



Optimization Over Families of Quasi-Periodic Orbits

David Lujan¹ · Daniel J. Scheeres¹

Accepted: 11 August 2024
© The Author(s) 2024

Abstract

Quasi-periodic orbit families in astrodynamics are usually studied from a global standpoint without much attention to the specific orbits which are computed. Instead, we focus on the computation of particular quasi-periodic orbits and develop tools to do so. These tools leverage the parametric structure of families of quasi-periodic orbits to treat orbits only as a set of orbit frequencies instead of states in phase space. We develop a retraction on the family of quasi-periodic orbits to precisely navigate through frequency space, allowing us to compute orbits with specific frequencies. The retraction allows for movements in arbitrary directions. To combat the effects of resonances which slice through frequency space we develop resonance avoidance methods which detect resonances during continuation procedures and change the step size accordingly. We also develop an augmented Newton's method for root-finding and an augmented gradient descent method for unconstrained optimization over a family of quasi-periodic orbits. Lastly, we implement an augmented Lagrangian method to solve constrained optimization problems. We note that many of the tools developed here are applicable to a wider range of solutions defined implicitly by a system of equations, but focus on quasi-periodic orbits.

Keywords Quasi-periodic orbits · Invariant tori · Orbit families · Optimization · Resonance

1 Introduction

Trajectory design of 1-parameter families of solutions is simpler than multi-parameter families since there is only one dimension in the space of solutions to search: forward and backward. Multi-parameter families introduce a larger solution

✉ David Lujan
david.lujan@colorado.edu

Daniel J. Scheeres
scheeres@colorado.edu

¹ Aerospace Engineering Sciences, University of Colorado Boulder, 3775 Discovery Dr., Boulder, CO 80303, USA

space and increase the complexity of finding solutions since there are multiple dimensions to search. Most studies utilizing methods to compute quasi-periodic orbits don't focus on the computation of specific family members, but instead are used to research entire families of orbits [1–7]. Knowing the characteristics and limitations of each orbit family is the first step in determining which orbit types should be used in designing a mission. The next step is finding explicit members from the candidate families to provide options that meet initial mission constraints. Quasi-periodic orbit families generally exist in multi-parameter families in astrodynamics models, and the ability to search through these higher dimensional spaces to pick out a specific orbit is not trivial.

A family of quasi-periodic orbits satisfying the Diophantine conditions¹ has been shown to be a nearly continuous family called a Cantor family (Cantor manifold) in [8–10]. A Cantor manifold has gaps in it where the torus frequencies do not satisfy the Diophantine conditions. It has been shown that a Cantor manifold can be treated as a smooth manifold where smoothness is defined in the sense of Whitney² [9–13]. The smoothness of families of quasi-periodic orbits is a powerful notion as it allows one to draw ties to smooth manifolds defined by the implicit function theorem, and utilize all the rich theory and tools developed for smooth manifolds [14, 15]. One such tool is the gradient; an indispensable tool enabling gradient-based optimization over families of orbits, adding to the list of available tools for trajectory designers. This work focuses on the optimization of cost functions over a family of quasi-periodic orbits.

Optimization can be done on smooth and non-smooth manifolds using non-gradient-based methods, such as genetic algorithms and particle swarm optimizers [16], however in general, it is beneficial to utilize the topology and smoothness of a space when available. The exceptions to this are when the cost of evaluating the gradient is prohibitive, the gradient is not well-behaved over the space, or when there are many local minima.

The standard optimization formulation

$$\begin{aligned} & \min_{z \in \mathbb{R}^d} f(z) \\ & \text{subject to } g_i(z) = 0, \quad i = 1, \dots, r \\ & \quad \quad \quad h_j(z) \leq 0, \quad j = 1, \dots, s, \end{aligned} \quad (1)$$

where $z = [\mathbf{x}^T, \boldsymbol{\omega}^T]^T$ treats the orbit states \mathbf{x} and the orbit frequencies $\boldsymbol{\omega}$ as independent variables which need to satisfy equality constraints defining a family of quasi-periodic orbits. Instead, we reformulate the optimization problem as

¹ Also called a non-resonance condition. The Diophantine conditions ensure the torus frequencies stay sufficiently far from resonances.

² Whitney's theorem in Reference [11] says that a differentiable function $f(x)$ defined on a compact subset $U \subset \mathbb{R}^n$ can be extended to a differentiable function $F(x)$ defined on \mathbb{R}^n .

$$\begin{aligned} & \min_{\omega \in \Omega} f(\mathbf{x}, \omega) \\ & \text{subject to } h_i(\mathbf{x}, \omega) \leq 0, \quad i = 1, \dots, s, \end{aligned} \quad (2)$$

to leverage the fact that each orbit is uniquely parameterized by the orbit frequencies such that the optimization occurs over the set of orbit frequencies in a family of quasi-periodic orbits, Ω . Optimizing over the orbit frequencies greatly reduces the number of optimization variables. The negative gradient of a cost function, $-\nabla_{\omega} f$, dictates the direction in Ω to move locally to minimize f . We use a retraction developed here to explore the descent direction and compute the corresponding quasi-periodic orbits. Using the retraction derives several benefits. First, and foremost, the retraction ensures the invariance equations and any additional parametric constraints for the computation of quasi-periodic orbits are always satisfied. Moreover, the retraction is specialized to handle the subtleties of computing quasi-periodic orbits, such as avoiding resonances, efficiently computing quasi-periodic orbits. Lastly, the retraction uses continuation which is leveraged to perform line searches in a gradient descent method.

Optimizing over families of quasi-periodic orbits, and even periodic orbit, closely relates to a class of problems called PDE-constrained optimization (PDECO). PDECO problems occur when there are constraints governed by physical laws described by partial differential equations (PDEs). PDECO is a well-developed and active field which supports research in a variety of disciplines. One popular use is for topology (shape) optimization which aims to find the optimal shape of a surface given constraints within a dynamical environment. The most well-known use of topology optimization in aerodynamics might be to design aircraft. Other applications of PDECO include crystal growth [17], cooling of electronic components [18], drug transport [19–21], image denoising [22], and uncertainty [23]. The reader is directed to the recent book by Antil and Leykekhman in [24] for a deeper dive into the subject of PDECO and its applications. The following papers contain additional applications [25–27]. While no PDEs are used here, the form of the resulting problem is similar. The equations defining a family of quasi-periodic orbits serve the same purpose as PDEs with the primary constraint being the quasi-periodicity constraint involving the solution flow of an ordinary differential equation (ODE). The process to compute families of solutions in ODEs involves the use of continuation. The idea of using continuation to solve optimization problems seems to have originated in the work of Kernévez and Doedel [28].

Kernévez and Doedel devise an optimization routine to find the solution pair $(\lambda_0, \mathbf{u}_0)$ of control variables and state variables which maximizes a cost functional $f(\lambda, \mathbf{u})$ such that the state equation $\mathbf{g}(\lambda, \mathbf{u}) = \mathbf{0}$ is satisfied and subject to inequality constraints $h_i(\lambda, \mathbf{u}) \leq 0$. Their approach freezes all λ except one element, employs continuation to compute a branch of solutions to $\mathbf{g}(\lambda, \mathbf{u}) = \mathbf{0}$, and then pick the solution from the branch which maximizes $f(\lambda, \mathbf{u})$. They repeat this process in succession and change which control variable is left unfrozen. Their approach is applied to a family of dynamical systems which depend on λ and experience bifurcations in these variables. The successive use of continuation is called successive continuation and has been used in a variety of disciplines [29–33],

however they find optimal control variables which change the dynamical system. In essence, they are solving for a dynamical system which provides favorable behaviors to optimize their processes. The application here considers optimizing over families of solutions within a single dynamical system using successive continuation, but for the simplicity of presentation we mostly restrict our discussion to families of 2-dimensional quasi-periodic invariant tori.

We assume the reader is familiar with quasi-periodic orbits and generally understand how they are computed using continuation methods, therefore we only provide the equations we use to solve for quasi-periodic orbits and the dynamical system in which they reside in Sect. 2. We proceed to develop the optimization problem in Sect. 3 and describe how derivatives of functions are taken with respect to orbit frequencies in Sect. 4. From there we develop a modified continuation procedure, which we call a retraction, to target a set of orbit frequencies in Sect. 5. Due to the freedom the retraction has to move in frequency space we develop methods to reduce the effects of resonances on the computation of quasi-periodic orbits in Sect. 6. We then describe the ways in which we solve optimization problems over a family of quasi-periodic orbits and provide results for various test cases optimizing over the Earth–Moon L_2 quasi-halo orbit family. Finally, we conclude the paper with an overview of the work and provide a discussion in Sect. 9.

2 Preliminaries

2.1 Continuation of Quasi-Periodic Orbits

Let X be a vector of points in phase space which partition an invariant curve of a 2-dimensional quasi-periodic invariant torus with frequencies

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix}. \quad (3)$$

The corresponding stroboscopic time, T , and rotations number, ρ_1 are computed as

$$T = \frac{2\pi}{\omega_0} \quad (4)$$

$$\rho_1 = T\omega_1 \quad (5)$$

We define the vector

$$z = \begin{bmatrix} X \\ T \\ \rho_1 \\ \boldsymbol{\omega} \end{bmatrix}. \quad (6)$$

The equations defining a family of 2-dimensional quasi-periodic invariant tori [34–36] are given by

$$F(\mathbf{z}) = \begin{bmatrix} R_{-\rho_1} \varphi_T(X) \\ \left\langle X - \tilde{X}, \frac{\partial \tilde{X}}{\partial \theta_0} \right\rangle \\ \left\langle X - \tilde{X}, \frac{\partial \tilde{X}}{\partial \theta_1} \right\rangle \\ T\omega_0 - 2\pi \\ \rho_1 - T\omega_1 \end{bmatrix} = \mathbf{0}, \tag{7}$$

where $R_{-\rho_1}$ is a rotation operator that rotates points in the Fourier domain by an angle $-\rho_1$, $\varphi_T(X)$ is the solution flow of each point in X from time 0 to time T , and \tilde{X} is a previously computed invariant curve. The second and third equation in F are called phase constraints and give a unique parameterization to the torus. The last two equations are consistency constraints capturing the dependence between the orbit frequencies and the stroboscopic time and rotation number. The implicit function theorem states that if $\dim(\text{null}(DF)) = 2$ then there are 2 independent parameters to the 2-parameter family. We choose the orbit frequencies to be these independent parameters and define the dependent state variables of a quasi-periodic orbit as

$$\mathbf{x} = \begin{bmatrix} X \\ T \\ \rho_1 \end{bmatrix}. \tag{8}$$

We define the following sets

$$\mathcal{M} = \{z \mid F(z) = \mathbf{0}\} \tag{9}$$

$$\mathcal{X} = \{x \mid F(x) = \mathbf{0}\} \tag{10}$$

$$\Omega = \{\omega \mid F(\omega) = \mathbf{0}\}. \tag{11}$$

If $\dim(\text{null}(DF)) = 2$ for all points in \mathcal{M} then each of \mathcal{M} , \mathcal{X} , and Ω are implicitly defined smooth embedded submanifolds, and refer to them simply as smooth manifolds here.

Suppose we add a single parametric constraint, g_1 to F , such as a constant frequency constraint, then the dimension of the family generally reduces from 2 to 1. We let the new generating equations be

$$\bar{F}(z) = \begin{bmatrix} F(z) \\ g_1(z) \end{bmatrix} = \mathbf{0} \tag{12}$$

and define the 1-dimensional smooth manifolds

$$\bar{\mathcal{M}} = \{z \mid \bar{F}(z) = \mathbf{0}\} \tag{13}$$

$$\bar{\mathcal{X}} = \{x \mid \bar{F}(x) = \mathbf{0}\} \tag{14}$$

$$\bar{\Omega} = \{\omega \mid \bar{F}(z) = \mathbf{0}\}. \tag{15}$$

Single-parameter continuation is typically employed to compute $\bar{\mathcal{M}}$. Let z_0 be a solution to $\bar{F} = \mathbf{0}$. Computing the tangent space of $\bar{\mathcal{M}}$, $T_{z_0}\bar{\mathcal{M}}$, provides a family tangent vector, z'_0 , to predict a nearby solution. The tangent space is computed by computing the nullspace of DF . Given a small step size Δs a guess of a new solution $z_1 \in \bar{\mathcal{M}}$ is given by

$$\hat{z}_1 = z_0 + \Delta s z'_0. \tag{16}$$

The 1-parameter family is mapped out by including the pseudo-arclength constraint³ [37]

$$s(z) = \frac{1}{N_1} \langle X - \tilde{X}, \tilde{X}' \rangle + (T - \tilde{T})\tilde{T}' + (\rho_1 - \tilde{\rho}_1)\tilde{\rho}'_1 + \langle \omega - \tilde{\omega}, \tilde{\omega}' \rangle - \Delta s \tag{17}$$

and solving the system of equations

$$H(z) = \begin{bmatrix} \bar{F}(z) \\ s(z) \end{bmatrix} = \mathbf{0} \tag{18}$$

for z_1 . The process of prediction and correction continues until user-defined conditions are met, or a solution fails to be found.

2.2 Circular Restricted Three-Body Problem

The dynamical system of study here is the circular restricted three-body problem [38]. It is the study of motion of a massless particle under the gravitational forces of two massive bodies P_1 and P_2 in circular orbits about their common center of mass with $m_1 \geq m_2$. The dynamics are stated in a rotating frame such that the x -axis points from P_1 to P_2 , the z -axis is aligned with the orbital angular momentum vector, and the y -axis completes the right-handed coordinate system. The equations are written in a non-dimensional (ND) form where the distance between P_1 and P_2 and the mean motion are equal to one. The dimensionless mass parameter is defined as $\mu = m_2/(m_1 + m_2)$. The equations of motion take the following form

$$\begin{aligned} \ddot{x} &= x + 2\dot{y} - \frac{(1 - \mu)(x + \mu)}{r_1^3} - \frac{\mu(x - 1 + \mu)}{r_2^3} \\ \ddot{y} &= y - 2\dot{x} - \frac{(1 - \mu)y}{r_1^3} - \frac{\mu y}{r_2^3} \\ \ddot{z} &= -\frac{(1 - \mu)z}{r_1^3} - \frac{\mu z}{r_2^3} \end{aligned} \tag{19}$$

³ Note that $z' = [\tilde{X}', \tilde{T}', \tilde{\rho}'_1, \tilde{\omega}']$.

where $r_1 = \sqrt{(x + \mu)^2 + y^2 + z^2}$ is the distance to P_1 and $r_2 = \sqrt{(x - 1 + \mu)^2 + y^2 + z^2}$ is the distance to P_2 . The value of μ used is 0.012153599037880 which describes the Earth–Moon system.

