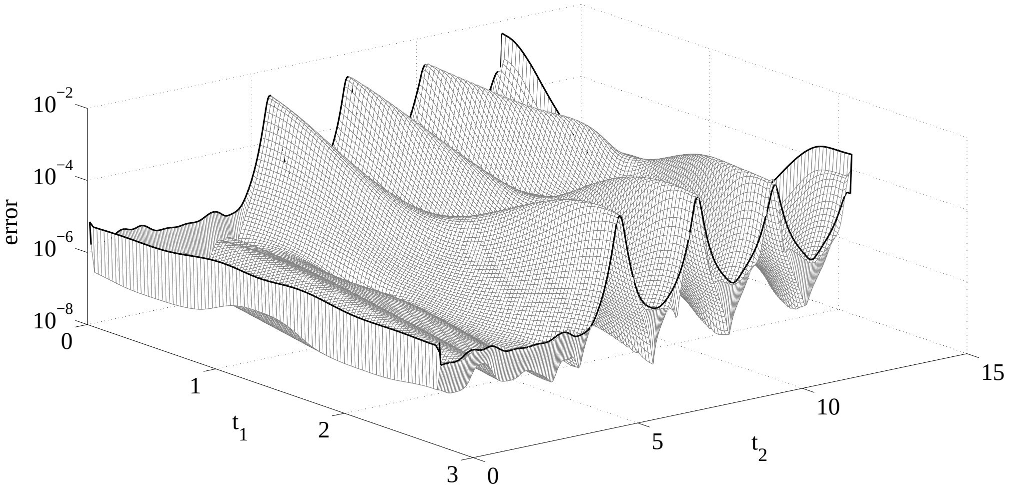


Figure 6: Approximation error for case b).

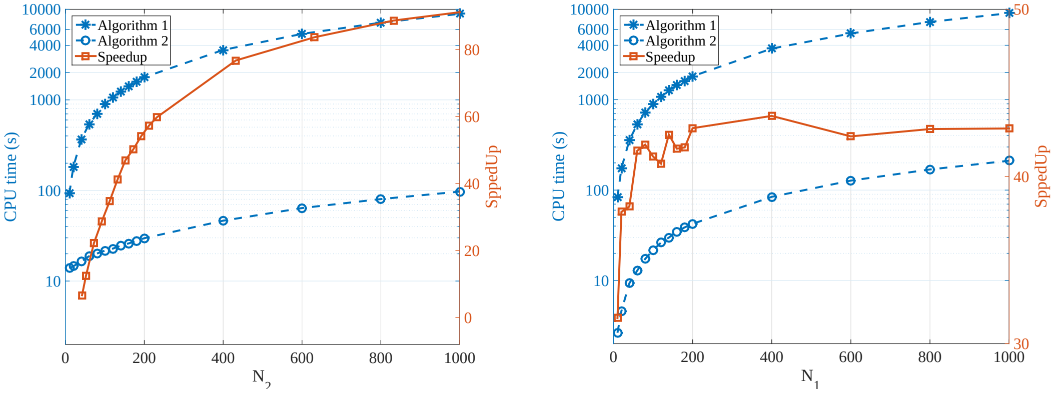
In Algorithm 1, the most expensive tasks are those in lines 8 and 10. Integrating the variational equation (line 8) $N_1 - 1$ times with variable final time $iT_1/(N_1-1)$, $i = 1, \dots, N_1 - 1$, is equivalent to integrate it once with final time $\sum_{i=1}^{N_1-1} iT_1/(N_1-1) = N_1T_1/2$. Since this is repeated $N_2 - 1$ times, the complexity is $k_1(N_1N_2 - N_1)$ (all factors k_i from now on are constant). Analogously, integrating the dynamics in line 10 $N_2 - 1$ times with variable final time $jT_2/(N_2-1)$, $j = 1, \dots, N_2 - 1$, and repeating it $N_1 - 1$ times involves an effort of $k_2(N_1N_2 - N_1)$. The rate of growth of Algorithm 1 is $\mathcal{O}(N_1N_2)$.

