The Astronomical Journal, 164:179 (16pp), 2022 November https://doi.org/10.3847/1538-3881/ac8d01

celmech: A Python Package for Celestial Mechanics

Sam Hadden[1] and Daniel Tamayo[2,3]
[1] Canadian Institute for Theoretical Astrophysics, 60 St George Street, Toronto, ON M5S 3H8, Canada; hadden@cita.utoronto.ca
[2] Department of Astrophysical Sciences, Princeton University, Princeton, NJ 08544, USA
[3] Physics Department, Harvey Mudd College, Claremont, CA 91711, USA
Received 2022 May 20; revised 2022 August 16; accepted 2022 August 24; published 2022 October 7

Abstract

We present celmech, an open-source Python package designed to facilitate a wide variety of celestial mechanics calculations. The package allows users to formulate and integrate equations of motion incorporating user-specified terms from the classical disturbing function expansion of the interaction potential between pairs of planets. The code can be applied, for example, to isolate the contribution of particular resonances to a system's dynamical evolution and develop simple analytical models with the minimum number of terms required to capture a particular dynamical phenomenon. Equations and expressions can be easily manipulated by leveraging the extensive symbolic mathematics capabilities of the sympy Python package. The celmech package is designed to interface seamlessly with the popular *N*-body code REBOUND to facilitate comparisons between calculation results and direct *N*-body integrations. The code is extensively documented, and numerous example Jupyter notebooks illustrating its use are available online.

Unified Astronomy Thesaurus concepts: Celestial mechanics (211); Perturbation methods (1215); Computational methods (1965)

1. Introduction

Celestial mechanics is the oldest branch of physics, and increasingly sophisticated analytical techniques for predicting planetary motions have been developed over the course of the subject's long history. While precise orbital solutions are now typically found through direct numerical integrations of the equations of motion, analytic methods remain central to solving open problems in dynamics. We are often more interested in developing theoretical insight into the qualitative elements (e.g., mean-motion resonances, hereafter MMRs, and secular evolution) responsible for dynamical phenomena such as chaos, transit timing variations (TTVs), extreme orbital eccentricities, etc. This is especially true when studying exoplanetary systems, where precise mass and orbital determinations are often unavailable, and analytical intuition can guide the construction and interpretation of suites of *N*-body integrations across a relevant subset of the vast parameter space.

The fact that planetary masses are much smaller than the central stellar mass, and that orbital eccentricities and inclinations are typically small, often makes it possible to model planetary dynamics using only a small number of slowly varying terms in the disturbing function expansion of the gravitational interaction potential between planet pairs (a Fourier series in the angular orbital elements and a power series in eccentricities and inclinations). For example, numerous authors have derived analytic models for MMRs in this manner (e.g., Mustill & Wyatt 2011; Batygin & Morbidelli 2013; Batygin 2015; Delisle et al. 2015; Delisle 2017; Hadden 2019; Hadden & Payne 2020). These simplified models can be used to study the process of resonance capture and constrain migration histories of resonant exoplanetary systems since they predict how the resonance capture probabilities and equilibrium configurations reached by systems of migrating planets depend on migration timescales, which are otherwise poorly constrained. Using similar disturbing function expansions to estimate the locations and widths of multiple MMRs, various authors have applied the resonance overlap criterion (e.g., Chirikov 1979) to derive analytic criteria for the onset of chaos in multiplanet systems (e.g., Wisdom 1980; Quillen 2011; Deck et al. 2013; Hadden & Lithwick 2018; Petit et al. 2020; Tamayo et al. 2021; Rath et al. 2022). The classic Laplace–Lagrange solution for the evolution of planets' eccentricities and inclinations is derived by taking the lowest-order secular terms appearing in the disturbing function (e.g., Murray & Dermott 1999). This approximate solution has found myriad applications in the study of exoplanets, from assessing the influence of unseen companions on multitransiting orbital configurations (e.g., Becker & Adams 2017; Read et al. 2017) to analyzing the interplay between orbital precession and the evolution of planetary obliquities in multiplanet systems (e.g., Millholland & Laughlin 2019; Saillenfest et al. 2019; Su & Lai 2022). Disturbing function methods have also be applied to the dynamical modeling of exoplanet transit timing observations, where they can significantly reduce computational costs relative to a purely *N*-body approach while also elucidating important degeneracies that may not otherwise be evident (e.g., Nesvorný & Morbidelli 2008; Lithwick et al. 2012; Agol & Deck 2016; Deck & Agol 2016; Hadden & Lithwick 2016).

The nonexhaustive list of applications given above serves to highlight the importance of an analytic, perturbative approach to studying planetary dynamics. Nevertheless, this long-standing approach can be tedious when formulating even relatively simple models as it entails identifying the appropriate disturbing function terms, finding the expressions for their coefficients, and then evaluating them by means of a variety of special functions. Examining any of the various high-order disturbing function expansions published during the nineteenth century (e.g., Peirce 1849; Le Verrier 1855; Boquet 1889), one quickly

appreciates that formulating any sophisticated, high-order analytic theories by hand represents a monumental undertaking.

Since the sixties, such detailed work has been performed using specialized codes dedicated to computing and manipulating disturbing function expansions (see references in Henrard 1989; Brumberg 1995). The state-of-the-art TRIP code (Gastineau & Laskar 2011, 2021), a dedicated computer algebra system for celestial mechanics, has proven especially valuable for studying the solar system's long-term dynamics (Hoang et al. 2021, 2022; Mogavero & Laskar 2022). However, such analyses have largely been limited to the specialized groups with the expertize to develop these highly technical routines.[4]

By contrast, the last decade has clearly demonstrated the impact open-source scientific software can have on the field (Tollerud et al. 2019), both by opening up specialized calculations to a wide base of users, and by providing application programming interfaces (APIs) that enable the interconnection of separate codes in a broad range of new and creative ways. In particular, successful cases in the exoplanet field include, e.g., ttvfast (Deck et al. 2014), EXOFAST (Eastman et al. 2013), batman (Kreidberg 2015), radvel (Fulton et al. 2018), REBOUND (Rein & Liu 2012), REBOUNDx (Tamayo et al. 2020), exostriker (Trifonov 2019), NbodyGradient (Agol et al. 2021), and exoplanet (Foreman-Mackey et al. 2021).

Ellis & Murray (2000) provided an important early step toward making computer-assisted disturbing function expansions more accessible to the wider research community. They derived a method for calculating coefficients associated with specific Fourier terms in the disturbing function. This offers an efficient alternative to computing complete high-order disturbing function expansions by the means of dedicated computer algebra. Mathematica notebooks implementing their algorithm were made available as part of the online material accompanying the popular textbook Murray & Dermott (1999).

In this paper, we present celmech (Hadden & Tamayo 2022), an open-source Python package, which facilitates the manipulation of disturbing function expansions to any order in the eccentricities and inclinations, and enables a broad range of analytical investigations. celmech is built upon on the disturbing function expansion algorithm derived by Ellis & Murray (2000), although with important modifications described in Section 4 that allow the equations of motion to be formulated in a powerful Hamiltonian framework instead of in orbital elements. We have designed the code's API to easily interface with the REBOUND package, streamlining comparisons between semianalytic celmech models and direct N-body integrations. Additionally, celmech includes functionality for performing canonical transformations, which are useful for both extracting conserved quantities and reducing the dimensionality of various problems (Section 5.3). We also plan to incorporate additional modules in upcoming releases.

The plan of the paper is as follows: We present a simple example application in Section 2 with the goal of introducing readers to the code and illustrating a typical workflow before proceeding through more in-depth discussions of technical details in subsequent sections. Section 3 reviews the coordinate system and Hamiltonian framework that celmech uses to formulate the equations of motion. Section 4 describes celmech's capabilities for performing disturbing function expansions in terms of both classical Keplerian orbital elements (Section 4.1), as well as canonical variables (Section 4.2). Section 5 introduces celmech's object-oriented approach to modeling Hamiltonian systems in general (Section 5.1) as well as those specific to planetary systems (Section 5.2). It also describes celmech's ability to manipulate Hamiltonian models via the application canonical transformations (Section 5.3), including Lie series transformations, a class of transformations that arise in Hamiltonian perturbation theory (Section 5.4). To aid the reader, we provide a list with the main mathematical symbols appearing in the paper along with their definitions in Appendix A.

[4] A limited version of TRIP is available as a binary at www.imcce.fr/Equipes/ASD/trip.

2. A Simple Example

We begin with a practical example to highlight some of the central machinery in celmech, demonstrate the code's API, and illustrate a possible workflow. We consider a system of three Earth-mass planets around a solar-mass star. The inner two are initialized at the 3:2 MMR with orbital periods of 1 and 1.5 yr respectively, while the third planet's orbital period is 3.2 yr and does not form any simple integer ratios with the inner two. The outermost orbit begins with zero eccentricity and inclination to the reference plane. The innermost orbit has both its eccentricity and inclination (in radians) set to 0.02, while the middle orbit has these quantities both set to 0.03. The longitudes of ascending node, and the pericenters of the inner two orbits lie along the positive x-axis, and both planets start at conjunction at their respective apocenters where their orbits are spaced farthest apart (Figure 1). The accompanying Jupyter notebook shows how celmech conveniently interfaces with the REBOUND N-body package, which provides extensive functionality for initializing orbital configurations that can then be converted to a celmech PoincareHamiltonian object. In the present example, we begin by creating a rebound.Simulation object comprised of a star and three planets in the orbital configuration described above, which we name sim. This Simulation instance is then used to initialize a new PoincareHamiltonian object, which we name Hp, as follows:

```
poincare_variables=Poincare.from_Simula-
tion(sim)
Hp=PoincareHamiltonian
(poincare_variables).
```

The poincare_variables object is an instance of the Poincare class, which is used to store the masses and orbital configurations of particles in an N-body system. In this example, the poincare_variables instance inherits the masses and orbital configurations of the particles stored by the rebound.Simulation instance sim. The poincare_variables object is then used to create the PoincareHamiltonian object Hp. The PoincareHamiltonian class, detailed below in Section 5.2, will be used throughout the present example to build and manipulate a Hamiltonian model for the system's dynamics.

2.1. First-order Approximation

The PoincareHamiltonian class has several functions for adding specific terms from the disturbing function between pairs of planets in a system that can be used build up a dynamical model. We expect that the 3:2 MMR between the

inner two planets will be important, so we call the following method of our PoincareHamiltonian object, Hp:

```
Hp.add_MMR_terms(p=3,  q=1,  indexIn=1,
indexOut=2).
```

This adds the lowest-order disturbing function terms in eccentricities and inclinations that are associated with a 3:2 MMR between planets 1 and 2 (the star has index 0) to the Hamiltonian represented by the PoincareHamiltonian object Hp. More generally, the arguments p and q are used to specify terms associated with any p:$p-q$ resonance, and the arguments indexIn and indexOut specify the pair of planets for which terms should be added.

We can check the terms added to the disturbing function terms by inspecting the Hp.df attribute, which prints them with their associated normalization in units of energy (this disambiguation is useful in systems with more than one pair of planets):

$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0),(0,0)}_{(3,-2,0,-1,0,0)}(\alpha_{1,2})e_2\cos(3\lambda_2-2\lambda_1-\varpi_2)$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0),(0,0)}_{(3,-2,-1,0,0,0)}(\alpha_{1,2})e_1\cos(3\lambda_2-2\lambda_1-\varpi_1), \quad (1)$$

with the gravitational constant G, masses m_i, reference semimajor axes $a_{i,0}$ (assumed constant), and $\alpha_{i,j}=a_{i,0}/a_{j,0}$. At first order in eccentricities (e_i) and inclinations (I_i), there are only two disturbing function terms associated with the 3:2 MMR with the resonant angles $3\lambda_2-2\lambda_1-\varpi_1$ and $3\lambda_2-2\lambda_1-\varpi_2$, where the λ_i are the mean longitudes, and ϖ_i are the longitudes of pericenter. These terms would typically be found by hand, e.g., in the Appendix of Murray & Dermott (1999). In the notation[5] of Murray & Dermott (1999), the expression would read

$$\mathrm{MD}: -\frac{Gm_1m_2}{a_{2,0}}[f^{(3)}_{27}(\alpha_{1,2})e_1\cos(3\lambda_2-2\lambda_1-\varpi_1)$$
$$+f^{(3)}_{31}(\alpha_{1,2})e_2\cos(3\lambda_2-2\lambda_1-\varpi_2)]. \quad (2)$$

Our $\tilde{C}$ coefficients correspond to the f coefficients of Murray & Dermott (1999), where we choose to include some additional granularity in our $\tilde{C}$ indices for specifying higher-order terms. The index convention is explained in the following subsection (Section 2.2).

The PoincareHamiltonian object also allows us to integrate the equations of motion using only the terms one has added to the disturbing function. We show the evolution for the middle planet's orbital eccentricity in Figure 1, comparing an N-body integration using REBOUND to our simple celmech model incorporating only the leading-order resonant terms between the inner two planets.

We see that the oscillation amplitude and frequency is reproduced to within ≈10%, although this causes the two models to quickly lose phase coherence. For many applications, this level of agreement is sufficient, but we can improve the agreement by incorporating more terms at higher order in the eccentricities and inclinations.