The circular restricted three-body problem admits one integral of motion called the Jacobi constant, an energy-like quantity that determines which areas of phase space are accessible and which other orbits can be reached without changing energy levels. Equation (20) defines the Jacobi constant given a state vector \mathbf{x} in non-dimensional units. In the given form a lower value of J correlates to higher energies at which more areas of space can be accessed.

$$J(\mathbf{x}) = 2 \left(\frac{1 - \mu}{r_1} + \frac{\mu}{r_2} \right) + x^2 + y^2 - (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) \tag{20}$$

The circular restricted three-body problem is an autonomous dynamical system. In these systems periodic orbits form 1-parameter families while n -dimensional quasi-periodic orbits form n -parameter families. In this work we are concerned with the Earth–Moon L_2 2-parameter family of quasi-halo orbits, and has been studied in great detail by Gómez and Mondelo [2], and Lujan and Scheeres [5, 6].

3 Problem Statement

We proceed with a general statement of the problem to be solved. Let $F : \mathbb{R}^D \times \mathbb{R}^p \rightarrow \mathbb{R}^D$ be a system of equations defining a p -parameter family of orbits. We are interested in finding a solution $\mathbf{z}^* \in \mathcal{M}$ such that \mathbf{z}^* minimizes a cost function $f : \mathbb{R}^D \times \mathbb{R}^p \rightarrow \mathbb{R}$ subject to equality and inequality constraints. The optimization problem can be stated as

$$\begin{aligned} & \min_{\mathbf{z} \in \mathbb{R}^{D+p}} f(\mathbf{z}) \\ & \text{subject to } \mathbf{F}(\mathbf{z}) = \mathbf{0} \\ & \quad g_i(\mathbf{z}) = 0, \quad i = 1, \dots, r \\ & \quad h_j(\mathbf{z}) \leq 0, \quad j = 1, \dots, s. \end{aligned} \tag{21}$$

In Problem (21), \mathbb{R}^{D+p} is the space over which the optimization occurs, treating \mathbf{z} as the optimization variables, and \mathbf{x} and $\boldsymbol{\omega}$ as independent quantities. From the implicit function theorem we know \mathbf{x} and $\boldsymbol{\omega}$ are not independent. Each \mathbf{x} is uniquely parameterized by $\boldsymbol{\omega}$ and the frequencies $\boldsymbol{\omega}$ lie on the manifold Ω . We recast Problem (21) as the following optimization problem

$$\begin{aligned} & \min_{\boldsymbol{\omega} \in \Omega} f(\mathbf{x}, \boldsymbol{\omega}) \\ & \text{subject to } g_i(\mathbf{x}, \boldsymbol{\omega}) = 0, \quad i = 1, \dots, r \\ & \quad h_j(\mathbf{x}, \boldsymbol{\omega}) \leq 0, \quad j = 1, \dots, s \end{aligned} \tag{22}$$

to remove the dependence on \mathbf{x} from the problem. As we will see, we can do this because we develop a function which computes \mathbf{x} given $\boldsymbol{\omega}$. We can further

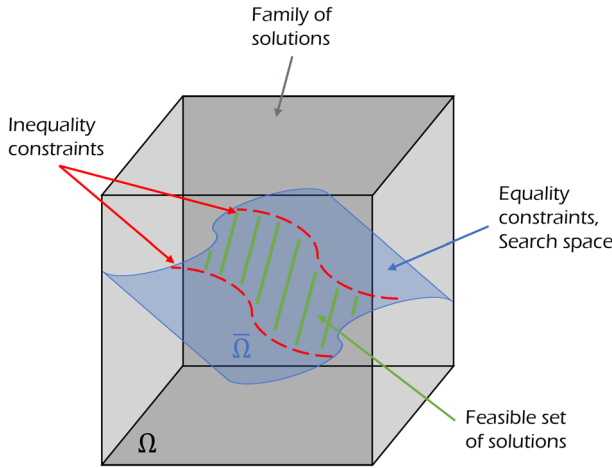


Fig. 1 Equality and inequality constraints reduce the feasible set of solutions

remove the remaining equality constraints by considering the fact that the equality constraints perform the same function as parametric constraints, leading to the new system of equations $\bar{F} : \mathbb{R}^D \times \mathbb{R}^p \rightarrow \mathbb{R}^{D+r}$

$$\bar{F}(z) = \begin{bmatrix} F(z) \\ g_1(z) \\ \vdots \\ g_r(z) \end{bmatrix} = \mathbf{0}. \tag{23}$$

According to the implicit function theorem, if $D_x \bar{F}$ is full rank for all z satisfying $\bar{F}(z) = \mathbf{0}$ then each set

$$\bar{\mathcal{M}} = \{(x, \omega) \mid \bar{F}(x, \omega) = \mathbf{0}\} \tag{24}$$

$$\bar{\mathcal{X}} = \{x \mid \bar{F}(x, \omega) = \mathbf{0}\} \tag{25}$$

$$\bar{\Omega} = \{\omega \mid \bar{F}(x, \omega) = \mathbf{0}\} \tag{26}$$

defines a $(p - r)$ -dimensional smooth manifold. It is evident that $0 \leq r \leq (p - 1)$, since for $r = p$ there is at most a single solution in the space to optimize over, and $r > p$ results in no solutions. Then Problem (22) reduces to

$$\begin{aligned} & \min_{\omega \in \bar{\Omega}} f(x, \omega) \\ & \text{subject to } h_i(x, \omega) \leq 0, \quad i = 1, \dots, s. \end{aligned} \tag{27}$$

Now each $\omega \in \bar{\Omega}$ satisfies F and the equality constraints. Problem (27) effectively removes the equality constraints by considering a space of parameters which identically satisfy the equality constraints. We note that the equality constraints decrease

the dimension of the feasible set of solutions while the inequality constraints reduce the measure of the feasible set of solutions (Fig. 1). Using the retraction developed in Sect. 5 we ensure F and g are automatically satisfied, removing that burden from the optimizer, and leading to an unconstrained optimization problem even in the presence of equality constraints.

4 Derivative With Respect to the Frequencies

Gradient-based methods are used to solve Problem (27). In the optimization problem we find ω^* minimizing a cost function f . We also consider the case of finding ω^* , the roots of a vector function g . Solving for the roots of a vector function is closely related to minimizing a cost function with Newton’s method. In the latter case, g can be thought of as the gradient of an unknown cost function. In both cases we take derivatives with respect to the frequencies. We proceed with an explanation of the derivative of g with respect to ω .

Let $g : \mathbb{R}^D \times \mathbb{R}^p \rightarrow \mathbb{R}^t$ with $t \geq 1$. Then

$$\begin{aligned} \frac{dg}{d\omega} &= \frac{\partial g}{\partial x} \frac{\partial x}{\partial \omega} + \frac{\partial g}{\partial \omega} \frac{\partial \omega}{\partial \omega} \\ &= \begin{bmatrix} \frac{\partial g}{\partial x} & \frac{\partial g}{\partial \omega} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial \omega} \\ \mathbb{1}_p \end{bmatrix} \\ &= \frac{dg}{dz} \frac{dz}{d\omega}. \end{aligned} \tag{28}$$

Here, we make two observations: First, $\frac{dg}{d\omega}$ is the standard Euclidean gradient. It is defined on Ω and not necessarily on $\bar{\Omega}$. Second, $\frac{dz}{d\omega}$ facilitates how z changes given unit changes in each frequency. The matrix $\frac{dz}{d\omega}$ is formed from a particular set of basis vectors spanning $T_z\mathcal{M}$, which we call the principle tangent basis, V_p . Let $z \in \mathcal{M}$ and

$$V = \begin{bmatrix} V_x \\ V_\omega \end{bmatrix} \tag{29}$$

be a matrix whose columns are linearly independent vectors spanning $T_z\mathcal{M}$. Furthermore, we require the $p \times p$ matrix V_ω be full rank. Then the principle tangent basis is found as

$$V_p = V[V_\omega^{-1}\mathbb{1}_p]. \tag{30}$$

Column i of V_p gives how z changes with unit changes in ω_i , or equivalently gives $\frac{dz}{d\omega_i}$. Therefore, we have

$$V_p = \frac{dz}{d\omega}, \tag{31}$$

and the Euclidean derivative of g with respect to ω is

$$\frac{d\mathbf{g}}{d\boldsymbol{\omega}} = \frac{d\mathbf{g}}{dz} V_p \quad (32)$$

the matrix multiplication between the Euclidean derivative of \mathbf{g} , with respect to \mathbf{z} , and V_p . The derivative $\frac{d\mathbf{g}}{dz}$ is readily obtained through analytical or numerical means, such as finite differencing or automatic differentiation. The principle tangent basis is readily obtained from the tangent space of \mathcal{M} . The tangent space is an output of the retraction.

We still require a step to transform the Euclidean derivative $\frac{d\mathbf{g}}{d\boldsymbol{\omega}}$ defined on Ω to a manifold derivative defined on $\bar{\Omega}$. The transformation of the derivative is achieved by projecting the derivative onto $T_z\bar{\mathcal{M}}$ through the projection operator. Now, we require $\mathbf{z} \in \bar{\mathcal{M}}$, and let \bar{V} be a matrix whose columns are linearly independent basis vectors spanning $T_z\bar{\mathcal{M}}$. Then the projection operator of a matrix onto $T_\omega\bar{\Omega}$ is given as

$$\text{Proj}_\omega(A) = \bar{V}_\omega [\bar{V}_\omega^\dagger A], \quad (33)$$

and the manifold derivative of \mathbf{g} with respect to the parameters is

$$\frac{d\bar{\mathbf{g}}}{d\boldsymbol{\omega}} = \text{Proj}_\omega \left(\frac{d\mathbf{g}}{d\boldsymbol{\omega}} \right), \quad (34)$$

where $\bar{\mathbf{g}} = \mathbf{g}|_{\bar{\mathcal{M}}}$ is the restriction of \mathbf{g} to $\bar{\mathcal{M}}$. When g is a scalar function, g , the Euclidean gradient of g with respect to $\boldsymbol{\omega}$ is given by

$$\nabla_\omega g(\mathbf{z}) = \frac{dg}{d\boldsymbol{\omega}}^T. \quad (35)$$

The manifold gradient is then

$$\nabla_\omega \bar{g}(\mathbf{z}) = \text{Proj}_\omega (\nabla_\omega g(\mathbf{z})). \quad (36)$$

In the constrained and unconstrained optimization problems the choice of derivative type is dependent on whether equality constraints are included. Equality constraints reduce the dimension of the search space, so the Euclidean gradient may point in a direction not tangent to $\bar{\Omega}$, showing the need for the manifold gradient.

5 The Retraction

Research investigating orbit families usually relies on the use of single-parameter continuation where the continuation is restricted to follow some rule based on the hard coded constraints [1, 2, 5–7, 34–36]. In these cases, the computed family members primarily serve to be representative members from the family to gain insight into the family as a whole. Intentionally computing a single, specific family member with continuation methods is not an ability we have seen in the astrodynamics community. In 2001 Gómez and Mondelo presented the idea of computing a periodic orbit with given energy level and a prescribed value of

a coordinate [2], however the idea is applied to the refinement of an orbit, which uses differential corrections, not continuation, so one must have an (approximated) orbit very close to the desired one. Leveraging the implicit function theorem, we develop a function, called a retraction, to compute orbits with a desired set of orbit frequencies, allowing for initial guesses to be far from the true orbit. We do this by augmenting a standard continuation method to travel a specified distance in a specified direction in the parameter space of an implicit manifold \mathcal{M} defined by $F(x, \omega) = \mathbf{0}$. The augmentation comes in the form of a line search over step sizes to minimize the difference between the distance in the parameter space which is desired and the distance which has actually been traveled by the continuation method. The retraction takes advantage of the tangent space of \mathcal{M} to continue solutions in the appropriate direction.

5.1 Continuation Equations

The aim of the tool is to target a solution z_t satisfying either Eqs. (7) or (12) by specifying the target frequencies ω_t . To do this we must begin with an initial solution z_0 from within the family.

In the case of satisfying Eq. (7) this is simple. We only require the line

$$\omega(t) = \omega_0 + t\delta\omega_t, \quad t \in [0, 1] \tag{37}$$

be entirely contained in Ω where $\delta\omega_t = \omega_t - \omega_0$. Then it is sufficient to use continuation to compute the branch of solutions lying on $\omega(t)$. We append Eq. (7) with the following scalar parametric constraint, called a direction constraint,

$$\begin{aligned} d(z) &= \frac{\check{\omega} - \check{\omega}_0}{\omega - \omega_0} - \frac{\check{\omega}_t - \check{\omega}_0}{\omega_t - \omega_0} \\ &= \frac{\delta\check{\omega}}{\delta\omega} - \frac{\delta\check{\omega}_t}{\delta\omega_t} \end{aligned} \tag{38}$$

where ω is the coordinate of ω such that

$$\omega = \arg \max_{i=0,1} |\omega_{0,i} - \omega_{t,i}|, \tag{39}$$

and $\check{\omega}$ are the remaining coordinates of ω . Choosing ω and $\check{\omega}$ in this way ensures there are no issues with singularities. Moreover, we append the pseudo-arclength equation $s(z)$ from Eq. (17) so that

$$\mathbf{H}(z) = \begin{bmatrix} F(z) \\ d(z) \\ s(z) \end{bmatrix} = \mathbf{0} \tag{40}$$

has a full rank Jacobian matrix. Equation (40) is used in a continuation method and ensures we compute solutions which lie along the straight line connecting ω_0 to ω_t . However, we need to modulate the step size so that the continuation can converge to

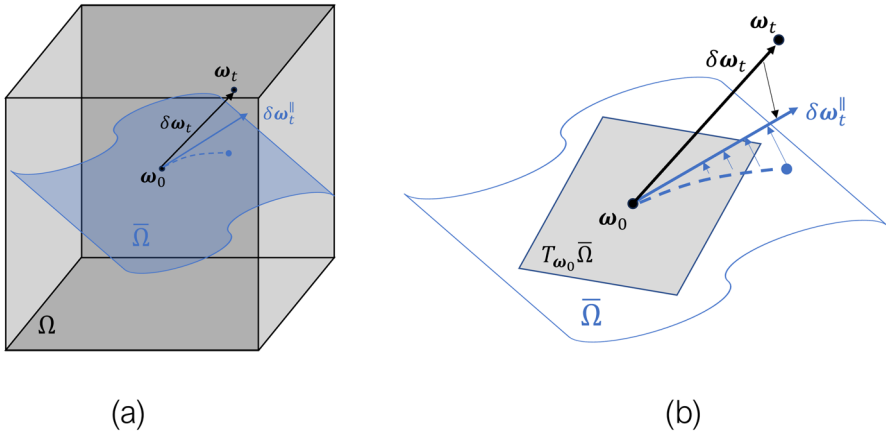


Fig. 2 Depiction of the general direction constraint. The manifold $\bar{\Omega}$ shown against the manifold Ω in (a). The projection of $\delta\omega$, onto the tangent space, resulting in $\delta\omega_t^\parallel$, and the projection of the continuation path onto $\delta\omega_t^\parallel$ in (b)

the target solution z_t and stop computing solutions. We will return to this in Section (5.4).

