On the computation of reducible invariant tori in a parallel computer

Àngel Jorba and Estrella Olmedo
Departament de Matemàtica Aplicada i Anàlisi
Universitat de Barcelona
Gran Via 585, 08007 Barcelona, Spain
E-mails: <angel@maia.ub.es>, <estrella@maia.ub.es>

19th May 2008

Abstract

We present an algorithm for the computation of reducible quasi-periodic solutions of discrete dynamical systems. The method is based on a quadratically convergent scheme that approximates, at the same time, the Fourier series of the torus, its Floquet transformation and its Floquet matrix. The Floquet matrix describes the linearization of the dynamics around the torus and, hence, its linear stability.

The algorithm presents a high degree of parallelism and the computational effort grows linearly with the number of Fourier modes needed to represent the solution. For these reasons it is a very good option to compute quasi-periodic solutions with several basic frequencies. The paper includes some examples to show the efficiency of the method in a parallel computer.

Contents

1 Introduction

Let us consider a discrete dynamical system that depends on time in a quasi-periodic way,

$$\left.\begin{array}{rcl} \bar{x} & = & f(x,\theta), \\ \bar{\theta} & = & \theta+\omega. \end{array}\right\} \tag{1}$$

We assume that $x \in \mathbb{R}^n$, $\theta \in \mathbb{T}^d$ and f is a smooth diffeomorphism. The components of the frequency vector ω are assumed to be linearly independent over the rationals, that is, $\langle k, \omega \rangle = k_1\omega_1 + \cdots + k_d\omega_d \neq 0$ for all $k \in \mathbb{Z}^d \setminus \{0\}$. Note that, if there exists a $\bar{k} \in \mathbb{Z}^d$ such that $\langle \bar{k}, \omega \rangle = 0$, then one of the components of θ can be expressed as a linear combination of the remaining ones so that the new expression for (1) has the same dynamics as before but the new set of angles has dimension $d-1$.

Note that, due to the translation in the variable θ, the dynamical system (1) cannot have fixed points, nor periodic orbits. The simplest invariant set for this system is an invariant torus parametrized by the angle θ: we will say that (1) has an invariant torus of dimension d, with frequency vector ω, if there exists a smooth injective map $u : \mathbb{T}^d \rightarrow \mathbb{R}^n$ such that

$$u(\theta+\omega) = f(u(\theta), \theta), \quad \forall \theta \in \mathbb{T}^d. \tag{2}$$

Systems of the form (1) appears in many applications. For instance, it is quite common that the first (simplified) model of a physical situation is an autonomous system, and that succesive improvements are based on adding perturbations that depend on time in a periodic way. If they have incommensurable periods, then the resulting perturbation is quasi-periodic and the model (after a suitable section in the case of a flow) takes the form of (1). Good examples come from Celestial Mechanics, where the effect of perturbing bodies can be modelled as the sum of periodic perturbations with incommensurable periods (see, for instance, [GLMS01a, GLMS01b, GJMS01a, GJMS01b]).

The knowledge of these solutions (and their stable and unstable manifolds) is a key step in the study of the dynamics of these systems. As it has been mentioned above, they are the simplest solutions and, in this sense, they play the same role as the equilibrium points in autonomous systems ([DJS91]).

In this paper we focus on the computation of reducible invariant tori of maps, focusing on the case in which the dimension of the tori is larger than one. As we will see in the examples, the method can be used to look for invariant tori of flows by means of a suitable Poincaré section. Let us start by introducing the main concepts involved in this problem.

As for fixed points or periodic orbits, it is natural to consider the linearized normal behaviour around a torus, since it provides the linear approximation to its stable, unstable and centre manifolds. If $h(\theta) \in \mathbb{R}^n$ denotes a small displacement with respect to a point $u(\theta)$ on the torus, we have that

$$f(u(\theta)+h(\theta),\theta) = f(u(\theta),\theta) + D_x f(u(\theta),\theta)h(\theta) + O(\|h\|^2).$$

If we define $A(\theta) = D_x f(u(\theta), \theta)$ and we rename h as x, the linearized normal behaviour around the torus is described by the following linear skew-product,

$$\left.\begin{array}{rcl} \bar{x} & = & A(\theta)x, \\ \bar{\theta} & = & \theta+\omega. \end{array}\right\} \qquad (3)$$

Definition 1 *The system (3) is said to be reducible if and only if there exists a change of variables of the form $z = C(\theta)y$ such that (3) becomes*

$$\left.\begin{array}{rcl} \bar{x} & = & Bx, \\ \bar{\theta} & = & \theta+\omega, \end{array}\right\}$$

where the matrix $B = C^{-1}(\theta+\omega)A(\theta)C(\theta)$ does not depend on θ. The matrix B is called the Floquet matrix and $z = C(\theta)y$ is the Floquet transformation.

Note that the dynamics of reducible systems is easily described by computing the (eigenvalues of the) reduced Floquet matrix. Hence, the linear stability of the torus is completely described by (the eigenvalues of) this Floquet matrix.

In this work we describe a numerical procedure to compute reducible invariant tori and their Floquet matrix. The method is very suitable for a parallel computer, and we will explain our implementation for a cluster of PCs and how it scales with the number of processors. As it has mentioned before, this method can also be applied to flows by using a suitable Poincaré section. Note that a suitable section of the flow reduces the dimension of the computed torus and, as we will see, it also reduces significatively the amount of computations compared to a similar method for flows.

There are several methods in the literature for the computation of two dimensional invariant tori of flows (or, equivalently, invariant curves for maps), among them we mention [DLR91, DJS91, Sim94, DL95, Sim98, CJ00, ERS00, SOV05, SVSO06, HdlL06, AKL07]. There are also numerical methods to approximate the Floquet matrix and Floquet transformation of linear skew-products, see [Jor01, WM06]. We stress that to deal with tori of dimensions larger than 2 for flows (or larger than 1 for maps) is more difficult, essentially because of the increase of the computational effort with the dimension of the torus. On the other hand, the recent growth of computing power is not coming from an increase of the speed of the processors, but on the number of processors available, either in a single computer or through a fast network. For this reason it is quite natural to focus on parallelization issues. A possibility is to implement the same algorithms as before, but taking advantage of a parallel environment (for an example, see [CJ08]). Here we have focus in an algorithm, that requires the torus to be reducible, that offers a high degree of parallelism.

Our algorithm is derived from the proof of the main theorem given in [JS96]. Although the results of [JS96] are for flows (not necessarily Hamiltonian flows), they can be easily translated for maps. In our approach, we represent the parametrization $u(\theta)$ of the torus by means of Fourier series, and we apply a Newton method to look for these Fourier coefficients. It turns out that, if the torus is reducible and we know an approximation to the reducing change and the Floquet matrix, the linear system that appears at each step of the Newton process can be written in block diagonal form (or even in diagonal form!), which is a huge reduction in the computational effort. Therefore, here we try to compute not only the torus but also the Floquet reduction.

As it has been mentioned above, one of the main issues we have addressed here is parallelization. When computing invariant tori of dimensions larger than 1, computing resources usually becomes the main constraint. The method considered here allows for a high degree of parallelism, specially in the parts with higher computation. This includes the evaluation of the equation to solve and its Jacobian, and the solution of the linear system. We have implemented this method to run on a cluster. The coding has been done in ANSI C, with calls to the PVM library for the communications. We include several examples showing the performance of our implementation. A preliminar version of this work can be found in [JO05].

One of the main drawbacks of this approach is that not all the tori are reducible (for concrete examples, see [BS98, Jor01]). However, reducibility is quite common in many situations and, as we show here, taking advantage of it can allow huge savings in computer time and memory.