In Algorithm 2, constructing the database of points (lines 5–9) requires integrating the dynamics in line 7 N_1 times for fixed final time T_2 (the value of the state at $t_{2,j}$ is retrieved by interpolation). Thus, this part has complexity k_3N_1. The interpolation and nonlinear correction in lines 14–17 is performed $(N_1 - 1) \times (N_2 - 1)$ times, and thus its effort is $k_4N_1N_2 - k_4(N_1 + N_2)$. The rate of growth of Algorithm 2 is $\mathcal{O}(N_1N_2)$ as well.

Since both algorithms behave as $\mathcal{O}(N_1N_2)$, their performances depend on the coefficients k_i, the speedup being

$$\mathrm{SU} = \frac{(k_1 + k_2)N_1N_2 - k_1N_1 - k_2N_2}{k_4N_1N_2 + (k_3 - k_4)N_1 - k_4N_2}. \tag{21}$$

An assessment of the CPU time has been performed by varying N_1 and N_2 one at a time, while keeping the same settings of the previous section. The outcome is reported in Figures 7, where both the CPU time and the speedup are reported for the two cases. It can be seen that: 1) the CPU time increases linearly in both algorithms, so confirming that they are $\mathcal{O}(N_1N_2)$; 2) although the effort of both algorithms grows at the same rate, Algorithm 2 takes always less CPU time than Algorithm 1; 3) for fixed N_1, SU in (21) is $(\alpha_1N_2 + \beta_1)/(\gamma_1N_2 + \delta_1)$, and therefore it approaches $\alpha_1/\gamma_1 \simeq 90$ for increasing N_2; 4) for fixed N_2, the SU in (21) is $(\alpha_2N_1 + \beta_2)/(\gamma_2N_1 + \delta_2)$ and tends to $\alpha_2/\gamma_2 \simeq 43$.



(a) Fixed N_1 ($N_1 = 100$)

(b) Fixed N_2 ($N_2 = 100$)

Figure 7: Runtime analysis for fixed N_1 and N_2 (speedup reported on the right y-axis).

6. Conclusions

In this paper a method to approximate the invariant manifolds in the restricted three-body problem has been developed. The method is based on a cubic convolution interpolation followed by a nonlinear correction, which can be implemented once a data base of sample values is given. The method is particularly suitable in those frameworks in which a fast evaluation of the manifold states is wanted for any combination of the design variables (two scalars in the present implementation). The method is suitable in approximating well-behaved manifolds, for which reasonable accuracy and computational efficiency have been proven.

Acknowledgments

The author would like to express his gratitude to Alexander Wittig for the useful suggestions, to Renyong Zhang for supporting the development of the code, and to Marco Morandini for helping with the runtime analysis.

References

[1] E. M. Alessi, G. Gómez, and J. Masdemont. Leaving the Moon by means of Invariant Manifolds of Libration Point Orbits. *Communications in Nonlinear Science and Numerical Simulation*, 14(12):4153–4167, 2009.

[2] Y. Ren, J. Masdemont, G. Gómez, and E. Fantino. Two Mechanisms of Natural Transport in the Solar System. *Communications in Nonlinear Science and Numerical Simulation*, 17(2):844–853, 2012.

[3] G. Mingotti, F. Topputo, and F. Bernelli-Zazzera. Efficient Invariant-Manifold, Low-Thrust Planar Trajectories to the Moon. *Communication in Nonlinear Science and Numerical Simulation*, 17(2):817–831, 2012.

[4] Y. Ren and J Shan. Low-Energy Lunar Transfers using Spatial Transit Orbits. *Communications in Nonlinear Science and Numerical Simulation*, 19(3):554–569, 2014.

[5] D. Dunham and R. Farquhar. Libration Point Missions, 1978–2002. In G. Gómez, M. Lo, and J. Masdemont, editors, *Proceedings of the Conference on Libration Point Orbits and Applications*, pages 45–73. World Scientific, 2003.

[6] G. Gómez, J. Masdemont, and J. Mondelo. Libration Point Orbits: A Survey from the Dynamical Point of View. In G. Gómez, M. Lo, and J. Masdemont, editors, *Proceedings of the Conference on Libration Point Orbits and Applications*, pages 311–372. World Scientific, 2003.