2.2. Going to Second Order

Even going to just the next (second) order in the eccentricities and inclinations introduces a large number of additional terms. The disturbing function terms required to account for the 3:2 MMR interactions between the inner planet pair and the secular interactions between all planet pairs at second order in eccentricities and inclinations are given by

$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0)}_{(3,-2,0,-1,0,0)}(\alpha_{1,2})e_2\cos(3\lambda_2-2\lambda_1-\varpi_2)$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0)}_{(3,-2,-1,0,0,0)}(\alpha_{1,2})e_1\cos(3\lambda_2-2\lambda_1-\varpi_1)$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0)}_{(6,-4,0,-2,0,0)}(\alpha_{1,2})e_2^2\cos(6\lambda_2-4\lambda_1-2\varpi_2)$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0)}_{(6,-4,-1,-1,0,0)}(\alpha_{1,2})e_1e_2\cos(6\lambda_2-4\lambda_1-\varpi_1-\varpi_2)$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0)}_{(6,-4,-2,0,0,0)}(\alpha_{1,2})e_1^2\cos(6\lambda_2-4\lambda_1-2\varpi_1)$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0)}_{(6,-4,0,0,0,-2)}(\alpha_{1,2})s_2^2\cos(6\lambda_2-4\lambda_1-2\Omega_2)$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0)}_{(6,-4,0,0,-1,-1)}(\alpha_{1,2})s_1s_2\cos(6\lambda_2-4\lambda_1-\Omega_1-\Omega_2)$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0)}_{(6,-4,0,0,-2,0)}(\alpha_{1,2})s_1^2\cos(6\lambda_2-4\lambda_1-2\Omega_1)$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,1)}_{(0,0,0,0,0,0,0)}(\alpha_{1,2})e_2^2-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,1,0)}_{(0,0,0,0,0,0,0)}(\alpha_{1,2})e_1^2$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,1,0,0)}_{(0,0,0,0,0,0,0)}(\alpha_{1,2})s_2^2-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(1,0,0,0)}_{(0,0,0,0,0,0,0)}(\alpha_{1,2})s_1^2$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0)}_{(0,0,-1,1,0,0)}(\alpha_{1,2})e_1e_2\cos(-\varpi_1+\varpi_2)$$
$$-\frac{Gm_1m_2}{a_{2,0}}\tilde{C}^{(0,0,0,0)}_{(0,0,0,0,-1,1)}(\alpha_{1,2})s_1s_2\cos(-\Omega_1+\Omega_2)$$
$$-\frac{Gm_1m_3}{a_{3,0}}\tilde{C}^{(0,0,0,1)}_{(0,0,0,0,0,0,0)}(\alpha_{1,3})e_3^2-\frac{Gm_1m_3}{a_{3,0}}\tilde{C}^{(0,0,1,0)}_{(0,0,0,0,0,0,0)}(\alpha_{1,3})e_1^2$$
$$-\frac{Gm_1m_3}{a_{3,0}}\tilde{C}^{(0,1,0,0)}_{(0,0,0,0,0,0,0)}(\alpha_{1,3})s_3^2-\frac{Gm_1m_3}{a_{3,0}}\tilde{C}^{(1,0,0,0)}_{(0,0,0,0,0,0,0)}(\alpha_{1,3})s_1^2$$
$$-\frac{Gm_1m_3}{a_{3,0}}\tilde{C}^{(0,0,0,0)}_{(0,0,-1,1,0,0)}(\alpha_{1,3})e_1e_3\cos(-\varpi_1+\varpi_3)$$
$$-\frac{Gm_1m_3}{a_{3,0}}\tilde{C}^{(0,0,0,0)}_{(0,0,0,0,-1,1)}(\alpha_{1,3})s_1s_3\cos(-\Omega_1+\Omega_3)$$
$$-\frac{Gm_2m_3}{a_{3,0}}\tilde{C}^{(0,0,0,1)}_{(0,0,0,0,0,0,0)}(\alpha_{2,3})e_3^2-\frac{Gm_2m_3}{a_{3,0}}\tilde{C}^{(0,0,1,0)}_{(0,0,0,0,0,0,0)}(\alpha_{2,3})e_2^2$$
$$-\frac{Gm_2m_3}{a_{3,0}}\tilde{C}^{(0,1,0,0)}_{(0,0,0,0,0,0,0)}(\alpha_{2,3})s_3^2-\frac{Gm_2m_3}{a_{3,0}}\tilde{C}^{(1,0,0,0)}_{(0,0,0,0,0,0,0)}(\alpha_{2,3})s_2^2$$
$$-\frac{Gm_2m_3}{a_{3,0}}\tilde{C}^{(0,0,0,0)}_{(0,0,-1,1,0,0)}(\alpha_{2,3})e_2e_3\cos(-\varpi_2+\varpi_3)$$
$$-\frac{Gm_2m_3}{a_{3,0}}\tilde{C}^{(0,0,0,0)}_{(0,0,0,0,-1,1)}(\alpha_{2,3})s_2s_3\cos(-\Omega_2+\Omega_3), \quad (3)$$

where the inclinations are expressed through $s_i=\sin(I_i/2)$, and the Ω_i are the longitudes of ascending node. We have gone from just 2 terms in Equation (1) to 26 terms in Equation (3).

[5] We modify the notation of Murray & Dermott (1999) slightly by subscripting the inner and outer planet's variables by their indices, and labeling their f_i as $f_i^{(j)}(\alpha)$ to make explicit the coefficients' dependence on the semimajor axis ratio $\alpha_{1,2}=a_1/a_2$ and the coefficient, j, multiplying the outer planet's mean longitude, λ_2, in the cosine argument.

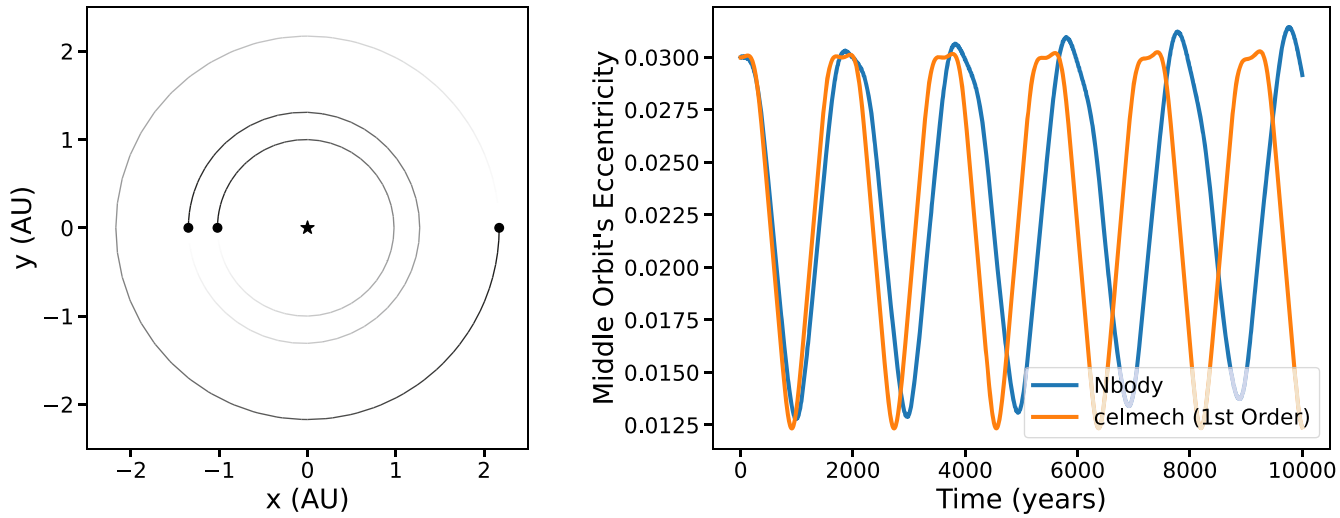

Figure 1. Left panel: initial orbital configuration in the reference $x - y$ plane (inclinations to this plane are small $\lesssim 2°$). Right panel: comparison of the evolution of the middle planet's orbital eccentricity between a celmech model including the leading-order 3:2 MMR terms in the disturbing function between the inner two planets (orange, Equation (1)) to an N-body integration (blue).

While the coefficients of such an expansion would be extremely tedious to look up by hand, these terms are easy to incorporate with celmech.

As can be verified in Equation (3), the $\tilde{C}$ subscript indices specify the coefficients of the cosine arguments

$$(k_1, k_2, k_3, k_4, k_5, k_6) \longleftrightarrow \cos\left(k_1\lambda_{\rm out} + k_2\lambda_{\rm in} + k_3\varpi_{\rm in} + k_4\varpi_{\rm out} + k_5\Omega_{\rm in} + k_6\Omega_{\rm out}\right), \quad (4)$$

where the "in" and "out" subscripts refer to the inner and outer body, respectively, and the ordering of the k coefficients has been chosen to match the ordering conventionally used in the cosine arguments (e.g., Murray & Dermott 1999). The amplitude of each cosine term is a power series in the eccentricities and inclinations. We choose to separate the terms in these expansions, and specify them by the $\tilde{C}$ superscript indices, as detailed in Section 4.2. Most terms in Equation (3) are the leading-order contributions with superscripts (0, 0, 0, 0), although we see examples of higher-order contributions in the secular terms without a cosine argument (all $k_i = 0$ in Equation (4)).

The most important second-order terms for our setup are the second-order 3:2 MMR terms, i.e., the 6:4 cosine terms above, involving the combination $6\lambda_2 - 4\lambda_1$. In celmech, we would incorporate MMR terms up to a specified maximum order by adding the max_order keyword to our call above:

```
Hp.add_MMR_terms(p=3, q=1, max_order=2,
indexIn=1, indexOut=2).
```

The secular terms (terms above that do not depend on the mean longitudes λ, including those with no cosine factors) also make a small but noticeable impact. For each pair of planet indices (idx1, idx2), the secular terms can be added by

```
Hp.add_secular_terms(max_order=2, indexI-
n=idx1, indexOut=idx2).
```

Comparing Figure 1 with the left panel of Figure 2, we see that this second-order model (orange curve) does improve the agreement with N-body.

2.3. Mean Variables

While it is tempting to simply continue adding higher-order terms, in this case it would not help. There is a separate issue causing a discrepancy with the N-body integrations, which can be fixed (at negligible computational cost) by converting from *osculating* to *mean* variables. Any time one takes only a subset of terms in the disturbing function, one is considering a (slightly) different problem from the full one. This distinction is particularly important, given our rather special initial conditions.

Each time an inner planet overtakes its outer neighbor at a conjunction, it gains orbital energy on approach as it is being pulled forward by its outer neighbor, only to lose that energy approximately symmetrically on the way out. These encounters cause short-period *spikes* in the orbits' semimajor axes, and the fact that we started our planets at a conjunction, means that we are starting at the peak of one of those *spikes*. In other words, our initial semimajor axes are, in a sense, spurious, and not reflective of the mean values they have over the vast majority of the orbit. This in turn impacts the period ratio between the inner two planets and their resonant dynamics.

Rather than incorporating even more terms in our disturbing function, the most elegant way of handling these (and other) ignored terms is through a near-identity canonical transformation from "osculating" to "mean" variables using the Lie series method (Morbidelli 2002; our Section 5.4). One can do this in celmech by creating a FirstOrderGeneratingFunction object, and specifying the terms one wants to average out. The terms in the disturbing function causing the semimajor axis spikes discussed above are of the form $\cos\left[k\left(\lambda_2 - \lambda_1\right)\right]$, and can be analytically summed over (Malhotra 1993; our Section 5.4). These terms are independent of the eccentricities and inclinations, and thus of zeroth order, and can be added in celmech starting from a set of poincare_variables specifying the (osculating) initial conditions via

```
chi=FirstOrderGeneratingFunction
(poincare_variables)
chi.add_zeroth_order_term().
```

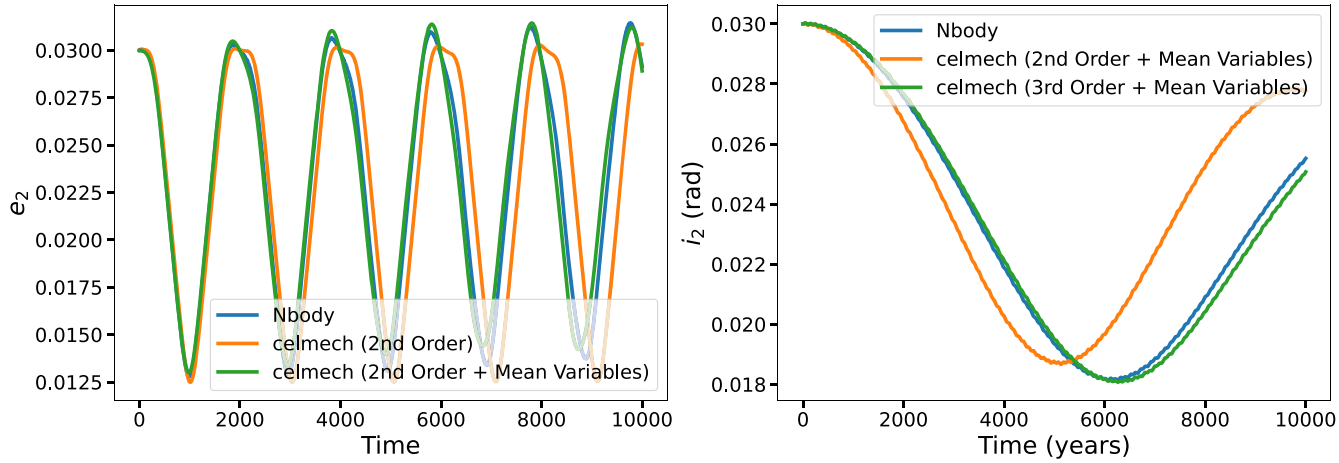

Figure 2. Left panel: comparison of the evolution of the middle orbit's eccentricity in our example between *N*-body (blue) and increasingly sophisticated celmech models. The orange curve incorporates the most important terms to second order in the eccentricities and inclinations, while the green curve additionally incorporates a correction from osculating to mean variables (see text). Right panel: comparison of the evolution of the evolution of the middle orbit's inclination between *N*-body (blue) and a second-order (orange) and third-order (green) celmech model (both corrected to mean variables).

The effects of unmodeled, nearby MMRs are much smaller for this problem than the zeroth-order terms above, but still make a noticeable difference, and are added by calls of the form

```
chi.add_MMR_terms(p=2,q=1,l_max=1,
indexIn=1, indexOut=2)
chi.add_MMR_terms(p=4,q=1,l_max=1,
indexIn=1, indexOut=2).
```

Once we have incorporated all the terms for which we want to correct, we transform to mean variables with chi.osculating_to_mean(), and the resulting integration (green curve in left panel of Figure 2) yields excellent agreement with *N*-body.