In this paper we only focus on non-autonomous systems (2), and we look for invariant tori parametrized by the forcing angle θ. In a forthcoming paper ([JO08]) we will show how this method can also be used for autonomous systems,

$$\bar{x} = f(x), \quad x \in \mathbb{R}^n,$$

where the frequency of the torus does not appear explicitly.

The paper is organized as follows: Section 2 contains the description of the method, Section 3 discusses the implementation in a parallel computer and Section 4 contains some examples to illustrate the method.

2 The main iteration

We will assume that there exists an invariant torus for (1), that is, that there exists a smooth function $u : \mathbb{T}^d \to \mathbb{R}^n$ verifying (2). The numerical method is based on a Newton iteration, with quadratic convergence. Therefore, it needs an initial approximation to the desired torus. For the moment being, and to simplify the presentation, we will simply assume that such approximation is available. In the examples we will discuss some possibilities to derive this initial approximation.

Let $\|.\|$ be a norm over $\mathbb{R}^n$ and, if $u : \mathbb{T}^d \to \mathbb{R}^n$ is a smooth function, we use the standard notation $\|u\|_\infty = \sup_\theta \|u(\theta)\|$. We assume that we have a parametrization $x_0 : \mathbb{T}^d \to \mathbb{R}^n$ such that, if we define

$$y_0(\theta) = x_0(\theta + \omega) - f(x_0(\theta)), \tag{4}$$

then $\|y_0\|_\infty$ is small. We will denote this as $\|y_0\|_\infty \approx \varepsilon$.

We also assume that we know a $n \times n$ matrix $C_0(\theta)$ that is also a good approximation to the Floquet transformation. This means that the matrix $C_0^{-1}(\theta+\omega) D_x f(x_0(\theta), \theta) C_0(\theta)$

is close to a constant matrix B_0 or, in other words, if we denote

$$Q_0(\theta) = C_0^{-1}(\theta+\omega) D_x f(x_0(\theta)) C_0(\theta) - B_0, \tag{5}$$

what we are assuming is that $\|Q_0\|_\infty \approx \varepsilon$.

In one step of the iterative procedure, we want to compute a new approximation x_1 to the torus, a new approximation C_1 to its Floquet change and a new approximation B_1 to its reduced matrix, such that the new "remainders"

$$\begin{aligned} y_1(\theta) &= x_1(\theta+\omega) - f(x_1(\theta)), && (6)\\ Q_1(\theta) &= C_1^{-1}(\theta+\omega) D_x f(x_1(\theta)) C_1(\theta) - B_1, && (7) \end{aligned}$$

are of second order w.r.t. the previous ones, that is, $\|y_1\|_\infty \approx \varepsilon^2$ and $\|Q_1\|_\infty \approx \varepsilon^2$.

In a practical implementation of the method we will use truncated Fourier series to represent the approximations to the torus and the Floquet change. For simplicity, we will first describe it using infinite series. The details of the representation in truncated Fourier series are dicussed in Section 3.1.

2.1 On the hypotheses

As it has been mentioned before, the method is based on a proof of existence of such tori. Here we discuss the role of these hypotheses and how they affect the convergence of the numerical method.

A first hypothesis refers to the smoothness of the map. It is quite common to require analyticity, since it simplifies the proofs and it is enough for most of the applications. For these reasons, we will assume that the map (1), and the functions $x_0(\theta)$ and $C_0(\theta)$ giving the initial approximations to the torus and to the Floquet change are real analytic.

A second hypothesis is a Diophantine condition on the frequency vector ω and the eigenvalues $\lambda_1, \dots, \lambda_n$ of the matrix B_0: we assume that there exists real constants $c > 0$ and $\gamma > d-1$ such that

$$\begin{aligned} |\exp(\langle k, \omega\rangle\, \mathrm{i}) - \lambda_j| &> \frac{c}{|k|^\gamma}, \quad \forall k \in \mathbb{Z}^d \setminus \{0\}, \quad j = 1, \dots, n, && (8)\\ |\exp(\langle k, \omega\rangle\, \mathrm{i}) - \frac{\lambda_j}{\lambda_\ell}| &> \frac{c}{|k|^\gamma}, \quad \forall k \in \mathbb{Z}^d \setminus \{0\}, \quad j, \ell = 1, \dots, n, && (9) \end{aligned}$$

where "i" denotes the complex unit. It is clear that condition (8) is always satisfied if all the eigenvalues λ_j have modulis different from 1, and that (9) is always satisfied if all the eigenvalues have different modulus. If there are eigenvalues with modulus 1 and/or eigenvalues with the same modulus, these conditions can be still satisfied (for a general reference, see [Lan91]).

As we will see in the next two sections, the iterative method is based in computing corrections for the approximations to the torus and the Floquet transformation. These corrections are obtained by solving a linear equation that comes from the linearization of the problem. As usual, this equation is solved by expanding in Fourier series w.r.t.

θ. Then, the coefficients of the Fourier series of these corrections contain the divisors $\exp(\langle k, \omega\rangle\, \mathrm{i}) - \lambda_j$ (for the torus) and $\lambda_j \exp(\mathrm{i}\, \langle k, \omega\rangle) - \lambda_\ell$ (for the Floquet change). Hence, we not only need that these values are different from zero but that they cannot get too close to zero when $|k|$ increases (k is the index of the Fourier expansion), because this could destroy the convergence of these series. This is the so-called small divisors problem.[1]

The last hypothesis is a non-degeneracy condition. At each step of the iterative process, the approximation to the reduced Floquet matrix B changes and so do its eigenvalues. This means that asking for conditions (8) and (9) at the beginning it is enough for the first step of the iterative process, but it does not guarantee that they are satisfied for the following steps. To deal with this situation, it is common to ask that the eigenvalues λ_j depend on parameters. Then, playing with the parameters one can recover properties (8) and (9). This allows to show that, for a (Cantor) set of positive Lebesgue measure of values of the parameters, these conditions are satisfied in all the steps and the proof can be completed.

For more details on these topics, we suggest the general works [BHS96, Lla01, Sev03], and references therein.

In principle, to be sure that the method will converge one should check all the conditions of the theorems, including that the value ε that measures the accuracy of the initial approximation is small enough. As we will only apply a finite number of iterations of the method, we can check for the conditions at each step. In particular, for the range of considered Fourier coeffcients (i.e, the range of indices k) we check if the values $|\exp(\langle k, \omega\rangle\, \mathrm{i}) - \lambda_j|$ and $|\lambda_j \exp(\mathrm{i}\, \langle k, \omega\rangle) - \lambda_\ell|$ are very small. If so, the program stops with a message. The error due to the use of truncated Fourier series is discussed in Section 3.2.

2.2 Correcting the approximation to the torus

We assume that we know an approximation to the torus, x_0, and an approximation to its Floquet transformation, C_0, and to its Floquet matrix, B_0. If we define y_0 and Q_0 as in (4) and (5), then we are assuming that $\|y_0\|_\infty$ and $\|Q_0\|_\infty$ are of order ε (here ε denotes an small parameter). To simplify notation, we write this as $\|y_0\|_\infty \approx \varepsilon$ and $\|Q_0\|_\infty \approx \varepsilon$. The first step of the iteration is based in the following property:

Lemma 1 *Let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of B_0, and we assume that they satisfy the Diophantine condition (8) for some constants $c > 0$ and $\gamma > d$. Let us define $g(\theta) = -C_0^{-1}(\theta + \omega) y_0(\theta)$. Then, there exists a function $u : \mathbb{T}^d \to \mathbb{R}^n$ satisfying*

$$u(\theta + \omega) = B_0 u(\theta) + g(\theta), \tag{10}$$

and such that, if $x_1(\theta) = x_0(\theta) + C_0(\theta) u(\theta)$ and y_1 is defined as in (6), we have that

$$\|y_1\|_\infty \approx \varepsilon^2.$$

[1] If all the eigenvalues have modulus different from one, and there are no two of them with the same modulus, these divisors can be bounded from below by a constant and the study becomes much simpler. See, for instance, [BMS76].

