Numerical computation of high-order expansions of invariant manifolds of high-dimensional tori

Joan Gimeno* Àngel Jorba† Begoña Nicolás‡ Estrella Olmedo§

November 9, 2021

Abstract

In this paper we present a procedure to compute reducible invariant tori and their stable and unstable manifolds in Poincaré maps. The method has two steps. In the first step we compute, by means of a quadratically convergent scheme, the Fourier series of the torus, its Floquet transformation, and its Floquet matrix. If the torus has stable and/or unstable directions, in the second step we compute the Taylor-Fourier expansions of the corresponding invariant manifolds up to a given order. The paper also discusses the case in which the torus is highly unstable so that a multiple shooting strategy is needed to compute the torus.

If the order of the Taylor expansion of the manifolds is fixed and N is the number of Fourier modes, the whole computational effort (torus and manifolds) increases as $\mathcal{O}(N \log N)$ and the memory required behaves as $\mathcal{O}(N)$. This makes the algorithm very suitable to compute high-dimensional tori for which a huge number of Fourier modes are needed. Besides, the algorithm has a very high degree of parallelism. The paper includes examples where we compute invariant tori (of dimensions up to 5) of quasi-periodically forced ODEs. The computations are run in a parallel computer and its efficiency with respect to the number of processors is also discussed.

Keywords: Parametrization method, Quasi-periodic Floquet Theory, Jet transport, and Parallel computing.

1 Introduction

The understanding of the dynamics of a given system is usually based on the knowledge of a subset of solutions that serve as skeleton for the global properties. This includes equilibrium points, periodic and quasi-periodic solutions, and their invariant manifolds.

On the other hand, the accurate description of realistic phenomena usually requires of more sophisticated models. In several cases, some effects are introduced in the mathematical model as a combination of different periodic time-dependent perturbations. For example, the description of the motion of a spacecraft with some level of accuracy requires the introduction of several effects in the model: the effect of the Sun can be introduced as a periodic time-dependence ([Hua60, CRR64, And98]), and adding some other effects (like the effects coming from the non-circular motion of the Moon) results in a quasi-periodic

*Department of Mathematics, University of Rome Tor Vergata, Via della Ricerca Scientifica 1, 00133 Rome, Italy, gimeno@mat.uniroma2.it

†Departament de Matemàtiques i Informàtica, Universitat de Barcelona, Gran Via de les Corts Catalanes, 585, 08007 Barcelona, Spain, angel@maia.ub.es

‡Departament de Matemàtiques i Informàtica, Universitat de Barcelona, Gran Via de les Corts Catalanes, 585, 08007 Barcelona, Spain, bego@maia.ub.es

§Barcelona Expert Center (BEC), Institute of Marine Sciences (ICM) and Consejo Superior de Investigaciones Científicas (CSIC), P. Marítim de la Barceloneta, 37-49, 08003 Barcelona, Spain, olmedo@icm.csic.es

time-dependence. As examples, in [GLMS01a, GLMS01b, GJMS01a, GJMS01b], the authors examine the effect of perturbing bodies as the sum of periodic perturbations with incommensurable periods, or in [GMM02], where very detailed models are presented for the motion of a particle in the Solar system and in the Earth-Moon system, taking into account up to five natural frequencies of the latter system. In these cases, we usually have a differential equation,

$$\dot{x} = f(x,t), \tag{1}$$

where $x \in \Omega = \mathring{\Omega} \subset \mathbb{R}^n$ and $f\colon \Omega \times \mathbb{R} \to \mathbb{R}^n$ is a smooth function that we assume to depend quasi-periodically on t. This means that there exists a function $F\colon \Omega \times \mathbb{T}^{d+1} \to \mathbb{R}^n$, with $d \geqslant 1$, such that

$$f(x,t) = F(x,\theta_0,\ldots,\theta_d), \qquad \theta_i = \omega_i t, \quad i = 0,1,\ldots,d.$$

Then (1) can be rewritten as

$$\begin{cases} \dot{x} = F(x,\theta), \\ \dot{\theta} = \omega, \end{cases} \tag{2}$$

where $\theta = (\theta_0,\ldots,\theta_d) \in \mathbb{T}^{d+1}$ is an angular variable vector and $\omega = (\omega_0,\ldots,\omega_d)$ is a frequency vector, of dimension $d+1$, whose components are considered to be linearly independent over the rationals. Since F is periodic in each angular variable, a temporal Poincaré section can be defined using one of the angles of the system, for example $\theta_0 = 0 \mod 2\pi$. The associated Poincaré map is the flow of the differential equation from initial time at 0 to the final time at time $\frac{2\pi}{\omega_0}$. Then, redefining the vector θ as $(\theta_1,\ldots,\theta_d)$ we can write the Poincaré map as

$$\begin{cases} \bar{x} = P(x,\theta), \\ \bar{\theta} = \theta + \rho, \end{cases} \tag{3}$$

where ρ is a d-dimensional vector with components $\rho_i = \frac{2\pi\omega_i}{\omega_0}$ for $i = 1,\ldots,d$. The upper bar, $\bar{}$, denotes the image under the Poincaré map P.

For the discrete system (3), the simplest possible invariant sets are invariant tori of dimension d parametrized by the angle θ, and with frequencies ρ. Each of these tori can be represented by a smooth injective map $\varphi\colon \mathbb{T}^d \to \mathbb{R}^n$, which satisfies the following invariance condition

$$P(\varphi(\theta),\theta) = \varphi(\theta+\rho), \qquad \forall\, \theta \in \mathbb{T}^d. \tag{4}$$

Here we are interested in these tori and in an efficient computation of them. Note that, as the dimension of the torus is higher, the computational effort increases. If the torus is reducible (see Section 2), there are very efficient numerical methods that allow to compute the torus jointly with the Floquet change of variables and the reduced Floquet matrix. One of these methods is presented in [JO09], and it is based on the proofs made in [JS96]. It is remarkable that the number of operations and storage requirements are proportional to $N \log N$ and N respectively, where N denotes the number of Fourier modes used to represent the torus. A side benefit is that the method has a high degree of parallelism so it can take advantage of modern computers.

In the present work, we extend the methods in [JO09] so that now: *i)* the method provides a high-order parametrization of the stable and/or unstable manifolds of the tori if they exist; *ii)* we implement a C code that runs in a computer with several processors between which the computations are done concurrently using OpenMP [DM98] instead of the PVM library [GBD$^+$95] used in [JO09].

On the one hand, we implement a parametrization method for computing the Taylor-Fourier expansion of the un/stable invariant manifolds of invariant tori. To implement this parametrization method on a Poincaré map we need to estimate the high-order derivatives of this Poincaré map, which requires to compute high-order derivatives of the flow of the ODE. This may be very tedious and tricky if we have to compute them by hand. In particular, since here we focus on the analysis of invariant tori of maps, we

follow and apply the ideas developed in [GJJC+21], where authors explain how to numerically integrate high-order derivatives of maps using automatic differentiation with respect to initial data; a technique called "jet transport" [AFJ+08, AFJ+09].

On the other hand, the use of parallel computing allows to reduce the total computational time by using several processors. The parallelization presented in [JO09] was implemented on a cluster of PCs (with a distributed memory), making use of the PVM library for the communications through an Ethernet network using a master-slave scheme. Therefore, there was always a penalty time to pay for those communications. For this reason, the reduction in temporal costs of the parallelization was relevant for a small number of processors, but at some point, this reduction stagnates. Nowadays, the processors involved in our parallelizations belong to the same computer (hence, with a shared memory), so there is not communication penalty between the threads. Both, the computation of the tori and their invariant manifolds are implemented in parallel.

There are many other papers in the literature devoted to the computation of invariant tori and their manifolds. The computation of a quasi-periodic solution (with two basic frequencies) of a flow is done, for instance, in [DJS91] by means of a collocation method. The approximation of an invariant tori of a flow is also considered in [DLR91, DL95, HKM97] using a PDE approach. The computation of 1D invariant tori of maps is discussed in [Sim98, CJ00], and an algorithm to obtain their linear stability is contained in [Jor01]. For algorithms taking advantage of some properties of the linearized normal behaviour, see [HdlL06, HLS12, CH17, KAdlL21]. The main differences between these papers and the present one is that here we focus on higher dimensional tori of flows, that require a large amount of computations (and hence, to take advantage of parallel computers) and the use of jet transport to compute high-order derivatives of Poincaré maps.

The structure of this paper is as follows. In Section 2, the reducibility of a quasi-periodic solution is explained along with a brief summary of the process developed in [JO09] for computing invariant tori for maps and for obtaining their stability information at the same time. In Section 3, we introduce the parametrization of hyperbolic invariant manifolds of those tori, that will be combined with multiple shooting methods, explained in Section 4, in order to improve the accuracy of the computations, specially when the instability of the invariant objects is strong. Details concerning to the implementation of the computations are included in Section 5. Section 6 is devoted to two different applications of the presented technique. And finally, Section 7 presents some conclusions and future work.

2 Reducibility of the system

We can analyze the linear behaviour around a torus satisfying (4) by taking a small displacement $h(\theta) \in \mathbb{R}^n$ from the torus and applying the map:

$$P(\varphi(\theta) + h(\theta), \theta) = P(\varphi(\theta), \theta) + D_x P(\varphi(\theta), \theta) h(\theta) + \mathcal{O}(\|h\|^2),$$

where $A(\theta) = D_x P(\varphi(\theta), \theta)$ is the Jacobian of the Poincaré map on the torus $\varphi(\theta)$.

Renaming $h(\theta)$ as x, this linear behaviour can be expressed by the linear skew-product

$$\left.\begin{aligned} \bar{x} &= A(\theta) x, \\ \bar{\theta} &= \theta + \rho. \end{aligned}\right\} \tag{5}$$

Definition 2.1 (Discrete reducible torus)**.** The system (5) is said to be reducible if, and only if, there exists a continuous change of variables $x = C(\theta) y$ such that (5) becomes

$$\left.\begin{aligned} \bar{y} &= B y, \\ \bar{\theta} &= \theta + \rho, \end{aligned}\right\} \tag{6}$$

where the matrix

$$B = C^{-1}(\theta + \rho)A(\theta)C(\theta), \tag{7}$$

does not depend on θ. The matrix B is the *Floquet matrix* and $x = C(\theta)y$ is the *Floquet transformation*.

Notice that the Floquet matrix B is a constant matrix that contains the dynamical information of the system in (5), and then, its eigenvalues provide the linear stability around the torus.

Remark 2.1 (Continuous reducible torus). Analogously, being Ψ a quasi-periodic solution, of $d+1$ dimensions, of the system (1), let us consider the linearization of the system around the solution as

$$\dot{x} = \mathfrak{a}(\Psi, t)x \tag{8}$$

where $\mathfrak{a}(\Psi, t) = D_x f(\Psi(t), t)$. We will say that Ψ is a reducible quasi-periodic solution of (1) if, and only if, there exists a quasi-periodic change of variables $x = \mathfrak{c}(t)y$ that transforms (8) into

$$\dot{y} = \mathfrak{b}y, \tag{9}$$

where $\mathfrak{b}$ is a constant matrix.

Then, Floquet matrix B for transforming the system (5) into (6) corresponds to

$$B = \exp\{\delta\mathfrak{b}\}$$

where δ denotes the time used to define the temporal Poincaré map P and $\mathfrak{b}$ is the Floquet matrix resulting from transforming (8) into (9).

Remark 2.2. There is some freedom in choosing the form of matrix B. For instance, it can be chosen to be in diagonal (or Jordan) form by composing $C(\theta)$ with a suitable linear (may be complex) transformation, or to be block diagonal to keep B (and $C(\theta)$) real.