For the case of satisfying Eq. (12) we have the 1-dimensional manifold \mathcal{M} and only need to append $s(z)$ to $\bar{F}(z)$ to generate the continuation equations, $\bar{H}(z)$. However, suppose $\dim\mathcal{M} = p$ (i.e. we are interested in a p -parameter family of p -dimensional quasi-periodic orbits) and $\dim\bar{\mathcal{M}} = p - c$, then the direction constraint becomes a vector equation and modifications need to be made. The reader is pointed to Fig. 2 to follow the proceeding discussion of this case.

We append the pseudo-arclength constraint to Eq. (12), however it is likely that $\omega(t) \notin \bar{\Omega}$ for all $t \in [0, 1]$, so we must project the vector $\delta\omega_t$ into the tangent space $T_{\omega_0}\bar{\Omega}$. We then want to ensure that the path the continuation follows on $\bar{\Omega}$ is aligned with the projected vector $\delta\omega_t^\parallel$. That is to say the vector pointing from the initial frequencies to the current frequencies $\delta\omega = \omega - \omega_0$ projected into $T_{\omega_0}\bar{\Omega}$ is parallel to $\delta\omega_t^\parallel$. The projection is accomplished as follows. Let \bar{V}_0 be a matrix whose columns are linearly independent vectors spanning $T_{z_0}\bar{\mathcal{M}}$. Furthermore, we let

$$\bar{V}_0 = \begin{bmatrix} \bar{V}_{x_0} \\ \bar{V}_{\omega_0} \end{bmatrix}, \tag{41}$$

where \bar{V}_{ω_0} is a $p \times (p - c)$ matrix with linearly independent column vectors⁴ and \bar{V}_{x_0} is the remaining elements of \bar{V}_0 . The projection operator is given as

$$\text{Proj}_{\omega_0}(\mathbf{v}) = \bar{V}_{\omega_0} \left[\bar{V}_{\omega_0}^\dagger \mathbf{v} \right]. \tag{42}$$

Now the projected vector we wish to follow along the manifold is

⁴ \bar{V}_0 having linearly independent column vectors does not imply any subset of the rows also have linear independence. Thus, we require \bar{V}_{ω_0} to also have linearly independent columns.

$$\delta\omega_t^{\parallel} = \text{Proj}_{\omega}(\delta\omega_t). \tag{43}$$

The projected vector from the initial frequencies to the current frequencies in the continuation is given by

$$\delta\omega^{\parallel} = \text{Proj}_{\omega_0}(\delta\omega), \tag{44}$$

leading to the following direction constraints

$$\bar{d}(z) = \frac{\delta\check{\omega}^{\parallel}}{\delta\omega^{\parallel}} - \frac{\delta\check{\omega}_t^{\parallel}}{\delta\omega_t^{\parallel}}. \tag{45}$$

It is important to note that the projection operator uses \bar{V}_{ω_0} from the initial solution z_0 for the entirety of the computation of the branch.

We do not append all the direction constraints to $\bar{F}(z)$. Instead, only $p - 1 - c$ constraints are appended. The choice of which equations to discard does not matter. Any $p - 1 - c$ of the scalar equations suffices to construct a full rank Jacobian matrix. Let \bar{d} be the $p - 1 - c$ direction equations, then the continuation equations become

$$\bar{H}(z) = \begin{bmatrix} \bar{F}(z) \\ s(z) \\ \bar{d}(z) \end{bmatrix} = \mathbf{0}. \tag{46}$$

The original target solution is likely not able to be achieved. Instead, we must settle for a *nearby* solution which satisfies Eq. (12). To this end, let

$$d = \|\delta\omega_t^{\parallel}\|. \tag{47}$$

Then traveling in the direction of $\delta\omega_t^{\parallel}$ on $\bar{\Omega}$ a distance d should provide a nearby solution as long as d isn't too large.

5.2 Initial Direction

From Eq. (43) we know the initial direction on $\bar{\Omega}$ needed to continue solutions which satisfy the direction constraints of Eq. (46). However, to initialize the continuation method we need to know the direction for both the states and the frequencies. We generate the initial direction z'_0 to move on $\bar{\mathcal{M}}$ as

$$\begin{aligned} z'_0 &= \bar{V}_0 \begin{bmatrix} V_{\omega_0}^{\dagger} \delta\omega_t \end{bmatrix} \\ &= \begin{bmatrix} \bar{V}_{x_0} \bar{V}_{\omega_0}^{\dagger} \delta\omega_t \\ \bar{V}_{\omega_0} \bar{V}_{\omega_0}^{\dagger} \delta\omega_t \end{bmatrix} \\ &= \begin{bmatrix} \delta x_t^{\parallel} \\ \delta\omega_t^{\parallel} \end{bmatrix}. \end{aligned} \tag{48}$$

In the case with no additional parametric constraints $V_{\omega_0}^{\dagger} = V_{\omega_0}^{-1}$, reducing Eq. (48) to

$$\begin{aligned}
 z'_0 &= V_0 \left[V_{\omega_0}^{-1} \delta \omega_t \right] \\
 &= \begin{bmatrix} V_{x_0} & V_{\omega_0}^{-1} \delta \omega_t \\ V_{\omega_0} & V_{\omega_0}^{-1} \delta \omega_t \end{bmatrix} \\
 &= \begin{bmatrix} \delta x_t \\ \delta \omega_t \end{bmatrix}.
 \end{aligned} \tag{49}$$

5.3 Stopping Criterion

A standard continuation method computes branches until either the maximum number of solutions has been computed or when a new solution fails to be found. However, to target a specific solution we must introduce a stopping criterion which stops on the correct solution. In the case with no parametric constraints we know the frequencies ω_t and can stop when those frequencies are met, however when parametric constraints are introduced the target frequencies are likely not in $\bar{\Omega}$. Thus, we introduce a stopping criterion based on distance traveled on the frequency manifold.

The target distance to travel is given in Eq. (47). We need a way to compute and keep track of the distance traveled, so that we can end the continuation when the distance traveled equals the target distance within tolerance. To describe the distance computation we describe how the distance is computed on the first continuation step. Let z_0 be the initial solution with frequencies ω_0 , Δs_0 be the initial step size, and let z_1 be the first solution from the continuation. Then we define the distance traveled on $\bar{\Omega}$ be $d_1 = \text{sign}(\Delta s_0) \|\omega_1 - \omega_0\|$. Extending this to the k^{th} solution leads to

$$d_k = \sum_{j=1}^k \text{sign}(\Delta s_{j-1}) \|\omega_j - \omega_{j-1}\|. \tag{50}$$

The distance d_k is an approximation of the length of the continuation curve starting at ω_0 in $\bar{\Omega}$ at the k^{th} continuation step. Positive step sizes represent forward progression along the continuation curve and thus accumulate positive distance. On the contrary, negative step sizes represent backtracking along the continuation curve, and thus subtract off from the total distance traveled. In the case of no parametric constraints the distance computation can be simplified to $d_k = \|\omega_k - \omega_0\|$. Computing d_k allows us to keep track of the progression of the continuation. Leveraging this knowledge we can intelligently choose step sizes such that $d_k \rightarrow d$.

5.4 Step Size Computation

A standard continuation method utilizes some nominal step size controller based on the number of iterations needed to converge on a solution. This works well for computing branches of solutions to study families as a whole, but does not perform

well when a specific solution from a single- or multi-parameter family is desired. A nominal step size update is adequate until the solutions approach the target solution, at which point the step size needs to be fine-tuned.

The method to choose the step size to converge on the target distance is a line search over step sizes, given in Algorithm 1. Let ω_k and ω'_k be the frequencies and their family tangent values, respectively, at the current continuation solution. Additionally, let Δs_k be the current nominal step size, and d_k be the current distance traveled on the parameter manifold. A step size s is chosen from a uniform partition on the interval $[-1.2|\Delta s_k|, 1.2|\Delta s_k|]$ with N points. We choose $N = 1000$, allowing the step size to potentially decrease by 3 orders of magnitude each continuation step. The step size is multiplied by $\|\omega'_k\|$ to generate an estimate $\Delta\omega$ of how much the frequencies will change by on the next continuation solution. $\Delta\omega$ is also the predicted distance traveled on the parameter manifold from solution k to solution $k + 1$. The distance is added to the current distance traveled, d_k , to generate a guess of the total distance traveled upon convergence to solution z_{k+1} . The absolute value of the difference between the target distance d and the predicted total distance traveled gives an error, or the distance left to go before converging to the target distance. We choose the step size s which minimizes this error.

Algorithm 1 Line search to determine the step size to get closest to the target distance

```

1: procedure TARGETSTEPsize( $\Delta s_k, \omega'_k, d_k, d, N$ )
2:    $\Delta d_{\min} = |d - d_k|$ 
3:   for  $j \in \{0, 1, \dots, N\}$  do
4:      $s = -1.2|\Delta s_k| + 2.4|\Delta s_k| \frac{j}{N}$ 
5:      $\Delta\omega = s\|\omega'_k\|$ 
6:      $d_j = d_k + \Delta\omega$ 
7:      $\Delta d_j = |d - d_j|$ 
8:     if  $\Delta d_j < \Delta d_{\min}$  then
9:        $\Delta s_{\min} = s$ 
10:       $\Delta d_{\min} = \Delta d_j$ 
11:    end if
12:  end for
13:  return  $\Delta s_{\min}$ 
14: end procedure

```

We note there are more sophisticated line search methods to pick the optimal step size, however we are not interested in getting the exact optimal step size because the true frequencies will differ from the predicted frequencies. Moreover, to truly know the optimal step size it is necessary to solve the continuation equations for each step size. This is extremely prohibitive, so we rely solely on the predicted error rather than the true error to pick the optimal step size. We comment on the range of step sizes; by allowing the lower bound of the step sizes to be negative we allow the possibility of backtracking in

the event the continuation has traveled too far, ultimately allowing for convergence to the target distance.

5.5 The Algorithm

Thus far we have seen that by projecting the vector from the frequencies of the initial solution to the target frequencies to the parameter tangent space and using the distance constraints we can control the direction of the continuation. Furthermore, by modulating the step size the continuation can achieve a target solution.

Algorithm 2 Retract from $\mathcal{T}_{\omega_0} \bar{\Omega}$ down to $\bar{\Omega}$ and find the corresponding solution on $\bar{\mathcal{X}}$

```

1: procedure RETRACT( $\delta\omega_t, z_0, \bar{V}_0, \Delta s_0, \varepsilon, N$ )
2:    $d = \|\delta\omega_t\|$ 
3:    $z'_0 = \bar{V}_0 [\bar{V}_{\omega_0}^\dagger \delta\omega_t]$   $\triangleright$  Automatically get  $\delta\omega_t^\parallel$  from Equation (48).
4:    $z'_0 = \frac{z'_0}{\|z'_0\|}$ 
5:    $k = 0$ 
6:    $(\omega_k, \omega'_k, \Delta s_k, d_k) = (\omega_0, \omega'_0, \Delta s_0, 0)$ 
7:   while  $|d - d_k| > \varepsilon$  do
8:      $\Delta s_k = \text{TARGETSTEP SIZE}(\Delta s_k, \omega'_k, d_k, d, N)$ 
9:      $(z_{k+1}, \bar{V}_{k+1}, \Delta s_{k+1}) = \text{SOLVER}(z_k, z'_k, \Delta s_k)$ 
10:     $d_{k+1} = d_k + \|\omega_{k+1} - \omega_k\|$ 
11:     $z_{k+1} = z_{k+1} - z_k$ 
12:     $k \leftarrow k + 1$ 
13:   end while
14:    $(z^*, \bar{V}^*) = (z_k, \bar{V}_k)$ 
15:   return  $(z^*, \bar{V}^*)$ 
16: end procedure

```

Algorithm 2 takes as inputs the direction to travel in the frequency space, $\delta\omega_t$, the initial solution, z_0 , the initial tangent space, \bar{V}_0 , the initial step size, Δs_0 , the error tolerance on the distance, ε , and the number of points for the routine TARGETSTEP SIZE, N . The distance, d , to travel is computed as the norm of $\delta\omega_t$. Then the family tangent vector is computed according to Eq. (48) and normalized with the appropriate norm $\|\cdot\|$ consistent with the inner product used in the pseudo-arclength equation. The algorithm proceeds to go into the continuation loop. In each iteration the step size is modulated with Algorithm 1, the next solution is computed with the routine SOLVER. The routine SOLVER solves the continuation equations and returns the next solution, its tangent space, and the nominal step size. The cumulative distance in the frequency space is computed. The continuation loop continues until either the distance traveled is within the error tolerance or an updated solution is not found, and returns the final solution.

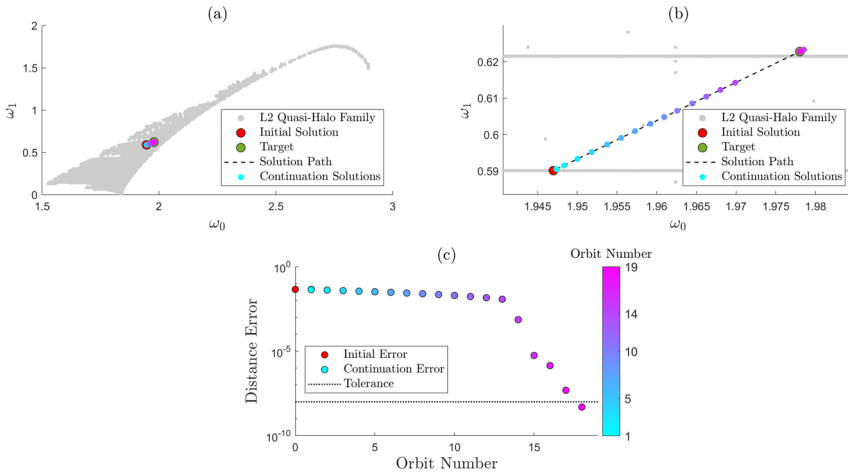


Fig. 3 Targeting the frequencies of a quasi-periodic orbit

5.6 Validation

The retraction is tested on the 2-parameter family of the Earth–Moon L_2 quasi-halo orbits in the circular restricted three-body problem. For the case of no additional parametric constraints we use the retraction to compute an orbit with a specific set of orbit frequencies. For the case with a parametric constraint we include a Jacobi constant constraint, constraining the possible set of frequencies. The only way to know the achievable set of frequencies is to compute the entire constant energy branch. Without computing the entire branch we settle for targeting the distance along the branch to produce an orbit close to a desired set of orbit frequencies.