Proof: Writing $x_1(\theta) = x_0(\theta) + h(\theta)$, we can use the smoothness of f to look for a linear approximation to h: as

$$y_1(\theta) = x_0(\theta+\omega) + h(\theta+\omega) - f(x_0(\theta)) - D_x f(x_0(\theta))h(\theta) + O_2(h),$$

we look for an h satisfying

$$h(\theta+\omega) = D_x f(x_0(\theta))h(\theta) - y_0(\theta). \tag{11}$$

If we can find such h, with a norm of the same order as y_0, it is clear that we will have $\|y_1\|_\infty \approx \varepsilon^2$. To solve (11), we apply the transformation $h(\theta) = C_0(\theta)u(\theta)$ to obtain

$$u(\theta+\omega) = (B_0 + Q_0(\theta))u(\theta) + g(\theta), \tag{12}$$

where Q_0 has been defined in (5) and $g(\theta) = -C_0^{-1}(\theta+\omega)y_0(\theta)$. It is easy to check that, to get a remainder y_1 of order ε^2, it is enough to compute an approximate solution of (12) with an error of the order of ε^2. Therefore, since it is enough to compute u with an accuracy of order ε, we drop Q_0 and we focus on equation (10).

To see that (10) has a solution, we expand u and g in Fourier series. Although it is usual in theoretical works to expand in complex series, here we expand in real Fourier series. The reason is that we have used real expansions in the computer programs. Hence,

$$\begin{aligned} g(\theta) &= \frac{a_0}{2} + \sum_{k\neq 0} a_k \cos\langle k,\theta\rangle + \sum_{k\neq 0} b_k \sin\langle k,\theta\rangle, \\ u(\theta) &= \frac{\alpha_0}{2} + \sum_{k\neq 0} \alpha_k \cos\langle k,\theta\rangle + \sum_{k\neq 0} \beta_k \sin\langle k,\theta\rangle, \end{aligned}$$

where $\langle k,\theta\rangle = k_1\theta_1 + \cdots + k_d\theta_d$. If we plug these expansions in (10), it is not difficult to check that the unknown Fourier coefficients α_k, β_k must satisfy the following equations,

$$\begin{aligned} (\mathrm{Id} - B_0)\frac{\alpha_0}{2} &= \frac{a_0}{2}, \\ (B_0^2 - 2\cos\langle k,\omega\rangle B_0 + \mathrm{Id})\alpha_k &= (\cos\langle k,\omega\rangle \mathrm{Id} - B_0)a_k - \sin\langle k,\omega\rangle b_k, \\ (B_0^2 - 2\cos\langle k,\omega\rangle B_0 + \mathrm{Id})\beta_k &= (\cos\langle k,\omega\rangle \mathrm{Id} - B_0)b_k + \sin\langle k,\omega\rangle a_k. \end{aligned} \tag{13}$$

The eigenvalues of the matrices $B_0^{(k)} = B_0^2 - 2\cos\langle k,\omega\rangle B_0 + \mathrm{Id}$ ($k \neq 0$) are of the form $\lambda^2 - 2\cos\langle k,\omega\rangle \lambda + 1$, for $\lambda \in \mathrm{Spec}\,(B_0)$, which implies that $B_0^{(k)}$ is regular if all the eigenvalues of B_0 are different from $\exp(\pm\langle k,\omega\rangle \mathrm{i})$. For the convergence of the series we use condition (8) to show the analyticity of u in a subset of the analyticity domain of g, in the same way it is done in [JS96] or [Lla01]. ■

2.3 Correcting the approximation of the Floquet transformation

The linearized flow around the initial approximation is given by

$$\left.\begin{aligned} \bar{x} &= D_x f(x_0(\theta))x, \\ \bar{\theta} &= \theta + \omega. \end{aligned}\right\}$$

We have an approximation for its Floquet change and the reduced Floquet matrix, $C_0(\theta)$ and B_0, satisfying (5) with $\|Q_0\|_\infty \approx \varepsilon$. After the previous step, we have a new approximation to the solution, x_1, and we want to find a new transformation $C_1(\theta)$ and a new reduced matrix B_1 such that (7) is satisfied with $\|Q_1\|_\infty \approx \varepsilon^2$ (this situation has already been considered in several places, see [BMS76, JS92, JRV97]). Let us introduce the matrices R and B_1 as

$$\begin{array}{rcl} R(\theta) & = & C_0^{-1}(\theta+\omega)D_x f(x_1(\theta))C_0(\theta) - B_0, \\ B_1 & = & B_0 + \operatorname{Avg}(R), \end{array} \tag{14}$$

where $\operatorname{Avg}(R)$ denotes the average of the map $\theta \mapsto R(\theta)$, that is,

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

Lemma 2 *Let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of B_1, and we assume that they satisfy the Diophantine condition (9) for some constants $c > 0$ and $\gamma > d$. Let us define $\widetilde{R}(\theta) = R(\theta) - \operatorname{Avg}(R)$. Then, there exists a matrix valued function $H : \mathbb{T}^d \to \mathbb{R}^{n\times n}$ satisfying*

$$H(\theta+\omega)B_1 - B_1 H(\theta) = \widetilde{R}(\theta), \tag{15}$$

and such that, if $C_1(\theta) = C_0(\theta)(\mathrm{Id} + H(\theta))$, the matrix Q_1 defined as in (7) satisfies

$$\|Q_1\| \approx \varepsilon^2.$$

Proof: We look for a new approximation to the Floquet transformation, ε-close to the previous one,

$$C_1(\theta) = C_0(\theta)\left(\mathrm{Id} + H(\theta)\right), \tag{16}$$

where $H(\theta)$ denotes a matrix with norm of order ε. The new transformation $x = C_1(\theta)z$ must reduce the linear skew product

$$\left.\begin{array}{rcl} \bar{x} & = & D_x f(x_1(\theta))x, \\ \bar{\theta} & = & \theta+\omega, \end{array}\right\} \tag{17}$$

to the form

$$\left.\begin{array}{rcl} \bar{z} & = & (B_1 + Q_1(\theta))z, \\ \bar{\theta} & = & \theta+\omega, \end{array}\right\} \tag{18}$$

being Q_1 a matrix with norm of order ε^2.

Let us introduce the matrices R and B_1 as in (14). Applying the transformation $x = C_0(\theta)y$ to (17), we obtain

$$\left.\begin{array}{rcl} \bar{y} & = & \left(B_1 + \widetilde{R}(\theta)\right)y, \\ \bar{\theta} & = & \theta+\omega, \end{array}\right\} \tag{19}$$

where $\widetilde{R}(\theta) = R(\theta) - \operatorname{Avg}(R(\theta))$. Now, if we denote a near identity transformation as

$$y = \left(\mathrm{Id} + H(\theta)\right)u,$$

(H is supposed to be small), and we apply it to (19), we obtain

$$\left.\begin{array}{rcl}\bar{u} & = & (\mathrm{Id}+H(\theta+\omega))^{-1}\left(B_1+\widetilde{R}(\theta)\right)(\mathrm{Id}+H(\theta))\,u\\ \bar{\theta} & = & \theta+\omega.\end{array}\right\} \qquad (20)$$

We are interested in reducing (20) to order ε^2. Note that $(I+H(\theta+\omega))^{-1}$ can be approximated by

$$(\mathrm{Id}+H(\theta+\omega))^{-1}\approx I-H(\theta+\omega),$$

with an error of $O(\|H\|^2)$. This means that (20) can be written as

$$\left.\begin{array}{rcl}\bar{u} & = & \left(B_1-H(\theta+\omega)B_1+B_1H(\theta)+\widetilde{R}(\theta)+O_2\right)u\\ \bar{\theta} & = & \theta+\omega.\end{array}\right\}$$

Hence, if $H(\theta)$ satisfies (15), it is clear that the transformation (16) reduces (17) to the form (18).