Assuming the existence of a torus for system (3) that satisfies invariance condition (4) and that is reducible, [JO09] developed an iterative method based on Newton iteration (quadratically convergent) for finding the torus and the Floquet change at the same time. For this, it is necessary to know suitable seeds for $\varphi(\theta)$ and $C(\theta)$ (namely $x_0(\theta)$ and $C_0(\theta)$ respectively), such that $C_0^{-1}(\theta + \rho)D_x P(x_0(\theta), \theta)C_0(\theta)$ is close to a constant matrix B_0. Let us suppose by now that these approximations are chosen. Then, residual magnitudes $y_0(\theta)$ and $Q_0(\theta)$ indicate the error of these approximations to the real solution, i.e.,

$$y_0(\theta) = x_0(\theta + \rho) - P(x_0(\theta), \theta), \tag{10}$$

$$Q_0(\theta) = C_0^{-1}(\theta + \rho)D_x P(x_0(\theta), \theta)C_0(\theta) - B_0. \tag{11}$$

Then, the norm of these magnitudes is a small quantity of order say ε. Let us take, for example, infinite norm, $\|y_0\|_\infty \approx \varepsilon$ and $\|Q_0\|_\infty \approx \varepsilon$.

The idea is to use the reducibility assumption for finding a better approximation of the invariant torus, and with it, to improve the Floquet change, iteratively until the precision of both parametrizations is good enough. As all details are carefully described in [JO09], very brief description of the procedure is given here, we address the interested reader to that reference.

Note that in the present work we assume that the torus exists and that it is real analytic. In [JS96] the proof of its existence is given for flows and it can be easily translated for maps, as long as some hypothesis are satisfied. First hypothesis is referred to the smoothness of the map. Second hypothesis is that some Diophantine conditions, involving the frequency vector ρ and the eigenvalues $\lambda_1, \ldots, \lambda_n$ of the matrix B_0 are satisfied. Concretely, it is assumed that there exist real constants $c > 0$ and $\gamma > d - 1$ such that

$$|\exp(\langle\kappa, \rho\rangle \mathrm{i}) - \lambda_j| > \frac{c}{|\kappa|^\gamma}, \qquad \forall\kappa \in \mathbb{Z}^d\backslash\{0\}, \quad j = 1, \ldots, n, \tag{12}$$

$$\left|\exp(\langle\kappa, \rho\rangle \mathrm{i}) - \frac{\lambda_j}{\lambda_l}\right| > \frac{c}{|\kappa|^\gamma}, \qquad \forall\, \kappa \in \mathbb{Z}^d\backslash\{0\}, \quad j, l = 1, \ldots, n, \tag{13}$$

where i denotes the complex unit, $\langle \cdot, \cdot \rangle$ the standard scalar product, and $|\kappa| = |\kappa_1| + \cdots + |\kappa_d|$. Note that condition (12) is satisfied if all the eigenvalues λ_j have modulus different from 1, and condition (13) is satisfied if all eigenvalues have different modulus. As at every step of the iterative procedure, the Floquet matrix changes, a non-degeneracy condition on the eigenvalues is needed. For this, it is common to make eigenvalues depending on parameters such that Diophantine condition holds at every iterative step. In practice, we do not need to verify this condition but what we have to do is to check if the left-hand side of (12) and (13) is small. This can be done indirectly by checking the size of the Fourier modes of the correction given by the Newton method is not too big.

Assuming all these hypothesis to hold and that $x_0(\theta)$ and $C_0(\theta)$ are available (with $\|y_0\|_\infty \approx \varepsilon$ and $\|Q_0\|_\infty \approx \varepsilon$), let us summarize the iterative scheme in [JO09] to find good approximations of the torus and of the Floquet change. The iterative scheme is divided in two steps. The first one focuses on computing a better approximation of the torus, and the second one, on improving the Floquet change and Floquet matrix.

Algorithm 2.2 (Computation of invariant torus, Floquet change, and Floquet matrix)**.**

- ⋆ *Input: Discrete system as* (3)*, initial guesses* $x_0(\theta)$*,* $C_0(\theta)$*, and* B_0*.*
- ⋆ *Output: Torus* $\varphi(\theta)$*, Floquet change* $C(\theta)$*, and Floquet matrix* B*.*

First step:

1. *Compute the error* $y_0(\theta) = x_0(\theta + \rho) - P(x_0(\theta), \theta)$*.*
2. *Compute the function* $g(\theta) = -C_0^{-1}(\theta + \rho) y_0(\theta)$*.*
3. *Find* u *that verifies* $u(\theta + \rho) = B_0 u(\theta) + g(\theta)$*. For this, we expand functions* g *and* u *in real Fourier series (the expansion can be done in complex Fourier series, but we work with real expansions in the computer programs):*

$$g(\theta) = \frac{g^{(0)}}{2} + \sum_{\kappa \neq 0} g_\kappa^{(c)} \cos\langle \kappa, \theta \rangle + g_\kappa^{(s)} \sin\langle \kappa, \theta \rangle,$$
$$u(\theta) = \frac{u^{(0)}}{2} + \sum_{\kappa \neq 0} u_\kappa^{(c)} \cos\langle \kappa, \theta \rangle + u_\kappa^{(s)} \sin\langle \kappa, \theta \rangle,$$

where $\kappa \in \mathbb{N}^d$ *and* $\langle \kappa, \theta \rangle = \kappa_1\theta_1 + \cdots + \kappa_d\theta_d$*, and solve the following system to find Fourier coefficients* $u^{(0)}$*,* $u_\kappa^{(c)}$*, and* $u_\kappa^{(s)}$

$$\begin{aligned} (Id - B_0)\frac{u^{(0)}}{2} &= \frac{g^{(0)}}{2}, \\ (B_0^2 - 2\cos\langle \kappa, \rho \rangle B_0 + Id) u_\kappa^{(c)} &= (\cos\langle \kappa, \rho \rangle Id + B_0) g_\kappa^{(c)} - \sin\langle \kappa, \rho \rangle g_\kappa^{(s)}, \\ (B_0^2 - 2\cos\langle \kappa, \rho \rangle B_0 + Id) u_\kappa^{(s)} &= (\cos\langle \kappa, \rho \rangle Id + B_0) g_\kappa^{(s)} + \sin\langle \kappa, \rho \rangle g_\kappa^{(c)}, \end{aligned} \tag{14}$$

where Id *denotes the identity matrix.*

4. *Compute* $h(\theta) = C_0(\theta) u(\theta)$*.*
5. *Compute* $x_1(\theta) = x_0(\theta) + h(\theta)$*, that is the new approximation of the torus, such that* $\|y_1\|_\infty \approx \varepsilon^2$*, with* y_1 *defined like in (10).*

Second step:

1. *Compute the matrices* $R(\theta) = C_0^{-1}(\theta+\rho)D_x f(x_1(\theta),\theta)C_0(\theta) - B_0$, $\tilde{R}(\theta) = R(\theta) - Avg(R)$, *and* $B_1 = B_0 + Avg(R)$, *where* $Avg(R)$ *is the average of the map* $\theta \mapsto R(\theta)$, *that is,*

$$Avg(R) = \frac{1}{(2\pi)^d}\int_{\mathbb{T}^d} R(\theta)\, d\theta$$

and B_1 *is the new approximation to the Floquet matrix.*

2. *Find the matrix valued function* H *that verifies* $H(\theta+\rho)B_1 - B_1 H(\theta) = \tilde{R}(\theta)$. *For this, we expand* R *and* H *in real Fourier series:*

$$H(\theta) = \sum_{\kappa\neq 0} H_\kappa^{(c)}\cos\langle\kappa,\theta\rangle + H_\kappa^{(s)}\sin\langle\kappa,\theta\rangle,$$
$$\tilde{R}(\theta) = \sum_{\kappa\neq 0} R_\kappa^{(c)}\cos\langle\kappa,\theta\rangle + R_\kappa^{(s)}\sin\langle\kappa,\theta\rangle,$$

and we solve the following system to find Fourier coefficients $H_\kappa^{(c)}$ *and* $H_\kappa^{(s)}$ *(note that* $H^{(0)} = 0$*)*

$$\begin{aligned}(H_\kappa^{(c)}\cos\langle\kappa,\rho\rangle + H_\kappa^{(s)}\sin\langle\kappa,\rho\rangle)B_1 - B_1 H_\kappa^{(c)} &= R_\kappa^{(c)},\\ (H_\kappa^{(s)}\cos\langle\kappa,\rho\rangle - H_\kappa^{(c)}\sin\langle\kappa,\rho\rangle)B_1 - B_1 H_\kappa^{(s)} &= R_\kappa^{(s)}.\end{aligned} \tag{15}$$

3. *Compute* $C_1(\theta) = C_0(\theta)(Id + H(\theta))$, *that is the new approximation of the Floquet transformation, such that* $\|Q_1\|_\infty \approx \varepsilon^2$, *with* Q_1 *defined like in (11).*

Once we have $x_1(\theta)$, B_1, *and* $C_1(\theta)$, *we keep on iterating until either the norms of* y *and* Q *are small enough or the differences between one step and the previous one are small enough.*

Note that the computation of each pair of coefficients $(u_\kappa^{(c)}, u_\kappa^{(s)})$ in (14) is independent for each κ. The same happens with for each pair $(H_\kappa^{(c)}, H_\kappa^{(s)})$ in (15). That makes these linear systems very suitable for their computational resolution in parallel. With this, the dimension of each linear system depends on the dimension of the phase space and the number of linear systems to be solved on the number of Fourier modes used. On the other hand, the evaluation of the map P and $D_x P$ can be perform independently of each θ, which leads to a straightforward parallelization.

3 Parametrization of invariant manifolds

The quasi-periodic solutions of interest in this work are those that have a saddle part, i.e., those that have un/stable invariant manifolds associated. In that case, the use of the eigenfunctions and eigenvalues associated with their hyperbolic directions are typically used to compute linear approximations to the invariant manifolds. Here, we present a procedure to compute a high-order approximation of the stable or unstable invariant manifolds for a torus in the map (3). In fact, the procedure presented here can be seen as an extension of the one in [JN21] for the computation of the un/stable invariant manifolds of invariant curves.

Let us write the invariant manifold of an invariant torus as a formal Taylor-Fourier expansion in terms of two parameters, a parameter $\sigma \in \mathbb{R}$ and the angle vector $\theta \in \mathbb{T}^d$:

$$W(\theta,\sigma) = a_0(\theta) + a_1(\theta)\sigma + \sum_{k\geqslant 2} a_k(\theta)\sigma^k, \tag{16}$$

where a_k is a function from $\mathbb{T}^d$ to $\mathbb{R}^n$.

Here we consider the case in which σ is one-dimensional. The case of invariant manifolds with several hyperbolic directions requires a parameter σ_j for each direction. In such a case, k in (16) becomes a multi-index and the Taylor-Fourier polynomial manipulations have to be modified accordingly. The case of invariant manifolds with several hyperbolic directions associated with fixed points of Poincaré maps is detailed in [GJJC$^+$21].

In the particular case that the system is Hamiltonian, the stable eigenvalue corresponds to the inverse of the unstable one. And, since many of the classical mechanical systems show a symmetry when inverting the time, sometimes it is possible to have both parametrizations, the stable and the unstable, by just computing one of them and applying the corresponding symmetry. Here, we are going to explain the general case, valid for Hamiltonian and non-Hamiltonian systems and without considering, in advance, any symmetry.

3.1 Computation of the invariant manifold

The invariance condition for an invariant manifold W of a torus of a Poincaré map P like (3) can be written as

$$P(W(\theta,\sigma),\theta) = W(\theta+\rho,\lambda\sigma), \tag{17}$$

where ρ is the frequency vector and λ is a real hyperbolic eigenvalue. The right hand side of (17) can be written as

$$W(\theta+\rho,\lambda\sigma) = a_0(\theta+\rho) + a_1(\theta+\rho)\lambda\sigma + \sum_{k\geqslant 2} a_k(\theta+\rho)(\lambda\sigma)^k.$$

Each function a_k will be obtained by solving the invariance condition (17) order by order. Let us denote by $W_m(\theta,\sigma)$ the truncated Taylor-Fourier series up to order m in a neighbourhood of $\sigma = 0$,

$$W_m(\theta,\sigma) = a_0(\theta) + a_1(\theta)\sigma + \sum_{k=2}^{m} a_k(\theta)\sigma^k. \tag{18}$$

From equations (16) and (17), we see that a_0 is the parametrization of the invariant torus and a_1 is the hyperbolic eigenfunction, that are obtained as truncated real Fourier series following the iterative scheme in the previous section, see also Algorithm 3.2. More precisely, we have obtained the eigenvectors of the Floquet matrix B. Then, if v is the eigenvector of B associated with λ, the eigenfunction a_1 of the torus is $C(\theta)v$. The torus and its eigendirection give the linear approximation to the invariant manifolds, which is the expansion in (16) up to order one.