5.6.1 Targeting Orbit Frequencies in a Family of Quasi-Periodic Orbits

We choose an initial quasi-halo orbit with frequencies

$$\omega = (1.946982196701564, 0.590131700668313),$$

target frequencies

$$\omega_t = (1.978018524093039, 0.622782717360435),$$

and a distance tolerance of $\epsilon = 1e-8$. Therefore,

$$\delta\omega_t = (0.031036327391475, 0.032651016692122)$$

and $d = 0.045048224260009$. Based on $\delta\omega_t$, we have $\delta\dot{\omega} = \omega_0$ and $\delta\omega = \omega_1$.

Algorithm 2 produces a quasi-halo with frequencies

$$\omega^* = (1.978018520706167, 0.622782713797358).$$

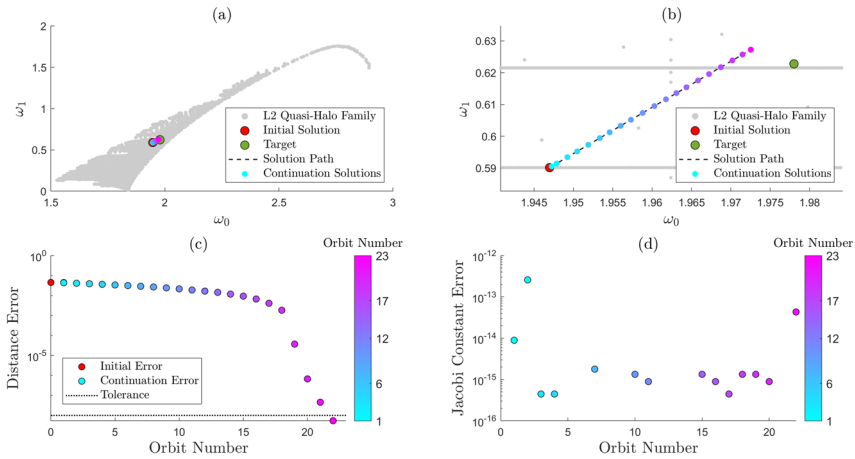


Fig. 4 Targeting the distance along a constant Jacobi energy line

The results are given in Fig. 3. Plots (a) and (b) show the continuation in frequency space with plot (b) being a zoomed in view of plot (a), while plot (c) shows the distance error versus the continuation solutions. Plot (a) shows how long the continuation path was relative to the size of the family in frequency space. In plot (b) it is seen that the solution jumps past the target solution and begins to backtrack. From there it takes only a few more solutions to hone in on a quasi-halo orbit with frequencies satisfying the tolerance. The entire algorithm computes 19 orbits and takes 105 s to run. The computations were performed in the Julia language and the integrations were not run in parallel. The runtime can be significantly reduced if the integrations are run in parallel.

Plot (c) shows a monotonically decreasing error as the orbits are successively computed. This behavior is expected for larger step sizes, but as the step size gets small enough it is not unexpected for the distance error to oscillate. The oscillation can occur because the predicted frequencies do not exactly determine the true frequencies. The Newton’s method can converge to an orbit further away from the target than expected based on the predicted frequencies.

5.6.2 Targeting Distance on a Subset of a Family of Quasi-Periodic Orbits

For this test case a Jacobi constant constraint is appended to F , creating a 1-parameter family of quasi-halo orbits. No direction constraints are necessary since the continuation method can trace out a 1-parameter branch.

The same initial quasi-halo orbit, target frequencies, distance error tolerance, and number of points discretizing the invariant curve are used as in Sect. 5.6.1. The only difference is the use of the Jacobi constant constraint. The value is constrained to match the Jacobi constant of the initial quasi-halo orbit.

Algorithm 2 produces a quasi-halo with frequencies

$$\omega^* = (1.978018524093039, 0.622782717360435).$$

The results are given in Fig. 4. Plots (a) and (b) show the continuation in frequency space, while plot (c) shows the distance error versus the continuation solutions, and plot (d) shows the error in the prescribed value of the Jacobi constant for each computed solution. The initial step size is $1e-3$ while the maximum nominal step size is $5e-3$. Keeping the maximum step size small when parametric constraints are included cultivates accurate distance computations. The number $5e-3$ is conservative since constant Jacobi energy lines in frequency space are nearly straight [5]. Though it cannot be seen in plot (b) the solution jumps past the target solution and begins to backtrack. The entire algorithm computes 23 orbits and takes 134 s to run.

Plot (c) shows the distance error decreasing monotonically again while plot (d) shows the computed orbits share the same Jacobi constant. Plot (b) shows the orbit a distance d along this constant Jacobi energy branch. If we compare the distance between the initial and the final parameters on Ω we get a distance of 0.04504817, but the distance on $\bar{\Omega}$ is 0.04504822. The distance on Ω does not quite meet the distance error tolerance of $1e-8$, while the distance on $\bar{\Omega}$ does.

6 Resonance Avoidance Methods

It is well known that resonances between the internal frequencies of invariant tori play a vital role in the theoretical study and numerical computation of quasi-periodic invariant tori. Greene [39] and MacKay [40] show invariant curves are most robust when the rotation number of the curve is a noble number. As the rotation number moves away from a noble number and approaches a rational number the robustness diminishes. At a resonance the frequencies of an invariant torus violate the non-resonance condition, and the dimension of the torus decreases, resulting in a submanifold of lower-dimensional invariant tori [9]. For a 2-dimensional quasi-periodic invariant torus with frequencies near a resonance the invariant curve no longer covers \mathbb{T} , and resembles an island chain on a Poincaré map [41], a behavior which extends to higher-dimensional quasi-periodic invariant tori. Both phenomena are destructive to the computation of quasi-periodic orbits.

The complications of computing invariant tori near resonances is numerically observed by Gómez and Mondelo in [2] wherein they compute a 2-parameter family of 2-dimensional quasi-periodic orbits. To avoid these challenges, they tactfully choose generating periodic orbits such that $\frac{2\pi}{\rho}$ is an integer plus the golden number $\frac{1}{2}(1 + \sqrt{5})$, leading to the most irrational set of frequencies. Gómez and Mondelo enforce the continuation to maintain this rotation number to ensure the family of computed quasi-periodic orbits exist in the nonlinear system.

McCarthy and Howell present a method to avoid resonances in a 3-parameter family of 3-dimensional quasi-periodic orbits [35]. They suggest fixing the rotation numbers such that they take values which are not multiples of 2π . Their method ensures the ratios of the frequencies stay away from integer values. They do not address the issue that many resonances occur when the ratios of the frequencies

are not integers. Picking the rotation numbers to ensure the frequency ratios avoid rational numbers is a more challenging task. However, avoiding resonances is exactly one of the issues addressed in this section.

Both resonance avoidance methods described above are sufficient for studying large portions of families of quasi-periodic orbits, but they are restrictive in the types of branches which can be computed, and do not allow the parameter space to be explored freely.

6.1 Irrational Numbers in Floating Point Arithmetic

To efficiently compute branches of quasi-periodic orbits in arbitrary directions we devise a methodology to avoid resonances. To avoid resonances it is necessary to identify when the torus frequencies are near a resonance. Identifying a resonance on a computer equates to determining whether the ratio of two frequencies in floating point arithmetic is well approximated by a rational number. Determining the irrationality of a frequency vector on a computer is not a trivial task since all floating point numbers are necessarily rational. While this is not a well-defined problem we leverage research that addresses the issue of detecting resonances on a computer.

A question addressed by Sander and Meiss in [42] is: Given a floating point number x and an interval $I_\delta = [x - \delta, x + \delta]$ what is the rational $\frac{n}{d}$ in this interval with the smallest denominator? They mention there are built in functions, such as in Mathematica and MATLAB, which return an n and d , but do not return the smallest denominator. Therefore, they present an algorithm which does return the smallest denominator such that $\frac{n}{d} \in I$. We use their algorithm, given in Algorithm 3, to determine the approximate rationality between the torus frequencies.

Algorithm 3 Compute the smallest denominator of the rational number approximating a floating point number within an interval

```

1: procedure SMALLDENOM( $x, \delta$ )
2:    $(n, d) = (p_l, q_l) = (0, 1)$ 
3:    $(p_r, q_r) = (1, 0)$ 
4:   while  $|x - \frac{n}{d}| \geq \delta$  do
5:      $(n, d) = (p_l + p_r, q_l + q_r)$ 
6:     if  $x < \frac{n}{d}$  then
7:        $(p_r, q_r) = (n, d)$ 
8:     else
9:        $(p_l, q_l) = (n, d)$ 
10:    end if
11:  end while
12:  return  $(n, d)$ 
13: end procedure

```

Sander and Meiss perform tests on a set of floating point numbers to determine which values of d make for good approximations to x . They found that $\log_{10}(d)$ has a mean value of $\log_{10}(\delta)/2$, and when $\log_{10}(d)$ deviates from the mean value then x is well approximated by a rational number. Their results lead to the development of the metric

$$\sigma = |\log_{10}(d) + \log_{10}(\delta)/2| \tag{51}$$

to rate how “rational” x is [42]. In their work they use $\delta = 10^{-8}$, and deem a point x to be rational when the log deviation σ is greater than 0.3375. We use the same δ value in this work, but use different values of σ to determine when an orbit is near a resonance. It is not straightforward to determine what σ should be because each resonance has a region around it where the computation of orbits becomes difficult; the size of this region is different for each resonance.

6.2 Method 1: Picking Irrational Frequencies

In a continuation method to compute branches of quasi-periodic orbits we have a current solution $z_k = [x_k^T, \omega_k^T]^T$ and a family tangent vector z'_k which governs the prediction of the next solution z_k . Suppose a performance-based nominal step size Δs for the prediction is used, such as the step size controller in Chapter 4 of Seydel [37], then the nominal set of frequencies is

$$\omega_{k+1} = \omega_k + \Delta s_k \omega'_k. \tag{52}$$

The nominal frequencies may be nearly commensurate, thus making the next torus numerically challenging to compute. However, within any open interval around any number there are irrational numbers. Leveraging this information we conduct a line search to pick a new step size within the interval $[0.9\Delta s_k, 1.1\Delta s_k]$ so that the new predicted set of frequencies is the “most” irrational.

For a given value of $s \in [0.9\Delta s_k, 1.1\Delta s_k]$ we compute a set of predicted frequencies $\omega_p = \omega_k + s\omega'_k$. For each combination of ratios between the frequencies, excluding the inverse of the ratios, we rate the rationality of each ratio using SMALLDENOM and Eq. (51). These values are stored for each combination of ratios and for each set of predicted frequencies, producing a matrix of σ values. The best step size is the one that minimizes the average σ value for a fixed value of s . Picking the best step size this way minimizes the likelihood of being near a resonance, promoting well-conditioned computations of invariant tori. The method to choose the best step size to produce the most irrational set of frequencies is given in Algorithm 4.

We note that Algorithm 3 is not well-behaved over the interval $[0.9\Delta s_k, 1.1\Delta s_k]$ (see Figure 13 in [42]), so locating the optimal step size is reduced to performing a grid search with 1000 values of s equally spaced throughout the interval.

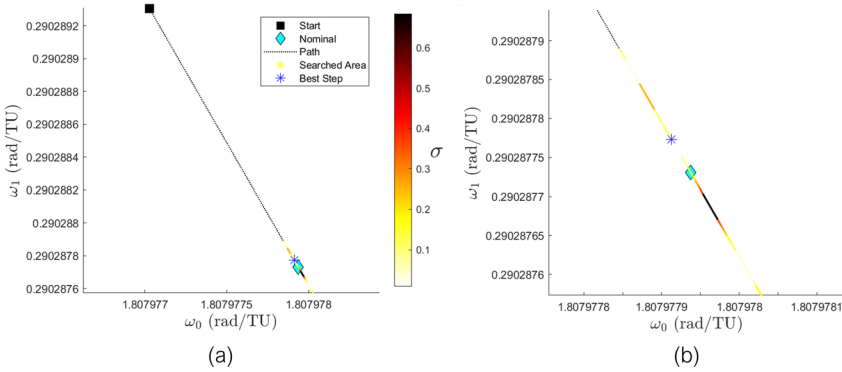


Fig. 5 Example of picking step size to minimize chance of stepping near a resonance. Plot (b) is a zoomed in copy of plot (a)

Algorithm 4 Compute best step size to avoid being near a resonance

```

1: procedure IRRATIONALSTEPSize( $\Delta s, \omega, \omega', \delta$ )
2:   for all  $s \in [0.9\Delta s, 1.1\Delta s]$  do
3:      $\omega_p = \omega + s\omega'$ 
4:     for all unique combinations of frequencies  $\omega_{p,i}, \omega_{p,j}$  do
5:        $\hat{r} = \frac{\omega_{p,i}}{\omega_{p,j}}$ 
6:        $(n, d) = \text{SMALLDENOM}(\hat{r}, \delta)$ 
7:        $\sigma_{(i,j),s} = |\log_{10}(d) + \log_{10}(\delta)|/2$ 
8:     end for
9:   end for
10:   $\sigma_s = E_{(i,j)}[\sigma_{(i,j),s}]$   $\triangleright$  Take the mean over the  $(i, j)$  dimension of the array.
11:   $\Delta s_{\min} = \arg \min_s \sigma_s$ 
12:  return  $\Delta s_{\min}$ 
13: end procedure

```

An example of using Algorithm 4 is given in Fig. 5. The black square represents the frequencies of the current solution. The cyan diamond represents the nominal frequencies using the nominal continuation step size. One thousand predicted frequencies are computed within the interval $[0.9\Delta s, 1.1\Delta s]$. Each pair of predicted frequencies are evaluated using Algorithm 3 and Eq. (51), and are colored according to their σ value. The blue star is the frequency pair which attains the minimum value of σ in the grid search. The step size corresponding

to that frequency pair is the step size chosen for the prediction step. Had the nominal step size been slightly larger the nominal set of frequencies could have been considered rather rational, possibly requiring more Newton iterations to converge.

6.3 Method 2: Hopping Beyond Resonances

With a step size chosen nominally, similar to the method presented in Chapter 4 of Seydel [37], it is easy for the continuation of quasi-periodic orbits to get stuck near a resonance. As the continuation approaches a strong resonance more iterations will necessarily be needed to converge to a true solution. By taking more iterations to converge the step size will decrease and the frequencies of the quasi-periodic orbits will ever so slowly approach the resonance, leading to stagnation and termination in the continuation procedure.