[7] D. Folta and M. Beckman. Libration Orbit Mission Design: Applications of Numerical and Dynamical Methods. In G. Gómez, M. Lo, and J. Masdemont, editors, *Proceedings of the Conference on Libration Point Orbits and Applications*, pages 85–113. World Scientific, 2003.

[8] M. Gidea and F. Deppe. Chaotic Orbits in a Restricted Three-Body Problem: Numerical Experiments and Heuristics. *Communications in Nonlinear Science and Numerical Simulation*, 11(2):161–171, 2006.

[9] H. Lei and B. Xu. High-Order Solutions around Triangular Libration Points in the Elliptic Restricted Three-Body Problem and Applications to Low Energy Transfers. *Communications in Nonlinear Science and Numerical Simulation*, 19(9):3374–3398, 2014.

[10] R. Farquhar. Lunar Communications with Libration-Point Satellites. *Journal of Spacecraft and Rockets*, 4(10):1383–1384, 1967.

[11] R. Farquhar. Future Missions for Libration-Point Satellites. *Journal of Spacecraft and Rockets*, 8(7):815–816, 1971.

[12] R. Farquhar. A Halo-Orbit Lunar Station. *Astronautics and Aeronautics*, 10(6): 59–63, 1972.

[13] T. Heppenheimer. Steps toward Space Colonization: Colony Location and Transfer Trajectories. *Journal of Spacecraft and Rockets*, 15(5):305–312, 1978.

[14] R. Farquhar. The Role of the Sun–Earth Collinear Libration Points in Future Space Exploration. *Space Times*, 39(1):10–12, November–December 2000.

[15] G. Condon and D. Pearson. The Role of Humans in Libration Point Missions With Specific Application to an Earth-Moon Libration Point Gateway Station. *Advances in the Astronautical Sciences*, 109(1):95–110, 2001.

[16] M. Lo and S. Ross. The Lunar L_1 Gateway: Portal to the Stars and Beyond. In *AIAA Space 2001 Conference and Exposition, Paper 2001-4768, Albuquerque, NM, USA*, pages 1–9, 2001.

[17] R. Farquhar, D. Dunham, Y. Guo, and J. McAdams. Utilization of Libration Points for Human Exploration in the Sun–Earth–Moon System and Beyond. *Acta Astronautica*, 55(3–9):687–700, 2004.

[18] C. Beichman, G Gómez, M. Lo, J. Masdemont, and L. Romans. Searching for Life with the Terrestrial Planet Finder: Lagrange Point Options for a Formation Flying Interferometer. *Advances in Space Research*, 34(3):637–644, 2004.

[19] K.E. Post, E. Belbruno, and F. Topputo. Efficient Cis-Lunar Trajectories. In *Global Space Exploration Conference, Paper GLEX-2012.02.3.6x12248, Washington, DC, USA*, pages 1–19. International Astronautical Federation, Paris, France, 2012.

[20] G. Gómez, A. Jorba, J. Masdemont, and C. Simó. Study of the Transfer from the Earth to a Halo Orbit around the Equilibrium Point L_1. *Celestial Mechanics and Dynamical Astronomy*, 56(4):541–562, 1993.

[21] K. Howell, B. Barden, and M. Lo. Application of Dynamical Systems Theory to Trajectory Design for Libration Point Missions. *The Journal of the Astronautical Sciences*, 45(2):161–178, 1997.

[22] A. Zanzottera, G. Mingotti, R. Castelli, and M. Dellnitz. Intersecting Invariant Manifolds in Spatial Restricted Three-Body Problems: Design and Optimization of Earth-to-Halo Transfers in the Sun-Earth-Moon Scenario. *Communications in Nonlinear Science and Numerical Simulation*, 17(2):832–843, 2012.

[23] J. Masdemont. High-Order Expansions of Invariant Manifolds of Libration Point Orbits with Applications to Mission Design. *Dynamical Systems: An International Journal*, 20(1):59–113, 2005.

[24] M. Dellnitz, K. Padberg, M. Post, and B. Thiere. Set Oriented Approximation of Invariant Manifolds: Review of Concepts for Astrodynamical Problems. *AIP Conference Proceedings*, 886:90–99, 2007.

[25] R. Thurman and P. Worfolk. The Geometry of Halo Orbits in the Circular Restricted Three-Body Problem. Technical report, Geometry Center Research Report, University of Minnesota, GCG95, 1996.