Assume that we know the parametrization W_{m-1} up to order $m-1$, i.e. the functions $a_0,\ldots,a_{m-1}$ are known. We want to find the function a_m involved in W_m. For this, we apply the Poincaré map to $W_m(\theta,\sigma) = W_{m-1}(\theta,\sigma) + a_m(\theta)\sigma^m$,

$$\begin{aligned} P(W_m(\theta,\sigma),\theta) &= P(W_{m-1}(\theta,\sigma),\theta) + D_xP(W_{m-1}(\theta,\sigma),\theta)a_m(\theta)\sigma^m + \mathcal{O}(\sigma^{m+1}) \\ &= P(W_{m-1}(\theta,\sigma),\theta) + D_xP(a_0(\theta),\theta)a_m(\theta)\sigma^m + \mathcal{O}(\sigma^{m+1}). \end{aligned} \tag{19}$$

The Taylor-Fourier expansion of the invariant manifold W_{m-1} under the Poincaré map is

$$P(W_{m-1}(\theta,\sigma),\theta) = W_{m-1}(\theta+\rho,\lambda\sigma) + b_m(\theta)\sigma^m + \mathcal{O}(\sigma^{m+1}),$$

where b_m is a θ-dependent function at order m that comes from the evaluation of the invariant manifold at order $m-1$. For the computation of this function we make use of the jet transport technique, detailed in Section 5.2. Note that we only need to compute the expansion of $P(W_{m-1}(\theta,\sigma),\theta)$ w.r.t. σ up to order m.

Now we insert the last expression in (19) and impose the invariance condition (17) up to order m:

$$b_m(\theta)\sigma^m + D_xP(a_0(\theta),\theta)a_m(\theta)\sigma^m = a_m(\theta+\rho)\lambda^m\sigma^m.$$

If the number of harmonics used is low, functions a_k can be computed by solving the above linear system in which the matrix $D_xP(a_0(\theta),\theta)$ is involved. If the dimension of the torus is high, so is the number of harmonics and then solving this system directly is not feasible. However, when the system is reducible we can express the previous equation in terms of the Floquet matrix B, such that the final system to solve offers again a high degree of parallelism.

Therefore, we introduce the Floquet change $a_m(\theta) = C(\theta)u_m(\theta)$,

$$b_m(\theta) + D_xP(a_0(\theta),\theta)C(\theta)u_m(\theta) = C(\theta+\rho)u_m(\theta+\rho)\lambda^m,$$

and multiplying by $C^{-1}(\theta+\rho)$ on both sides, it leads to

$$C^{-1}(\theta+\rho)b_m(\theta) + Bu_m(\theta) = u_m(\theta+\rho)\lambda^m. \tag{20}$$

Under the generic condition of non-resonance, detailed in the Lemma 3.1, (20) determines uniquely the function u_m, that gives a_m through the Floquet transformation. So, let us assume that $g_m(\theta) = C^{-1}(\theta+\rho)b_m(\theta)$ admits a (real) Fourier series expansion, that is,

$$g_m(\theta) = \frac{g^{(0)}}{2} + \sum_{\kappa\neq 0} g_\kappa^{(c)}\cos\langle\kappa,\theta\rangle + g_\kappa^{(s)}\sin\langle\kappa,\theta\rangle.$$

Then, we have to find the coefficients of another Fourier expansion

$$u_m(\theta) = \frac{u^{(0)}}{2} + \sum_{\kappa\neq 0} u_\kappa^{(c)}\cos\langle\kappa,\theta\rangle + u_\kappa^{(s)}\sin\langle\kappa,\theta\rangle,$$

such that (20) is satisfied. Imposing the equation (20) on the Fourier coefficients and using a Cramer-block method, we end up with the following linear system of cohomological equations depending on κ,

$$\begin{aligned}
(\lambda^m Id - B)\frac{u^{(0)}}{2} &= \frac{g^{(0)}}{2},\\
(B^2 - 2\lambda^m\cos\langle\kappa,\rho\rangle B + \lambda^{2m}Id)u_\kappa^{(c)} &= (\lambda^m\cos\langle\kappa,\rho\rangle Id + B)g_\kappa^{(c)} - \lambda^m\sin\langle\kappa,\rho\rangle g_\kappa^{(s)},\\
(B^2 - 2\lambda^m\cos\langle\kappa,\rho\rangle B + \lambda^{2m}Id)u_\kappa^{(s)} &= (\lambda^m\cos\langle\kappa,\rho\rangle Id + B)g_\kappa^{(s)} + \lambda^m\sin\langle\kappa,\rho\rangle g_\kappa^{(c)}.
\end{aligned} \tag{21}$$

The linear systems in (21) are solvable as long as $B^{(\kappa)} = B^2 - 2\lambda^m\cos\langle\kappa,\rho\rangle B + \lambda^{2m}Id$ is invertible for all κ. If μ is an eigenvalue of B, then $B^{(\kappa)}$ has eigenvalues of the form

$$\mu^2 - 2\lambda^m\cos\langle\kappa,\rho\rangle\mu + \lambda^{2m}$$

which makes $B^{(\kappa)}$ invertible whenever μ is different to $\lambda^m\exp(\pm\langle\kappa,\rho\rangle\mathrm{i})$.

We have then proved the following lemma:

Lemma 3.1. *Let B be a Floquet matrix associated with the frequency vector ρ on $\mathbb{T}^d$ and let $|\lambda| \neq 1$ be a real number satisfying that for each eigenvalue μ of B and a fixed $m \in \mathbb{N}$, $m \geqslant 2$,*

$$\mu \neq \lambda^m\exp(\pm\langle\kappa,\rho\rangle\mathrm{i}), \qquad \forall\,\kappa\in\mathbb{N}^d. \tag{22}$$

Then for all smooth function g_m on $\mathbb{T}^d$, there exists a unique smooth function u_m such that

$$\lambda^m u_m(\theta+\rho) = Bu_m(\theta) + g_m(\theta). \tag{23}$$

Remark 3.1. Note that (22) is always satisfied when λ is the dominant eigenvalue of the Floquet matrix.

Expressions in (21) recall those in (14) and (15). Therefore, it is clear that the computation of each pair of coefficients $(u_\kappa^{(c)}, u_\kappa^{(s)})$ of u_m is independent to each other. That makes the proposed invariant manifold computation highly parallelizable as it was the Algorithm 2.2. Hence, to find the unknowns $(u_\kappa^{(c)}, u_\kappa^{(s)})$, we solve a large number of small dimensional linear systems at the same time; the dimension of each linear system depends on the dimension of the phase space and the number of systems only depends on the number of Fourier modes used for the approximation of g_m and u_m.

Notice that, when we start the computation of the functions a_k for $k \geqslant 2$, we use the same number of Fourier modes N_i for each of the angular dimensions θ_i with $i = 1, \ldots, d$, as for the torus and the Floquet change Fourier series. However, it may happen that those numbers of modes, that were enough for distretizing accurately the torus and the Floquet change, may not be enough for discretizing some of the parametrization functions a_k for $k \geqslant 2$. If this happens for a given order, it is necessary to increase the number of Fourier modes from this order on. This has not been the case in the examples of Section 6, where we have checked the size of the Fourier modes after the computation. Varying the number of Fourier modes during the computation has an extra penalty that depends on the number of Fourier modes added. An example of a varying number of Fourier modes during the computation of the manifold can be found in [JN21].

Algorithm 3.2 summarizes the process explained above.

Algorithm 3.2 (Invariant manifold of a torus through its Floquet transformation)**.**

- ⋆ ***Input:*** *Discrete system as in* (3)*, torus* $\varphi(\theta)$*, Floquet change* $C(\theta)$*, Floquet matrix* B*, real eigenvalue* $|\lambda| \neq 1$ *of* B*, and its eigenvector* v*.*
- ⋆ ***Output:*** *Coefficients* $a_k(\theta)$ *for* $k \geqslant 2$ *verifying* (17)*.*

1. $a_0(\theta) \leftarrow \varphi(\theta)$.
2. $a_1(\theta) \leftarrow C(\theta)v$.
3. *For* $k = 2, 3, \ldots$
   a) $b_0(\theta) + \cdots + b_k(\theta)\sigma^k \leftarrow P(a_0(\theta) + \cdots + a_{k-1}(\theta)\sigma^{k-1} + \mathbf{0}\sigma^k, \theta)$ *using jet transport.*
   b) $g_k(\theta) \leftarrow C^{-1}(\theta + \rho)b_k(\theta)$.
   c) *Find* $u_k(\theta)$ *such that* $\lambda^k u_k(\theta + \rho) = Bu_k(\theta) + g_k(\theta)$ *using* (21)*.*
   d) $a_k(\theta) \leftarrow C(\theta)u_k(\theta)$.

3.2 Stable invariant manifold

The procedure introduced above is valid for both, the stable and the unstable invariant manifold computation. However, when the Poincaré map is applied forward in time to compute the stable manifold, it approaches the torus and also the unstable invariant manifold. This computation affects the numerical accuracy of the stable manifold by increasing the numerical errors due to the effect of the unstable direction. This effect is more relevant when the unstable direction is strong. Because of that, it is more accurate to obtain the parametrization of the stable invariant manifold using the inverse of the Poincaré map in (3),

$$\begin{cases} x = P^{-1}(\bar{x}, \bar{\theta}), \\ \theta = \bar{\theta} - \rho. \end{cases} \tag{24}$$

In the case of the stable invariant manifold, i.e. the reals $|\lambda| < 1$ eigenvalues of B in (6), the invariance condition (17) is written for (24) as,

$$P^{-1}(W(\theta + \rho, \sigma), \theta + \rho) = W\big(\theta, \frac{\sigma}{\lambda}\big). \tag{25}$$

We proceed as before, we consider a formal power expansion of W in (16) and solve (25) order by order to obtain the functions a_k that parametrize the stable invariant manifold.

Notice that in that case, we must introduce a Floquet change that removes the angle dependence when the dynamics is moving backward in time, i.e. when we apply P^{-1}. The Floquet transformation for the torus, and the torus itself, in the map P and in the map P^{-1} are related through a phase equal to the vector ρ.

Following the Algorithm 2.2, the invariant torus and its eigenfunction (order zero and one of the parametrization) are obtained by application of P. If we want to use them for the parametrization of the stable manifold, where we apply P^{-1}, we have to re-parametrize them as $a_0(\theta+\rho)$ and $a_1(\theta+\rho)$. Therefore, in this case we look for the functions a_k with $k \geqslant 2$ shifted a quantity ρ.

Then, assuming that we know the parametrization up to order $m-1$, in order to find the function $a_m(\theta) = C(\theta)u_m(\theta)$, such that $u_m(\theta)$ satisfies

$$C^{-1}(\theta)b_m^-(\theta) + B^{-1}u_m(\theta+\rho) = \lambda^{-m}u_m(\theta), \tag{26}$$

where $b_m^-(\theta)$ denotes the term of order m resulting from the evaluation of the invariant manifold up to order $m-1$ by the inverse Poincaré map. Now, multiplying by B and by λ^m the last expression, we have

$$\lambda^m u_m(\theta+\rho) = Bu_m(\theta) + g_m(\theta), \tag{27}$$

that has the same form as (23) with $g_m(\theta) = -\lambda^m BC^{-1}(\theta)b_m^-(\theta)$. Therefore, relaying on Lemma 3.1 and the solution of the linear systems in (21), there exists the function $u_m\colon \mathbb{T}^d \mapsto \mathbb{R}^n$, evaluated in $u_m(\theta+\rho)$, that satisfies (27). Note that, the condition in (22) remains the same.