To circumvent the issue of premature termination due to resonances one needs a way of identifying when the continuation is stagnating at a particular resonance. From there we can make an informed decision on how to adjust the step size. Tracking the σ values alone to determine the likelihood of being near a resonance is not sufficient. The σ value is only able to detect when the torus frequencies are near a resonance, however the continuation of a branch of quasi-periodic orbits may be progressing sufficiently and just happen to converge with frequency ratios near different rational values.

A method to hop beyond resonances requires not only tracking the σ values, but also requires tracking the ratio of the frequencies to determine progress in the continuation. Therefore, we present the following heuristic method to identify and jump past resonances in Algorithm 5 and Algorithm 6. Let ω_k be the frequencies of the current converged solution with the new nominal step size Δs_k . Let r be a vector which records the value of each combination of frequency ratios, and let c be a vector of positive integers which keeps a count of the consecutive occurrences when each frequency ratio is near a resonance. At the start of the continuation, for $k = 0$, r is composed of the values of frequency ratios for the initial quasi-periodic orbit, while c starts at all zeros. Once the next solution z_{k+1} has been found then Algorithm 6 is called to determine whether the step size should be increased to move beyond a resonance.

Algorithm 5 Determine if continuation of quasi-periodic orbits are near a resonance

```

1: procedure CHECKRESONANCES( $\mathbf{r}, \mathbf{c}, \boldsymbol{\omega}, \delta$ )
2:    $k = 1$ 
3:   for all unique combinations of frequencies  $\omega_i, \omega_j$  do
4:      $\hat{r} = \frac{\omega_i}{\omega_j}$ 
5:      $(n, d) = \text{SMALLDENOM}(\hat{r}, \delta)$ 
6:      $\sigma = |\log_{10}(d) + \log_{10}(\delta)/2|$ 
7:     if  $\sigma > 0.1$  and  $|\mathbf{r}[k] - \frac{n}{d}| < 1e - 3$  then
8:        $\mathbf{c}[k] = \mathbf{c}[k] + 1$ 
9:     else
10:       $\mathbf{c}[k] = 0$ 
11:    end if
12:     $\mathbf{r}[k] = \hat{r}$ 
13:     $k = k + 1$ 
14:  end for
15:  return  $\mathbf{r}, \mathbf{c}$ 
16: end procedure

```

Algorithm 6 first calls Algorithm 5 to update \mathbf{r} and \mathbf{c} . Algorithm 5 goes through each combination of frequency ratios of $\boldsymbol{\omega}_{k+1}$, computes the rational with the smallest denominator using Algorithm 3, and computes σ . For a given frequency ratio we increase the counter associated with that ratio by one if two conditions are met. The first is that σ needs to be sufficiently large to indicate the presence of a resonance. Trial and error has resulted in a threshold value of 0.1. The second condition is the frequency ratio of the current continuation solution must be sufficiently close to the frequency ratio of the previous continuation solution. We let the threshold value be $1e-3$. If either of these conditions are not met then the counter associated with that frequency ratio is reset to zero. The new values of the frequency ratios are recorded and compared with the next continuation solution.

Algorithm 6 Decide whether to increase step size to hop a resonance

```

1: procedure HOPRESONANCE( $\mathbf{r}, \mathbf{c}, \boldsymbol{\omega}, \Delta s, \delta$ )
2:    $(\mathbf{r}, \mathbf{c}) = \text{CHECKRESONANCES}(\mathbf{r}, \mathbf{c}, \boldsymbol{\omega}, \delta)$ 
3:   if any element of  $\mathbf{c}$  is equal to 5 then
4:      $\Delta s = 0.05 * \text{sign}(\Delta s)$ 
5:   end if
6:   return  $\mathbf{r}, \mathbf{c}, \Delta s$ 
7: end procedure

```

Algorithm 6 uses \mathbf{c} to decide if the continuation has stagnated at a resonance. If any value in \mathbf{c} has a value of 5, indicating the continuation has been near a particular resonance for 5 consecutive solutions, then we change the step size to be

0.05 multiplied by the sign of the step size. A step size magnitude of 0.05 has been deemed sufficient to be able to move beyond many resonances while also providing for good linear approximations in the prediction step of continuation. However, the value 0.05 should not be taken as absolute and is likely dependent on many factors such as the family of orbits, the dynamical system, and the dimension of the invariant tori. As it has been mentioned the method presented in this section is a heuristic method, and, as such, many values have been chosen to provide favorable performance for the applications encountered in this work.

6.4 Validation

We tested the performance between the use of both resonance avoidance methods together against using neither method. In the former case we call it the RAV variant, and in the latter case we call it the plain variant. The plain variant uses only the nominal step size update. The nominal step size update is also used in the RAV variant with the inclusion of Algorithm 4 to make little modifications to the step size and with the inclusion of Algorithm 6 to prevent stagnation at a resonance.

For the test we chose 7 quasi-halo orbits from within the Earth–Moon L_2 quasi-halo orbit family to act as initial solutions for continuing branches of solutions. About each initial solution 15 directions equally spaced out on a circle are chosen for the branches to follow. The frequencies are targeted according to the retraction developed in Sect. 5.

All together there are 105 branches of solutions spanning a large portion of the unstable 2-dimensional quasi-halo orbits for both variants. The branches are faced with resonances to get to the target set of frequencies. We are looking to see how many of the branches successfully reach the target set of frequencies, how much time the branch took to get there, and how many family members were computed along the way. Examining these three quantities gives insight into the performance of the resonance avoidance methods. Statistics relating to these three metrics are in Table 1.

Figure 6 shows the performance and convergence conditions of the RAV variant compared to the plain variant. In frequency space, the figure shows the family of quasi-halo orbits, the starting solutions, the target frequencies, and the final solution in each branch. Each final solution is colored according to the reason the branch terminated. Select resonance lines are laid on top of the family to help show interactions between the branches and resonances. The RAV variant has 77 branches which converge to solutions with the target set of frequencies, 22 branches which reach the boundary of the family, and 6 branches which are stopped by resonances. Therefore, there are 99 successful branches and 6 unsuccessful branches. The plain variant has 70 convergent branches, 21 which reach the boundary of the family, and 14 that are stopped by resonances, showing the RAV variant performs better at moving past resonances.

For the RAV variant the average time per branch is 98 s, the average number of solutions per branch is 15, and the average time per solution is 6.5 s. For the plain variant the average time per branch is 102.5 s, the average number of solutions per

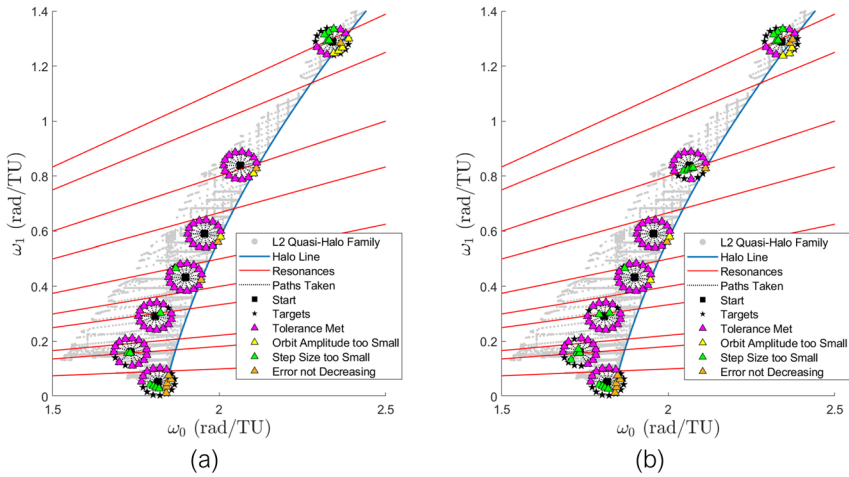


Fig. 6 Comparing the solutions in frequency space between the resonance avoidance variant (a) to the plain variant (b)

Table 1 Performance of RAV and Plain variants

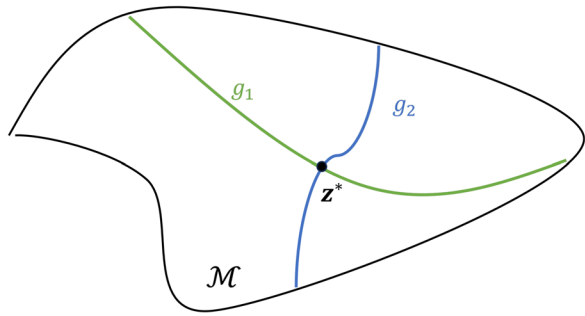
Variant	RAV	Plain
# Hit Converge	77	70
# Hit Boundary	22	21
# Hit Resonance	6	14
Avg. Time per Branch	98 sec	102.5 sec
Avg. # QPOs per Branch	15	17
Avg. Time per QPO	6.5 sec	6 sec

branch is 17, and the average time per solution is 6 s. The results show the RAV variant increases the average time per solution, but decreases the average number of solutions per branch and has an overall decrease in the average time per branch.

7 Targeting Orbital Characteristics

In this section we consider the root-finding problem to find a solution within a 2-parameter family of solutions with desired characteristics. We do this by modifying a Newton’s method to solve for the roots of a function $g : \mathcal{M} \rightarrow \mathbb{R}^t$, for $t = 1, 2$. The function g can be thought of as equality constraints or parametric constraints where each g_i specifies a characteristic we wish the solution to have. Specifying 2 parametric constraints for a 2-dimensional manifold generally leads to a single solution on the manifold (Fig. 7). Other cases include multiple, isolated solutions, no

Fig. 7 Example of two parametric constraints on a 2-dimensional manifold, leading to a single solution z^*



solutions, and infinite solutions. For $t = 1$ there are generally an infinite number of solutions.

A Newton’s method is used to update ω rather than z . The updated frequencies are used with the retraction to compute the solution at the updated frequency values. The reason for updating ω instead of z is Newton’s method is only able to converge on the solution, z^* , satisfying

$$H(z) = \begin{bmatrix} F(z) \\ g(z) \end{bmatrix} = \mathbf{0}$$

if the initial guess, z_0 , is reasonably close to the true solution. We call this the refinement of a solution. We leverage the retraction to remove F , focusing solely on g . The retraction ensures $F(z) = \mathbf{0}$ is satisfied at all times, and uses continuation which allows z_0 to be far from z^* .

The next section presents the algorithm used to target the characteristics of a solution within a 2-parameter family of solutions. We then validate the algorithm with various examples in Sect. 7.2, testing it on the 2-parameter family of quasi-halo orbits.

7.1 Modified Newton’s Method

We wish to find a solution $z^* \in \mathcal{M}$ such that $g(z^*) = \mathbf{0}$. The algorithm to target the characteristics of a solution from within a p -parameter family of solutions is given in Algorithm 7. Given an initial solution $z_0 \in \mathcal{M}$ and a matrix V_0 whose columns span the tangent space $T_{z_0} \mathcal{M}$, the principle tangent basis is computed according to Eq. (30), providing the derivative of z_0 with respect to the frequencies, $\frac{dz}{d\omega}|_{z_0}$. The derivative of g with respect to z is computed and evaluated at z_0 , $\frac{dg}{dz}|_{z_0}$. The Euclidean derivative of g with respect to ω is computed according to Eq. (28) and evaluated at z_0 .

A Newton update is computed to provide a change in frequencies $\delta\omega_1$. If the number of scalar equations composing g is equal to p then the inverse in the Newton update is the usual inverse. If the number of scalar equations is less than p then the inverse is the Moore-Penrose inverse. The change in frequencies dictates a direction and a distance to move in Ω , directly providing the set of frequencies ω_1 of the

solution \mathbf{z}_1 . The updated solution \mathbf{z}_1 and its tangent space V_1 is computed with the retraction in Algorithm 2. The actual updated frequencies are not exactly equal to the guess provided by the Newton step, so $\boldsymbol{\omega}_1$ is taken from the returned solution \mathbf{z}_1 . For brevity, we leave out certain parameters from Algorithm 7 and assume the appropriate parameters are passed into Algorithm 2.

The iterations continue until either $\|\mathbf{g}(\mathbf{z}_k)\| \leq \varepsilon$ or $\|\boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k\| \leq \varepsilon$, where ε is the error tolerance on the characteristics. The first termination condition guarantees the solution has characteristics close enough to the targeted characteristics. The second termination condition indicates convergence to a local minimum of \mathbf{g} .

Algorithm 7 IFT Newton's Method: Find a root of a function defined on an implicit manifold

```

1: procedure IFTNM( $\mathbf{z}_0, V_0, \mathbf{g}, \varepsilon$ )
2:    $k = 0$ 
3:    $(\mathbf{z}_k, V_k) = (\mathbf{z}_0, V_0)$ 
4:   while  $\|\mathbf{g}(\mathbf{z}_k)\| > \varepsilon$  do
5:      $\frac{d\mathbf{z}}{d\boldsymbol{\omega}}|_{\mathbf{z}_k} = V_k \left[ V_{\boldsymbol{\omega}, k}^{-1} \mathbb{I}_p \right]$ 
6:      $\frac{d\mathbf{g}}{d\boldsymbol{\omega}}|_{\mathbf{z}_k} = \left[ \frac{d\mathbf{g}}{d\mathbf{z}} \frac{d\mathbf{z}}{d\boldsymbol{\omega}} \right]_{\mathbf{z}_k}$ 
7:      $\delta\boldsymbol{\omega}_{k+1} = - \left[ \frac{d\mathbf{g}}{d\boldsymbol{\omega}} \right]_{\mathbf{z}_k}^{-1} \mathbf{g}(\mathbf{z}_k)$ 
8:      $(\mathbf{z}_{k+1}, V_{k+1}) = \text{RETRACT}(\delta\boldsymbol{\omega}_{k+1}, \mathbf{z}_k, V_k)$ 
9:     if  $\|\boldsymbol{\omega}_{k+1} - \boldsymbol{\omega}_k\| \leq \varepsilon$  then
10:      return  $(\mathbf{z}_{k+1}, V_{k+1})$ 
11:    end if
12:     $k \leftarrow k + 1$ 
13:  end while
14:   $(\mathbf{z}^*, V^*) = (\mathbf{z}_k, V_k)$ 
15:  return  $(\mathbf{z}^*, V^*)$ 
16: end procedure

```

7.2 Validation

The maximum number of characteristics we can choose an orbit to have is 2. For a single characteristic there are an infinite number of solutions which can be targeted. As long as the two characteristics generate intersecting contours on Ω there is a solution \mathbf{z}^* such that $\mathbf{g}(\mathbf{z}^*) = \mathbf{0}$.