2.4. Going to Higher Order

In the right panel of Figure 2, we can see that, even at second order (orange), there is a significant discrepancy with *N*-body (blue) in the orbital inclination evolution of the middle planet. In order to approach the *N*-body evolution, we have to include terms up to third order in the eccentricities and inclinations. This is the order at which the terms directly coupling the inclinations to the quickly varying eccentricities first appear. These are easily obtained by setting max_order = 3 in the calls above, although we spare the reader the details of the resulting 56 terms in the disturbing function. The add_secular_terms method accepts a similar max_order keyword argument, but the higher-order secular terms do not appear until fourth order in eccentricities and inclinations.

2.5. Limits

The degree of agreement between *N*-body and celmech models will depend on planet masses, with smaller masses generally giving better agreement. In particular, the conversion from *osculating* to *mean* variables is not perfect. The Lie series transformation (Section 5.4) corrects the unmodeled terms (e.g., conjunctions, nearby MMRs) to first order in the planetary masses. This generates second-order (and higher) mass terms that should be added to the equations of motion governing the evolution of the *mean* variables. For our low-mass planets, these corrections are negligible, but at higher mass ratios with the central star, these terms start to be significant (Sansottera & Libert 2019). Adding higher-order terms in the eccentricities and inclinations will help until one reaches the level at which these second-order mass terms become significant.

2.6. Applications

The consideration of only the leading-order terms is attractive for both analytical understanding and computational cost. As one incorporates more terms of progressively higher orders, celmech can become slower than direct *N*-body integrations. Nevertheless, there are several applications for high-order models. First, in many instances, it may be possible to deploy integration algorithms that are more efficient than celmech's default general-purpose integrator, and any such algorithms can potentially make use of the lower-level derivative and Jacobian information generated by the PoincareHamiltonian instance (i.e., Hp). This derivative and Jacobian information can also be valuable for locating equilibrium configurations such as periodic orbits via root-finding methods. High-order expansion can also be the starting point for developing accurate integrable approximations of the dynamics through canonical transformations (Hadden 2019). Finally, incremental comparisons of successively higher-order expansions with *N*-body integrations can also provide a quick guide to the degree of complexity required to model a particular dynamical phenomenon of interest.

3. Coordinate System

This section details the coordinate system and dynamical variables the celmech code uses for formulating Hamiltonian equations of motion. We begin by reviewing the Hamiltonian formulation of the planetary *N*-body problem, considering $N-1$ planets of mass m_i with $i = 1,\ldots,N-1$ orbiting a central star of mass M_*. Let $\mathbf{u}_*$ and $\boldsymbol{u}_i$ be the Cartesian position vectors of the star and the planets, respectively, in the inertial barycentric frame. We start by formulating the Hamiltonian in terms of canonical heliocentric coordinates, $(\boldsymbol{r}_i, \tilde{\boldsymbol{r}}_i)$, where $\boldsymbol{r}_i = \boldsymbol{u}_i - \boldsymbol{u}_*$ is the ith planet's heliocentric position vector, and $\tilde{\boldsymbol{r}}_i = m_i\dot{\boldsymbol{u}}_i$ is its canonically conjugate momentum (Laskar & Robutel 1995). In

these coordinates, the Hamiltonian reads

$$H = \sum_{i=1}^{N-1} H_{\mathrm{Kep},i} + \sum_{i=1}^{N-1}\sum_{j=1}^{i-1} H_{\mathrm{int.}}^{(i,j)} \tag{5}$$

where

$$H_{\mathrm{Kep},i} = \frac{1}{2}\frac{|\tilde{\boldsymbol{r}}_i|^2}{\mu_i} - \frac{GM_i\mu_i}{|\boldsymbol{r}_i|}, \tag{6}$$

$$H_{\mathrm{int.}}^{(i,j)} = -\frac{Gm_im_j}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} + \frac{\tilde{\boldsymbol{r}}_i\cdot\tilde{\boldsymbol{r}}_j}{M_*}, \tag{7}$$

$\mu_i = \frac{m_iM_*}{M_*+m_i}$, and $M_i = M_* + m_i$.[6]

We will perform a canonical transformation from the canonical heliocentric coordinates $(\boldsymbol{r}_i, \tilde{\boldsymbol{r}}_i)$ to the canonical modified Delaunay variables that constitute action-angle coordinates of the integrable two-body Hamilontians, $H_{\mathrm{Kep},i}$. In order to effect this transformation, we first define a set of *canonical heliocentric* Keplerian orbital elements, $(a_i, e_i, I_i, \lambda_i, \varpi_i, \Omega_i)$, which represent, respectively, the planetary orbit's semimajor axis, eccentricity, inclination, mean longitude, longitude of periapse, and longitude of ascending node. Standard orbital elements are computed from bodies' three-dimensional Cartesian positions and velocities as well as a "gravitational parameter" (e.g., Murray & Dermott 1999). For each planet, canonical heliocentric elements are computed from the Cartesian position vector $\boldsymbol{r}_i$, the *velocity* vector $\tilde{\boldsymbol{r}}_i/\mu_i$, and the gravitational parameter GM_i. While $\tilde{\boldsymbol{r}}_i/\mu_i$ does not correspond to any physical velocity vector in the system, this definition of canonical heliocentric elements ensures that the modified Delaunay variables, defined below, form a canonical set of action-angle coordinates for the two-body Hamiltonian, $H_{\mathrm{Kep},i}$. The celmech module nbody_simulation_utilities provides the function reb_calculate_orbits to compute canonical heliocentric orbital elements from a REBOUND simulation and the function reb_add_from_elements to add particles to a REBOUND simulation by specifying these elements (Jupyter Notebook example).

The canonical modified Delaunay coordinates[7] are defined in terms of the canonical heliocentric orbital elements. The action variables are

$$\begin{aligned}\Lambda_i &= \mu_i\sqrt{GM_ia_i},\\ \Gamma_i &= \mu_i\sqrt{GM_ia_i}(1-\sqrt{1-e_i^2}),\\ \text{and } Q_i &= 2\mu_i\sqrt{M_ia_i}\sqrt{1-e_i^2}\,\sin^2(I_i/2);\end{aligned} \tag{8}$$

and they have units of angular momentum. The angle variables canonically conjugated to the action variables Λ_i, Γ_i, and Q_i are λ_i, $\gamma_i = -\varpi_i$, and $q_i = -\Omega_i$, respectively. In order to avoid coordinate system singularities that leave the canonical variables γ_i and q_i undefined when $e_i = 0$ and $I_i = 0$, celmech formulates equations of motion in terms canonical coordinate–momentum pairs

$$\begin{aligned}(\eta_i,\,\kappa_i) &= \sqrt{2\Gamma_i}\times(\sin\gamma_i,\,\cos\gamma_i),\\ \text{and } (\rho_i,\,\sigma_i) &= \sqrt{2Q_i}\times(\sin q_i,\,\cos q_i)\end{aligned} \tag{9}$$

rather than (Γ_i, γ_i) and (Q_i, q_i). In modified Delaunay variables, the two-body Hamiltonian, Equation (6), simply becomes

$$H_{\mathrm{Kep},i} = \frac{G^2M_i^2\mu_i^3}{2\Lambda_i^2}. \tag{10}$$

By contrast, $H_{\mathrm{int.}}^{(i,j)}$ does not admit a simple expression in terms of the canonical variables. Rather, the benefit of expressing the Hamiltonian in terms of classical action-angle variables is that $H_{\mathrm{int.}}^{(i,j)}$ can be decomposed as a Fourier series in the canonical angle variables, enabling the application of methods from Hamiltonian perturbation theory.

4. Expansion of the Disturbing Function

The celmech.disturbing_function module provides tools for calculating coefficients appearing in the cosine series expansion of the interaction Hamiltonian, $H_{\mathrm{int.}}^{(i,j)}$, both in terms of both orbital elements as well as canonical variables. We begin by describing the expansion of the disturbing function in terms of orbital elements.

4.1. Expansion in Terms of Orbital Elements

In order to write the cosine series expansion, we will take $a_j > a_i$, define $\alpha_{i,j} = a_i/a_j$, $s_i = \sin(I_i/2)$, $\boldsymbol{\theta}_{i,j} = (\lambda_j,\ \lambda_i,\ -\gamma_i,\ -\gamma_j,\ -q_i,\ -q_j)$, and write

$$\begin{aligned}H_{\mathrm{int}}^{(i,j)} = &-\frac{Gm_im_j}{a_j}\sum_{\boldsymbol{k}}\sum_{\nu_1,\nu_2,\nu_3,\nu_4=0}^{\infty}\tilde{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha_{i,j})\\ &\times s_i^{|k_5|+2\nu_1}s_j^{|k_6|+2\nu_2}e_i^{|k_3|+2\nu_3}e_j^{|k_4|+2\nu_4}\cos(\boldsymbol{k}\cdot\boldsymbol{\theta}_{i,j})\end{aligned} \tag{11}$$

where $\boldsymbol{k} = (k_1,\ k_2,\ldots,k_6)$ and $\boldsymbol{\nu} = (\nu_1,\ldots,\nu_4)$ are integer multi-indices. Rotation and reflection symmetries of the planets' gravitational interactions dictate that $\tilde{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha) \neq 0$ only if $\sum_{l=1}^{6}k_l = 0$ and $k_5 + k_6 = 2n$ where n is an integer (e.g., Murray & Dermott 1999). For small eccentricities and inclinations, the coefficient of the cosine term $\cos(\boldsymbol{k}\cdot\boldsymbol{\theta}_{i,j})$ is of the order $\sim s_i^{|k_5|}s_j^{|k_6|}e_i^{|k_3|}e_j^{|k_4|}$, and the higher-order corrections are of degree 2 and greater in the planets' eccentricities and inclinations.

To compute the coefficients $\tilde{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha)$ appearing in Equation (11), we first represent the interaction Hamiltonian as the sum of two parts, $H_{\mathrm{int}}^{(i,j)} = -\frac{Gm_im_j}{a_j}(\mathcal{R}_{\mathrm{dir.}}^{(i,j)} + \mathcal{R}_{\mathrm{ind.}}^{(i,j)})$ where we define the direct and indirect parts of the disturbing function as[8]

$$\begin{aligned}\mathcal{R}_{\mathrm{dir.}}^{(i,j)} &= \frac{a_j}{|\boldsymbol{r}_j - \boldsymbol{r}_i|},\\ \text{and } \mathcal{R}_{\mathrm{ind.}}^{(i,j)} &= -\frac{a_j}{GM_*m_im_j}\tilde{\boldsymbol{r}}_i\cdot\tilde{\boldsymbol{r}}_j,\end{aligned} \tag{12}$$

[6] Strictly speaking, Hamiltonian (Equation (6)) should also include the term $-\frac{\tilde{\boldsymbol{r}}_0}{M_*}\cdot\sum_{i\neq 0}\tilde{\boldsymbol{r}}_i$ where $\tilde{\boldsymbol{r}}_0$ is the total momentum of the system and is conjugate to the canonical coordinate $\boldsymbol{r}_0 = \boldsymbol{u}_*$ (see, e.g., Equation (27) of Hernandez & Dehnen (2017)). This term is necessary to derive the correct canonical equation of motion for $\boldsymbol{r}_0$ but is not needed to get the correct canonical equations for $\boldsymbol{r}_i$ with $i > 0$ because $\tilde{\boldsymbol{r}}_0 \equiv 0$. The time evolution of $\boldsymbol{r}_0$ can be calculated without explicitly integrating its equation of motion by noting that $\boldsymbol{r}_0 = -\sum_{i=1}^{N-1}m_i\boldsymbol{r}_i/(M_* + \sum_{i=1}^{N-1}m_i)$.

[7] Different literature sources use different names for these canonical variables. Murray & Dermott (1999) refer to these variables as "Poincaré" variables whereas Morbidelli (2002) refers to these variables as modified Delaunay variables and reserves the name "Poincaré" variables for the set of canonical variables introduced in Equation (9).

[8] Note that our definition of the indirect disturbing function, $\mathcal{R}_{\mathrm{ind.}}$, differs from Murray & Dermott (1999) because we formulate our equations of motion in terms of canonical heliocentric variables. Murray & Dermott (1999) formulate equations in terms of heliocentric position and velocity vectors that do not constitute canonically conjugate variable pairs. Consequently, their formulation of the equations of motion requires different indirect terms to be computed depending on whether a planet is subject to an interior or exterior perturber.

and we denote the contributions of $\mathcal{R}^{(i,j)}_{\rm dir.}$ and $\mathcal{R}^{(i,j)}_{\rm ind.}$ to the value of the coefficient $\tilde{C}^{\nu}_{k}(\alpha)$ as $[\tilde{C}^{\nu}_{k}(\alpha)]_{\rm ind.}$ and $[\tilde{C}^{\nu}_{k}(\alpha)]_{\rm ind.}$, respectively. Given the values of $\boldsymbol{k}$ and $\boldsymbol{\nu}$, the celmech code implements the algorithm described in Ellis & Murray (2000) to compute $[\tilde{C}^{\nu}_{k}(\alpha)]_{\rm dir.}$ in terms of Laplace coefficients and their derivatives while the calculation of $[\tilde{C}^{\nu}_{k}(\alpha)]_{\rm ind.}$ is detailed in Appendix B. The celmech.disturbing_function module's df_coefficient_Ctilde function can be used to compute the coefficients $\tilde{C}^{\nu}_{k}(\alpha)$. The function takes two integer tuples, k = (k_1, k_2,…,k_6) and nu = (ν_1,…,ν_4), as arguments and returns a python dictionary representation of the coefficient with key-value pairs in the form {(p_1, (j_1, s_1, n_1)): A_1,…, (p_N, (j_N, s_N, n_N)): A_N, ('indirect', $p_{\rm ind}$): $A_{\rm ind}$} with

$$\tilde{C}^{\nu}_{k}(\alpha) = A_{\rm ind}\,\alpha^{-p_{\rm ind}/2} + \sum_{l=1}^{N} A_l \alpha^{p_l} \frac{d^{n_l}}{d\alpha^{n_l}} b^{(j_l)}_{s_l}(\alpha) \tag{13}$$

where

$$b^{(j)}_{s}(\alpha) = \frac{1}{\pi}\int_{-\pi}^{\pi} \frac{\cos(j\theta)}{(1+\alpha^2-2\alpha\cos\theta)^{s}} d\theta\,,$$

and p_l, n_l, and j_l are integers, and s_l is a half-integer.