To show that (15) has a solution, we expand H and $\widehat{R}$ in complex Fourier series (although for the computations we will use a real expansion, as it will be discussed later). Therefore, we denote

$$H(\theta)=\sum_k H_k\exp(\mathrm{i}\,\langle k,\theta\rangle),\quad \widetilde{R}(\theta)=\sum_{k\neq 0}R_k\exp(\mathrm{i}\,\langle k,\theta\rangle),$$

and we insert these expansions in (15). Equating for each k we have that, for $k=0$, it is enough to take $H_0=0$. For $k\neq 0$, H_k has to satisfy

$$H_kB_1\exp(\mathrm{i}\,\langle k,\omega\rangle)-B_1H_k=R_k. \qquad (21)$$

For each $k\neq 0$, let us define the linear map L_k, acting on the space of constant matrices, as $L_k(H)=HB_1\exp(\mathrm{i}\,\langle k,\omega\rangle)-B_1H$. Without loss of generality, we can assume that B_1 is already in diagonal form, $B_1=\mathrm{diag}\,(\lambda_1,\dots,\lambda_n)$. Then, it is not difficult to show that

$$\ker(L_k)=\{0\}\iff \text{there exist } j,\ell \text{ such that } \lambda_j\exp(\mathrm{i}\,\langle k,\omega\rangle)-\lambda_\ell=0.$$

Taking into account condition (9), it is clear that (21) determines the matrices H_k for all $k\neq 0$. Moreover, (9) allows to prove the convergence of these series and to show that H is analytic on a subset of the domain of analyticity of R. As in Lemma 1, the details can be found in [JS96] or [Lla01]. ■

For the computations, we will use real Fourier series. As $H_0=0$, we write

$$\begin{array}{rcl}H(\theta) & = & \displaystyle\sum_{k\neq 0}H_k^{(c)}\cos\langle k,\theta\rangle+H_k^{(s)}\sin\langle k,\theta\rangle\,,\\ R(\theta) & = & \displaystyle\sum_{k\neq 0}R_k^{(c)}\cos\langle k,\theta\rangle+R_k^{(s)}\sin\langle k,\theta\rangle\,.\end{array}$$

Then, it is not difficult to derive the following expressions

$$\begin{array}{rcl}(H_k^{(c)}\cos\langle k,\omega\rangle+H_k^{(s)}\sin\langle k,\omega\rangle)B_1-B_1H_k^{(c)} & = & R_k^{(c)},\\ (H_k^{(s)}\cos\langle k,\omega\rangle-H_k^{(c)}\sin\langle k,\omega\rangle)B_1-B_1H_k^{(s)} & = & R_k^{(s)}.\end{array} \qquad (22)$$

For each, k, this is a linear system (of dimension $2n^2$) for the unknowns $H_k^{(c)}$ and $H_k^{(s)}$.

2.4 The iterative scheme

This process is repeated until the norm of the remainders, $\|y_k\|_\infty$ and $\|Q_k\|_\infty$, is small enough. The scheme is summarized as follows: We assume we have an approximation $x_0(\theta)$ of a torus such that $\|y_0(\theta)\|_\infty = \|x_0(\theta+\omega) - f(x_0(\theta))\|_\infty \approx \varepsilon$, and an approximation $y = C_0(\theta)x$ to its Floquet transformation, such that it transforms the linear system

$$\begin{aligned} \bar{x} &= D_x f(x_0(\theta))x, \\ \bar{\theta} &= \theta + \omega, \end{aligned}$$

into

$$\begin{aligned} \bar{y} &= (B_0 + Q_0(\theta))y, \\ \bar{\theta} &= \theta + \omega, \end{aligned}$$

with $\|Q_0\|_\infty \approx \varepsilon$. Each iteration of the process has two steps. The first step consists of the following operations:

1.a) Compute $g(\theta) = -C_0^{-1}(\theta + \omega)y_0(\theta)$.

1.b) Find u such that $u(\theta + \omega) = B_0 u(\theta) + g(\theta)$. This is done by expanding g and u in real Fourier series and using (13).

1.c) Compute $h(\theta) = C_0(\theta)u(\theta)$.

1.d) The new approximation to the torus is $x_1(\theta) = x_0(\theta) + h(\theta)$.

The second step corrects the approximation to the Floquet matrix and Floquet transformation. It consists of the following operations:

2.a) Compute $R(\theta) = C_0^{-1}(\theta + \omega)D_x f(x_1(\theta))C_0(\theta) - B_0$, $\widetilde{R}(\theta) = R(\theta) - \mathrm{Avg}\,(R)$ and $B_1 = B_0 + \mathrm{Avg}\,(R)$. B_1 is the new approximation to the Floquet matrix of the torus.

2.b) Find a matrix valued function H such that $H(\theta + \omega)B_1 - B_1 H(\theta) = \widetilde{R}(\theta)$. This is done by expanding R and H in real Fourier series, see (22).

2.c) Compute $C_1(\theta) = C_0(\theta)(\mathrm{Id} + H(\theta))$. C_1 is the new approximation to the Floquet change.

Note that the dimension of the linear systems in the method only depends on the dimension of the phase space, and their number only depends on the number of Fourier modes used in the approximation. As the number of Fourier modes needed to approximate a torus grow exponentially with the dimension of the torus, the method seems suitable for computing tori of higher dimensions.

3 Computer implementation

In this section, we discuss the computer implementation of the algorithm. As it is usual in numerical implementations, we will use truncated Fourier series to represent tori and their Floquet transformations, although for some operations it is better to represent them by a table of values (see Section 3.1). The number of Fourier modes considered (or the number of points in the tables) is determined from the level of accuracy, as discussed in Section 3.2. Finally, in Section 3.3, we discuss some details of the parallel implementation.

3.1 Table of values and Fourier coefficients

Some of the operations in the algorithm are better performed on Fourier series (see (13) and (22)), while some others are better done on tables of values of the functions (the computation of the matrices $D_x f(x_j(\theta))$ and $R(\theta)$, and the products involving them).

To perform these transformations, we have used a multidimensional Discrete Fourier Transform algorithm, as explained in [FJ05]. These techniques are implemented in the well-known library FFTW3 (its home page is `http://www.fftw.org/`), included as an optional package in several GNU/Linux distributions.

We note that the two sets (table of values and truncated Fourier series) have the same size. Hence, truncated series or tables of values will be stored indistinctly in the same array, depending on the step where the algorithm is.

3.2 The control of the error

We have two main sources of error that affect the result:

- The error of the invariance condition on the table of values of the torus (and of the Floquet transformation). This error is easy to control: the iterations explained in Section 2 are stopped when the remainders on this mesh are small.

- The interpolation error, due to the substitution of the torus (and the Floquet change) for a mesh of points. To control this second source of errors we need first to estimate it, and then to be able to change the mesh when this error is too large. In what follows we will discuss these issues.