We test the algorithm on six different characteristic functions. The cases, run time, and number of Newton iterations are given in Table 2. Case 1 targets a solution with a specified invariant curve amplitude A_1 (see Lujan and Scheeres [6] for the amplitude computations). Case 2 targets a solution with a specified Jacobi constant. Case 3 targets a solution with a specified amplitude size and Jacobi

Table 2 Test cases for targeting solution characteristics

Case #	$g(z)$	Run Time (min)	# Iterations
1	$A(X) - A_1^*$	3.7	4
2	$\bar{J}(X) - J^*$	2.4	3
3	$\begin{bmatrix} A(X) - A_1^* \\ \bar{J}(X) - J^* \end{bmatrix}$	3.2	4
4	$\begin{bmatrix} A(X) - A_1^* \\ \omega_0 - \omega_0^* \end{bmatrix}$	4.1	4
5	$\lambda_u(z) - \lambda_u^*$	12	3
6	$\begin{bmatrix} \bar{J}(X) - J^* \\ \lambda_u(z) - \lambda_u^* \end{bmatrix}$	19.2	4

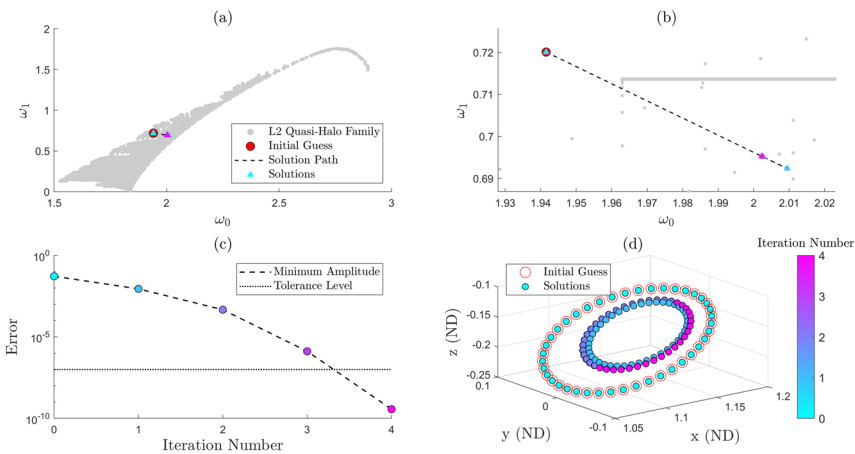


Fig. 8 Results of case 1

constant. Case 4 targets a solution with a specified orbit amplitude and frequency ω_0 . Case 5 targets a solution with a specified value of the unstable eigenvalue. Lastly, case 6 targets a solution with a specified Jacobi constant and unstable eigenvalue. Cases 1, 2, and 5 have an infinite number of solutions, while cases 3, 4, and 6 have a single unique solution. In cases 3, 4, and 6 we choose characteristic values from pre-computed quasi-halo orbits, so there is a truth target orbit to compare to. In all cases we use an error tolerance, ϵ_0 , of $1e-7$ for Algorithm 7 and $1e-8$ for Algorithm 2.

The results of cases 1, 3, and 6 are given in Figs. 8, 9, 10. Plots (a) and (b) show the iterates ω_k . Plot (c) shows the error of each characteristic function versus the iteration number. Plot (d) shows the invariant curve of each solution X_k . Cases 3 and 6 show the truth target invariant curve in plot (d). All cases converge in 3 or 4 iterations. Cases 1 through 4 converge in 4 min or less, while cases 5 and 6 take longer. The increased time of cases 5 and 6 are due to the computation time required to compute the derivative of the unstable eigenvalue with respect to z .

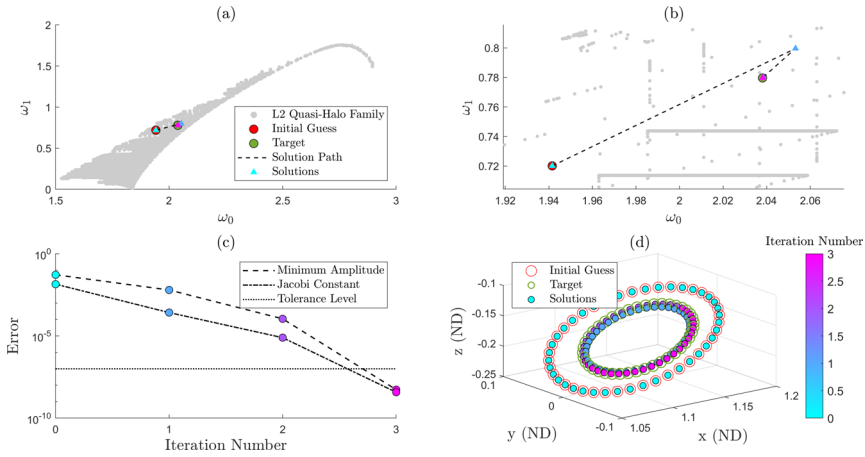


Fig. 9 Results of case 3

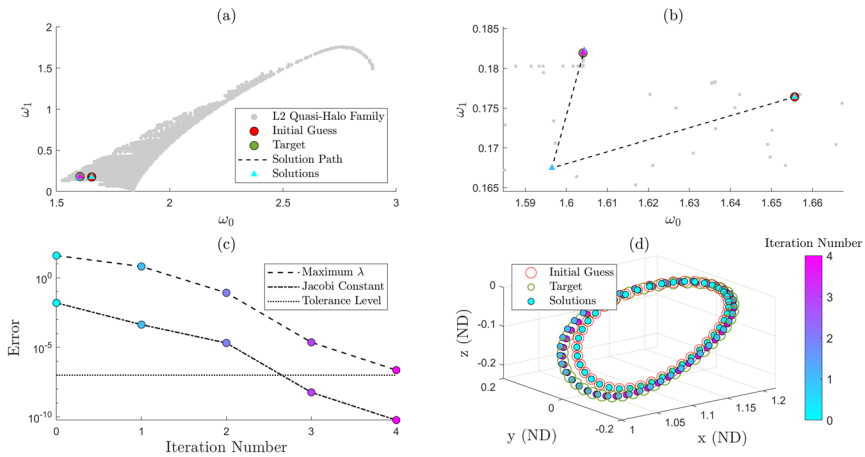


Fig. 10 Results of case 6

There is no analytical means to compute the derivative of the unstable eigenvalue, so we must resort to numerical derivatives. Finite differencing is prohibitively slow, so we use automatic differentiation to perform the numerical derivatives.

Cases 1 through 4 find solutions within the error tolerance. Cases 5 and 6 terminate before the error tolerance is reached because the difference between the current and updated frequencies are within tolerance. Even with the premature termination of the algorithm, the last iteration of case 6 is nearly identical to the truth target solution as seen in plot (d) of Fig. 10. In case 4 the error in ω_0 reached tolerance on the first iteration, and exactly matched the targeted value in

the subsequent iterations. In cases 1, 2, and 5 a solution is still found even though $\frac{dg}{d\omega}$ is not invertible in the usual sense.

Algorithm 7 converges on the target solution quickly even with initial solutions not near the truth solution. Treating ω as the free vector and x as a dependent vector to be found by the retraction shows improved performance over a standard Newton’s method which treats z as the free vector.

8 Optimization

In this section we solve Problem (27) with and without inequality constraints. For the unconstrained problem, we develop a modified gradient descent algorithm developed specifically for the use of the retraction to efficiently perform line searches. For the constrained problem, we use a standard augmented Lagrangian method from [43]. The augmented Lagrangian method constructs a series of unconstrained optimization problems which are solved with the modified gradient descent algorithm. The modified gradient descent is tested on various unconstrained problems, and then the augmented Lagrangian method is used to solve three constrained problems.

8.1 Modified Gradient Descent

The standard gradient descent method assumes all the variables in z are independent optimization variables and optimizes over a Euclidean space. For solutions $z \in \mathcal{M}$ or $z \in \bar{\mathcal{M}}$ only ω are the independent optimization variables, and the space $\bar{\Omega}$ is not Euclidean, requiring manifold derivatives. Equality constraints are absorbed by the retraction, reducing the search space to the only space the optimization is aware of. Therefore, including equality constraints still results in the unconstrained optimization problem

$$\min_{\omega \in \bar{\Omega}} f(x, \omega). \tag{53}$$

We call the modified gradient descent method the IFT gradient descent method because it is applicable to implicitly defined smooth manifolds which satisfy the conditions of the implicit function theorem. The IFT gradient descent is similar to the Euclidean gradient descent except the gradient is taken only with respect to the frequencies ω and the gradient is a manifold gradient when there are equality constraints, g . The IFT gradient descent aims to find a local minimum solution z^* of the function $f : \bar{\mathcal{M}} \rightarrow \mathbb{R}$. Beginning with an initial solution $z_0 \in \bar{\mathcal{M}}$ and \bar{V}_0 we compute the gradient of f with respect to the frequencies ω_0 with Eq. (36). The initial descent direction, d , is given as

$$d = -\nabla_{\omega} \bar{f}(z_0).$$

The descent direction provides the direction to search on $\bar{\Omega}$ to find solutions which reduce the value of the cost function. While the magnitude of d provides a distance to travel, the descent direction only provides local information. Traveling exactly a

distance $d = \|\mathbf{d}\|$ may result in a solution with a cost larger than the initial solution. To ensure the next solution, \mathbf{z}_1 , has a cost less than the initial solution a line search is employed. The descent direction is computed for \mathbf{z}_1 and the process continues until either $\|\nabla_{\omega} \bar{f}(\mathbf{z}_k)\| \leq \varepsilon$ or $\|\mathbf{z}_{k+1} - \mathbf{z}_k\| \leq \varepsilon$.

Ideally, we want the next iterate to be the solution in the direction of \mathbf{d} which has the lowest cost associated with it. That is we want to find the solution which attains the minimum of the set of points

$$L_{\alpha} = \{f(\mathbf{z}) \mid \mathbf{z} = \text{RETRACT}(\gamma \mathbf{d}, \mathbf{z}_k, \bar{V}_k), \gamma \in [0, \alpha]\} \quad (54)$$

for a sufficiently large $\alpha > 0$. Finding the exact optimal solution of L_{α} is computationally expensive since each solution must be found with the retraction function. The retraction uses continuation, so we leverage this to construct L_{α} . A small initial step size, Δs_0 , is chosen so that many solutions are computed near \mathbf{z}_k , giving a fine resolution near \mathbf{z}_k and less resolution further away as the step sizes increases. We don't want to compute too many solutions in the continuation, and we don't want to pass over the optimal solution to Problem 53 when \mathbf{z}_k is near the optimal solution. The cost is computed for each solution in the continuation and the solution with the minimum cost is chosen as the next iterate in the gradient descent algorithm.

Algorithm 8 IFT Gradient Descent: Find minimum of a cost function defined over an implicit manifold

```

1: procedure IFTGD( $f, \mathbf{z}_0, \bar{V}_0, \varepsilon$ )
2:    $k = 0$ 
3:    $(\mathbf{z}_k, \bar{V}_k) = (\mathbf{z}_0, \bar{V}_0)$ 
4:   while  $\|\nabla_{\omega} f(\mathbf{z}_k)\| > \varepsilon$  do
5:      $\mathbf{d} = -\nabla_{\omega} \bar{f}(\mathbf{z}_k)$ 
6:      $(\mathbf{z}_{k+1}, \bar{V}_{k+1}) = \text{LINESEARCH}(f, \mathbf{d}, \mathbf{z}_k, \bar{V}_k)$ 
7:     if  $\|\mathbf{z}_{k+1} - \mathbf{z}_k\| \leq \varepsilon$  then
8:       return  $(\mathbf{z}_{k+1}, \bar{V}_{k+1})$ 
9:     end if
10:     $k \leftarrow k + 1$ 
11:  end while
12:   $(\mathbf{z}^*, \bar{V}^*) = (\mathbf{z}_k, \bar{V}_k)$ 
13:  return  $(\mathbf{z}^*, \bar{V}^*)$ 
14: end procedure

```

8.2 Augmented Lagrangian Method

An ALM constructs a modified cost function called the augmented Lagrangian given by

$$\mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\eta}, \boldsymbol{\mu}) = f(\mathbf{x}) + \frac{\rho}{2} \sum_{i=1}^s \left[\max \left(0, h_i(\mathbf{x}) + \frac{\mu_i}{\rho} \right) \right]^2, \tag{55}$$

where $\rho > 0$ is a penalty parameter and $\boldsymbol{\mu} \in \mathbb{R}_+^s$ are Lagrange multipliers. An ALM solves the unconstrained subproblem

$$\min_{\mathbf{x} \in \mathbb{R}^d} \mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\mu}) \tag{56}$$

successively while updating ρ and $\boldsymbol{\mu}$ after each solve. The method to solve the subproblem can be any method to solve unconstrained optimization problems. We call the method to solve the subproblems the SUBSOLVER routine. We refer the reader to the book by Birgin and Martinez [44] and to Chapter 17 in [45] to learn more about ALMs. For implicitly defined smooth manifolds the unconstrained subproblem becomes

$$\min_{\boldsymbol{\omega} \in \Omega} \mathcal{L}_\rho(\mathbf{z}, \boldsymbol{\mu}). \tag{57}$$

Before introducing the algorithm we present the clip operator defined by

$$\text{clip}_{[a,b]}(x) = \max(a, \min(b, x)), \tag{58}$$

define $\mathcal{I} = 1, \dots, s$ to be the set of indices for the inequality constraints, and let

$$\mathbf{h}(\mathbf{z}) = [h_1(\mathbf{z}), \dots, h_s(\mathbf{z})]^T.$$

The algorithm used to solve the constrained optimization problems is given in Algorithm 9 and pulled from Liu and Boumal in [43], however slight modifications have been made to it.