Algorithm 3.3 (Stable invariant manifold of a torus through its Floquet transformation)**.**

- ⋆ ***Input:*** *Discrete system as in* (3)*, torus* $\varphi(\theta)$*, Floquet change* $C(\theta)$*, Floquet matrix* B*, real eigenvalue* $|\lambda| < 1$ *of* B*, and its eigenvector* v*.*
- ⋆ ***Output:*** *Coefficients* $a_k(\theta+\rho)$ *for* $k \geqslant 2$ *verifying* (25)*.*

1. $a_0(\theta+\rho) \leftarrow \varphi(\theta+\rho)$.
2. $a_1(\theta+\rho) \leftarrow C(\theta+\rho)v$.
3. *For* $k = 2, 3, \ldots$
   a) $b_0(\theta) + \cdots + b_k(\theta)\sigma^k \leftarrow P^{-1}(a_0(\theta+\rho) + \cdots + a_{k-1}(\theta+\rho)\sigma^{k-1} + \mathbf{0}\sigma^k, \theta+\rho)$ *using jet transport.*
   b) $g_k(\theta) \leftarrow -\lambda^k BC^{-1}(\theta)b_k(\theta)$.
   c) *Find* $u_k(\theta)$ *such that* $\lambda^k u_k(\theta+\rho) = Bu_k(\theta) + g_k(\theta)$ *using* (21)*.*
   d) $u_k(\theta+\rho) \leftarrow u_k(\theta)$.
   e) $a_k(\theta+\rho) \leftarrow C(\theta+\rho)u_k(\theta+\rho)$.

3.3 Scaling factor

The parametrization of the invariant manifold is not unique. In particular, if $W(\theta,\sigma)$ is a parametrization, then $W(\theta, c\sigma)$ for $c \neq 0$ is also a parametrization. The role of c is to rescale the coefficients $a_k(\theta)$ in (16) as $c^k a_k(\theta)$. To minimize the error propagation we want to avoid the norms of the coefficients to grow or decrease too fast with k [Ric80, FdlL92].

If the radius of convergence w.r.t. σ is known, ϱ, we can use the scaling $c = 1/\varrho$. If ϱ is not known, we can run the algorithm for the manifold twice; one to estimate it and a second one to rescale. Note that if after the first run the estimated radius of convergence is not far from 1, we accept the computed expansion

without the second run. On the other hand, if the radius is far from 1, we can decide to recompute the manifold with a proper scaling.

Finally, a simple way to rescale the expansion is to scale the eigenvector of the Floquet matrix to have norm c and run the algorithm.

4 Multiple shooting

There are invariant objects so unstable that it is impossible to integrate accurately the flow around them during the time involved in the Poincaré map. In these cases, it is convenient to split the time into a certain number of temporal sections, such that the integration time between each two consecutive sections is considerably reduced and so the propagation of the numerical error. These methods are commonly known as multiple shooting or parallel shooting methods and they have been widely used to compute highly unstable periodic orbits [SB02]. A version of this methodology applied to the computation of invariant tori and their Floquet changes can be found in [Olm07].

A multiple shooting method splits the Poincaré map into a finite number of maps such that the composition of them gives the original one. Let $\phi(t;0,x,\theta)$ be the time t flow of (2) from initial time 0 and initial conditions $x \in \mathbb{R}^n$ and $\theta \in \mathbb{T}^d$. The temporal Poincaré map is defined as

$$P(x,\theta) = \phi(\delta;0,x,\theta),$$

with $\delta = 2\pi/\omega_0$. Let us assume that P admits a reducible torus $\varphi\colon \mathbb{T}^d \to \mathbb{R}^n$ with frequency vector $\rho \in \mathbb{T}^d$ given by $\rho = \omega\delta$ being ω in the given model (2). That is, φ must satisfy (4). A multiple shooting method with r sections splits the map P into $P_1,\dots,P_r$ new maps such that their composition gives P. Therefore, the problem consists in finding a reducible torus given in r pieces $\varphi_1,\dots,\varphi_r$ that are the intersections of the torus of the flow with respectively each temporal section.

A standard multiple shooting method is one that considers the r sections equidistantly in time. Let us then define

$$P_j(x,\theta) = \phi(j\delta/r;(j-1)\delta/r,x,\theta), \qquad j=1,\dots,r.$$

There are several options to define the new invariance equations associated with $\varphi_1,\dots,\varphi_r$. For instance,

$$\begin{aligned} &P_j(\varphi_j(\theta),\theta) = \varphi_{j+1}(\theta), \qquad j=1,\dots,r-1,\\ &P_r(\varphi_r(\theta),\theta) = \varphi_1(\theta+\rho), \end{aligned}$$

or another example can be,

$$\begin{aligned} &P_j(\varphi_j(\theta),\theta+(j-1)\rho/r) = \varphi_{j+1}(\theta+j\rho/r), \qquad j=1,\dots,r-1,\\ &P_r(\varphi_r(\theta),\theta+(r-1)\rho/r) = \varphi_1(\theta+\rho). \end{aligned}$$

However, these two examples do not scale with the Algorithms 2.2, 3.2, and 3.3 because, in both cases, the right hand side has no a uniform rotation angle for each $j=1,\dots,r$. That forces us to change other steps in those algorithms besides the only evaluation of the Poincaré map.

Our goal is then to choose an expression that only change the evaluation of the P and D_xP and the other steps in Algorithms 2.2, 3.2, and 3.3 remain the same. The previous proposed invariance equations require to consider a rotation vector of dimension d for each shooting. To keep the same shifting over the r new sections, we propose

$$\begin{aligned} &P_j(\varphi_j(\theta),\theta-(j-1)\rho/r) = \varphi_{j+1}(\theta+\rho/r), \qquad j=1,\dots,r-1,\\ &P_r(\varphi_r(\theta),\theta-(r-1)\rho/r) = \varphi_1(\theta+\rho/r). \end{aligned} \tag{28}$$

Note that one can see a multiple shooting approach as a single shooting but with a larger phase space. The equations (28) do not recover P in (4) by a direct composition since they must be alternated with rotation

operators. Similarly, the φ in (4) can be obtained from the $\varphi_1, \ldots, \varphi_r$ in (28) by undoing rotations. The Lemma 4.1 makes explicit all these rotations and, in particular, says that $D_x P$ is a product of differentials of P_j with interlaced rotations.

Lemma 4.1. *Let α be an angle and let T_α be the operator defined as $T_\alpha x(\theta) = x(\theta + \alpha)$. Then the map P at the torus φ in (4) and P_j in (28) at the torus φ_j are related by*

$$P = T_{\rho-\rho/r} \circ P_r \circ T_{-\rho/r} \circ P_{r-1} \circ \cdots \circ T_{-\rho/r} \circ P_1.$$

4.1 Reducibility and multiple shooting

The linear skew-product associated to the linearization around a torus found with this multiple shooting has a uniform rotation over the sections. Thus, (28) has a linear behaviour expressed by

$$\begin{aligned} \bar{X}_{j+1} &= A_j(\theta) X_j, \qquad j = 1, \ldots, r-1, \\ \bar{X}_1 &= A_r(\theta) X_r, \\ \bar{\theta} &= \theta + \rho/r, \end{aligned} \tag{29}$$

with $A_j(\theta) = D_x P_j(\varphi_j(\theta), \theta - (j-1)\rho/r)$ for $j = 1, \ldots, r$.

Following the Definition 2.1, (29) is reducible if, and only if, there exist a change of variables of the form $X_j = C_j(\theta) Y_j$ for $j = 1, \ldots, r$ such that (29) becomes

$$\begin{aligned} \bar{Y}_{j+1} &= B_j Y_j, \qquad j = 1, \ldots, r-1 \\ \bar{Y}_1 &= B_r Y_r, \\ \bar{\theta} &= \theta + \rho/r, \end{aligned} \tag{30}$$

where the matrices $B_1, \ldots, B_r \in \mathbb{R}^{n \times n}$ are defined by

$$\begin{aligned} B_j &= C_{j+1}(\theta + \rho/r)^{-1} A_j(\theta) C_j(\theta), \qquad j = 1, \ldots, r-1 \\ B_r &= C_1(\theta + \rho/r)^{-1} A_r(\theta) C_r(\theta), \end{aligned} \tag{31}$$

and they do not depend on θ.

Remark 4.1 (Matrix-form). The linearization around a torus can also be formulated in a matrix-block form of a higher dimensional problem, that is, $n \cdot r$ dimension. Thus in practice, Algorithm 2.2 can be used with multiple shooting just considering ρ/r instead of ρ and evaluating P_j and $D_x P_j$ following (28). Indeed, if we consider the block matrices

$$\tilde{A}(\theta) = \begin{pmatrix} A_1 & & & \\ & \ddots & & \\ & & A_{r-1} & \\ & & & A_r \end{pmatrix}(\theta), \qquad \tilde{B} = \begin{pmatrix} B_1 & & & \\ & \ddots & & \\ & & B_{r-1} & \\ & & & B_r \end{pmatrix}, \tag{32}$$

$$\tilde{C}(\theta) = \begin{pmatrix} & C_1 & & \\ & & \ddots & \\ & & & C_{r-1} \\ C_r & & & \end{pmatrix}(\theta), \qquad \tilde{C}^{-1}(\theta) = \begin{pmatrix} C_1^{-1} & & & \\ & \ddots & & \\ & & C_{r-1}^{-1} & \\ & & & C_r^{-1} \end{pmatrix}(\theta),$$

then using a little bit more memory to keep the zeros for each θ we can directly use Algorithm 2.2.

As a consequence of the Remark 4.1, we have the following straightforward lemma.

Lemma 4.2. *Let $B_1, \dots, B_r$ be the matrices in* (31) *and let $\tilde{B}$ be the matrix in* (32)*. Then μ is an eigenvalue of $\tilde{B}$ if, and only if, μ^r is an eigenvalue of $B_r B_{r-1} \cdots B_1$. In other words, the eigenvalues of $\tilde{B}$ are the complex roots of the eigenvalues of $B_r B_{r-1} \cdots B_1$.*

Lemma 4.3 (see §51 in [Wil65])**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be square matrices. Then the spectrum of $\mathcal{AB}$ is the same as the spectrum of $\mathcal{BA}$.*

Combining Lemmas 4.1, 4.2, and 4.3, we prove Proposition 4.4. That result links the relation between the Floquet matrix with the one using the multiple shooting. To prove it, it is enough *i)* to observe that from Lemma 4.1 the spectrum of $D_x P$ on the torus φ has the same spectrum as $(D_x P_r) \cdots (D_x P_1)$, respectively on $\varphi_r, \dots, \varphi_1$. *ii)* The different rotation operators $T_{-\rho/r}$ do not change the spectrum because of the Lemma 4.3. *iii)* The Floquet C_j do not change the spectrum of B_j. Therefore, Lemma 4.2 allows to finish the proof of the Proposition 4.4.

Proposition 4.4. *The eigenvalues of the Floquet matrix of a multiple shooting* (30) *with r sections are the complex r roots of the eigenvalues of the Floquet matrix with single shooting* (6)*.*

4.2 Multiple shooting applied to invariant manifolds

If the torus is very hyperbolic, the linear approximation to the manifold in one of the sections of the torus (single shooting) is enough to globalize the manifold with a good level of accuracy. This is because, as the manifold is very unstable, it is sufficient to use the unstable direction of the torus in one of the sections, say φ_1, to grow numerically the manifold [Ros20]. Other works, as [Dua20], use multiple shooting to compute the linear approximation to the invariant manifold. As here we are interested in a high-order approximation to these manifolds, we need to compute high-order derivatives of the map. Due to the strong instability of the torus, we have to continue with the multiple shooting scheme in order to compute the derivatives of the maps P_j accurately. Therefore, we will compute the Taylor-Fourier expansions for the torus φ_j, $j = 1, \dots, r$.