The function celmech.disturbing_function module's evaluate_df_coefficient_dict can be used to compute a numerical value of a coefficient, $\tilde{C}^{\nu}_{k}$, for a particular choice of α. The function deriv_df_coefficient computes derivatives of disturbing function coefficients with respect to α. This function takes a dictionary of the form returned by df_coefficient_Ctilde as an argument and returns a new dictionary of the same form representing the derivative of the coefficient with respect to α. Some basic calculations of disturbing function coefficients are demonstrated in the Jupyter example notebook DisturbingFunctionCoefficients.ipynb.

4.2. Expansion in Terms of Canonical Variables

While our $\tilde{C}^{\nu}_{k}$ coefficients for the direct part of the disturbing function match those of Ellis & Murray (2000) derived in terms of orbital elements, formulating Hamiltonian equations of motion requires a disturbing function expansion expressed in terms of canonical variables. To derive an expansion in terms of canonical variables, we first introduce some auxiliary variables: we introduce the reference semimajor axes $a_{i,0}$ about which the expansion of coefficients will be centered along with $\alpha_{ij,0} = a_{i,0}/a_{j,0}$, $\Lambda_{i,0} = \Lambda_i\sqrt{a_{i,0}/a_i}$, and $\delta_i = (\Lambda_i - \Lambda_{i,0})/\Lambda_{i,0}$. Next, we define $X_i = \sqrt{\frac{1}{\Lambda_{i,0}}}(\kappa_i - i\eta_i)$ and $Y_i = \frac{1}{2}\sqrt{\frac{1}{\Lambda_{i,0}}}(\sigma_i - i\rho_i)$, where we use a Roman " i "to denote the imaginary unit, i.e., $\mathrm{i} \equiv \sqrt{-1}$. The complex variables $z_i = e_i e^{i\varpi_i}$ and $\zeta_i = s_i e^{\mathrm{i}\Omega_i}$ can then be expressed explicitly in terms of the variables X_i, Y_i, and δ_i according to

$$z_i = \frac{1}{\sqrt{1+\delta_i}} X_i \sqrt{1 - \frac{1}{4(1+\delta_i)} X_i \bar{X}_i}\,, \tag{14}$$

and

$$\zeta_i = \frac{1}{\sqrt{1+\delta_i}} \frac{Y_i}{\sqrt{1 - \frac{1}{2(1+\delta_i)} X_i \bar{X}_i}} \tag{15}$$

where overlines denote complex conjugates. To express the amplitude of a single disturbing function cosine term, $\cos(\boldsymbol{k}\cdot\boldsymbol{\theta}_{i,j})$, in terms of the canonical variables introduced in Section 3, we seek an expression for coefficients $C^{\nu,\boldsymbol{l}}_{k}$ such that, when k_3, k_4, k_5, and $k_6 > 0$, the following relationship holds:

$$\begin{aligned}\left(\frac{1}{a_j}\right) z_i^{k_3} z_j^{k_4} \zeta_i^{k_5} \zeta_j^{k_6} \textstyle\sum_{\nu=0}^{\infty} \tilde{C}^{\nu}_{k}(\alpha_{ij}) s_i^{2\nu_1} s_j^{2\nu_2} e_i^{2\nu_3} e_j^{2\nu_4} \\ = \left(\frac{1}{a_{j,0}}\right) X_i^{k_3} X_j^{k_4} Y_i^{k_5} Y_j^{k_6} \textstyle\sum_{l=0}^{\infty}\sum_{\nu=0}^{\infty} C^{\nu,\boldsymbol{l}}_{k} \\ \times (\alpha_{ij,0})|Y_i|^{2\nu_1}|Y_j|^{2\nu_2}|X_i|^{2\nu_3}|X_j|^{2\nu_4}\delta_i^{l_1}\delta_j^{l_2}.\end{aligned} \tag{16}$$

When $k_3 < 0$, $z_i^{k_3}$ and $X_i^{k_3}$ in Equation (16) are replaced with $\bar{z}_i^{-k_3}$ and $\bar{X}_i^{-k_3}$, respectively; and analogous substitutions for variables (z_j, ζ_i, ζ_j) and (X_j, Y_i, Y_j) are made when any of k_4, k_5, or $k_6 < 0$. In Appendix C, we work out explicit expressions for the coefficients $C^{\nu,\boldsymbol{l}}_{k}$ in terms of terms of the $\tilde{C}^{\nu}_{k}$ coefficients and their derivatives. While these expressions are, in general, complicated, note that $C^{(0,0,0,0),(0,0)}_{k} = \tilde{C}^{(0,0,0,0)}_{k}$. The disturbing_function module's df_coefficient_C function can be used to obtain disturbing function coefficients $C^{\nu,\boldsymbol{l}}_{k}$.

4.3. Generating Disturbing Function Arguments of a Given Order

In many applications, one is interested in gathering a series of particular disturbing function terms, such as those associated with a particular resonance, up to a maximum order, $N_{\rm max}$, in eccentricities and inclinations. The disturbing_function module provides the function df_arguments_dictionary that allows the user to enumerate all cosine arguments, $\boldsymbol{k}\cdot\boldsymbol{\theta}_{ij}$, appearing in the disturbing function expansion up to a given order $N_{\rm max}$ in inclinations and eccentricities. In order to do so, the function enumerates tuples (k_3, k_4, k_5, k_6) that satisfy $|k_3| + |k_4| + |k_5| + |k_6| \leqslant N_{\rm max}$, along with the condition $k_5 + k_6 = 2n$ for integer n. The latter condition ensures $C^{\boldsymbol{n},\boldsymbol{l}}_{k} \neq 0$ (see Section 4.1). These terms are further grouped by their value of $\sum_{i=3}^{6} k_i$, which must be equal to $-(k_1+k_2)$ to ensure $C^{\boldsymbol{n},\boldsymbol{l}}_{k} \neq 0$. The Jupyter notebook example DisturbingFunctionArguments.ipynb illustrates how the results returned by df_arguments_dictionary are structured.

The disturbing_function module also provides the convenience function list_resonance_terms to create a list of all $\boldsymbol{k}$ and $\boldsymbol{\nu}$ combinations associated with a particular MMR that appear in the expansion of the disturbing function up to a user-specified order. In particular, given the input p and q specifying the p:$p-q$ MMR along with a maximum order, list_resonance_terms returns a python list of tuples in the form ((k_1, k_2,…,k_6), (ν_1,…,ν_4)) that includes all terms that satisfy

$$\begin{gathered}(q-p)k_1 + pk_2 = 0, \text{ and} \\ |k_3| + |k_4| + |k_5| + |k_6| + 2(\nu_1+\nu_2+\nu_3+\nu_4) \leqslant N_{\rm max}\end{gathered} \tag{17}$$

in addition to the usual rules of $\sum_{i=1}^{6} k_i = 0$ and $k_5 + k_6 = 2n$ for integer n. Similarly, the function list_secular_terms can be used to create a list of all $\boldsymbol{k}$ and $\boldsymbol{\nu}$ combinations that appear as secular terms with $k_1 = k_2 = 0$ in the disturbing function expansion up to a user-specified order. Both the list_resonance_terms and list_secular_terms functions are illustrated in a Jupyter notebook example.

4.4. Oblateness and General Relativity

The effects of primary oblateness and general relativistic precession can be important for the dynamical evolution of planetary systems. Here we give expressions for the orbit-averaged potentials of these effects in terms of the canonical variables used by celmech. The gravitational potential due to the quadrupole moment of an oblate central primary at a heliocentric position $\boldsymbol{r}$ is given by

$$\phi_{J_2}(\boldsymbol{r}) = J_2 \frac{GM_* R_*^2}{r^3} P_2(\hat{\boldsymbol{r}} \cdot \hat{\boldsymbol{z}}) \tag{18}$$

where $r = |\boldsymbol{r}|$, $\hat{\boldsymbol{r}} = \boldsymbol{r}/r$, and $\hat{\boldsymbol{z}}$ is the direction perpendicular to the primary's equatorial plane. In terms of orbital elements, $\hat{\boldsymbol{r}} \cdot \hat{\boldsymbol{z}} = \sin(I)\sin(f + \varpi - \Omega) = 2s\sqrt{1-s^2}\sin(f + \varpi - \Omega)$ where f is the true anomaly, and the inclination, I, is measured relative to the primary's equatorial plane. Denoting orbit-averaged values by $<\cdot>$, we use $<r^{-3}> = a^{-3}(1-e^2)^{-3/2}$ and $<r^{-3}\sin^2(f+\varpi-\Omega)> = \frac{1}{2} < r^{-3}>$ to find that the orbit-averaged value of the quadrupole potential is given by $<\phi_{J_2}> = -J_2 \frac{GM_*}{a}\left(\frac{R_*}{a}\right)^2 (1-e^2)^{-3/2}(1/2 + 3(s^4 - s^2))$. The corresponding contribution to the Hamiltonian can be written in terms of canonical variables as

$$H_{J_2}(\Lambda, \eta, \kappa, \rho, \sigma) = -\frac{G^4 M_*^4 \mu^6 J_2 R_*^2}{\Lambda^3\left(\Lambda - \frac{1}{2}(\eta^2+\kappa^2)\right)^3} \times \left[\frac{1}{2} - \frac{3}{4}\left(\frac{\rho^2+\sigma^2}{\Lambda - \frac{1}{2}(\eta^2+\kappa^2)}\right) + \frac{3}{16}\left(\frac{\rho^2+\sigma^2}{\Lambda - \frac{1}{2}(\eta^2+\kappa^2)}\right)^2\right]. \tag{19}$$

Nobili & Roxburgh (1986) describe how the apsidal precession caused by general relativity (GR) can be approximated by including a potential term equal to

$$H_{\rm GR}(\eta, \kappa) = -\int_{-\pi}^{\pi} \frac{3G^2M_*^2}{c^2 r^2} d\lambda = -\frac{3G^2M_*^2}{c^2 a^2\sqrt{1-e^2}} = -\frac{3G^4M^4\mu^4}{\Lambda^3 c^2\left(\Lambda - \frac{1}{2}(\eta^2+\kappa^2)\right)}. \tag{20}$$

Primary oblateness and GR effects can be modeled by adding terms in the form of Equations (19) and (20) to the Hamiltonian represented by a PoincareHamiltonian object, described below in Section 5.2, using the add_orbit_average_J2_terms and add_gr_potential_terms methods.

5. Hamiltonian and Poincare Modules

This section begins in Section 5.1 with a description of how the celmech code represents general Hamiltonian systems. In Section 5.2, we describe the PoincareHamiltonian class, which is capable of representing the Hamiltonian of a planetary system by the inclusion of a collection of disturbing function terms capturing interplanet gravitational interactions. Section 5.3 describes how canonical transformations, which can be applied to modify any Hamiltonian represented by celmech, are implemented. Section 5.4 describes the implementation of the FirstOrderGeneratingFunction class that can be used to apply techniques from canonical perturbation theory to *average* away fast oscillations in the instantaneous, *osculating* orbital elements and transform to *mean* elements that better track the system's long-term evolution.

5.1. Representing Hamiltonian Systems

The celmech.hamiltonian module defines the PhaseSpaceState and Hamiltonian classes, which serve as the basic objects used by celmech to represent Hamiltonian dynamical systems. The Jupyter example notebook SimplePendulum.ipynb provides a basic example that applies these classes to model the motion of a simple pendulum. PhaseSpaceState instances represent points in the phase space of a Hamiltonian dynamical system and store symbolic variables along with numerical values for canonical variables as key-value pairs of an ordered dictionary under the qp attribute. The Hamiltonian class can be used to define a Hamiltonian function of the phase space coordinates and momenta of a particular PhaseSpaceState instance. A Hamiltonian instance is initialized by passing a symbolic expression for the Hamiltonian along with a PhaseSpaceState instance. The PhaseSpaceState will then be stored by the new Hamiltonian object as its state attribute. The symbolic expression for the Hamiltonian should depend on the same canonical variables stored by the PhaseSpaceState instance. The Hamiltonian represented by a Hamiltonian class can also depend on any number of symbolic parameters whose numerical values should be set at initialization by passing a dictionary that pairs parameter symbols with their numerical values. The numerical parameter values are stored as the H_params attribute and can be updated at any point after initializing a Hamiltonian instance to explore how a system's dynamics depends on parameter values.

In addition to storing a symbolic representation of the Hamiltonian function and its associated flow, a Hamiltonian object can be used to integrate the equations of motion and evolve a phase space trajectory in time. In order to do so, a Hamiltonian instance implements the functions flow_func and jacobian_func that take numerical values of the canonical variables as input and return the numerical value of the flow given by Hamilton's equations, and its Jacobian with respect to the canonical variables. These functions are used in conjunction with the scipy package's (Virtanen et al. 2020) integrate.ode class[9] to evolve the phase space state of the system.