The variable $\boldsymbol{\mu}_0$ is the initial values of the Lagrange multipliers. We set these initial values to 1e-8. The variables $\boldsymbol{\mu}_{\min}$ and $\boldsymbol{\mu}_{\max}$ are minimum and maximum bounds to the Lagrange multipliers, and set $\boldsymbol{\mu}_{\min} = 1\text{e-}12$ and $\boldsymbol{\mu}_{\max} = 1\text{e+}9$. The variable ρ_0 is an initial penalty parameter, and we set $\rho_0 = 1\text{e-}4$. The multiplier $\theta_\rho > 1$ determines the rate of growth of the penalty parameter, and we set $\theta_\rho = 20$. The variables $\epsilon_0, \epsilon_{\min}$, and $\theta_\epsilon \in (0, 1)$ play a role in determining how many iterations are necessary before convergence can be declared. We set $\epsilon_0 = 1, \epsilon_{\min} = 1\text{e} - 8$, and $\theta_\epsilon = 0.005$. The multiplier θ_σ determines a threshold for the relative amount of movement in the inequality

constraints in order to adjust the penalty parameter. Essentially, if the values of \mathbf{h}_{k+1} do not change much from the previous iteration then the penalty parameter is increased. We set this value to be 0.8. The constant d_{\min} is an error bound. When the distance between two consecutive solutions is less than this amount then it is likely that the algorithm is converging to an optimal solution. However, the convergence decision is balanced with the number of iterations which have satisfied $\|\mathbf{z}_{k+1} - \mathbf{z}_k\| < d_{\min}$, thereby increasing the confidence that the returned solution is actually an optimal solution to Problem (27). We set $d_{\min} = 1e-6$. Finally, N_{\max} limits the number of iterations, so the algorithm cannot continue forever. We set $N_{\max} = 50$.

Algorithm 9 Augmented Lagrangian Method

```

1: procedure ALM( $f, \mathbf{h}, \mathbf{z}_0, \bar{V}_0, \boldsymbol{\mu}_0, \boldsymbol{\mu}_{\min}, \boldsymbol{\mu}_{\max}, \rho_0, \theta_\rho, \varepsilon_0, \varepsilon_{\min}, \theta_\varepsilon,$ 
    $\theta_\sigma, d_{\min}, N_{\max}$ )
2:    $(\mathbf{z}_k, \bar{V}_k) = (\mathbf{z}_0, \bar{V}_0)$ 
3:   for  $k = 0, 1, \dots, N_{\max}$  do
4:      $(\mathbf{z}_{k+1}, \bar{V}_k) = \text{SUBSOLVER}(\mathcal{L}_{\rho_k}(\mathbf{z}, \boldsymbol{\mu}_k), \mathbf{z}_k, \bar{V}_k)$ 
5:     if  $\|\mathbf{z}_{k+1} - \mathbf{z}_k\| < d_{\min}$  and  $\varepsilon_k \leq \varepsilon_{\min}$  then
6:       return  $\mathbf{z}_{k+1}$ 
7:     end if
8:      $\mu_{k+1}^i = \text{clip}_{[\mu_{\min}^i, \mu_{\max}^i]}(\mu_k^i + \rho_k h_i(\mathbf{z}_{k+1}))$ , for  $i \in \mathcal{I}$ 
9:      $\sigma_{k+1}^i = \max\left(h_i(\mathbf{z}_{k+1}), -\frac{\mu_k^i}{\rho_k}\right)$  for  $i \in \mathcal{I}$ 
10:    if  $\|\mathbf{z}_{k+1} - \mathbf{z}_k\| < d_{\min}$  and  $\rho > 1$  then
11:       $\varepsilon_{k+1} = \max(\varepsilon_{\min}, \theta_\varepsilon \varepsilon_k)$ 
12:    end if
13:    if  $k = 0$  or  $\max_{i \in \mathcal{I}}(|\sigma_{k+1}^i|) \leq \theta_\sigma \max_{i \in \mathcal{I}}(|\sigma_k^i|)$  then
14:       $\rho_{k+1} = \rho_k$ 
15:    else
16:       $\rho_{k+1} = \theta_\rho \rho_k$ 
17:    end if
18:  end for
19:   $\mathbf{z}^* = \mathbf{z}_{k+1}$ 
20:  return  $\mathbf{z}^*$ 
21: end procedure

```

After Algorithm 9 returns a solution \mathbf{x}^* we check the gradient of $\mathcal{L}_{\rho^*}(\mathbf{z}^*, \boldsymbol{\mu}^*)$ to determine whether the returned solution is optimal. If $\|\nabla \mathcal{L}_{\rho^*}(\mathbf{z}^*, \boldsymbol{\mu}^*)\| \leq \varepsilon_{\min}$ and each $h_i(\mathbf{z}^*) \leq 0$ for $i \in \mathcal{I}$, then \mathbf{z}^* is declared to be a solution to Problem (27).

Table 3 Test cases for unconstrained optimization

Case #	$f(z)$	$g(z)$	Run Time (min)	# Iterations
1	$\frac{1}{2}(\bar{J}(X) - J^*)^2$	-	29.3	-
2	$\frac{1}{2}(\bar{J}(X) - J^*)^2$	$\omega_0 - \omega_0^*$	4	-

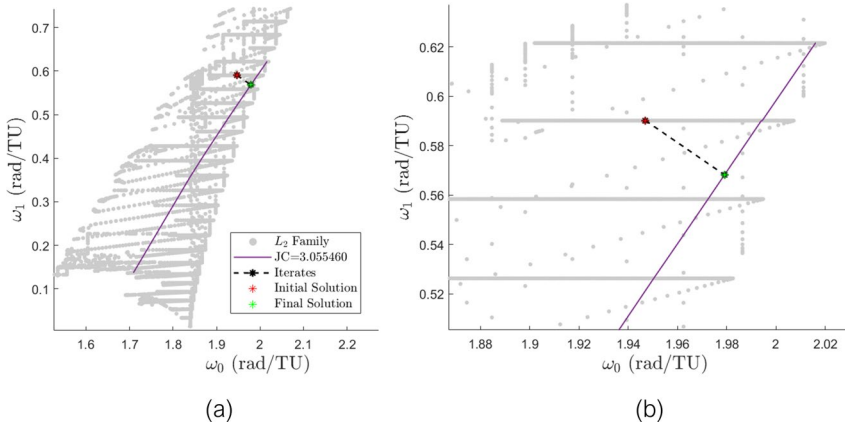


Fig. 11 Results of case 1. Plot **a** is a zoomed out version of plot **b**

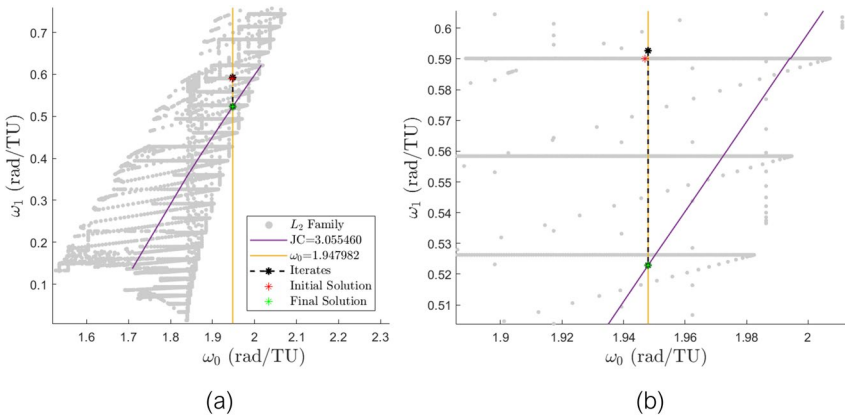


Fig. 12 Results of case 2. Plot **a** is a zoomed out version of plot **b**

8.3 Validation

8.3.1 Unconstrained Optimization

We test Algorithm 8 on the family of quasi-halo orbits to find an orbit which minimizes the distance squared to the desired Jacobi constant value of $J^* = 3.055460$. The first case does not include equality constraints, having two significant effects.

Table 4 Test cases for constrained optimization

Case #	$f(\mathbf{z})$	$\mathbf{g}(\mathbf{z})$	$\mathbf{h}(\mathbf{z})$	Run Time (hr)	# Iterations
1	$\frac{1}{2}(\bar{J}(\mathbf{X}) - J^*)^2$	–	$\omega_1 - \omega_1^*$	2.1	9
2	$\frac{1}{2}(\bar{J}(\mathbf{X}) - J^*)^2$	$\omega_0 - \omega_0^*$	$\omega_1 - \omega_1^*$	0.4	20
3	$\frac{1}{2} \log_{10}(\lambda_u(\mathbf{z}))^2$	–	$\begin{bmatrix} \omega_0 - \omega_0^* \\ \hat{\omega}_0 - \omega_0 \\ \bar{J}(\mathbf{X}) - J^* \\ \hat{J} - \bar{J}(\mathbf{X}) \end{bmatrix}$	24.7	18

First, we only need to take a Euclidean gradient. Second, there are an infinite number of solutions to Problem 53 having a Jacobi constant with a specific value. In the second case we include a constant ω_0 equality constraint, identifying a single solution to Problem 53. The two cases are given in Table 3 along with the run time. Due to the specific implementation of Algorithm 8 we did not record the number of gradient descent iterations needed for convergence.

The results of case 1 and 2 are in Fig. 11 and 12, respectively. In Fig. 11 we see the descent direction lies perpendicular to the Jacobi constant contour, showing the continuation path follows the path of steepest descent. It is not clear why the run time is nearly 30 min, however this time can surely be decreased by tuning the line search. We note the use of Algorithm 7 is likely to be more efficient to find a solution with the desired Jacobi constant.

In case 2 we constrain the optimal solution to have an ω_0 value of 1.947982. The initial solution does not meet this constraint, so Algorithm 7 is first used to find a solution with the desired ω_0 to initialize Algorithm 8. The gradient of f is projected to the tangent space of the 1-dimensional manifold $\bar{\mathcal{M}}$, pointing parallel to the constant ω_0 line. The continuation path follows the constant ω_0 constraint until the gradient vanishes at the optimal solution on the Jacobi constant line. Case 2 has a 7x speed-up over case 1, showing that equality constraints speed up the optimization.

8.3.2 Constrained Optimization

We validate Algorithm 9, using Algorithm 8 in place of SUBSOLVER, with the three test cases. The cases are given in Table 4 along with the run time and the number of iterations for the ALM to converge. In cases 1 and 2 we minimize the distance squared to the desired Jacobi constant value of $J^* = 3.055460$. Case 1 does not have an equality constraint, while case 2 has a constant ω_0 equality constraint. Both cases constrain ω_1 to be less than 0.518223. In case 3 we minimize the square of the logarithm of the unstable eigenvalue and place constraints on the minimum and maximum values of both ω_0 and the Jacobi constant.

In all three cases we know a priori where the optimal solution lies. In case 1, any quasi-halo orbit with the desired Jacobi constant and $\omega_1 \leq 0.518223$ is an optimal solution. In case 2, there is a single optimal solution. The optimal solution is the quasi-halo with orbit frequencies $\omega = (1.947982, 0.518223)$. In case 3, the optimal solution lies on the boundary of the feasible set at the intersection of the lines

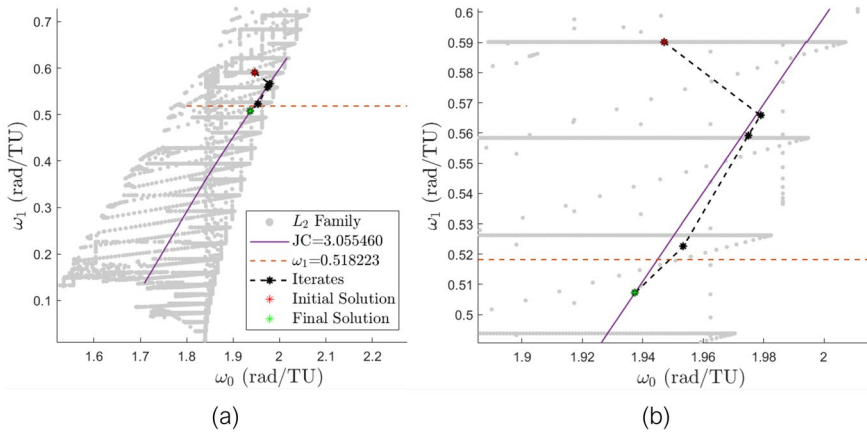


Fig. 13 Orbit frequency iterates to minimize the difference in the Jacobi constant with an ω_1 inequality constraint

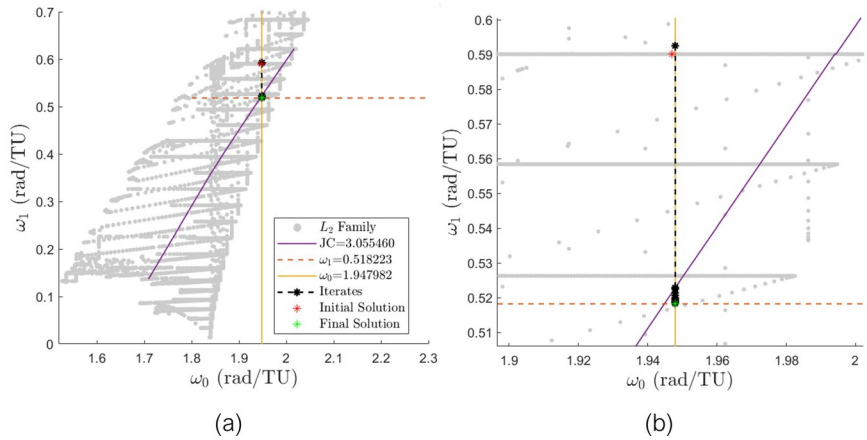


Fig. 14 Orbit frequency iterates to minimize the difference in the Jacobi constant with an ω_0 equality constraint and ω_1 inequality constraint

$$\bar{J}(X) = 3.030083$$

$$\omega_0 = 1.884444.$$

The location of the optimal solution can be inferred from the stability surface plots in Lujan and Scheeres in [5].

The results of case 1 through 3 are in Figs. 13, 14, 15, respectively. The black stars are solutions to each subproblem in the ALM, beginning with the initial solution and ending with the optimal solution. In case 1 the feasible set of solutions lies below the $\omega_1 = 0.518223$ line. In Fig. 13 we see the first iteration of the ALM finds a solution near the desired Jacobi constant because the penalty parameter and Lagrange multipliers are

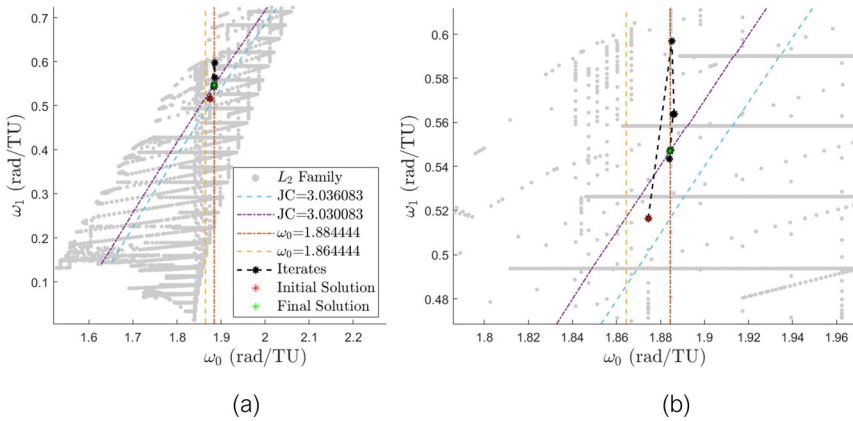


Fig. 15 Orbit frequency iterates to minimize the unstable eigenvalue with Jacobi constant and ω_0 inequality constraints

very small. At the first solution the inequality constraint is not satisfied, so the penalty parameter and Lagrange multipliers are modified to construct a new augmented Lagrangian with a different vector field. As the iterations continue the penalty parameter and Lagrange multipliers continue to be modified until an optimal solution is found.