The parametrization of the manifold, as explained in Section 3, is done at each of the r sections. Let W_j be a formal power expansion for each $j = 1, \dots, r$ of the form

$$W_j(\theta, \sigma) = \sum_{k \geqslant 0} a_{j,k}(\theta) \sigma^k, \qquad \theta \in \mathbb{T}^d.$$

We denote the truncated power expansion of W_j of order m by $W_{j,m}$.

Let us assume, by simplicity, that $|\mu| \neq 1$ is real. Then, applying the invariance condition of the torus to the invariant manifold leads to the equations

$$\begin{aligned} &P_j(W_j(\theta, \sigma), \theta - (j-1)\rho/r) = W_{j+1}(\theta + \rho/r, \mu\sigma), \qquad j = 1, \dots, r-1, \\ &P_r(W_r(\theta, \sigma), \theta - (r-1)\rho/r) = W_1(\theta + \rho/r, \mu\sigma). \end{aligned} \tag{33}$$

The zeroth order in σ of (33) is just the torus φ_j in (28), that is, $a_{j,0} = \varphi_j$. The first order in σ in (33) has the form

$$\begin{aligned} &D_x P_j(a_{j,0}(\theta), \theta - (j-1)\rho/r) a_{j,1}(\theta) = a_{j+1,1}(\theta + \rho/r)\mu, \qquad j = 1, \dots, r-1, \\ &D_x P_r(a_{r,0}(\theta), \theta - (r-1)\rho/r) a_{r,1}(\theta) = a_{1,1}(\theta + \rho/r)\mu. \end{aligned}$$

Using the change $a_{j,1}(\theta) = C_j(\theta) v_j$ for $j = 1, \dots, r$ and the definition of B_j in (31), we end up with

$$\begin{aligned} &B_j v_j = \mu v_{j+1}, \qquad j = 1, \dots, r-1, \\ &B_r v_r = \mu v_1, \end{aligned}$$

and by the matrix-block form in Remark 4.1, we conclude that $v = (v_1, \dots, v_r)$ is an eigenvector of eigenvalue μ of $\tilde{B}$ that, by Proposition 4.4, means that μ is a r root of an eigenvalue of (6).

Let us now assume that we know the functions $a_{j,k}$ for $j = 1, \dots, r$ and $k = 0, \dots, m-1$. Then, using the induction hypothesis, for $j = 1, \dots, r-1$, (and similarly for $j = r$)

$$\begin{aligned} &P_j(W_{j,m-1}(\theta,\sigma), \theta - \frac{j-1}{r}\rho) \\ &= P_j(W_{j,m-1}(\theta,\sigma), \theta - \frac{j-1}{r}\rho) + D_x P_j(W_{j,m-1}(\theta,\sigma), \theta - \frac{j-1}{r}\rho) a_{j,m}(\theta)\sigma^m + \mathcal{O}(\sigma^{m+1}) \\ &= P_j(W_{j,m-1}(\theta,\sigma), \theta - \frac{j-1}{r}\rho) + A_j(\theta) a_{j,m}(\theta)\sigma^m + \mathcal{O}(\sigma^{m+1}) \\ &= W_{j+1,m-1}(\theta + \frac{\rho}{r}, \mu\sigma) + b_{j,m}(\theta)\sigma^m + A_j(\theta) a_{j,m}(\theta)\sigma^m + \mathcal{O}(\sigma^{m+1}), \end{aligned} \tag{34}$$

with $A_j(\theta) = D_x P_j(a_{j,0}(\theta), \theta - (j-1)\rho/r)$. Equating the order σ^m in (34) with the right hand side in (33), we end up with the expressions

$$\begin{aligned} b_{j,m}(\theta) + A_j(\theta) a_{j,m}(\theta) &= a_{j+1,m}(\theta + \rho/r)\mu^m, \qquad j = 1, \dots, r-1, \\ b_{r,m}(\theta) + A_r(\theta) a_{r,m}(\theta) &= a_{1,m}(\theta + \rho/r)\mu^m. \end{aligned}$$

Introducing the Floquet change $a_{j,m}(\theta) = C_j(\theta) u_{j,m}(\theta)$ for all $j = 1, \dots, r$, and using (31), we deduce

$$\begin{aligned} C_{j+1}(\theta + \rho/r)^{-1} b_{j,m}(\theta) + B_j u_{j,m}(\theta) &= \mu^m u_{j+1,m}(\theta + \rho/r), \qquad j = 1, \dots, r-1, \\ C_1(\theta + \rho/r)^{-1} b_{r,m}(\theta) + B_r u_{r,m}(\theta) &= \mu^m u_{1,m}(\theta + \rho/r). \end{aligned} \tag{35}$$

The system of equations (35) can directly be solved as the one in (23) and independently on j.

This scheme works for real un/stable manifolds. For the reasons discussed in Section 3.2, to compute stable manifolds is numerically more precise to consider the inverse Poincaré map. We can then write similar conditions to (33) for P_j^{-1}.

Remark 4.2. The parametrization (33) can also been seen as a single shooting in higher dimension. Thus we can skip the detailed expressions in (35) for each index j and apply directly the Section 3 but with the matrix-block in the Remark 4.1 and with ρ/r instead of ρ. In particular, the Algorithms 3.2 and 3.3 can be used with the penalty of more memory usage.

5 Computer implementation

This section is devoted to provide some technical details and information concerning to the computer implementation of the introduced algorithms.

First, we give some explanations about how to work with Fourier series of several variables and the package used for this aim. Section 5.2 details the idea of the jet transport technique that we use to obtain high-order derivatives of the Poincaré map through automatic differentiation. Then, since this work is focused on the parallelism of computations, Section 5.3 is devoted to the technical details in the implementation of our computations and an analysis of the degree of parallelism achieved. Finally, we include some numerical tests to analyse the accuracy of the obtained results.

5.1 Manipulation of Fourier series in several variables

An effective manipulation of Fourier series has a crucial impact on the performance of Algorithms 2.2, 3.2, and 3.3 to do the steps of shifting by ρ and to solve cohomological equations.

We do not only need to able to express the series in its Fourier coefficients and its tabulation on a grid of angles $\theta \in \mathbb{T}^d$; a process based on the Discrete Fourier Transform. We have to know how to perform

operations affecting to its coefficients. In particular, we need to be able to know at each memory location which is its coefficient and vice versa. These two operations; to know the tuple $(\kappa_1, \dots, \kappa_d)$ from its index and to know index from its tuple, are simple and essential and they are encoded in Algorithms 5.2 and 5.3.

The former depends on how the coefficients are packed, since it is highly dependent on the package. We are going to assume that the coefficients and its tabulation are contiguously allocated in memory but they may be interleaved and/or split in memory. Moreover, we will also assume, for simplicity, that the mesh will have odd size in each of the directions in order to avoid discussions about aliasing phenomena and Nyquist frequencies.

5.1.1 Truncated Fourier series representation

Let x be a real-periodic function admitting a Fourier representation. That this,

$$x(\theta) = \sum_{\kappa \in \mathbb{Z}^d} c_\kappa \exp(-2\pi \mathrm{i} \langle \kappa, \theta \rangle), \qquad c_\kappa \in \mathbb{C} \text{ and } \theta \in \mathbb{T}^d. \tag{36}$$

where $\mathbb{T}$ is identified with $[0, 1)$ or $[-\frac{1}{2}, \frac{1}{2})$. Note that (36) can easily be expressed in terms of sum of $\sin / \cos$ with real coefficients by a change of the basis. It is standard in FFT (Fast Fourier Transform) algorithms to truncate the series in (36) in an even equispaced mesh $\bar{N} = (\bar{N}_1, \dots, \bar{N}_d) \in \mathbb{N}^d$, with $N_j = 2\bar{N}_j + 1$ and $N = (N_1, \dots, N_d)$. Thus, the mesh points are $\{\kappa/N \colon \kappa \in I_N\}$, being I_N the set of indices defined by

$$I_N = \{(\kappa_1, \dots, \kappa_d) \in \mathbb{Z}^d \colon -\bar{N}_j \leqslant \kappa_j \leqslant \bar{N}_j \text{ for all } j = 1, \dots, d\} \tag{37}$$

and, whose cardinal is

$$M = \prod_{j=1}^{d} (2\bar{N}_j + 1) = \prod_{j=1}^{d} N_j. \tag{38}$$

Hence, numerically (36) is equivalent to

$$x(\theta) = \frac{1}{M} \sum_{\kappa \in I_N} \widehat{x}_\kappa \exp(-2\pi \mathrm{i} \langle \kappa, \theta \rangle) \tag{39}$$

where $\widehat{x}_\kappa / M$ is close to the exact Fourier coefficient c_κ.

Since $x(\theta)$ is always real some of the coefficients are redundant, more explicitly $\widehat{x}_\kappa = \widehat{x}_\kappa^*$ with * representing the conjugate as complex numbers. This fact allows us to reduce the coefficients storage to

$$\bar{M} = (\bar{N}_d + 1) \prod_{j=1}^{d-1} N_j. \tag{40}$$

The truncated Fourier series (36), in the equispaced mesh, is in bijection with the values of the function at those points. That is,

$$\frac{1}{M} \sum_{\kappa \in I_N} \widehat{x}_\kappa \exp(-2\pi \mathrm{i} \langle \kappa, \theta \rangle) = \sum_{\kappa \in I_N} x(\theta) \exp(2\pi \mathrm{i} \langle \kappa, \theta \rangle), \qquad \theta \in \{\kappa/N \colon \kappa \in I_N\}.$$

Therefore, numerically it is equivalent to store the values $\{x(\kappa/N) \colon \kappa \in I_N\}$ or the Fourier coefficients $\{\widehat{x}_\kappa \colon \kappa \in I_N\}$. The former contains N real numbers given in (38) and the latter $2\bar{N}$ given in (40) real numbers. Note that N and $2\bar{N}$ differ, and then, both representation can be stored in-place if a padding for the table of values is considered.

5.1.2 Major orders of Fourier coefficients

In the literature one can mainly find two approaches to store the Fourier coefficients depending on the memory order of them. In this work, we have used the one followed in the FFTW3 (Fastest Fourier Transform in the West) package [FJ05] and, in particular, in its `dft_c2r` and `dft_r2c` plans.

In computer science, the ordering of data can generically be in row-major or col-major orders. To detail the access in the Fourier coefficients for an arbitrary dimension, and following the approach in the FFTW3 package, we use the row-major order.

Definition 5.1 (Row-major and col-major orders)**.** Let x be a contiguous allocated N-dimensional vector and let $(N_1, \ldots, N_d)$ be in $\mathbb{N}^d$ so that $M = \prod_{j=1}^{d} N_j$.

1. x is said to be in row-major order if, and only if, for each index $\ell = 0, \ldots, M-1$ of x, there is a unique $(\kappa_1, \ldots, \kappa_d)$ in $\prod_{j=1}^{d} \{0, \ldots, N_j - 1\}$ such that

$$\ell = (\cdots(\kappa_1 N_2 + \kappa_2)N_3 + \cdots)N_d + \kappa_d. \tag{41}$$

2. x is said to be in col-major order if, and only if, for each index $\ell = 0, \ldots, M-1$ of x, there is a unique $(\kappa_1, \ldots, \kappa_d)$ in $\prod_{j=1}^{d} \{0, \ldots, N_j - 1\}$ such that

$$\ell = (\cdots(\kappa_d N_{d-1} + \kappa_{d-1})N_{d-2} + \cdots)N_1 + \kappa_1.$$

Note that we can go from ℓ to its tuple $(\kappa_1, \ldots, \kappa_d)$ using just integer operations. Algorithms 5.2 details the steps to get the unique tuple corresponding to a certain index of a vector x in row-major order (similarly for col-major order).