5.2. Constructing Hamiltonians for Planetary Systems

The celmech.poincare module provides tools to model planetary systems' dynamics by building a Hamiltonian from the disturbing function expansion described in Section 4. To build these Hamiltonian models, the module defines the Poincare class, a special subclass of the PhaseSpaceState class, and PoincareHamiltonian, a special subclass of the Hamiltonian class. The Poincare class represents the phase space state of an N-body planetary system in terms of the canonically conjugate coordinates $\boldsymbol{q} = (\lambda_1, \eta_1, \rho_1, \ldots, \lambda_{N-1}, \eta_{N-1}, \rho_{N-1})$ and momenta $\boldsymbol{p} = (\Lambda_1, \kappa_1, \sigma_1, \ldots, \Lambda_{N-1}, \kappa_{N-1}, \sigma_{N-1})$. In addition to storing these canonical variables, a Poincare instance owns a collection of PoincareParticle objects through the Poincare.particles attribute. The

[9] See docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.ode.html.

individual members of this collection represent the individual bodies of the N-body system, i.e., the star and planets. Each planet PoincareParticle object records the values of its mass and six canonical variables as well as various quantities, including orbital elements, derived from these values. The API for accessing the mass and orbital-element values of individual PoincareParticle instances is designed to closely mirror that of the REBOUND particle class.

A PoincareHamiltonian is initialized by passing a Poincare object instance, which is then stored as the PoincareHamiltonian's state attribute. Upon initialization, the symbolic Hamiltonian stored by a PoincareHamiltonian object for an N-body planetary system is simply the sum of $N-1$ individual Keplerian Hamiltonians. In other words, the Hamiltonian is given by $H = \sum_{i=1}^{N-1} H_{\mathrm{Kep},i}$ where $H_{\mathrm{Kep},i}$ is given by Equation (10). The user can then add specific interaction terms between the pairs of planets from the disturbing function expansion described in Section 4 using a variety of methods of the PoincareHamiltonian class. The most basic method for adding disturbing function terms is PoincareHamiltonian.add_cosine_term. This method takes the integers $\boldsymbol{k} = (k_1,\ldots,k_6)$ along with the inner and outer planet indices, i and j, as input and adds the terms

$$-\frac{Gm_im_j}{a_{j,0}}\sum_{\boldsymbol{l},\boldsymbol{\nu}} C_{\boldsymbol{k}}^{\boldsymbol{\nu},\boldsymbol{l}}(\alpha_{i,j})|Y_i|^{|k_5|+2\nu_1}|Y_j|^{|k_6|+2\nu_2} \times |X_i|^{|k_3|+2\nu_3}|X_j|^{|k_4|+2\nu_4}\delta_i^{l_i}\delta_j^{l_j}\cos[\boldsymbol{k}\cdot\boldsymbol{\theta}_{ij}] \quad (21)$$

to the Hamiltonian.[10] The combinations of $\boldsymbol{l}$ and $\boldsymbol{\nu}$ occurring in the sum are controlled by the user through a variety of optional keyword arguments. If no keyword arguments are passed, the default behavior is to include only the lowest-order contribution from the disturbing function, i.e., a single term with $\boldsymbol{l} = (0,\ 0)$ and $\boldsymbol{\nu} = (0,\ 0,\ 0,\ 0)$. The PoincareHamiltonian methods add_MMR_terms and add_secular_terms combine this basic add_cosine_term method with the functions for enumerating the particular disturbing function arguments described in Section 4.3 to add collections of resonant or secular terms to the Hamiltonian up to a user-specified order in inclinations and eccentricities.

5.3. Canonical Transformations

The study of Hamiltonian systems is often greatly simplified by canonical transformations from one set of canonical variables to another that leave Hamilton's equations unchanged. This allows for the algorithmic determination of constants of motion that can sometimes be used to greatly reduce the dimensionality of the problem (e.g., Hadden 2019). The celmech CanonicalTransformation class provides a general framework for implementing canonical transformations. Instances of this class allow the user to apply canonical transformations to expressions by applying substitution rules that replace any occurrences of old canonical variables with new ones and vice versa. A CanonicalTransformation instance can also be used to generate new Hamiltonian objects by applying its transformation to an existing Hamiltonian instance's expression for the Hamiltonian, stored as the H attribute. The resulting Hamiltonian object's state attribute will be a new PhaseSpaceState object representing the new canonical variables, and its H attribute will be the transformed Hamiltonian.

Users can create canonical transformation objects by supplying sets of old and new canonical variables along with substitution rules.[11] Users can also use one the following class methods for initializing some frequently used canonical transformations:

1. from_type2_generating_function allows the user to specify a generating function $F_2(\boldsymbol{q}, \boldsymbol{P})$ producing a canonical transformation that satisfies the equations
$$\boldsymbol{Q} = \nabla_{\boldsymbol{P}} F_2\ ;\ \boldsymbol{p} = \nabla_{\boldsymbol{q}} F_2\ . \quad (22)$$
Provided with an expression for the generating function, F_2, this method will attempt to automatically determine the corresponding transformation rules by applying sympy's solve function to express the new variables $(\boldsymbol{Q}, \boldsymbol{P})$ in terms of the old variables $(\boldsymbol{q}, \boldsymbol{p})$ and vice versa.
2. cartesian_to_polar produces a canonical transformation that transforms user-specified conjugate coordinate–momentum pairs of old variables (q_i, p_i) to new conjugate pairs $(Q_i, P_i) = \left(\mathrm{atan2}(q_i, p_i), \frac{1}{2}(q_i^2 + p_i^2)\right)$, where atan2 denotes the two-argument arctangent function that returns a value in the range $(-\pi, \pi]$. All other conjugate pairs will remain unchanged.
3. polar_to_cartesian produces a canonical transformation that transforms user-specified conjugate pairs, (q_i, p_i), to new conjugate coordinate–momentum pairs $(Q_i, P_i) = (\sqrt{2p_i}\sin q_i, \sqrt{2p_i}\cos q_i)$. All other conjugate pairs will remain unchanged.
4. from_linear_angle_transformation produces the canonical transformation $(\boldsymbol{q}, \boldsymbol{p}) \rightarrow (\boldsymbol{Q}, \boldsymbol{P})$ given by
$$\boldsymbol{Q} = T\cdot\boldsymbol{q}\ ;\ \boldsymbol{P} = (T^{-1})^{\mathrm{T}}\cdot\boldsymbol{p} \quad (23)$$
where T is a user-supplied, invertible matrix.
5. from_poincare_angles_matrix takes a Poincare instance as input, along with an invertible matrix T, and produces a transformation where the new canonical coordinates are linear combinations of the planets' angular orbital elements given by $\boldsymbol{Q} = T\cdot(\lambda_1,\ldots,\lambda_N,\ -\varpi_1,\ldots,\ -\varpi_N,\ -\Omega_1,\ldots,\ -\Omega_N)$, and the corresponding new conjugate momenta are given in terms of the canonical modified Delaunay variables described in Section 3 by $\boldsymbol{P} = (T^{-1})^{\mathrm{T}}\cdot(\Lambda_1,\ldots,\Lambda_N,\ \Gamma_1,\ldots,\Gamma_N,\ Q_1,\ldots,Q_N)$.
6. Lambdas_to_delta_Lambdas takes a Poincare instance as input and implements the canonical transformation replacing the Λ_i variables with $\delta\Lambda_i = \Lambda_i - \Lambda_{i,0}$ as the momenta canonically conjugate to the planets' mean longitudes, λ_i.

[10] The reference semimajor axes a_{i0} occurring in Equation (21), both explicitly as well as implicitly via the variables α_{ij} and δ_i, are stored as parameters of the PoincareHamiltonian object as parameters in the H_params attribute. The reference semimajor axis values are set to the instantaneous semimajor axes of the particles at the time when the PoincareHamiltonian is initialized. Users should be aware that altering the values of the a_{i0} parameters stored by a PoincareHamiltonian object will *not* automatically update the values of the $\Lambda_{i,0}$ or the disturbing function coefficient parameters stored by a PoincareHamiltonian object. The simplest way to update all parameters depending on the reference semimajor axes in a self-consistent fashion is initialize a new PoincareHamiltonian object from a set of particles possessing instantaneous semimajor axes equal to the desired reference values.

[11] It is the user's responsibility to ensure that the supplied rules actually constitute a canonical change of variables, but the method test_canonical() can be used to test whether the supplied rules constitute a canonical transformation.

7. rescale_transformation produces a canonical transformation that simultaneously rescales the Hamiltonian and all canonical momentum variables by a common factor, f. In other words, if H is the old Hamiltonian and $_{pi}$ are the old momentum variables, the new Hamiltonian will be $H' = fH$, and $p_i' = fp_i$ will replace $_{pi}$ as momentum variables conjugate to the old coordinates q_i. The user can optionally specify a subset of canonical variable pairs (y_i, x_i) as Cartesian pairs, which will be transformed so that the new, rescaled variables are instead $(y'_i, x'_i) = (\sqrt{f}y_i, \sqrt{f}x_i)$.
8. composite combines a series of canonical transformations.

While time-dependent canonical transformations are not currently supported, they can always be implemented by reformulating the Hamiltonian under consideration in the extended phase space that includes time and the negative of the system's energy as an additional conjugate variable pair.

Often canonical transformations are applied to eliminate the explicit dependence of the transformed Hamiltonian on one or more canonical variable. This is because, when the transformed Hamiltonian does not depend on one of the angle variables, the corresponding conjugate momentum variable is conserved, and Hamilton's equations for the remaining degrees of freedom are simplified. To make the discussion more concrete, suppose we have a canonical transformation of the canonical variables of an N degree of freedom system $(q_i, _{pi}) \rightarrow (\phi_1,\ldots,\phi_M, Q_1, \ldots,Q_{N-M}, I_1,\ldots,I_M, P_1,\ldots,P_{N-M})$ where the transformed Hamiltonian, $H'(Q_i, P_i; I_j)$, is independent of the coordinate variables ϕ_j. so that the quantities $I_1,\ldots,I_M$ are constant. When such an elimination can be accomplished, it is frequently convenient to consider a system's dynamics in the reduced phase space comprised of the remaining $N-M$ degrees of freedom with canonical variables $(Q_i, _{Pi})$ and treat the M conserved quantities, I_j, as parameters of the Hamiltonian. When transforming a Hamiltonian by applying the old_to_new_hamiltonian method of a CanonicalTransformation object, the user has the option to automatically detect whether the resulting transformed Hamiltonian is independent of any of the new canonical variables and, if so, produce a Hamiltonian object with a state attribute representing a point in the reduced phase space with canonical variables $(Q_i, _{Pi})$ and with the quantities I_j stored as parameters under the Hamiltonian object's H_params attribute. To enable the application of the inverse transformation from the new canonical variables back to old ones in this situation, the new Hamiltonian object possesses a full_qp attribute that stores the full set of variables, $(\phi_1,\ldots,\phi_M, Q_1, \ldots,Q_{N-M}, I_1,\ldots,I_M, P_1,\ldots,P_{N-M})$, while the qp attribute[12] only stores the variables of the reduced phase space, $(Q_i, _{Pi})$. The reduced Hamiltonian object's full_qp attribute will only store the initial values of the ignorable coordinates, ϕ_j, determined at the time that the transformation was applied. It is important to recognize that the actual time evolution of these coordinates, given by $\frac{d}{dt}\phi_j = \frac{\partial}{\partial I_j}H'(Q_i, P_i; I_j)$, is generally nontrivial and depends on the specific trajectory, $(Q_i(t), _{Pi}(t))$, of the system in the reduced phase space. Consequently, it will usually not be possible to reconstruct the full phase space state of the system in the old canonical variables $(q_i, _{pi})$ if only the reduced equations of motion governing the $(Q_i, _{Pi})$ are integrated. Nonetheless, in many applications, the values of these ignorable coordinates are unimportant for the aspects of the dynamics one wishes to study, and the application of the inverse transformation to the data stored in the full_qp attribute can provide useful information about how the system evolves in the original canonical variables. A basic illustration of these principles is provided in an example Jupyter notebook[13].

5.4. Lie Series Transformations

Modeling the dynamics of a planetary system with a Hamiltonian that includes only a finite number of terms from a disturbing function can be justified because the infinite number of ignored terms are rapidly oscillating and have negligible average influence on the motion, at least over sufficiently short timescales. Nevertheless, these fast oscillations in the full problem can make it challenging to match the initial conditions in an *averaged* formulation, and such discrepancies can make comparisons between analytical and N-body models challenging, especially near regions of phase space where the dynamical behavior changes sharply, e.g., near resonance boundaries. Additionally, while typically we are interested in methods for *removing* these nuisance terms in the disturbing function, there are also cases where these fast oscillations are of primary interest. An important example is for transiting pairs of planets near (but not in) MMR, where the "fast" oscillations due to the resonant terms are observable as TTVs (for a Lie series approach to TTVs, see Nesvorný & Morbidelli 2008).

A powerful approach for transforming back and forth between the instantaneous, or *osculating*, canonical variables (which are subject to a multitude of fast oscillations) and the *mean* variables that remove these fast modes is through the Lie series method. This technique introduces a near-identity canonical transformation (ensuring that the *mean* variables remain canonical) that, by construction, can cancel any set of rapidly oscillating terms in the Hamiltonian to first order in the planet–star mass ratio. This transformation is often referred to as *averaging* since, at this leading order in the masses, the technique is equivalent to averaging the problem over the rapidly oscillating angles. An advantage of the Lie series method is that it explicitly constructs the resultant terms at higher orders in the masses, which are not available through an averaging approach. A thorough discussion of canonical perturbation theory using the Lie series method, including important issues related to chaos, nonintegrability, and the (non)convergence of these series, is beyond the scope of this paper and can be found in references such as Morbidelli (2002) or Lichtenberg & Lieberman (2013). We will limit our discussion to the construction of canonical transformations to first order in the planet–star mass ratios, which are equivalent to transformations back and forth between osculating and mean variables, and are implemented in celmech.

To introduce Lie-series generating functions, consider splitting the Hamiltonian of a N-body system into three pieces: a Keplerian piece given by $H_{\rm Kep} = \sum_{i=1}^{N-1} H_{{\rm Kep},i}$, a term $H_{\rm slow}$ containing slowly varying disturbing function terms such as those with (near) resonant cosine arguments or secular terms, and a term $H_{\rm osc}$ that contains rapidly oscillating terms with

[12] The qp attribute of a Hamiltonian object is actually just a shortcut for Hamiltonian.state.qp where Hamiltonian.state is the PhaseSpaceState object representing the state of the system. See Section 5.1.