Case 2 begins with the same initial solution as case 1 and optimizes the same cost function, however case 2 has an equality constraint. The feasible set of solutions is the set of all quasi-halo orbits with $\omega_0 = 1.947982$ and $\omega_1 \leq 0.518223$. In Fig. 14 we see the initial quasi-halo does not satisfy the equality constraint, so Algorithm 7 is used to initialize Algorithm 9. The first iteration of the ALM finds a quasi-halo orbit near the desired Jacobi constant. The remaining iterates are found with ever smaller ω_1 values as the penalty parameter and the Lagrange multipliers are modified until the final solution satisfies the inequality constraint such that $\omega_1 = 0.518223$.

For case 3 we adjust the initial value of the penalty parameter to be equal to 1000 because we know the optimal solution to the unconstrained problem is far from the feasible set of solutions. The feasible set of solutions lies on the interior and boundary of the area encompassed by the four inequality constraints. The time to find the optimal solution is quite high, at 24 h, due to the high cost of evaluating the gradient of the augmented Lagrangian. There are other ways to compute the optimal solution in this situation. We could have used Algorithm 7 to compute the solution with $\omega_0 = 1.884444$ and a Jacobi constant of 3.030083. However, in other cases the location of the optimal solution may not be available. The family may not be known in great detail ahead of time, or the location of the optimal solution may not be easy to determine.

9 Conclusion

We developed tools to perform root-finding, and unconstrained and constrained optimization over families of quasi-periodic orbits by leveraging the smoothness of quasi-periodic orbit families. These tools enable orbits to be accurately chosen

based on orbital characteristics and constraints, and requires little knowledge of the orbit family to function. We developed a modified Newton's method to perform the root-finding of quasi-periodic orbits with specified values of orbital characteristics. This is useful when one knows the exact values of the desired parameters. Taking this a step further, one can construct an organized grid of solutions in various parameter spaces with arbitrary spacing, possibly improving interpolation methods over families of quasi-periodic orbits. Organized grids of solutions serve as efficient orbit databases, and reduce the number of solutions needed to store and represent a family. For cases when one does not know the exact parameter values or wants to include inequality constraints then one can use the optimization tools. By leveraging continuation methods we transform optimization problems with equality constraints to unconstrained optimization problems on submanifolds. For these cases we developed a modified gradient descent algorithm. The inclusion of inequality constraints leads to constrained optimization problems. To solve constrained optimization problems we implemented an augmented Lagrangian method, though other methods to solve constrained optimization problems would suffice.

The tools developed here provide novel tools for trajectory design for multi-parameter families of orbits. To create these tools we first created a retraction, allowing us to take advantage of the manifold structure of families of orbits and reduce the state space to search for solutions. The retraction augments a continuation method to control the direction of continuation and stop at a specific point in the frequency space. Moreover, we develop two methods to reduce the effects of resonances on the computation of quasi-periodic orbits. These methods do not impose restrictions on the continuation and use information about resonances to change the step size. All tools and methods in this paper are tested with the 2-parameter Earth-Moon L_2 quasi-halo orbit family in the circular restricted three-body problem. We would like to point out that much of the mathematical development in this paper can be applied to p -parameter families of solutions satisfying $F(\mathbf{z}) = \mathbf{0}$, not just 2-parameter families of quasi-periodic orbits.

Acknowledgements This research was partially supported by AFOSR grant FA9550-21-1-0332.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no Conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bosanac, N.: Bounded motions near resonant orbits in the earth-moon and sun-earth systems. In: AAS/AIAA Astrodynamics Specialist Conference, Snowbird, UT (2018)
2. Gómez, G., Mondelo, J.M.: The dynamics around the collinear equilibrium points of the RTBP. *Phys. D: Nonlinear Phenom.* **157**(4), 283–321 (2001). [https://doi.org/10.1016/S0167-2789\(01\)00312-8](https://doi.org/10.1016/S0167-2789(01)00312-8)
3. Gómez, G., Marcote, M., Mondelo, J.M.: The invariant manifold structure of the spatial Hill's problem. *Dyn. Syst.* **20**(1), 115–147 (2005). <https://doi.org/10.1080/14689360412331313039>
4. Haro, A., Mondelo, J.M.: Flow map parameterization methods for invariant tori in Hamiltonian systems. *Commun. Nonlinear Sci. Numer. Simul.* **101**, 105859 (2021). <https://doi.org/10.1016/j.cnsns.2021.105859>
5. Lujan, D., Scheeres, D.J.: Earth-Moon L_2 quasi-halo orbit family: characteristics and manifold applications. *J. Guid. Control. Dyn.* **45**(11), 2029–2045 (2022). <https://doi.org/10.2514/1.G006681>
6. Lujan, D., Scheeres, D.J.: Dynamics in the vicinity of the stable halo orbits. *J. Astronaut. Sci.* **70**, 20 (2023). <https://doi.org/10.1007/s40295-023-00379-7>
7. Ming, W., Yang, C., Zhang, H.: Family of resonant quasi-periodic distant retrograde orbits in cislunar space. In: The 28th International Symposium on Space Flight Dynamics, Beijing, China (2022)
8. Jorba, A., Villanueva, J.: The fine geometry of the Cantor families of invariant tori in Hamiltonian systems. In: Casacuberta, C. et al. (eds.) *European Congress of Mathematics*. Birkhäuser, Basel, pp. 557–564 (2001). https://doi.org/10.1007/978-3-0348-8266-8_48
9. Broer, H.W., Huitema, G.B., Sevryuk, M.B.: *Quasi-Periodic Motions in Families of Dynamical Systems*. Springer, Berlin (1996). <https://doi.org/10.1007/978-3-540-49613-7>
10. Kuksin, S., Pöschel, J.: Invariant cantor manifolds of quasi-periodic oscillations for a nonlinear Schrödinger equation. *Ann. Math.* **143**(1), 149 (1996)
11. Whitney, H.: Analytic extensions of differentiable functions defined in closed sets. *Trans. Am. Math. Soc.* **36**(1), 63–89 (1934)
12. Pöschel, J.: Integrability of Hamiltonian systems on cantor sets. *Commun. Pure Appl. Math.* **35**(5), 653–696 (1982)
13. de la Llave, R.: A tutorial on KAM theory. In: Katok, A. et al. (eds.) *Smooth Ergodic Theory and Its Applications*. American Mathematical Society, Providence, RI, pp. 175–292 (2001). <https://doi.org/10.1090/pspum/069>
14. Lee, J.M.: *Introduction to Smooth Manifolds*. Springer, New York (2012). <https://doi.org/10.1007/978-1-4419-9982-5>
15. Boumal, N.: *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, Cambridge (2023). <https://doi.org/10.1017/9781009166164>. <https://www.nicolasboumal.net/book>
16. Aguiar e Oliveira, H.: *Evolutionary Global Optimization Manifolds and Applications*, 1st edn. Springer, Berlin (2015)
17. Ng, J., Dubljevic, S.: Optimal boundary control of a diffusion-convection-reaction PDE model with time-dependent spatial domain: Czochralski crystal growth process. *Chem. Eng. Sci.* **67**(1), 111–119 (2012). <https://doi.org/10.1016/j.ces.2011.06.050>. (Dynamics, Control and Optimization of Energy Systems)
18. Queipo, N., Devarakonda, R., Humphrey, J.A.C.: Genetic algorithms for thermosciences research: application to the optimized cooling of electronic components. *Int. J. Heat Mass Transf.* **37**(6), 893–908 (1994). [https://doi.org/10.1016/0017-9310\(94\)90215-1](https://doi.org/10.1016/0017-9310(94)90215-1)
19. Chakrabarty, S.P., Hanson, F.B.: Optimal control of drug delivery to brain tumors for a distributed parameters model. In: *Proceedings of the 2005, American Control Conference*, pp. 973–978 (2005). <https://doi.org/10.1109/ACC.2005.1470086>
20. Somayaji, M.R., Xenos, M., Zhang, L., Mekarski, M., Linninger, A.A.: Systematic design of drug delivery therapies. *Comput. Chem. Eng.* **32**(1), 89–98 (2008). <https://doi.org/10.1016/j.compchemeng.2007.06.014>. (Process Systems Engineering: Contributions on the State-of-the-Art)
21. Antil, H., Nochetto, R.H., Venegas, P.: Optimizing the Kelvin force in a moving target subdomain. *Math. Models Methods Appl. Sci.* **28**(01), 95–130 (2018). <https://doi.org/10.1142/S0218202518500033>
22. de los Reyes, J.C., Schönlieb, C.B.: Image denoising: learning the noise model via nonsmooth PDE-constrained optimization. *Inverse Problems Imaging* **7**, 1183–1214 (2013)
23. Garreis, S., Surowiec, T.M., Ulbrich, M.: An interior-point approach for solving risk-averse PDE-constrained optimization problems with coherent risk measures. *SIAM J. Optim.* **31**(1), 1–29 (2021). <https://doi.org/10.1137/19M125039X>

24. Antil, H., Leykekhman, D.: *Frontiers in PDE-Constrained Optimization*. Springer, Berlin (2018). <https://doi.org/10.1007/978-1-4939-8636-1>
25. Allaire, G., Jouve, F., Toader, A.M.: A level-set method for shape optimization. *C.R. Math.* **334**(12), 1125–1130 (2002). [https://doi.org/10.1016/S1631-073X\(02\)02412-3](https://doi.org/10.1016/S1631-073X(02)02412-3)
26. Antil, H., Heinkenschloss, M., Hoppe, R.H.W., Linsenmann, C., Wixforth, A.: Reduced order modeling based shape optimization of surface acoustic wave driven microfluidic biochips. *Math. Comput. Simul.* **82**(10), 1986–2003 (2012). <https://doi.org/10.1016/j.matcom.2010.10.027>
27. Jameson, A.: *Aerodynamic shape optimization using the adjoint method*. Lecture Series at the Von Karman Institute, February 6, 2003, Brussels, Belgium (2003). <http://aero-comlab.stanford.edu/publications.html>
28. Kernévez, J.P., Doedel, E.J.: *Optimization in Bifurcation Problems using a Continuation Method*, pp. 153–160. Birkhäuser Basel, Basel (1987). https://doi.org/10.1007/978-3-0348-7241-6_16
29. Wyczalkowski, M., Szeri, A.J.: Optimization of acoustic scattering from dual-frequency driven microbubbles at the difference frequency. *J. Acoust. Soc. Am.* **113**(6), 3073–3079 (2003). <https://doi.org/10.1121/1.1570442>
30. Toilliez, J.O., Szeri, A.J.: Optimized translation of microbubbles driven by acoustic fields. *J. Acoust. Soc. Am.* **123**(4), 1916–1930 (2008). <https://doi.org/10.1121/1.2887413>
31. D’Avino, G., Crescitelli, S., Maffettone, P.L., Grosso, M.: On the choice of the optimal periodic operation for a continuous fermentation process. *Biotechnol. Prog.* **26**(6), 1580–1589 (2010). <https://doi.org/10.1002/btpr.461>
32. Ahsan, Z., Dankowicz, H., Sieber, J.: Optimization along families of periodic and quasiperiodic orbits in dynamical systems with delay. *Nonlinear Dyn.* (2020). <https://doi.org/10.1007/s11071-019-05304-y>
33. Li, M., Dankowicz, H.: Staged construction of adjoints for constrained optimization of integro-differential boundary-value problems. *SIAM J. Appl. Dyn. Syst.* **17**(2), 1117–1151 (2018). <https://doi.org/10.1137/17M1143563>
34. Baresi, N., Olikara, Z.P., Scheeres, D.J.: Fully numerical methods for continuing families of quasi-periodic invariant tori in astrodynamics. *J. Astronaut. Sci.* **65**(2), 157–182 (2018). <https://doi.org/10.1007/s40295-017-0124-6>
35. McCarthy, B.P., Howell, K.C.: Leveraging quasi-periodic orbits for trajectory design in Cislunar space. *Astrodynamics* **5**(2), 139–165 (2021). <https://doi.org/10.1007/s42064-020-0094-5>
36. Olikara, Z.P., Scheeres, D.J.: Numerical method for computing quasi-periodic orbits and their stability in the restricted three-body problem. *Adv. Astronaut. Sci.* **145**, 911–930 (2012)
37. Seydel, R.: *Practical Bifurcation and Stability Analysis*. Springer, New York (2010). <https://doi.org/10.1007/978-1-4419-1740-9>
38. Szebehely, V.: *Theory of Orbits: The Restricted Problem of Three Bodies*. Academic Press, New York and London (1967)
39. Greene, J.M.: A method for determining a stochastic transition. *J. Math. Phys.* **20**(6), 1183–1201 (1979). <https://doi.org/10.1063/1.524170>
40. MacKay, R.S.: A renormalization approach to invariant circles in area-preserving maps. *Phys. D* **7**(1), 283–300 (1983). [https://doi.org/10.1016/0167-2789\(83\)90131-8](https://doi.org/10.1016/0167-2789(83)90131-8)
41. Mondelo, J.M., Barrabés, E., Gómez, G., Ollé, M.: Fast numerical computation of lissajous and quasi-halo libration point trajectories and their invariant manifolds. In: 63rd International Astronautical Congress, Naples, Italy (2012)
42. Sander, E., Meiss, J.D.: Birkhoff averages and rotational invariant circles for area-preserving maps. *Phys. D* **411**, 132569 (2020). <https://doi.org/10.1016/j.physd.2020.132569>
43. Liu, C., Boumal, N.: Simple algorithms for optimization on Riemannian manifolds with constraints. *Appl. Math. Optim.* (2020). <https://doi.org/10.1007/s00245-019-09564-3>
44. Birgin, E.G., Martínez, J.M.: *Practical Augmented Lagrangian Methods for Constrained Optimization*. Society for Industrial and Applied Mathematics, Philadelphia (2014). <https://doi.org/10.1137/1.9781611973365>
45. Nocedal, J., Wright, S.J.: *Numerical Optimization*, 2nd edn. Springer, Berlin (2006). <https://doi.org/10.1007/978-0-387-40065-5>