Algorithm 5.2 (From index to tuple in row-major order)**.**

⋆ ***Input:*** *Integer ℓ and odd integers $(N_1, \ldots, N_d)$.*

⋆ ***Output:*** *Integers $(\kappa_1, \ldots, \kappa_d)$ such that (41) is verified.*

⋆ ***Notation:*** */ denotes the integer division and % the integer modulus.*

1. $j \leftarrow \ell$.

2. For $i = d, \ldots, 1$

a) $\kappa_i \leftarrow j \% N_i$.
b) $j \leftarrow j / N_i$.

Algorithms 5.2 is suitable to recover the tuple of mesh points from the contiguously allocated vector of them. However, the Fourier coefficients are indexed by I_N defined in (37) and the Algorithm 5.2 does not apply directly. A slightly modified version given in Algorithm 5.3 takes into account the different values of the tuple $(\kappa_1, \ldots, \kappa_d)$ in I_N. The inverse process of the Algorithm 5.3 is now straightforward, in practice, this process of given an element in I_N to determine the corresponding index ℓ of the vector (in row-major order) has not been needed in all the implementation of computing the invariant torus, its Floquet, and its invariant manifolds.

Algorithm 5.3 (Fourier coefficients: from index to tuple)**.**

⋆ ***Input:*** *Integer ℓ and odd integers $(N_1, \ldots, N_d)$.*

⋆ ***Output:*** *Integers $(\kappa_1, \ldots, \kappa_d)$ in I_N defined in (37).*

⋆ *Notation:* / *denotes the integer division and* % *the integer modulus.*

1. $j \leftarrow \ell$.
2. *For* $i = d, \dots, 1$
   a) $\kappa_i \leftarrow j\%N_i$.
   b) *If* $\kappa_i > N_i/2$, *then* $\kappa_i \leftarrow N_i - \kappa_i$.
   c) $j \leftarrow j/N_i$.

Algorithm 5.3 allows us to easily apply a rigid rotation to the periodic orbit $x\colon \mathbb{T}^d \to \mathbb{R}$. Indeed, if ρ is in $\mathbb{R}^d$, then

$$x(\theta + \rho) = \sum_{\kappa \in I_N} \exp(-2\pi \mathrm{i} \langle \kappa, \rho \rangle) \widehat{x}_\kappa \exp(-2\pi \mathrm{i} \langle \kappa, \theta \rangle).$$

and with Algorithm 5.3 we can get the tuple κ from the index ℓ of the contiguously allocated vector of size $\bar{M}$ encoding the Fourier coefficients. Similarly, we can easily solve the different cohomology equations in Algorithms 2.2, 3.2, and 3.3.

5.2 Jet transport

Jet transport is a computational technique to obtain high-order derivatives of the flow of an ODE with respect to initial data and/or parameters [BM98, AFJ+08, AFJ+09, ADLBZB10, WZ12]. It is based on applying automatic differentiation [Gri00] to a numerical integrator of ODEs. This is done by substituting the basic arithmetic by an arithmetic of (truncated) formal power series in several variables. A formal power series codifies the value of a function (the constant term) and their derivatives (the coefficients of each monomial) up to a given order (the truncation order of the series), and the propagation of these power series through the numerical integration produces exactly the same results as the integration of the corresponding high-order variational equations of the ODE (see [GJJC+21] for more details).

Here we approximate the un/stable invariant manifolds by truncated Taylor-Fourier series of $W(\theta, \sigma)$ in (16) and we need to compute $P(W(\theta, \sigma), \theta)$ as a truncated Taylor-Fourier series of

$$P(W(\theta, \sigma), \theta) = \sum_{k \geqslant 0} b_k(\theta) \sigma^k, \qquad \theta \in \mathbb{T}^d,$$

where P is a Poincaré map. The idea is to use the equivalence between trigonometric polynomials and a suitable table of values. As the Fourier series a_k in (16) are, in fact, trigonometric polynomials, we can represent them as a suitable tabulation, $\{a_k^{(\ell)}\}_\ell = \{a_k(\theta_\ell)\}_\ell$. Then, the Taylor-Fourier series can be represented as a set of Taylor expansions,

$$W^{(\ell)}(\sigma) = \sum_{k \geqslant 0} a_k^{(\ell)} \sigma^k, \qquad \ell = 0, 1, \dots,$$

Then, we use jet transport to compute $P(W^{(\ell)}(\sigma), \theta_\ell)$,

$$P(W^{(\ell)}(\sigma), \theta_\ell) = \sum_{k \geqslant 0} b_k^{(\ell)} \sigma^k, \qquad \ell = 0, 1, \dots,$$

and, from the coefficients of these Taylor expansions, we can use Fourier transforms to recover the corresponding Fourier series b_k so that we have $P(W(\theta, \sigma), \theta)$. If, for some b_k, the size of the last Fourier modes is not small, it means that more Fourier modes are needed. Then, we restart the calculation using a finer discretization for the corresponding a_k (i.e. a larger number of points θ_ℓ) so that more Fourier modes for b_k are obtained. Consequently, function a_k is also approximated by a higher number of Fourier modes.

5.3 Parallelism

Algorithms 2.2, 3.2, and 3.3 contain steps that are highly parallelizable. Such a parallelism was already exploited in [JO09] in Algorithm 2.2 using the PVM library [GBD+95] running on a cluster of PCs connected through an Ethernet network. Here we use OpenMP 4.5 [Ope15] which runs concurrently in a PC with several CPUs and it provides an easier and efficient parallelism programming.

The use of profilers for the experiments in Section 6 shows that more than the 98% (in both algorithms) is spent in the evaluation of the discrete map P and its derivatives, which involves ODE integrations, and a lower percentage is required to solve the cohomological equations. Therefore, the parallelism strategy has consisted in running the evaluation of the ODE integrator, Taylor [JZ05] and Runge-Kutta-Verner 8(9) [Ver78] in the experiments, sequentially and independently in each of the different available CPUs of the PC. This provides an automatic parallelism since the algorithms require to evaluate the discrete map for each of the different angle values in the mesh in $\mathbb{T}^d$. Note that with this approach the use of jet transport does not provide any downside because we do not parallelize the integrator itself.

The second level of parallelism is in the cohomological equations, the shifting by ρ, and some of the matrix-solvers that are independent to each other either in a Fourier representation or in a table of values.

Finally, we parallelize the transformation between the table of values and the Fourier coefficients (and vice versa). That has been done by the feature already provided in the FFTW3 package and in combination with the OpenMP. We did not detect a major improvement because the package itself is already optimized enough and already the profiler indicated that these transformations do not contribute too much in the performance when one use the FFTW3.

We took care of the potential overhead in the initialization of the threads, that is, the different (sub)processes that are executed in the CPUs. Thus, we initialize the threads at the beginning of the algorithms to have ready the pool of threads and bifurcate the code execution when we reach those parallelizable steps.

5.4 Accuracy tests

In order to ensure that the computations are correct we implement some tests. First, regarding to the torus and the Floquet change, we implement three tests, two of them already introduced in [Olm07], let us call them Test 1, Test 2, and Test 3. Secondly, Test 4, is implemented to assess the parametrization of the hyperbolic invariant manifolds. The Tests 2 and 3 are run after the solutions have been obtained, which means that they can be used to check how good are these solutions in terms of the invariance equations they must satisfied.

In all the tests we are going to use norms and tolerances that must be chosen depending on the model, precision arithmetic, and matching with other tolerances in the algorithms, such as, the one for the Newton's process or the ODE integrations. In Section 6, we will made explicit all these freedoms.

5.4.1 The invariance equation

Algorithm 2.2 stops when the invariance condition (4) for the torus φ and for the Floquet change (7) are satisfied within a certain threshold. On the other hand, Algorithm 3.2 and Algorithm 3.3 are not iterative processes and the steps in each algorithm are deduced by imposing (by power matching in σ) the invariance equations (17) and (25), respectively.

Test 1. Let $\mathcal{A}$ be a mesh in $\mathbb{T}^d$. A function z is said to verify the equation $\mathcal{I}(z(\theta)) = 0$ with tolerance τ if, and only if,

$$\max_{\theta \in \mathcal{A}} \|\mathcal{I}(z(\theta))\| \leqslant \tau.$$

Note that Test 1 can be defined in terms of the relative error instead of the absolute error.

5.4.2 The tail of the Fourier discretization

The test consists in checking that the truncated Fourier representation is accurate enough with the mesh size. We use the fact that, under a smoothness assumption, the Fourier coefficients decay. In the applications in Section 6, these functions are analytic and then their Fourier coefficients decay exponentially. The truncation error is approximated by the size of the last Fourier coefficients in its representation. To prevent potential symmetries that make zero some of the entries, we check the last two indexed coefficients.

Test 2. A truncated real Fourier representation given by

$$x(\theta) = \frac{x^{(0)}}{2} + \sum_{|\kappa|=1}^{N} x_\kappa^{(c)} \cos\langle \kappa, \theta\rangle + x_\kappa^{(s)} \sin\langle \kappa, \theta\rangle, \qquad \theta \in \mathbb{T}^d.$$

is said to verify the Test 2 with tolerance τ if, and only if, for all $\kappa \in \mathbb{N}^d$ such that $|\kappa| = N$ or $|\kappa| = N-1$,

$$\|(x_\kappa^{(c)}, x_\kappa^{(s)})\|_2 \leqslant \tau.$$

We apply the Test 2 for each of the Fourier series involved in the torus, in its Floquet change, and its parametrized manifold. Moreover, Test 2 can be used to keep track which of the components of the angular variables vector $\theta \in \mathbb{T}^d$ have the biggest tail size in norm, and then increase the mesh size on that direction until either we reach a maximum mesh size or we reach the desired tolerance.

5.4.3 The mesh

The third test is computationally more expensive, it consists in checking the function that we want to make zero in a different mesh but with the same size. A way to do this check without the need of using more computational sources is just to perform a fixed shift by an angle, say γ, and then check if the equation is still verified with a prescribed tolerance.

Test 3. Let $\mathcal{A}$ be a mesh in $\mathbb{T}^d$. A function z is said to verify the equation $\mathcal{I}(z(\theta)) = 0$ with tolerance τ and shifting $\gamma \in \mathbb{T}^d$ if, and only if,

$$\max_{\theta \in \mathcal{A}} \|\mathcal{I}(z(\theta + \gamma))\| \leqslant \tau.$$

Note that as Test 1, Test 3 can be defined in terms of the relative error instead of the absolute error. In the case of the torus, the Test 3 consists first in performing the shift $\psi(\theta) = \varphi(\theta+\gamma)$ and then checking (4) but for ψ and with the same original mesh in θ, that is,

$$\max_{\theta \in \mathcal{A}} \|\psi(\theta + \rho) - P(\psi(\theta), \theta + \gamma)\| \leqslant \tau.$$

Similarly, we can apply Test 3 for the Floquet change C, and for the coefficients a_k in the expansion of (18).

5.4.4 The invariant manifolds

Once we have computed the parametrization of the manifold up to the desired order, we check the final accuracy of the approximation of the invariant manifold. This is done by comparing the error of the invariance condition for the parametrization of the invariant manifold, $W(\theta, \sigma)$, at a given angle vector θ, but at two different values of σ, say σ_1 and $\sigma_2 = \sigma_1/2$.

Notice that, when computing the parametrization of the manifold up to order m, the truncation error depends on the power $m + 1$ of the parameter σ.

Test 4. For σ_i, with $i = 1, 2$, we would have that

$$\epsilon_i = |P(W_m(\theta, \sigma_i), \theta) - W_m(\theta + \rho, \lambda\sigma_i)| \approx c\sigma_i^{m+1},$$

where c is a constant. The relation between the two errors is

$$\frac{\epsilon_1}{\epsilon_2} \approx \frac{\sigma_1^{m+1}}{(\frac{\sigma_1}{2})^{m+1}} \approx 2^{m+1}.$$

Therefore, we check that the quantity

$$\frac{\log(\epsilon_1/\epsilon_2)}{\log(2)},$$

has a value close to $m + 1$.