Let us denote by $\mathcal{A} \subset \mathbb{T}^d$ the set of points corresponding to the tables of values used in the algorithm. Assume that after some iterations we have obtained two sets of values, one for the torus and one for the Floquet transformation, $\{x(\theta_i)\}_{\theta_i \in \mathcal{A}}$ and $\{C(\theta_i)\}_{\theta_i \in \mathcal{A}}$, and a constant Floquet matrix B satisfying

$$\begin{aligned} \max_{\theta_i \in \mathcal{A}} \{\|x(\theta_i + \omega) - f(x(\theta_i))\|\} &< \delta, \\ \max_{\theta_i \in \mathcal{A}} \{\|C^{-1}(\theta_i + \omega) D_x f(x(\theta_i)) C(\theta_i) - B\|\} &< \delta, \end{aligned} \tag{23}$$

where δ is a fixed tolerance. Let us define $\delta_{1,2}$ as

$$\delta_1 = \max_{\theta\in\mathbb{T}^d}\{\|x(\theta+\omega) - f(x(\theta))\|\},$$
$$\delta_2 = \max_{\theta\in\mathbb{T}^d}\{\|C^{-1}(\theta+\omega)D_xf(x(\theta))C(\theta) - B\|\}.$$

Note that the computation of $\delta_{1,2}$ is a difficult task (specially for $d \geq 2$). We will use two different methods to estimate these values. The first method is very fast (it requires almost no work) but the value provided is a rough estimate of the true value. The second method requieres a much higher effort and the result is more reliable. Let us first explain these two methods and then we will discuss how we use them.

The first method consists in looking at the norm of some of the "last" coefficients of the Fourier series, and use it as an estimate for the truncation error of the series. As we are looking for a fast indicator, we have used the last two coefficients along the "lines" $k = (0,\dots,0,k_j,0,\dots,0)$ (for $j = 1,\dots,d$). Once the Newton iteration has converged on a given mesh, we check the size of these coefficients. If one of them is larger than a prescribed threshold, we assume that the interpolation error is too big and we increase the number of Fourier series in the direction of these large coefficients (this is equivalent to refine the mesh of points in the direction of the corresponding angles θ_j). Then, we start again the Newton iterations from the previous approximation, with the new Fourier modes set to zero.

The second test is to evaluate the estimation of the error (23) on a set of values $\tilde{\mathcal{A}} \subset \mathbb{T}^d$ different from the set $\mathcal{A}$ used for the computations. A first option is to use a thinner partition $\tilde{\mathcal{A}}$, to produce a better estimate of the invariance errors $\delta_{1,2}$. The main inconvenience is that the number of points in the mesh grows very fast. For instance, if we double the number of points in each direction θ_j, $j = 1,\dots,d$, the total number of points in the mesh grows by a factor 2^d. Although this test has a natural parallelism, it can take a long time, specially in those cases where the map is given by the Poincaré section of a flow. For this reason, we propose the following alternative: During the iterative procedure, we will use a mesh $\tilde{\mathcal{A}}$ with the same number of points as $\mathcal{A}$. This mesh is obtained by adding a small vector $\gamma = (\gamma_1,\dots,\gamma_d)$ to each point of $\mathcal{A}$. The values γ_j are one half of the distance between the points of $\mathcal{A}$ in the direction θ_j. In this way the new mesh $\tilde{\mathcal{A}}$ is interlaced with the initial mesh $\mathcal{A}$. Then, we check the conditions

$$\max_{\theta_i\in\tilde{\mathcal{A}}}\{\|x(\theta_i+\omega) - f(x(\theta_i),\theta_i)\|\} < \delta,$$
$$\max_{\theta_i\in\tilde{\mathcal{A}}}\{\|C^{-1}(\theta_i+\omega)D_xf(x(\theta_i),\theta_i)C(\theta_i) - B\|\} < \delta.$$

If this test is not satisfied, we add some Fourier coefficients (in all the "directions" θ_i), and we go back to the Newton iteration to refine the solution. If it is satisfied, we can either stop the algorithm and accept the solution or to check it again with a thinner mesh. The idea is to avoid checking with finer meshes during the computations (it is too costly) and to do a single check at the end, to be sure of the accuracy.

3.3 Parallelism

The method has mainly two parallelisable points: the simultaneous resolution of the different linear systems that appear in (13) and (22), and the multiple evaluations of the map f and its differential. This last point could be the most computing intensive part if the evaluation of f and Df requires a lot of operations (think of a Poincaré map of an ODE). These are the places where most of the computational work is done.

Our implementation runs in a cluster of PCs, connected through an Ethernet network, and using the PVM library for the communications (see [GBD$^+$94]). The PVM library is freely accesible from the Internet, and it also comes as an option in several GNU/Linux distributions.

The programming model used is the so called master-slave model: there is a main program (the master) that splits the work into independent pieces and sends them to a set of programs (usually called slaves, but we prefer to refer to them as secondary programs) that do the computations and send the results back to the master (we prefer to call it primary program). This is one of the simplest paradigms for parallel computing, and it is enough for our situation.

In our implementation, the primary program runs in one of the nodes of the cluster, while the secondary programs run in different nodes, on a CPU-per-program basis. We use three different classes of secondary programs:

- Secondary programs of the kind A (SPA). Their task consists in creating and solving the linear systems (13). In this way each SPA receives from the main program a subset of multiindices k. For each k, SPA computes the linear system in (13) and solves it. Then, it returns to the main program an integer. This integer means that it has finished the task. If there are more systems to be solved, the main program send to the SPA another subset of multiindices, otherwise it sends an integer to tell SPA that there are no more systems to solve. Then, SPA returns all the computed solutions to the main program.

- Secondary programs of the kind B (SPB). They take care of the construction and resolution of the linear systems (22). As the SPA, each SPB receives from the main program a subset of multiindices k. Then, for each k, the SPB constructs the linear system in (22) and solves it. After that, it returns to the main program an integer saying that this task has been finished. As before, the main program knows if there are more systems to solve and, if so, it send two integers more. If all the systems have been solved, the main program sends an integer to indicate to SPB that the computation is finished, and then SPB sends all the results to the main program.

- Secondary programs of kind C (SPC). They have the task of evaluating over $\mathcal{A}$ the application f and its differential $D_x f$. SPC also receives from the main program a set of values of the angles θ. SPC computes the evaluation of $f(\theta)$ and $D_x f(x(\theta), \theta)$ for all these θ and asks to the main program for another set. When all the evaluations have been done, SPC sends all the results to the main program.

4 Examples

In this section we present several examples to show the effectiveness of the method.

4.1 A quasi-periodically forced pendulum

Let us consider the following system of ordinary differential equations:

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -\alpha \sin x + \varepsilon q(\theta_0, \dots, \theta_d), \\ \dot{\theta}_i &= \omega_i, \quad i = 0, \dots, d, \end{aligned} \tag{24}$$

where $x \in \mathbb{R}$, $y \in \mathbb{R}$ and, for $i = 0, \dots, d$, $\theta_i \in \mathbb{T}$ and $\omega_i \in \mathbb{R}$. We will use the following function q:

$$q(\theta_0, \dots, \theta_d) = \left[d + 2 + \sum_{i=0}^{d} \cos \theta_i \right]^{-1}. \tag{25}$$

The reason for using this expression and not something simpler like $\sum \cos \theta_i$ is to have a perturbing function with many harmonics of a relevant amplitude. A natural way to get this is to invert a trigonometric polynomial with a zero in the complex plane, not far from the real line (think of the Fourier expansion of the function q in (25)).

Note that, for $\varepsilon = 0$, system (24) has the d-dimensional invariant torus $x = y = 0$, $\theta \in \mathbb{T}^d$. It is known ([JS96, JV97]) that there exists a Cantor set $\mathcal{E}$ of values of ε for which the system (24) has a $(d+1)$-dimensional invariant torus, with frequencies $\omega_0, \dots, \omega_d$. Moreover, the Lebesgue measure of the set $\mathcal{E} \cap [0, \varepsilon_0]$ is exponentially small with ε_0. When ε is small but outside of the set $\mathcal{E}$, the torus can still exist, but with different stability properties and/or does not need to be unique. For a discussion of this, see [BHJ$^+$03].