[13] https://github.com/shadden/celmech/blob/master/jupyter_examples/BasicCanonicalTransformation.ipynb

negligible influence on the dynamics. We seek a canonical transformation that eliminates $H_{\rm osc}$ to first order in planet masses. In other words, we seek a canonical transformation T: $(q, \boldsymbol{p}) \rightarrow (\boldsymbol{q}', \boldsymbol{p}')$ such that the transformed Hamiltonian is $H' \equiv H \circ T^{-1} = H_{\rm Kep} + H_{\rm slow} + \mathcal{O}((m/M_*)^2)$. We use the Lie series method to construct such a transformation. The Lie transformation, $\exp[\mathcal{L}_\chi]$, of a function of phase space coordinates, f, is defined in terms of the Lie derivative operator, $\mathcal{L}_\chi \equiv [\cdot, \chi]$, as

$$\exp[\mathcal{L}_\chi] f = \sum_{n=0}^{\infty} \frac{1}{n!} \mathcal{L}_\chi^n f = f + [f, \chi] + \frac{1}{2}[[f, \chi], \chi] + \ldots . \tag{24}$$

Since a Lie transformation evolves f under the flow induced by Hamilton's equations for the "Hamiltonian" χ, it is canonical by construction. The Lie series method seeks to find a generating function χ such that the Lie transformation $\exp[\mathcal{L}_\chi]$ produces the desired transformation, T, that eliminates the unwanted terms. If the generating function, χ, is itself of order $\mathcal{O}(m/M_*)$, then the transformed Hamiltonian is $H' = \exp[\mathcal{L}_\chi] H \approx H_{\rm Kep} + H_{\rm slow} + H_{\rm osc} + [H_{\rm Kep}, \chi]$ after dropping terms $\mathcal{O}((m/M_*)^2)$, and the generating function χ should satisfy $[H_{\rm Kep}, \chi] = -H_{\rm osc}$. Noting that the Poisson bracket of $H_{\rm Kep}$ with all canonical variables except the mean longitudes vanishes and that

$$\left[H_{\rm Kep}, \frac{\sin(\boldsymbol{k} \cdot \boldsymbol{\theta}_{i,j})}{k_1 n_j + k_2 n_i} \right] = -\cos(\boldsymbol{k} \cdot \boldsymbol{\theta}_{i,j}) \tag{25}$$

where $n_i = \partial H_{\rm Kep} / \partial \Lambda_i$ is the ith planet's mean motion, we immediately see that any disturbing function terms in $H_{\rm osc}$ involving planets i and j written in the form of Equation (21) will be eliminated to first order in the planet masses by simply including a corresponding term in χ that replaces $\cos(\boldsymbol{k} \cdot \boldsymbol{\theta}_{i,j})$ with $\sin(\boldsymbol{k} \cdot \boldsymbol{\theta}_{i,j})/(k_1 n_j + k_2 n_i)$ provided $k_1 n_j + k_2 n_i \neq 0$. The condition that $k_1 n_j + k_2 n_i \neq 0$ should always be satisfied since $H_{\rm osc}$ is presumed to be comprised only of rapidly oscillating terms.

The celmech class FirstOrderGeneratingFunction allows users to construct a generating function χ that eliminates a user-specified collection of disturbing function terms in the transformed Hamiltonian to first order in planet masses following the approach just described above. Terms are added to the generating function in much the same way that terms are added to a PoincareHamiltonian instance. In fact, FirstOrderGeneratingFunction is a subclass of the PoincareHamiltonian class that overwrites the parent class's routines for adding cosine terms so that the trigonometric terms $\cos(\boldsymbol{k} \cdot \boldsymbol{\theta}_{i,j})$ are instead replaced with $\sin(\boldsymbol{k} \cdot \boldsymbol{\theta}_{i,j})/(k_1 n_j + k_2 n_i)$. The class provides a symbolic representations of the generating function under the chi and N_chi attributes, similar to the PoincareHamiltonian class's H and N_H attributes. The class also provides an array of methods for transforming canonical variables and deriving perturbative solutions to the equations of motion. The Jupyter example notebook FirstOrderGeneratingFunction.ipynb illustrates the basic application of the FirstOrderGeneratingFunction to more accurately predict the secular evolution of the mean eccentricities of a pair of planets near a 3:2 MMR. The Jupyter example notebook TransitTimingVariations.ipynb demonstrates the application of Lie series transformations to analytically predicting TTVs.

In addition to the term-wise elimination of individual rapidly oscillating disturbing function terms described above, it is possible to remove all harmonics depending solely on the angular combination $\psi = \lambda_i - \lambda_j$ to zeroth order in eccentricities and inclinations with a single transformation. A number of authors have previously exploited the fact that perturbative solutions for planets' mutual perturbation at zeroth order in eccentricities and inclinations can be expressed in a closed form without needing to resort to an expansion in Fourier harmonics (e.g., Malhotra 1993; Agol et al. 2005; Nesvorný & Vokrouhlický 2014; Hadden et al. 2019), although not all these studies deploy a Lie series formalism. To zeroth order in inclinations and eccentricities, the disturbing function, $\mathcal{R}_{\rm dir.}^{(i,j)} + \mathcal{R}_{\rm ind.}^{(i,j)}$, given in Equation (12) can be written as

$$\mathcal{R}_{\rm dir.}^{(i,j)} + \mathcal{R}_{\rm ind.}^{(i,j)} \approx P(\lambda_j - \lambda_i; \alpha) - \frac{1}{\sqrt{\alpha}} \cos(\lambda_i - \lambda_j) + \mathcal{O}(e, I^2) \tag{26}$$

where $P(\psi; \alpha) = (1 + \alpha^2 - 2\alpha \cos(\lambda_j - \lambda_i))^{-1/2}$. Therefore, a Lie series transformation with generating function

$$\chi = -\frac{G m_1 m_2}{a_2 (n_1 - n_2)} \left(-\frac{1}{\sqrt{\alpha}} \sin(\lambda_i - \lambda_j) + \int_0^{\lambda_i - \lambda_j} (P(\psi, \alpha) - \bar{P}(\alpha)) d\psi \right), \tag{27}$$

where $\bar{P}(\alpha) = \frac{1}{2\pi} \int_{-\pi}^{\pi} P(\psi, \alpha) d\psi$ is the mean value of $P(\psi, \alpha)$, will eliminate the disturbing function terms in Equation (26). The integral in Equation (27) can be written explicitly in terms of elliptic integrals as

$$\int_0^{\psi} (P(\psi', \alpha) - \bar{P}(\alpha)) d\psi' = \frac{2}{1 - \alpha} F\left(\frac{\psi}{2} \middle| -\frac{4\alpha}{(1 - \alpha)^2} \right) - \frac{2}{\pi} \mathbb{K}(\alpha^2) \psi \tag{28}$$

where $\mathbb{K}$ is the complete elliptic integral of the first kind, and $F(\cdot | m)$ is an incomplete elliptic integral of the first kind with modulus m. A term of the form given in Equation (27) can be added to the symbolic generating function stored by a FirstOrderGeneratingFunction object with the add_zeroth_order_term method.

6. Summary

In this paper, we presented celmech, an open-source python package for conducting celestial mechanics calculations. The code combines a disturbing function expansion algorithm based on a method originally devised by Ellis & Murray (2000) with the symbolic mathematics capabilities provided by the sympy library (Meurer et al. 2017) to enable users to efficiently create and integrate Hamiltonian models for planetary systems. The celmech API is designed so that users can easily compare these Hamiltonian models to direct N-body integrations with the REBOUND code (Rein & Liu 2012).

The code is available through the Python Package Index as well as on GitHub at github.com/shadden/celmech. The code can can be freely redistributed under the GPL-3.0 license. Online documentation can be found at celmech.readthedocs.io.

A library of Jupyter notebook examples illustrating various features and applications of the code is available at github.com/shadden/celmech/tree/master/jupyter_examples.

The celmech code's modular design will allow ongoing development. Some features currently implemented in celmech, in addition to the modules and capabilities detailed in this paper, include tools for formulating and integrating the equations of motion governing MMRs that are derived from numerically averaged disturbing functions (rather than truncated power series expansions in eccentricity and inclination) and tools for formulating and integrating secular equations of motion. We plan to detail the additional features in forthcoming papers. Community contributions to the celmech code are also welcome and can be submitted using the GitHub pull request feature online at github.com/shadden/celmech/pulls.

We thank the anonymous referee whose comments helped improve this manuscript. We thank David Hernandez for helpful discussions during the development of the celmech code. Implementations of some celmech routines are based on the Mathematica package by Fabio Zugno available at library.wolfram.com/infocenter/MathSource/4256/. S.H. acknowledges support by the Natural Sciences and Engineering Research Council of Canada (NSERC), funding reference CITA 490888-16. D.T. is grateful for support from the Lyman Spitzer Jr. fellowship.

Software: matplotlib (Hunter 2007), NumPy (Oliphant 2006), REBOUND (Rein & Liu 2012), REBOUNDx (Tamayo et al. 2020), scipy (Jones et al. 2001), sympy (Meurer et al. 2017).

Appendix A
Table of Symbols

Table A1 lists the main mathematical symbols used in the text along with brief descriptions as appropriate.

Table A1
Summary of Mathematical Notation Used Throughout the Paper: We Include Short Definitions When Appropriate

Name	Expression	Description
G		Newton's gravitational constant
M_*		Mass of central star
m_i		Mass of ith planet
$\boldsymbol{r}_i$		Heliocentric position of ith planet
$\tilde{\boldsymbol{r}}_i$		Barycentric momentum of ith planet
a_i		Semimajor axis of ith planet
e_i		Eccentricity of ith planet
I_i		Inclination of ith planet
λ_i		Mean longitude of ith planet
ϖ_i		Longitude of periapse ith planet
Ω_i		Longitude of ascending node ith planet
s_i	$\sin(I_i/2)$	Inclination variable appearing in many classic disturbing function expansions
μ_i	$m_i M_*/(M_* + m_i)$	Reduced mass of ith planet
M_i	$M_* + m_i$	Effective central mass for ith planet's orbit
Λ_i	$\mu_i\sqrt{GM_i a_i}$	Canonical momentum conjugate to λ_i
Γ_i	$\Lambda_i(1-\sqrt{1-e_i^2})$	Canonical momentum conjugate to $\gamma_i = -\varpi_i$
Q_i	$=2\Lambda_i\sqrt{1-e_i^2}\sin^2(I_i/2)$	Canonical momentum conjugate to $q_i = -\Omega_i$
κ_i	$=\sqrt{2\Gamma_i}\cos\varpi_i$	
η_i	$=-\sqrt{2\Gamma_i}\sin\varpi_i$	
σ_i	$=\sqrt{2Q_i}\cos\Omega_i$	
ρ_i	$=-\sqrt{2Q_i}\sin\Omega_i$	
$\mathcal{R}^{(i,j)}_{\text{dir.}}$	$=\frac{a_j}{\lvert \boldsymbol{r}_j-\boldsymbol{r}_i\rvert}$	Direct component of disturbing function
$\mathcal{R}^{(i,j)}_{\text{ind.}}$	$=-\frac{a_j}{GM_* m_j m_i}\tilde{\boldsymbol{r}}_j\cdot\tilde{\boldsymbol{r}}_i$	Indirect component of disturbing function
$\alpha_{i,j}$	$=a_i/a_j$	Semimajor axis ratio
$\boldsymbol{\theta}_{i,j}$	$=(\lambda_j, \lambda_i, \varpi_i, \varpi_j, \Omega_i, \Omega_j)$	Vector of angle variables
$\tilde{C}^{\nu}_{\boldsymbol{k}}(\alpha)$		Disturbing function coefficient of $\cos(\boldsymbol{k}\cdot\boldsymbol{\theta}_{i,j})$ for expansion in e_i and s_i
$b^{(s)}_{\nu}(\alpha)$	$\frac{1}{\pi}\int_{-\pi}^{\pi}\frac{\cos(j\theta)}{(1+\alpha^2-2\alpha\cos\theta)^s}$	Laplace coefficient
z_i	$=e_i\exp(\mathrm{i}\varpi_i)$	Complex eccentricity
ζ_i	$=s_i\exp(\mathrm{i}\Omega_i)$	Complex inclination variable
$\Lambda_{i,0}$		Reference value for Λ_i
δ_i	$\Lambda_i/\Lambda_{i,0}-1$	Fractional deviation of Λ_i from reference value $\Lambda_{i,0}$
X_i	$=\sqrt{\frac{1}{\Lambda_{i,0}}}(\kappa_i-\mathrm{i}\eta_i)$	$\approx e_i\exp(\mathrm{i}\varpi_i)+\mathcal{O}(e_i^3)$
Y_i	$=\sqrt{\frac{1}{4\Lambda_{i,0}}}(\sigma_i-\mathrm{i}\rho_i)$	$\approx\sin(I_i/2)\exp(\mathrm{i}\Omega_i)+\mathcal{O}(I_i e_i^2)$
$C^{\nu}_{\boldsymbol{k},l}(\alpha)$		Disturbing function coefficient of $\cos(\boldsymbol{k}\cdot\boldsymbol{\theta}_{i,j})$ for expansion in X_i,Y_i, and δ_i