The Test 4 will not numerically be satisfied for all values of σ_1 and m due to round-off and cancellations in ϵ_1, ϵ_2, and ϵ_1/ϵ_2. Thus, we must play with σ_1 and m in order to have enough significant digits to avoid these digit cancellations.

6 Applications

In this section we implement two different applications. The first example is a classical quasi-periodically forced pendulum and the second one is an application to celestial mechanics; a model for the Earth-Moon system subjected to five basic natural frequencies.

In order to stress the independence of the integration method, we use a Taylor integration with jet transport and tolerance 10^{-16} for the first method and a Runge-Kutta-Verner 8(9) with jet transport and tolerance 10^{-14} for the celestial mechanic one. To verify the different tests described in Section 5.4, we consider Euclidean norms for vectors, Fröbenius for matrices, and a generic test tolerance of $\tau = 10^{-10}$.

In all the experiments we have used the `gcc` compiler, version 8.3.0, on a Linux computer with two Intel(R) Xeon(R) CPU E5-2680 @2.70GHz processors, which give a total of 16 cores. For the sake of simplicity, in what follows we use the terms core and processor equally, to refer to a single computational unit.

6.1 A quasi-periodically forced pendulum

This first application considers one of the examples included in [JO09]. The system describes the movement of a quasi-periodically forced pendulum

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -\alpha \sin x + \varepsilon\zeta(\theta_0, \ldots, \theta_d), \\ \dot{\theta}_i &= \omega_i, \quad i = 0, \ldots, d \end{aligned} \tag{42}$$

where $x, y \in \mathbb{R}$, and α is a parameter whose value is chosen as 0.8. For $i = 0, \ldots, d$, $\theta_i \in \mathbb{T}$ and ε accounts for the weight of the forcing function ζ:

$$\zeta(\theta_0, \ldots, \theta_d) = \left[d + 2 + \sum_{i=0}^{d} \cos\theta_i \right]^{-1}.$$

As frequencies we have chosen, with $d = 4$,

$$\omega_0 = 1, \quad \omega_1 = \sqrt{2}, \quad \omega_2 = \sqrt{3}, \quad \omega_3 = \sqrt{5}, \quad \omega_4 = \sqrt{7}. \tag{43}$$

p	Total time	speed-up
1	8m3.286s	1.000
2	4m9.742s	1.934
4	2m18.073s	3.500
8	1m13.779s	6.550
16	39.234s	12.318

Table 1: Computational time needed for computing the torus of system (42) with frequency vector of dimension $d = 4$. First column corresponds to the number of processors used, second one to the total time employed according to the number of processors, and the last one designs the speed-up.

We have applied the methodology summarized in Section 2, Algorithm 2.2, to obtain the torus, the Floquet transformation, and the Floquet matrix near $x = \pi$, $y = 0$ for $\varepsilon = 0.01$. Recalling the Section 1, according to the dimension of the frequency vector selected, d, the dimension of the resulting torus of the flow (42) near the point $(\pi, 0)$ is $d + 1$. By defining a returning map P to the section $\theta_0 = 0 \mod 2\pi$, the dimension of the torus is reduced by one. As initial seeds we used the point $(\pi, 0)$ for the torus, the identity for the Floquet transform, and the differential of P at $(\pi, 0)$ for the Floquet matrix.

Each of the angles has been discretized using $N = 31$ Fourier modes, that results to a total of $2N^4 = 1847042$ unknowns for the torus and $4N^4 = 3694084$ for the Floquet change. Note that a direct method to compute the torus and not using the advantage of the Floquet change needs $4N^8$ memory space which is totally unfeasible.

The Algorithm 2.2 was run with a Newton threshold of 10^{-10} and the Test 1 for the torus is satisfied with 10^{-13} after 3 Newton's iteration and with 10^{-12} for the Floquet transformation and Floquet matrix.

After the Newton convergence and the success in the Test 1, we apply the Test 2 reporting the different values for each of the angular directions, that is, respectively, 10^{-10}, 10^{-11}, 10^{-10}, and 10^{-11}. The Test 3 is also satisfied with 10^{-11} for the torus and 10^{-12} for the Floquet transformation.

The Floquet matrix, B, has hyperbolic real eigenvalues $\lambda_s = 3.625204837874207 \times 10^{-3}$ and $\lambda_u = 2.758464817115549 \times 10^2$. In the Table 1, we show the computational time required for computing the torus using different number of processors. In the same table, the speed-up factor is included. This factor measures the relation between the time needed for solving the system with `p` processors with respect to the time of the linear resolution, that is, using just one processor (`p` = 1). Ideally, when the parallelization is performed with `p` processors, the time should be divided by `p`. We can see in the table that this does not happen, specially when the number of processors increases and so the overhead in each of the processors. Some checks have been done regarding to this; for example, disabling the Hyper-threading of the processors the computational times remained the same. It is noteworthy that the analysis of the profiler to our program shows that 99.76% of the computations have been parallelized.

We compute the approximations to the un/stable invariant manifolds up to order 10 following the Algorithms 3.2 and 3.3. Table 2 shows the required times for these computations using different number of processors and the corresponding values for the speed-up. The Test 1 is satisfied in relative error for each of the order in σ starting with a 10^{-14} at zeroth order to 10^{-11} at order 10. Tests 2, 3, and Test 4 have also been successful at each of the orders.

6.2 A quasi-periodically perturbed model for the Earth-Moon system

G. Gómez, J. J. Masdemont and J. M. Mondelo developed a methodology to generate simplified Solar Systems models (SSSM) using a set of basic frequencies, see [GMM02, Mon01]. The systems of equations introduced in those works describe the motion of a massless particle subjected to a series of time-periodic perturbations. These models are defined in such a way that if we remove all the time-periodic dependencies present in the SSSM, the resulting models correspond to the well-known Restricted Three-Body Problem

	unstable		stable	
p	Total time	speed-up	Total time	speed-up
1	2h33m13s	1.000	2h33m31s	1.000
2	1h19m08s	1.936	1h28m59s	1.725
4	43m12s	3.546	48m23s	3.173
8	22m13s	6.894	25m03s	6.129
16	11m09s	13.746	12m32s	12.245

Table 2: Computational time needed for the un/stable manifolds up to order 10 of the torus in Table 1. First column corresponds to the number of processors used.

(RTBP), [Sze67].

Among the simplified models introduced in [GMM02, Mon01], special attention is paid to the Earth-Moon case, including the gravitational effect of the Sun. For the description of this simplified model they use five basic frequencies for the accurate characterization of the lunar motion. The selection of these frequencies comes from the simplified Brown theory presented in [Esc68]. Their values in terms of cycles per lunar revolution (RTBP adimensional units) are the following:

- mean longitude of the Moon, $\omega_1 = 1$,
- mean elongation of the Moon from the Sun, $\omega_2 = 0.925195997455093$,
- mean longitude of the lunar perigee, $\omega_3 = 8.45477852931292 \times 10^{-3}$,
- longitude of the mean ascending node of the lunar orbit on the ecliptic, $\omega_4 = 4.01883841204748 \times 10^{-3}$,
- Sun's mean longitude of perigee, $\omega_5 = 3.57408131981537 \times 10^{-6}$.

So, this model includes the perturbative effect of the solar gravitational field, the lunar eccentricity, inclination between the orbital plane of the Moon and the ecliptic plane, and also between the orbital and equatorial planes.

In order to generate the model, the authors change these frequencies to a new basis $\nu = (\nu_1, \ldots, \nu_5)$ defined as $\nu_1 = \omega_2$, $\nu_2 = \omega_1 - \omega_3$, $\nu_3 = \omega_1 - \omega_2 + \omega_4$, $\nu_4 = \omega_1 - \omega_5$, and $\nu_5 = \omega_5 - \omega_2$, such that when the frequencies $\nu_1, \ldots, \nu_i$ are added to the unperturbed system (Earth-Moon RTBP), the simplified models SSSM_i are generated for $i = 1, \ldots, 5$, each of them subjected to $\nu_1, \ldots, \nu_i$ perturbations.

The equations of motion for an infinitesimal particle in these models SSSM_i, $i = 1, \ldots, 5$ are introduced in terms of time-dependent functions c_j^i, $j = 1, \ldots, 13$,

$$\begin{cases} \ddot{x} = c_1^i + c_4^i \dot{x} + c_5^i \dot{y} + c_7^i x + c_8^i y + c_9^i z + c_{13}^i \dfrac{\partial \Omega^i}{\partial x}, \\ \ddot{y} = c_2^i - c_5^i \dot{x} + c_4^i \dot{y} + c_6^i \dot{z} - c_8^i x + c_{10}^i y + c_{11}^i z + c_{13}^i \dfrac{\partial \Omega^i}{\partial y}, \\ \ddot{z} = c_3^i - c_6^i \dot{y} + c_4^i \dot{z} + c_9^i x - c_{11}^i y + c_{12}^i z + c_{13}^i \dfrac{\partial \Omega^i}{\partial z}, \end{cases} \tag{44}$$

being

$$\begin{aligned} \Omega^i = \frac{1-\mu}{\sqrt{(x-\mu)^2 + y^2 + z^2}} + \frac{\mu}{\sqrt{(x-\mu+1)^2 + y^2 + z^2}} \\ + \frac{\mu_S}{\sqrt{(x - x_S^i)^2 + (y - y_S^i)^2 + (z - z_S^i)^2}}, \end{aligned} \tag{45}$$

where μ is the Earth-Moon mass parameter, μ_S is the mass of the Sun with respect to the sum of masses of Earth and Moon, and x_S^i, y_S^i, z_S^i denote the positions of the particle with respect to the Sun.

The quasi-periodic time-dependent functions c_j^i can be computed in terms of the positions, velocities, accelerations, and over-accelerations of the two selected primaries. The description of these time-dependent functions as well as the positions x_S^i, y_S^i and z_S^i, consists on a refined Fourier analysis, detailed in [GMS10a, GMS10b]. Note that, regardless of the model i, taking $c_j = 0$ except for $c_5 = 2$, $c_7 = c_{10} = c_{13} = 1$ and omitting the last term in (45), the system of equations in (44) becomes that of the RTBP.

The RTBP presents five equilibrium points ([Sze67]), $L_{1,\ldots,5}$. In the Earth-Moon RTBP, $L_{1,2,3}$, are colinear of center×center×saddle type, and the other two, $L_{4,5}$, form an equilateral triangle having a dynamics of center×center×center. The dynamics of the saddle parts of $L_{1,2}$ are numerically difficult to compute because its unstable parts are of order 10^8 and 10^6 respectively.

The angular dimension of these points increases as the frequencies of the SSSM are included. In the SSSM_1 the equilibrium points become periodic orbits, in the SSSM_2 become two-dimensional quasi-periodic solutions (or 2D tori), and so on. A way of computing these quasi-periodic solutions is to continue them from one SSSM_i to SSSM_{i+1} as the number of considered frequencies increases. This continuation is sometimes difficult due to appearance of resonances. This phenomena was studied in works like [Olm07, HdlL07].

In order to avoid the continuation problems, we add a small dissipation parameter to the equations of the system when continuing from SSSM_i to SSSM_{i+1}. Thus, elliptic eigenvalues become hyperbolic and difficulties of convergence with the algorithm coming from possible resonances are likely removed. Once we have the invariant torus in the system SSSM_{i+1} plus the dissipation parameter, we remove that parameter and refine the invariant object in the original SSSM_{i+1}.

To prevent the numerical difficulties coming from the strong instability in $L_{1,2}$, we use multiple shooting with r sections, in particular, $r = 4$ and $r = 3$ respectively. Then we perform the computation of the torus, its Floquet change, and Floquet matrix until we reach the SSSM_3 model. With this, we have obtained the invariant tori that replace $L_{1,2}$ in the SSSM_3 model, which are tori of dimensions 3 for the flow, and their Floquet matrices. Note that, as we are using multiple shooting, we have computed r sections of the torus. Next, we have computed the unstable manifold of each torus. Table 3 shows the computational times and corresponding speed-up for the approximation of the unstable invariant manifolds up to order 10.