Here we select a small value for ε, and we will try to compute an invariant torus close to the origin, with frequencies $\omega_0, \dots, \omega_d$. For the computations we will use several values for d, to produce invariant tori of different dimensions. For $d = 1, \dots, 4$, the frequency vector ω is taken as

$$\omega_0 = 1, \quad \omega_1 = \sqrt{2}, \quad \omega_2 = \sqrt{3}, \quad \omega_3 = \sqrt{5}, \quad \omega_4 = \sqrt{7}. \tag{26}$$

Finally, we will use the parameters $\alpha = 0.8$ and $\varepsilon = 0.15$. To apply the methodology described before, let us consider the return map for the section $\theta_0 = 0 \pmod{2\pi}$. If we denote a point on this section by $z = (x, y)$ and $\theta = (\theta_1, \dots, \theta_d)$, we represent the associated return map to the section $\theta_0 = 0 \pmod{2\pi}$ as

$$\left. \begin{aligned} \bar{z} &= P(z, \theta), \\ \bar{\theta} &= \theta + \omega, \end{aligned} \right\} \tag{27}$$

being $\omega = (\tilde{\omega}_1, \dots, \tilde{\omega}_d)$ where, for $i = 1, \dots, d$, we define $\tilde{\omega}_i = 2\pi\omega_i$. The Jacobian of (27) is obtained by means of the numerical integration of the variational equations of (24).

d	Accuracy	Initial	Final
1	1.e-12	21	29
1	1.e-14	21	33
2	1.e-12	441	1225
2	1.e-14	441	1591
3	1.e-12	9261	53235
4	1.e-6	390625	531441
4	1.e-8	390625	1185921
4	1.e-10	390625	1975467

Table 1: The torus for the quasi-periodically forced pendulum. The first column shows the value of d; the second column corresponds to the accuracy of the solution; the third column contains the number of harmonics of the initial approximation; the last column shows the final number of harmonics used to approximate the solution within the accuracy shown in the second column.

We will focus on invariant tori of dimension d, that is, tori that can be parametrized as $z_0 : \mathbb{T}^d \to \mathbb{R}^2$ and that satisfy the equation $z_0(\theta+\omega) = P(z_0(\theta), \theta)$. They are the simplest invariant objects of a system like (27). As it has been mentioned before, the linearization around such a torus is given by (3), where $A(\theta) = D_z P(z_0(\theta), \theta)$.

When $\varepsilon = 0$, the point $x = y = 0$ is invariant by the flow of (24), and this implies that the map P satisfies $P(0) = 0$ (note that, if we consider the variable θ, this set is a d-dimensional invariant torus for (27)). To apply the previous algorithm, we will use $z_0(\theta) \equiv 0$ as an initial approximation to the torus and the Jacobian of P at $z = 0$ (for $\varepsilon = 0$) as an approximation to the reduced matrix (note that, for $\varepsilon = 0$, the Jacobian of P does not depend on θ) and the identity matrix as an approximation to the Floquet transformation. In other words, we use the torus for $\varepsilon = 0$ as an approximation to the torus for ε small.

As the number of Fourier terms needed to get an approximation with a prescribed accuracy is not known in advance, we start with a given number and the algorithm for the control of the error (see Section 3.2) takes care of increasing the number of Fourier terms. Table 1 shows the required number of terms for different values of d and different accuracies. Table 2 displays the eigenvalues of the Floquet matrix of the solutions for $d = 1, 2, 3$ and 4. Note that all these solutions are linearly stable.

Table 3 shows the computing time needed to obtain the solution in four different cases: $d = 1$ and $d = 2$ with accuracy 10^{-14}, $d = 3$ with accuracy 10^{-12}, and $d = 4$ with accuracy 10^{-10}. The second column shows the number of CPUs used (each is a 2.4 GHz Intel Xeon processor). Each processor runs one SPA, one SPB, and one SPC (see Section 3.3). Moreover, one of the processors runs the main program. The third column corresponds to the total time spent by the algorithm (i.e., from the beginning to the end of the run). The time spent by the main program (i.e., the part that has not been parallelized) is shown in the fourth column. The time used in the parallel part is obtained by substracting the third and fourth columns, and it is shown in the fifth one. The last

d	Modulus	Argument
1	1.0000000000000000e+00	-6.7997888309559873e-01
	1.0000000000000000e+00	6.7997888309559873e-01
2	9.9999999999999989e-01	-6.7049041137795473e-01
	9.9999999999999989e-01	6.7049041137795473e-01
3	1.0000000000000000e+00	-6.6756788267641354e-01
	1.0000000000000000e+00	6.6756788267641354e-01
4	1.0000000000000000e+00	-6.6631201608602053e-01
	1.0000000000000000e+00	6.6631201608602053e-01

Table 2: Stability of the torus for the quasi-periodically forced pendulum. The first column contains the value of d, and the second and third column show modulus and argument of the eigenvalues of the Floquet matrix, respectively.

column is the fourth column divided by p, and shows the time that the paralell part should take if the parallelism were perfect. The difference between the last two columns is due to the overheads added by the paralelization (they are mainly communications), and they increase with the number of processors used.

Figure 1 (left) shows the invariant curve of (27) for $d = 1$ in the (z_1, z_2) plane. Figure 1 (right) shows the torus of (27) for $d = 2$. To make this plot, we have taken a uniform "squared" mesh on $\mathbb{T}^2$ and we have plot the image of this mesh through the parametrization of the torus, to make a "geometric" plot. Another option would be to take a trajectory on $\mathbb{T}^2$ (a straight line with the slope given by the frequencies) and to plot the image of this line in the phase space, by means of the parametrization of the torus. Figure 2 shows three slices of the invariant torus for the case $d = 3$, this is, we fix the value of one of the angles (for instance, $\theta_j = 0$) and we draw the resultid 2-d torus as in the previous figure. Finally, Figure 3 shows six sections of the solution in the case $d = 4$. Of course, it is possible to use a 3-d mesh in Figure 2 (or a 4-d mesh in Figure 2) but the resulting plot is less clear. Another possibility is to visualise the torus making a movie of the section, when the value of the coordinate used for the section varies from 0 to 2π.

4.1.1 A disipative situation

Here we consider a disipative perturbation of (24):

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -y - \alpha \sin x + \varepsilon q(\theta_0, \dots, \theta_d), \\ \dot{\theta}_i &= \omega_i, \quad i = 0, \dots, d, \end{aligned} \tag{28}$$

where the function q is defined in (25) and the frequencies ω_i are defined in (26). We also fix $\varepsilon = 0.15$ and $d = 4$.

As it was done in Section 4.1, we use the Poincaré section $\theta_0 \equiv 0$ and we will compute invariant tori for the corresponding Poincaré map P. As before, we will obtain the initial

d	p	Total	Main	Parallel	Optimum
1	1	6.619s	0.502s	6.117s	6.117s
	2	3.675s	0.499s	3.176s	3.059s
2	1	11.214s	0.492s	10.722s	10.722s
	2	5.485s	0.469s	5.016s	5.361s
	3	4.288s	0.501s	3.787s	3.574s
3	1	8m59.267s	0m25.444s	8m33.823s	8m33.823s
	2	4m43.562s	0m25.388s	4m18.174s	4m16.912s
	3	3m17.493s	0m25.298s	3m17.493s	2m51.274s
	4	2m35.243s	0m25.285s	2m9.958s	2m8.455s
	5	2m11.122s	0m25.293s	1m45.829s	1m42.764s
	6	1m52.951s	0m25.445s	1m27.506s	1m25.637s
4	1	109m45.770s	5m47.956s	103m57.814s	103m57.814s
	2	58m5.449s	5m45.136s	52m20.313s	51m58.907s
	4	32m7.966s	5m47.030s	26m22.830s	25m59.454s
	6	23m28.204s	5m47.180s	17m41.024s	17m19.635s
	8	19m8.460s	5m46.636s	13m21.824	12m59.726s
	10	16m55.771s	5m48.277s	11m7.494s	10m23.781s

Table 3: Computing time for the torus of the quasi-periodically forced pendulum. The first column is the value of d, the second column (p) shows the number of processors used, the third column is the total computing time, the fourth is the cpu time of the main program (the non parallel part), the fifth is the difference between the two previous colums, and the last column is the fourt one divided by p.