[26] F. Topputo, M. Vasile, and F. Bernelli-Zazzera. Low Energy Interplanetary Transfers Exploiting Invariant Manifolds of the Restricted Three-Body Problem. *Journal of the Astronautical Sciences*, 53(4):353–372, October–December 2005.

[27] J. Senent, C. Ocampo, and A. Capella. Low-Thrust Variable-Specific-Impulse Transfers and Guidance to Unstable Periodic Orbits. *Journal of Guidance, Control, and Dynamics*, 28(2):280–290, 2005.

[28] G. Mingotti, F. Topputo, and F. Bernelli-Zazzera. Combined Optimal Low-Thrust and Stable-Manifold Trajectories to the Earth–Moon Halo Orbits. *AIP Conference Proceedings*, 886:100–110, February 2007.

[29] M. Ozimek and K. Howell. Low-Thrust Transfers in the Earth–Moon System Including Applications to Libration Point Orbits. *Journal of Guidance, Control, and Dynamics*, 33(2):533–549, 2010.

[30] C. Martin and B. Conway. *Space Trajectory Optimization*, chapter Optimal Low-Thrust Trajectories Using Stable Manifolds, pages 238–262. Cambridge University Press, 2010.

[31] G. Mingotti, F. Topputo, and F. Bernelli-Zazzera. Optimal Low-Thrust Invariant Manifold Trajectories via Attainable Sets. *Journal of Guidance, Control, and Dynamics*, 34(6):1644–1656, November–December 2011.

[32] J. Mondelo, E. Barrabés, G. Gómez, and M. Ollé. Numerical Parametrizations of Libration Point Trajectories and their Invariant Manifolds. *Advances in the Astronautical Sciences*, 129:1153–1168, 2007.

[33] K. Howell, C. Beckman, C. Patterson, and D. Folta. Representation of Invariant Manifolds for Applications in Three-Body Systems. *The Journal of the Astronautical Sciences*, 54(1):69–93, 2006.

[34] R. Keys. Cubic Convolution Interpolation for Digital Image Processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-29(6):1153–1160, 1981.

[35] V. Szebehely. *Theory of Orbits: The Restricted Problem of Three Bodies.* Academic Press Inc., 1967.

[36] A. Jorba and J. Masdemont. Dynamics in the Center Manifold of the Collinear Points of the Restricted Three Body Problem. *Physica D*, 132(1):189–213, 1999.

[37] G. Gómez and J.M. Mondelo. The Dynamics around the Collinear Equilibrium Points of the RTBP. *Physica D*, 157(4):283–321, 2001.

[38] K. Howell and H. Pernicka. Numerical Determination of Lissajous Trajectories in the Restricted Three-Body Problem. *Celestial Mechanics*, 41(1–4):107–124, 1988.

[39] K. Howell. Three-Dimensional, Periodic, 'Halo' Orbits. *Celestial Mechanics*, 32(1): 53–71, 1984.

[40] C. Conley. Low Energy Transit Orbits in the Restricted Three-Body Problem. *SIAM Journal on Applied Mathematics*, 16(4):732–746, 1968.

[41] J. Llibre, R. Martinez, and C. Simó. Transversality of the Invariant Manifolds Associated to the Lyapunov Family of Periodic Orbits Near L2 in the Restricted Three-Body Problem. *Journal of Differential Equations*, 58(1):104–156, 1985.

[42] T. Parker and L. Chua. *Practical Numerical Algorithms for Chaotic Systems.* Springer, New York, 1989.

[43] C. Simó. *Modern Methods in Celestial Mechanics*, chapter On the Analytical and Numerical Approximation of Invariant Manifolds, pages 285–329. Editions Frontiéres, 1990.

[44] S. DeSalvo, J. Essen, K.L. Ho, and G. Knight. A Method for Designing Impulsive Low-Energy Transfers Between the Earth and the Moon Using Invariant Manifolds. Technical report, Institute for Pure and Applied Mathematics, Research in Industrial Projects for Students, 2006.

[45] H. Späth. *Two Dimensional Spline Interpolation Algorithms.* A.K. Peters, Ltd., 1995.