	L_1 unstable		L_2 unstable	
λ_u	1.469645480926268e+02		1.343539917760893e+02	
p	Total time	speed-up	Total time	speed-up
1	16m29s	1.000	6h41m46s	1.000
2	9m44s	1.886	3h58m09s	1.912
4	5m20s	3.450	2h39m36s	3.456
8	2m30s	6.610	1h52m39s	6.596
16	1m18s	12.740	26m02s	13.090

Table 3: Computational time with p CPUs of the unstable manifolds of L_1 and L_2 of SSSM_3 using meshes $N = (43, 43)$ and $N = (223, 223)$, and parallel sections 4 and 3 respectively.

7 Conclusions and future work

This paper has shown that the computation of high-order Taylor-Fourier expansions of un/stable invariant manifolds associated with high-dimensional tori are, nowadays, feasible. Even when the instability of the torus is very strong, where we have combined the algorithms with multiple shooting methods. We have provided explicit algorithms to compute all these invariant objects.

The developed methods look suitable to address computation of invariant manifolds generated by several eigendirections. We plan to modify the current code for such a context as well as to manage some of the eigenvalue cases not included here. Similar ideas can be applied to the case when the frequency vector ρ is not known or even when the internal dynamics is not a fixed rotation ρ, in particular, in a context when the dynamical system is autonomous. Some results in these directions have already been worked in [Olm07].

The method is highly parallelizable to compute torus, the Floquet transformation, and its invariant manifolds. In the experiments, we used OpenMP showing a really good speed-up. We are also aware of other approaches that can take advantage of the intrinsic parallelism of the algorithms such as a GPU approach. We plan exploring in future works a GPU parallelization scheme and providing experiments showing that there is no relevant penalty in the communication between the CPU and the GPUs.

Acknowledgements

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734557, the Spanish grant PGC2018-100699-B-I00 (MCIU/AEI/FEDER, UE), and the Catalan grant 2017 SGR 1374. J.G. has also been supported by the Italian grant MIUR-PRIN 20178CJA2B "New Frontiers of Celestial Mechanics: theory and Applications". B. N. is supported by the Ministry of Economy, Industry and Competitiveness of Spain through the National Plan for I+D+i (MTM2015-67724-R) and through the national scholarship BES-2016-078722. E.O. acknowledges the 'Severo Ochoa Centre of Excellence' accreditation (CEX2019-000928-S).

References

[ADLBZB10] R. Armellin, P. Di Lizia, F. Bernelli-Zazzera, and M. Berz. Asteroid close encounters characterization using differential algebra: the case of Apophis. *Celestial Mech. Dynam. Astronom.*, 107(4):451–470, 2010.

[AFJ+08] E.M. Alessi, A. Farrés, À. Jorba, C. Simó, and A. Vieiro. Efficient usage of self validated integrators for space applications. Ariadna final report, contract no. 20783/07/NL/CB, ESTEC (European Space Agency), 2008.

[AFJ+09] E.M. Alessi, A. Farrés, À. Jorba, C. Simó, and A. Vieiro. Jet transport and applications to NEOs. In *Proceedings of the 1st IAA Planetary Defense Conference: Protecting Earth from Asteroids*, Granada (Spain), 2009.

[And98] M.A. Andreu. *The quasi-bicircular problem.* PhD thesis, Univ. Barcelona, 1998.

[BM98] M. Berz and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliab. Comput.*, 4:361–369, 1998.

[CH17] M. Canadell and À. Haro. Computation of quasi-periodic normally hyperbolic invariant tori: Algorithms, numerical explorations and mechanisms of breakdown. *J. Nonlinear Sci.*, 27:1829–1868, 2017.

[CJ00] E. Castellà and À. Jorba. On the vertical families of two-dimensional tori near the triangular points of the Bicircular problem. *Celestial Mech.*, 76(1):35–54, 2000.

[CRR64] J. Cronin, P.B. Richards, and L.H. Russell. Some periodic solutions of a four-body problem. *Icarus*, 3:423–428, 1964.

[DJS91] C. Díez, À. Jorba, and C. Simó. A dynamical equivalent to the equilateral libration points of the real Earth-Moon system. *Celestial Mech.*, 50(1):13–29, 1991.

[DL95] L. Dieci and J. Lorenz. Computation of invariant tori by the method of characteristics. *SIAM J. Numer. Anal.*, 32(5):1436–1474, 1995.

[DLR91] L. Dieci, J. Lorenz, and R.D. Russell. Numerical calculation of invariant tori. *SIAM J. Sci. Statist. Comput.*, 12(3):607–647, 1991.

[DM98] L. Dagum and R. Menon. OpenMP: an industry standard API for shared-memory programming. *Computational Science & Engineering, IEEE*, 5(1):46–55, 1998.

[Dua20] G. Duarte. *On the dynamics around the collinear points in the Sun-Jupiter System.* PhD thesis, Departament de Matemàtiques i Informàtica. Universitat de Barcelona, Nov 2020.

[Esc68] P. Escobal. *Methods of Astrodynamics.* J. Wiley & Sons, 1968.

[FdlL92] C. Falcolini and R. de la Llave. Numerical calculation of domains of analyticity for perturbation theories in the presence of small divisors. *J. Statist. Phys.*, 67(3-4):645–666, 1992.

[FJ05] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[GBD+95] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. PVM: Parallel Virtual Machine: a users' guide and tutorial for networked parallel computing. *Computers in Physics*, 9:607–607, 1995.

[GJJC+21] J. Gimeno, À. Jorba, M. Jorba-Cuscó, N. Miguel, and M. Zou. Numerical integration of high order variational equations of ODEs. Preprint, 2021.

[GJMS01a] G. Gómez, À. Jorba, J. Masdemont, and C. Simó. *Dynamics and mission design near libration points. Vol. III, Fundamentals: the case of collinear libration points*, volume 4 of *World Scientific Monograph Series in Mathematics.* World Scientific Publishing Co. Inc., 2001.

[GJMS01b] G. Gómez, À. Jorba, J. Masdemont, and C. Simó. *Dynamics and mission design near libration points. Vol. IV, Fundamentals: the case of collinear libration points*, volume 5 of *World Scientific Monograph Series in Mathematics.* World Scientific Publishing Co. Inc., 2001.

[GLMS01a] G. Gómez, J. Llibre, R. Martínez, and C. Simó. *Dynamics and mission design near libration points. Vol. I, Fundamentals: the case of collinear libration points*, volume 2 of *World Scientific Monograph Series in Mathematics.* World Scientific Publishing Co. Inc., 2001.

[GLMS01b] G. Gómez, J. Llibre, R. Martínez, and C. Simó. *Dynamics and mission design near libration points. Vol. II, Fundamentals: the case of collinear libration points*, volume 3 of *World Scientific Monograph Series in Mathematics.* World Scientific Publishing Co. Inc., 2001.

[GMM02] G. Gómez, J. J. Masdemont, and J. M. Mondelo. Solar system models with a selected set of frequencies. *Astronomy & Astrophysics*, 390(2):733–749, 2002.

[GMS10a] G. Gómez, J. M. Mondelo, and C. Simó. A collocation method for the numerical fourier analysis of quasi-periodic functions. i: Numerical tests and examples. *Discrete & Continuous Dynamical Systems - B*, 14(1):41–74, 2010.

[GMS10b] G. Gómez, J. M. Mondelo, and C. Simó. A collocation method for the numerical fourier analysis of quasi-periodic functions. ii: Analytical error estimates. *Discrete & Continuous Dynamical Systems - B*, 14(1):75–109, 2010.

[Gri00] A. Griewank. *Evaluating Derivatives*. SIAM, Philadelphia, Penn., 2000.

[HdlL06] À. Haro and R. de la Llave. A parameterization method for the computation of invariant tori and their whiskers in quasi-periodic maps: numerical algorithms. *Discrete Contin. Dyn. Syst. Ser. B*, 6(6):1261–1300, 2006.

[HdlL07] À. Haro and R. de la Llave. A parameterization method for the computation of invariant tori and their whiskers in quasi-periodic maps: explorations and mechanisms for the breakdown of hyperbolicity. *SIAM J. Appl. Dyn. Syst.*, 6(1):142–207, 2007.

[HKM97] M. Huang, T. Küpper, and N. Masbaum. Computation of invariant tori by the Fourier methods. *SIAM J. Sci. Statist. Comput.*, 18:918–942, 1997.

[HLS12] G. Huguet, R. de la Llave, and Y. Sire. Computation of whiskered invariant tori and their associated manifolds: New fast algorithms. *Discrete Contin. Dyn. Syst.*, 32(4):1309–1353, 2012.

[Hua60] S.S. Huang. Very restricted four-body problem. Technical note TN D-501, Goddard Space Flight Center, NASA, 1960.

[JN21] À. Jorba and B. Nicolás. Using invariant manifolds to capture an asteroid near the L_3 point of the Earth-Moon Bicircular model. *Commun. Nonlinear Sci. Numer. Simul.*, 102:105948, 2021.

[JO09] À. Jorba and E. Olmedo. On the computation of reducible invariant tori on a parallel computer. *SIAM J. Appl. Dyn. Syst.*, 8:1382–1404, 01 2009.

[Jor01] À. Jorba. Numerical computation of the normal behaviour of invariant curves of n-dimensional maps. *Nonlinearity*, 14(5):943–976, 2001.

[JS96] À. Jorba and C. Simó. On quasi-periodic perturbations of elliptic equilibrium points. *SIAM Journal on Mathematical Analysis*, 27(6):1704–1737, 1996.

[JZ05] À. Jorba and M. Zou. A software package for the numerical integration of ODEs by means of high-order Taylor methods. *Exp. Math.*, 14(1):99–117, 2005.

[KAdlL21] B. Kumar, R.L. Anderson, and R. de la Llave. High-order resonant orbit manifold expansions for mission design in the planar circular restricted 3-body problem. *Commun. Nonlinear Sci. Numer. Simul.*, 97:Paper No. 105691, 15, 2021.

[Mon01] J.M. Mondelo. *Contribution to the Study of Fourier Methods for Quasi–Periodic Functions and the Vicinity of the Collinear Libration Points*. PhD thesis, Departament de Matemàtica Aplicada i Anàlisi. Universitat de Barcelona, May 2001.

[Olm07] E. Olmedo. *On the parallel computation of invariant tori*. PhD thesis, Departament de Matemàtica Aplicada i Anàlisi. Universitat de Barcelona, May 2007.

[Ope15] OpenMP Architecture Review Board. OpenMP Application Programming Interface Version 4.5, November 2015.

[Ric80] D.L. Richardson. A note on a Lagrangian formulation for motion about the collinear points. *Celestial Mech.*, 22(3):231–236, 1980.

[Ros20] J.J. Rosales. *On the effect of the Sun's gravity around the Earth-Moon L_1 and L_2 libration points*. PhD thesis, Departament de Matemàtiques i Informàtica. Universitat de Barcelona, Jun 2020.

[SB02] J. Stoer and R. Bulirsch. *Introduction to numerical analysis*, volume 12 of *Texts in Applied Mathematics*. Springer-Verlag, New York, third edition, 2002.

[Sim98] C. Simó. Effective computations in celestial mechanics and astrodynamics. In V.V. Rumyantsev and A.V. Karapetyan, editors, *Modern Methods of Analytical Mechanics and their Applications*, volume 387 of *CISM Courses and Lectures*. Springer Verlag, 1998.

[Sze67] V. Szebehely. *Theory of Orbits*. Academic Press, 1967.

[Ver78] J.H. Verner. Explicit Runge-Kutta methods with estimates of the local truncation error. *SIAM J. Numer. Anal.*, 15(4):772–790, 1978.

[Wil65] J. H. Wilkinson. *The algebraic eigenvalue problem*. Clarendon Press, Oxford, 1965.

[WZ12] D. Wilczak and P. Zgliczyński. Cr-Lohner algorithm. *Schedae Informaticae*, 2011(20):9–42, 2012.