Appendix B
Expansion of the Indirect Disturbing Function

In this appendix, we detail the expansion of the indirect component of the disturbing function,

$$\mathcal{R}_{\rm ind.}^{(i,j)} = -\frac{a_j}{GM_* m_i m_j}\tilde{\boldsymbol{r}}_i \cdot \tilde{\boldsymbol{r}}_j \,,$$

closely following Laskar & Robutel (1995), who present a method for deriving a complete expansion of the indirect disturbing function, best accomplished with the aid of computer algebra. We extend their derivation in order to provide the explicit formulas for disturbing function coefficients associated with specific cosine arguments. Writing $\tilde{\boldsymbol{r}}_i = \mu_i n_i a_i \boldsymbol{v}_i$, the indirect disturbing function can be written as

$$\mathcal{R}_{\rm ind.}^{(i,j)} = -\frac{n_i n_j a_j^2 a_i}{GM_*}\boldsymbol{v}_i \cdot \boldsymbol{v}_j \approx -\frac{1}{\sqrt{\alpha}}\boldsymbol{v}_i \cdot \boldsymbol{v}_j + \mathcal{O}((m/M_*)^2) \,, \tag{B1}$$

where the $\mathcal{O}((m/M_*)^2)$ error terms arise from approximating $\mu_i \approx m_i$. The vectors $\boldsymbol{v}_i$ may be expressed in terms of orbital elements as

$$\boldsymbol{v}_i = \frac{1}{\sqrt{1-e_i^2}}\mathcal{R}_3(\Omega_i)\mathcal{R}_1(I_i)\mathcal{R}_3(-\Omega_i)\cdot\begin{pmatrix}\Re[\mathrm{i}\exp[\mathrm{i}\varpi_i](e^{\mathrm{i}f_i}+e_i)]\\ \Im[\mathrm{i}\exp[\mathrm{i}\varpi_i](e^{\mathrm{i}f_i}+e_i)]\\ 0\end{pmatrix}, \tag{B2}$$

$$= \Re\left[\mathrm{i}(e^{\mathrm{i}(f+\varpi)}+e_i e^{\mathrm{i}\varpi})\begin{pmatrix}\cos^2(I_i/2)+\bar{\zeta}^2\\ -\mathrm{i}(\cos^2(I_i/2)-\bar{\zeta}^2)\\ -2\mathrm{i}\cos(I_i/2)\bar{\zeta}\end{pmatrix}\right] \tag{B3}$$

where f_i is the planet's true anomaly, and $\Re[\cdot]$ and $\Im[\cdot]$ denote the real and imaginary parts of the enclosed expression, respectively. Using the identity $\Re[A]\Re[B] = \frac{1}{2}\Re(AB + A\bar{B})$ and denoting $\mathcal{Z}_i = \mathrm{i}(e^{\mathrm{i}(f+\varpi)} + e_i e^{\mathrm{i}\varpi})(1-e_i)^{-1/2}$, we have

$$\boldsymbol{v}_i \cdot \boldsymbol{v}_j = \Re[\underbrace{\mathcal{Z}_i\mathcal{Z}_j(\bar{\zeta}_i\cos(I_j/2) - \bar{\zeta}_j\cos(I_i/2))^2}_{\text{Term 1}} + \underbrace{\mathcal{Z}_i\bar{\mathcal{Z}}_j(\cos(I_j/2)\cos(I_i/2) + \bar{\zeta}_i\zeta_j)^2}_{\text{Term 2}}]. \tag{B4}$$

Below we will treat the disturbing function terms arising from the expressions labeled "Term 1" and "Term 2" in Equation (B4) separately. But before doing so, we express $\mathcal{Z}_i$ explicitly in terms of the mean longitude, λ_i, rather than the true anomaly, f_i, using Hansen coefficients (e.g., Hughes 1981) to write

$$e^{\mathrm{i}f} = \sum_{k=-\infty}^{\infty} X_k^{0,1}(e)e^{\mathrm{i}k(\lambda-\varpi)} = \sum_{k=-\infty}^{\infty}\left(e^{|k-1|}\sum_{\sigma=0}^{\infty}N_{k,\sigma}^{0,1}e^{2\sigma}\right)e^{\mathrm{i}(\lambda-\varpi)} \,, \tag{B5}$$

where the coefficients, $N_{k,\sigma}^{0,1}$, appearing in the series expansion of $X_k^{0,1}(e)$ can be calculated with the aid of the recursion relations described in Hughes (1981) and Murray & Dermott (1999).[14] We define the related set of coefficients $\mathcal{X}_k^{0,1}(e) = (1-e^2)^{-1/2}X_k^{0,1}(e)$ so that we may write

$$\mathcal{Z} = \mathrm{i}e^{\mathrm{i}\varpi}\sum_{k\neq 0}\mathcal{X}_k^{0,1}(e)e^{\mathrm{i}k(\lambda-\varpi)} \tag{B6}$$

after noting that no $k=0$ term appears in the sum because $X_0^{0,1}(e) = -e$. We also define coefficients $\mathcal{N}_{k,p}^{0,1} = \sum_{n=0}^{p}\binom{-1/2}{n}(-1)^n N_{k,p-n}^{0,1}$, so that we may write

$$\mathcal{X}_k^{0,1}(e) = e^{|k-1|}\sum_{p=0}^{\infty}\mathcal{N}_{k,p}^{0,1}e^{2p} \,. \tag{B7}$$

We now turn to deriving expressions for the terms labeled "Term 1" and "Term 2" in Equation (B4) in terms of orbital elements.

[14] Note that the coefficients $N_{k,\sigma}^{0,1}$ are *not* equivalent to the "Newcomb operators" defined in Hughes (1981) and Murray & Dermott (1999) and denoted as $X_{c,d}^{a,b}$. Rather, the coefficients $N_{k,\sigma}^{0,1}$ are related to the Newcomb operators according to $N_{k,\sigma}^{0,1} = X_{k-1+\sigma,\sigma}^{0,1}$ for $k \geqslant 1$ and $N_{k,\sigma}^{0,1} = X_{\sigma,1-k+\sigma}^{0,1}$ for $k < 1$.

Term 1. Expressing Term 1 in terms of orbital elements, we have

$$\mathcal{Z}_i\bar{\mathcal{Z}}_j(\cos(I_j/2)\cos(I_i/2) + \bar{\zeta}_i\zeta_j)^2 = e^{\mathrm{i}(\varpi_i-\varpi_j)}((1-s_i^2)(1-s_j^2) + 2s_is_j\sqrt{1-s_i^2}\sqrt{1-s_j^2}\,e^{\mathrm{i}(\Omega_j-\Omega_i)} + s_i^2s_j^2e^{2\mathrm{i}(\Omega_j-\Omega_i)}) \times \sum_{k,k'\neq 0}\mathcal{X}_k^{0,1}(e_i)\mathcal{X}_{k'}^{0,1}(e_j)e^{\mathrm{i}(k\lambda_i-k'\lambda_j-k\varpi_i+k'\varpi_j)}\,. \tag{B8}$$

We see from Equation (B8) that Term 1 contributes to the cosine amplitudes of Equation (11) for terms with cosine arguments satisfying

$$\pm\boldsymbol{k}\cdot\boldsymbol{\theta}_{i,j} = -k'\lambda_j + k\lambda_i + (1-k)\varpi_i + (k'-1)\varpi_j + m(\Omega_j-\Omega_i) \tag{B9}$$

with $m=0$, 1, 2, and k, $k'\neq 0$. The indirect contributions to the amplitudes will be

$$[\tilde{C}_{\boldsymbol{k}}^{(\nu_1,\nu_2,\nu_3,\nu_4)}(\alpha)]_{\mathrm{ind.}} = -\frac{1}{\sqrt{\alpha}}\mathcal{N}_{k',\nu_4}^{0,1}\mathcal{N}_{k,\nu_3}^{0,1}(-1)^{\nu_1+\nu_2}\binom{1-\frac{m}{2}}{\nu_1}\binom{1-\frac{m}{2}}{\nu_2}(1+\delta_{m,1}) \tag{B10}$$

for the terms with $\boldsymbol{k} = \pm(k', -k, k-1, 1-k', m, -m)$ with m=0, 1, or 2. In particular, indirect contributions to terms associated with p:$p-q$ MMRs will occur for coefficients $\tilde{C}^{\nu}_{(p,q-p,p-q-1,1-p,m,-m)}$ and $\tilde{C}^{\nu}_{(p,q-p,p-q+1,-1-p,-m,m)}$ when $(k', k) = (p, p-q)$ and $(k', k) = (-p, q-p)$, respectively.

Term 2. Writing Term 2 from Equation (B4) in terms of orbital elements, we obtain

$$\mathcal{Z}_i\mathcal{Z}_j(\bar{\zeta}_i\cos(I_j/2) - \bar{\zeta}_j\cos(I_i/2))^2 = e^{\mathrm{i}(\varpi_i+\varpi_j)}(s_i^2(1-s_j^2)e^{-2\mathrm{i}\Omega_i} - 2s_is_j\sqrt{(1-s_i^2)(1-s_j^2)}\,e^{-\mathrm{i}(\Omega_i+\Omega_j)} + s_j^2(1-s_i^2)e^{-2\mathrm{i}\Omega_j}) \times \sum_{k,k'\neq 0}\mathcal{X}_k^{0,1}(e_i)\mathcal{X}_{k'}^{0,1}(e_j)e^{\mathrm{i}(k\lambda_i+k'\lambda_j-k\varpi_i-k'\varpi_j)}\,. \tag{B11}$$

Indirect terms arising from Term 2 will contribute to the amplitudes of Equation (11) for cosine arguments satisfying

$$\pm\boldsymbol{k}\cdot\boldsymbol{\theta}_{i,j} = k'\lambda_j + k\lambda_i + (1-k)\varpi_i + (1-k')\varpi_j - 2\Omega_i + m(\Omega_i-\Omega_j) \tag{B12}$$

where $m=0$, 1, 2, and k, $k'\neq 0$. The corresponding indirect amplitudes are given by

$$[\tilde{C}_{\boldsymbol{k}}^{(\nu_1,\nu_2,\nu_3,\nu_4)}(\alpha)]_{\mathrm{ind.}} = -\frac{1}{\sqrt{\alpha}}\mathcal{N}_{k',\nu_4}^{0,1}\mathcal{N}_{k,\nu_3}^{0,1}(-1)^{\nu_1+\nu_2}\binom{\frac{m}{2}}{\nu_1}\binom{1-\frac{m}{2}}{\nu_2}(1+\delta_{m,1}). \tag{B13}$$

Term 2 contributes indirect terms to the cosine amplitudes $\tilde{C}^{\nu}_{(p,q-p,p-q+1,1-p,m-2,-m)}$ associated with p:$p-q$ resonances for $(k', k) = (p, q-p)$ and the amplitudes $\tilde{C}^{\nu}_{(p,q-p,p-q-1,-1-p,2-m,m)}$ for $(k', k) = (-p, p-q)$.

Appendix C
Relating Orbital-element and Canonical-variable Expansions

In this appendix, we derive expressions for the coefficients $C_{\boldsymbol{k}}^{\nu,I}(\alpha)$, expressing the expansion of the disturbing function in terms of canonical variables, in terms of the coefficients $\tilde{C}_{\boldsymbol{k}}^{\nu}(\alpha)$ that express the disturbing function's expansion in orbital elements. In order to do so, we proceed in two steps: First, in Section C.1, we express the disturbing function expansion in terms of the complex variables X_i, X_j, Y_i, and Y_j, which are related to celmech's default set of canonical variables according to $X_i = \sqrt{\frac{1}{\Lambda_{i,0}}}(\kappa_i - i\eta_i)$ and $Y_i = \frac{1}{2}\sqrt{\frac{1}{\Lambda_{i,0}}}(\sigma_i - i\rho_i)$. Then, in Section C.2, we expand the resulting expressions from Section C.1 in terms of $\delta_i = (\Lambda_i - \Lambda_{i,0})/\Lambda_{i,0}$ and δ_j.

C.1. Relating z *and* ζ *Expansion to* X *and* Y *Expansion*

Let us define the new variables $\hat{X}_i \equiv (1+\delta_i)^{1/2}X_i$ and $\hat{Y}_i \equiv (1+\delta_i)^{1/2}Y_i$ so that $z_i = \hat{X}_i\left(1-\frac{1}{4}|\hat{X}_i|^2\right)^{1/2}$ and $\zeta_i = \hat{Y}_i\left(1-\frac{1}{2}|\hat{X}_i|^2\right)^{-1/2}$ (see Equations (14) and (15)) along with the new disturbing function expansion coefficients $\hat{C}_{\boldsymbol{k}}^{\nu}(\alpha_{ij})$ that

satisfy

$$\left(\frac{1}{a_j}\right) z_i^{k_3} z_j^{k_4} \zeta_i^{k_5} \zeta_j^{k_6} \sum_{\boldsymbol{\nu}=0}^{\infty} \tilde{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha_{ij}) |\zeta_i|^{2\nu_1} |\zeta_j|^{2\nu_2} |z_i|^{2\nu_3} |z_j|^{2\nu_4} = \left(\frac{1}{a_j}\right) \hat{X}_i^{k_3} \hat{X}_j^{k_4} \hat{Y}_i^{k_5} \hat{Y}_j^{k_6} \sum_{\boldsymbol{\nu}=0}^{\infty} \hat{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha_{ij}) |\hat{Y}_i|^{2\nu_1} |\hat{Y}_j|^{2\nu_2} |\hat{X}_i|^{2\nu_3} |\hat{X}_j|^{2\nu_4} \tag{C1}$$

where we assume here and throughout this appendix that $k_m \geqslant 0$ for $m = 3,\ldots,6$.[15] In order to obtain explicit expressions for the coefficients $\hat{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha_{ij})$, we write the monomial $z_i^{k_3} \zeta_i^{k_5} |\zeta_i|^{2\nu_1} |z_i|^{2\nu_3}$ in terms of variables $\hat{X}_i$ and $\hat{Y}_i$ as