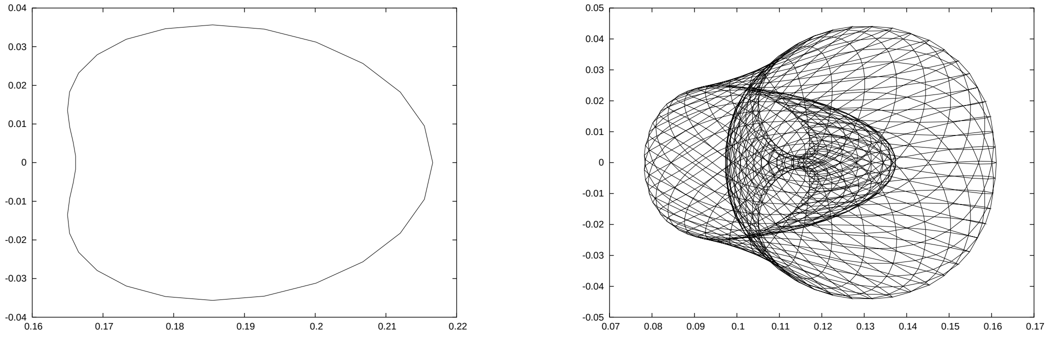

Figure 1: The picture of the found solution in the case of $d = 1$ (on left) and $d = 2$ (on right).

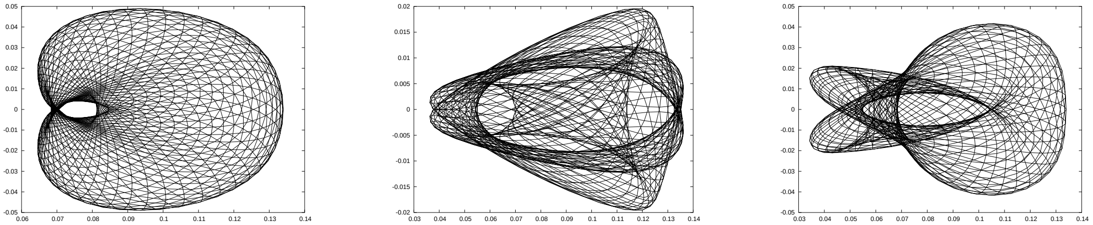

Figure 2: Some slices of the computed torus in the case $d = 3$. Left: $\theta_1 \equiv 0$. Middle: $\theta_2 \equiv 0$. Right: $\theta_3 \equiv 0$.

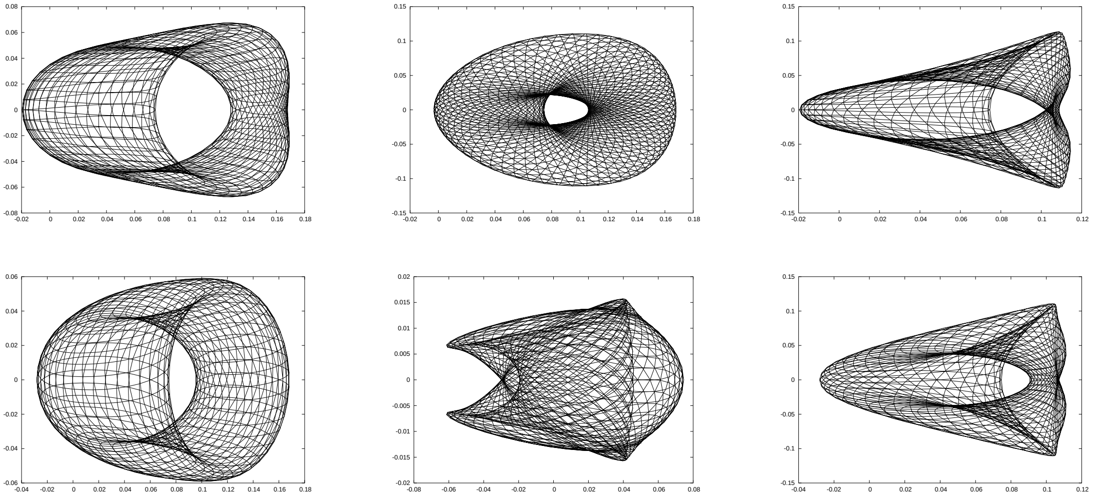

Figure 3: Some slices of the found solution in the case $d = 4$. The first row shows the slices $\theta_1 \equiv \theta_2 \equiv 0$ (left); $\theta_1 \equiv \theta_3 \equiv 0$ (middle); and $\theta_1 \equiv \theta_4 \equiv 0$ (right). The second row correspond to the slices $\theta_2 \equiv \theta_3 \equiv 0$ (left); $\theta_2 \equiv \theta_4 \equiv 0$ (middle); and $\theta_3 \equiv \theta_4 \equiv 0$ (right).

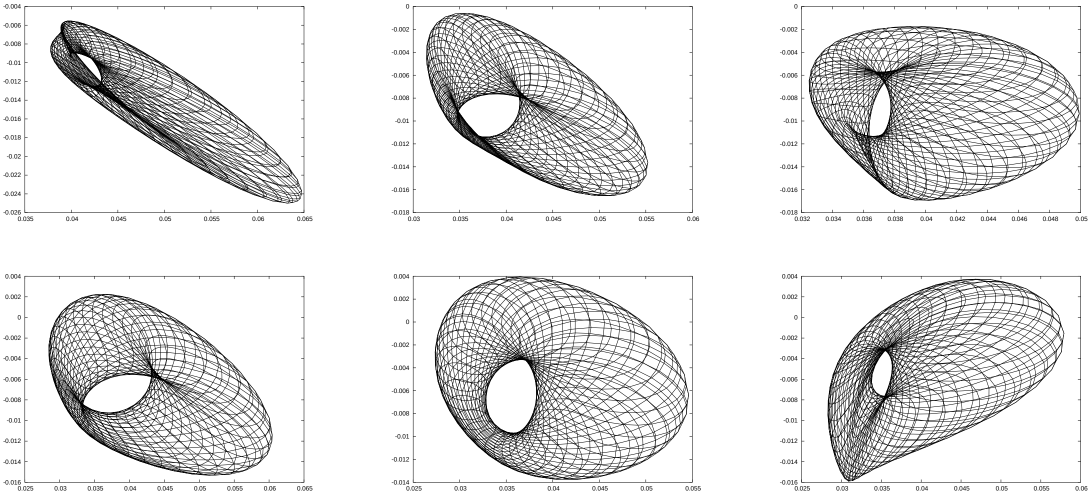

Figure 4: Slices of the found solution in the case $d = 4$ with a disipative perturbation. The first row shows the slices $\theta_1 \equiv \theta_2 \equiv 0$ (left); $\theta_1 \equiv \theta_3 \equiv 0$ (middle); and $\theta_1 \equiv \theta_4 \equiv 0$ (right). The second row correspond to the sections $\theta_2 \equiv \theta_3 \equiv 0$ (left); $\theta_2 \equiv \theta_4 \equiv 0$ (middle); and $\theta_3 \equiv \theta_4 \equiv 0$ (right).

approximation for the method from the case $\varepsilon = 0$: the fixed point $x = y = 0$ is used for the torus, the differential of P at this point for the Floquet matrix, and the identity matrix for the Floquet transformation.