$$\begin{aligned} z_i^{k_3} \zeta_i^{k_5} s_i^{2\nu_1} e_i^{2\nu_3} &= \hat{X}_i^{k_3} \hat{Y}_i^{k_5} |\hat{Y}_i|^{2\nu_1} |\hat{X}_i|^{2\nu_3} \left(1 - \frac{1}{4}|\hat{X}_i|^2\right)^{|k_3|/2+\nu_3} \left(1 - \frac{1}{2}|\hat{X}_i|^2\right)^{-|k_5|/2-\nu_1} \\ &\equiv \hat{X}_i^{k_3} \hat{Y}_i^{k_5} |\hat{Y}_i|^{2\nu_1} |\hat{X}_i|^{2\nu_3} \sum_{n=0}^{\infty} \mathcal{T}^{(n)}_{|k_3|/2+\nu_3, |k_5|/2+\nu_1} |\hat{X}_i|^{2n} \end{aligned} \tag{C2}$$

where the coefficients

$$\mathcal{T}^{(n)}_{p,q} = \left(-\frac{1}{2}\right)^n \sum_{l=0}^{n} \binom{p}{l} \binom{-q}{n-l} \left(\frac{1}{2}\right)^l$$

are obtained by expanding the factors $\left(1 - \frac{1}{4}|\hat{X}_i|^2\right)^{|k_3|/2+\nu_1}$ and $\left(1 - \frac{1}{2}|\hat{X}_i|^2\right)^{-|k_5|/2-\nu_3}$ in the first line of Equation (C2) and collecting terms $\propto |\hat{X}_i|^{2n}$. Using Equation (C2), we can write the individual terms appearing in the sum on the left-hand side of Equation (C1) in terms of the variables $\hat{X}_i$, $\hat{Y}_i$, $\hat{X}_j$, and $\hat{Y}_j$ as

$$\begin{aligned} &\tilde{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha_{ij}) z_i^{k_3} z_j^{k_4} \zeta_i^{k_5} \zeta_j^{k_6} s_i^{2\nu_1} s_j^{2\nu_2} e_i^{2\nu_3} e_j^{2\nu_4} = \\ &\tilde{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha_{ij}) \hat{X}_i^{k_3} \hat{X}_j^{k_4} \hat{Y}_i^{k_5} \hat{Y}_j^{k_6} |\hat{Y}_i|^{2\nu_1} |\hat{Y}_j|^{2\nu_2} \sum_{n,m=0}^{\infty} \mathcal{T}^{(n)}_{|k_3|/2+\nu_3, |k_5|/2+\nu_1} \mathcal{T}^{(m)}_{|k_4|/2+\nu_4, |k_6|/2+\nu_2} |\hat{X}_i|^{2n} |\hat{X}_j|^{2m} . \end{aligned} \tag{C3}$$

Collecting terms in Equation (C3) with $N_3 = \nu_3 + n$ and $N_4 = \nu_4 + n$, we arrive at the following expression relating $\hat{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha_{ij})$ to the coefficients $\tilde{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha_{ij})$:

$$\hat{C}_{\boldsymbol{k}}^{(N_1,N_2,N_3,N_4)}(\alpha_{ij}) = \sum_{n=0}^{N_3} \sum_{m=0}^{N_4} \mathcal{T}^{(n)}_{|k_3|/2+\nu_3, |k_5|/2+\nu_1} \mathcal{T}^{(m)}_{|k_4|/2+\nu_4, |k_6|/2+\nu_2} \tilde{C}_{\boldsymbol{k}}^{(N_1,N_2,n,m)}(\alpha_{ij}). \tag{C4}$$

C.2. Expansion in δ_i and δ_j

To derive explicit expressions for the coefficients $C_{\boldsymbol{k}}^{\boldsymbol{\nu},(l_1,l_2)}$ in terms of $\hat{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}$, we write out the δ_i and δ_j dependence of the terms appearing in the sum on the right-hand side of Equation (C4) explicitly as

$$\begin{aligned} &\left(\frac{1}{a_j}\right) \hat{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha_{ij}) \; \hat{X}_i^{k_3} \hat{X}_j^{k_4} \hat{Y}_i^{k_5} \hat{Y}_j^{k_6} |\hat{Y}_i|^{2\nu_1} |\hat{Y}_j|^{2\nu_2} |\hat{X}_i|^{2\nu_3} |\hat{X}_j|^{2\nu_4} \\ &= \left\{ \frac{1}{a_{j,0}} (1+\delta_i)^{-(|k_3|+|k_5|)/2-\nu_1-\nu_3} (1+\delta_j)^{2-(|k_4|+|k_6|)/2-\nu_2-\nu_4} \hat{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}} \left(\alpha_{ij,0} \left(\frac{1+\delta_i}{1+\delta_j} \right)^2 \right) \right\} \\ &\times X_i^{k_3} X_j^{k_4} Y_i^{k_5} Y_j^{k_6} |Y_i|^{2\nu_1} |Y_j|^{2\nu_2} |X_i|^{2\nu_3} |X_j|^{2\nu_4} . \end{aligned} \tag{C5}$$

Accordingly, we may write

$$C_{\boldsymbol{k}}^{\boldsymbol{\nu},(l_1,l_2)}(\alpha_{ij,0}) = \frac{1}{l_1! l_2!} \frac{\partial^{l_1+l_2}}{\partial \delta_i^{l_1} \partial \delta_j^{l_2}} \left[(1+\delta_i)^{-p_i/2} (1+\delta_j)^{-p_j/2} \hat{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}} \left(\alpha_0 \left(\frac{1+\delta_i}{1+\delta_j} \right)^2 \right) \right]_{(\delta_i,\delta_j)=0} \tag{C6}$$

where ${}_{pi} = |k_3| + |k_5| + 2\nu_1 + 2\nu_3$ and $p_j = 4 + |k_4| + |k_6| + 2\nu_2 + 2\nu_4$. For the case $l_1 = l_2 = 0$, from Equation (C6), we simply have $C_{\boldsymbol{k}}^{\boldsymbol{\nu},(0,0)}(\alpha_{ij,0}) = \hat{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}(\alpha_{ij,0})$. More generally, an explicit expression of $C_{\boldsymbol{k}}^{\boldsymbol{\nu},(l_1,l_2)}(\alpha_{ij,0})$ in terms of $\hat{C}_{\boldsymbol{k}}^{\boldsymbol{\nu}}$ and its derivatives can be derived by applying the product rule for derivatives along with di Bruno's formula (e.g., Johnson 2002) generalizing the chain rule to higher

[15] Whenever any $k_m < 0$, the appropriate replacements $Z^{k_m} \to \bar{Z}^{-k_m}$ should be made, where Z stands for one of X_i, X_j, Y_i, or Y_j.

derivatives to Equation (C6). A tedious but straightforward calculation yields

$$C_{\boldsymbol{k}}^{\nu,(l_1,l_2)}(\alpha_0) = \sum_{m_1=0}^{l_1}\sum_{r_1=1}^{l_1-m_1}\sum_{m_2=0}^{l_2}\sum_{r_2=1}^{l_2-m_2}\Psi_{l_1,l_2,p_1,p_2,m_1,m_2,r_1,r_2}\alpha_0^{(r_i+r_j)}\frac{d^{(r_1+r_2)}}{d\alpha^{(r_1+r_2)}}\hat{C}_{\boldsymbol{k}}^{\nu}(\alpha)\Big|_{\alpha=\alpha_0}$$

where

$$\Psi_{l_1,l_2,p_1,p_2,m_1,m_2,r_1,r_2} = \frac{1}{l_1!l_2!}\binom{l_2}{m_2}\binom{l_1}{m_1}(-2r_1-p_2/2)_{m_2}(-p_1/2)_{m_1}\mathcal{B}^{(+)}_{l_1-m_1,r_1}\mathcal{B}^{(-)}_{l_2-m_2,r_2},$$

$(x)_k \equiv x(x-1)...(x-k+1)$, and

$$\mathcal{B}^{(\pm)}_{k,n} = B_{n,k}\left(\binom{\pm 2}{1}1!,\ldots,\binom{\pm 2}{n-k+2}(n-k+2)!\right)$$

with $B_{n,k}$ denoting partial Bell polynomials.

ORCID iDs

Sam Hadden https://orcid.org/0000-0002-1032-0783
Daniel Tamayo https://orcid.org/0000-0002-9908-8705

References

Agol, E., & Deck, K. 2016, ApJ, 818, 177
Agol, E., Hernandez, D. M., & Langford, Z. 2021, MNRAS, 507, 1582
Agol, E., Steffen, J., Sari, R., & Clarkson, W. 2005, MNRAS, 359, 567
Batygin, K. 2015, MNRAS, 451, 2589
Batygin, K., & Morbidelli, A. 2013, A&A, 556, A28
Becker, J. C., & Adams, F. C. 2017, MNRAS, 468, 549
Boquet, F. 1889, AnPar, 11, B.1
Brumberg, V. A. 1995, Analytical Techniques of Celestial Mechanics (Berlin: Springer)
Chirikov, B. V. 1979, PhR, 52, 263
Deck, K. M., & Agol, E. 2016, ApJ, 821, 96
Deck, K. M., Agol, E., Holman, M. J., & Nesvorný, D. 2014, ApJ, 787, 132
Deck, K. M., Payne, M., & Holman, M. J. 2013, ApJ, 774, 129
Delisle, J. B. 2017, A&A, 605, A96
Delisle, J. B., Correia, A. C. M., & Laskar, J. 2015, A&A, 579, A128
Eastman, J., Gaudi, B. S., & Agol, E. 2013, PASP, 125, 83
Ellis, K. M., & Murray, C. D. 2000, Icar, 147, 129
Foreman-Mackey, D., Luger, R., Agol, E., et al. 2021, JOSS, 6, 3285
Fulton, B. J., Petigura, E. A., Blunt, S., & Sinukoff, E. 2018, PASP, 130, 044504
Gastineau, M., & Laskar, J. 2011, ACM Commun. Comput. Algebra, 44, 194
Gastineau, M., & Laskar, J. 2021, TRIP 1.4.120, TRIP Reference manual, IMCCE, Paris Observatory
Hadden, S. 2019, AJ, 158, 238
Hadden, S., Barclay, T., Payne, M. J., & Holman, M. J. 2019, AJ, 158, 146
Hadden, S., & Lithwick, Y. 2016, ApJ, 828, 44
Hadden, S., & Lithwick, Y. 2018, AJ, 156, 95
Hadden, S., & Payne, M. J. 2020, AJ, 160, 106
Hadden, S., & Tamayo, D. 2022, celmech, v1.0.2, Zenodo, doi:10.5281/zenodo.6984409
Henrard, J. 1989, CeMec, 45, 245
Hernandez, D. M., & Dehnen, W. 2017, MNRAS, 468, 2614
Hoang, N. H., Mogavero, F., & Laskar, J. 2021, A&A, 654, A156
Hoang, N. H., Mogavero, F., & Laskar, J. 2022, MNRAS, 514, 1342
Hughes, S. 1981, CeMec, 25, 101
Hunter, J. D. 2007, CSE, 9, 90
Johnson, W. P. 2002, Am. Math. Mon., 109, 217
Jones, E., Oliphant, T., Peterson, P., et al. 2001, SciPy: Open source scientific tools for Python, http://www.scipy.org/
Kreidberg, L. 2015, PASP, 127, 1161
Laskar, J., & Robutel, P. 1995, CeMDA, 62, 193
Le Verrier, U. J. 1855, AnPar, 1, 258
Lichtenberg, A. J., & Lieberman, M. A. 2013, Regular and Chaotic Dynamics, Vol. 38 (Berlin: Springer)
Lithwick, Y., Xie, J., & Wu, Y. 2012, ApJ, 761, 122
Malhotra, R. 1993, ApJ, 407, 266
Meurer, A., Smith, C. P., Paprocki, M., et al. 2017, PeerJ Comp. Sci., 3, e103
Millholland, S., & Laughlin, G. 2019, NatAs, 3, 424
Mogavero, F., & Laskar, J. 2022, A&A, 662, L3
Morbidelli, A. 2002, Modern Celestial Mechanics: Aspects of Solar System Dynamics (London: Taylor & Francis)
Murray, C. D., & Dermott, S. F. 1999, Solar System Dynamics (Cambridge: Cambridge Univ. Press)
Mustill, A. J., & Wyatt, M. C. 2011, MNRAS, 413, 554
Nesvorný, D., & Morbidelli, A. 2008, ApJ, 688, 636
Nesvorný, D., & Vokrouhlický, D. 2014, ApJ, 790, 58
Nobili, A., & Roxburgh, I. W. 1986, in IAU Symp. 114, Relativity in Celestial Mechanics and Astrometry. High Precision Dynamical Theories and Observational Verifications, ed. J. Kovalevsky & V. A. Brumberg (Dordrecht: D. Reidel), 105
Oliphant, T. E. 2006, A Guide to NumPy, Vol. 1 (USA: Trelgol Publishing)
Peirce, B. 1849, AJ, 1, 1
Petit, A. C., Pichierri, G., Davies, M. B., & Johansen, A. 2020, A&A, 641, A176
Quillen, A. C. 2011, MNRAS, 418, 1043
Rath, J., Hadden, S., & Lithwick, Y. 2022, ApJ, 932, 61
Read, M. J., Wyatt, M. C., & Triaud, A. H. M. J. 2017, MNRAS, 469, 171
Rein, H., & Liu, S. F. 2012, A&A, 537, A128
Saillenfest, M., Laskar, J., & Boué, G. 2019, A&A, 623, A4
Sansottera, M., & Libert, A. S. 2019, CeMDA, 131, 38
Su, Y., & Lai, D. 2022, MNRAS, 513, 3302
Tamayo, D., Murray, N., Tremaine, S., & Winn, J. 2021, AJ, 162, 220
Tamayo, D., Rein, H., Shi, P., & Hernandez, D. M. 2020, MNRAS, 491, 2885
Tollerud, E., Smith, A., Price-Whelan, A., et al. 2019, BAAS, 51, 180
Trifonov, T. 2019, The Exo-Striker: Transit and radial velocity interactive fitting tool for orbital analysis and N-body simulations, Astrophysics Source Code Library, record ascl:1906.004
Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, NatMe, 17, 261
Wisdom, J. 1980, AJ, 85, 1122