The invariant torus for $\varepsilon = 0.15$ has been computed with an accuracy of 10^{-12}. The number of Fourier terms required to get this accuracy is $1,185,921$. The eigenvalues of the Floquet matrix have modulus $4.3213918263772258 \times 10^{-2}$ and argument ± 1.5161316762139108. We note that, as the divergence of (28) is constant and equal to -1 and the matrix $A(\theta)$ is obtained by integrating the variational flow for 2π units of time (we recall that $\omega_0 = 1$), then $\det A(\theta) \equiv \exp(-2\pi)$. Therefore, the product of the modulus of the two computed eigenvalues should be equal to $\exp(-2\pi)$. As an extra test for the level of accuracy, we compute the difference between this product and $\exp(-2\pi)$, and the result is 7.1×10^{-19}. Figure 4 shows six sections of the solution.

4.2 The multi-circular model

Let us assume that Sun and Jupiter move in a circular orbit around their common centre of mass according to the Kepler laws. The Restricted Three Body Problem (RTBP) describes the motion of a third particle on the vector field produced by these two primaries (Sun and Jupiter). It is assumed that the mass of this third particle is so small that does not affect the motion of the two primaries. It is common to choose units of distance, time and mass such that the distance between the primaries is 1, the period of the primaries is 2π and the sum of masses of Sun and Jupiter is 1. It is also usual to use a rotating coordinate system with origin at the centre of mass, so that the two primaries are kept fixed on the X axis. In this reference system, there are 5 equilibrium points for the particle: three of them (L_1, L_2 and L_3) are on the X axis and the other two (L_4 and L_5) form an equilateral triangle with the primaries (for more details see, for instance, [Sze67]).

We are interested in finding quasi-periodic solutions near L_5 in models that are a perturbation of the RTBP. More concretely, we consider perturbations obtained by adding planets to the system. We will assume that the planets move in circular orbits around the centre of masses of Sun and Jupiter. In particular, we want to consider the direct gravitational effect on the particle due to Saturn, Uranus, Neptune and Earth. The equations of motion for the particle are:

$$\begin{aligned}
\dot{x} &= p_x + y, \\
\dot{y} &= p_y - x, \\
\dot{z} &= p_z, \\
\dot{p}_x &= p_y - (1-\mu)\frac{x-\mu}{r_S^3} - \mu\frac{x-(\mu-1)}{r_J^3} - \\
&\quad -\sum_{i=0}^{d} m_i \frac{x - a_i\cos\theta_i}{r_i^3} + \sum_{i=0}^{d} m_i \frac{\cos\theta_i}{a_i^2} \\
\dot{p}_y &= -p_x - (1-\mu)\frac{y}{r_S^3} - \mu\frac{y}{r_J^3} - \\
&\quad -\sum_{i=0}^{d} m_i \frac{y + a_i\sin\theta_i}{r_i^3} - \sum_{i=0}^{d} m_i \frac{\sin\theta_i}{a_i^2} \\
\dot{p}_z &= -(1-\mu)\frac{z}{r_S^3} - \mu\frac{z}{r_J^3} - \sum_{i=0}^{d} m_i \frac{z}{r_i^3}
\end{aligned} \tag{29}$$

where $\theta_i = \omega_i t + \theta_i^{(0)}$, and ω_i are the frequencies of the perturbing planets, m_i are their masses and a_i denotes their semimajor axis, in these coordinates. Moreover,

$$\begin{aligned}
r_S^2 &= (x-\mu)^2 + y^2 + z^2, \\
r_J^2 &= (x-(\mu-1))^2 + y^2 + z^2, \\
r_i^2 &= (x - a_i\cos\theta_i)^2 + (y + a_i\sin\theta_i)^2 + z^2,
\end{aligned}$$

for $i = 0, \ldots, d$. In our examples d takes values from 1 to 3 ($d+1$ is the number of perturbing planets). The mass parameter μ (that corresponds to the Sun-Jupiter problem) is $9.5388118036309677 \times 10^{-4}$. Table 4 shows the values of the constants, in adimensional units, that appears in the equation. The index $i = 0$ corresponds to Uranus, $i = 1$ to Saturn, $i = 2$ to Neptune and $i = 3$ to the Earth.

As in the previous examples, we will work on the section $\theta_0 \equiv 0$. The corresponding return map depends on d angles (because the system depends on $d+1$ and the section removes one of them). For simplicity, let us abuse the notation and denote by z the vector

i	m_i	ω_i	a_i
0	5.1464752074960787e-05	0.92801432138537394	5.7788760748791814
1	2.8561327940941584e-04	0.59716051258412511	1.8335242195766197
2	4.3620791653297386e-05	0.85880670883551138	3.6880511883932363
3	3.0375324347911519e-06	-0.1862408594090464	0.8923832582416737

Table 4: Masses, frequencies and radius for the planets used in (29).

d	1	2	3
Tolerance	1.e-12	1.e-10	1.e-9
Number of harmonics	89	3555	147825

Table 5: Accuracy and number of harmonics: the first row shows the value of d, the second row is the accuracy of the solution, and the third row contains the number of harmonics needed in the approximation of the solution.

$(x, y, z, p_x, p_y, p_z) \in \mathbb{R}^6$. Moreover, we define $\theta = (\theta_1, \dots, \theta_d)$ and, as before, we call P to the return map to the section $\theta_0 \equiv 0$. In this way, the system (29) can be represented by

$$\left.\begin{array}{rcl} \bar{z} & = & P(z, \theta), \\ \bar{\theta} & = & \theta + \omega, \end{array}\right\} \tag{30}$$

with $\omega = (\tilde{\omega}_i)$ with $\tilde{\omega}_i = 2\pi\omega_i/\omega_0$, for $i = 1, \dots, d$. The differential of P is obtained by means of the numerical integration of the variational equations of (29). Therefore, if $z_0(\theta)$ is a quasi-periodic solution of (30), then the linear dynamics around it is given by (3), where $A(\theta) = D_z P(z_0(\theta), \theta)$.

We are interested in finding an invariant torus of (29) near the equilibria L_5 of the unperturbed system, with the same frequencies as the perturbation. We will use as initial approximation the coordinates of the point L_5 of the Sun-Jupiter RTBP. The lienarization of P at L_5 (again for the Sun-Jupiter RTBP) will be used as the initial guess for the reduced Floquet matrix of the invariant torus of (29) that we want to compute. As for the unperturbed case (the RTBP) the Floquet change is the identity, we will also select the identity as a first approximation for the perturbed situation.

Table 5 shows the number of Fourier terms required to get an approximation of the torus with a given accuracy. In all the cases ($d = 1, 2, 3$) the six eigenvalues of the Floquet matrix are complex conjugated pairs with modulus one, so the tori are linearly stable. In Table 6 the positive argument of these eigenvalues is shown. The first row correspond to the case $d = 1$, the second to the case $d = 2$ and the last one to the case $d = 3$. As an extra test, we compute the difference between the product of the eigenvalues and 1 (the product has to be 1 due to the Hamiltonian structure). This difference is 4.66×10^{-15} for $d = 1$, 2.64×10^{-15} for $d = 2$ and 5.14×10^{-12} for $d = 3$. Finally, Figures 5 and 6 show the tori for the discrete system (30). The tori for the flow (29) can be easily obtained by means of numerical integrations starting on a mesh of points on the tori found for (30).

$d = 1$	$d = 2$	$d = 3$
5.4503461453063606e-01	4.6491650709992588e-01	4.8782691342003148e-01
5.4486199466481278e-01	4.6494463080617310e-01	4.8785025706880447e-01
5.4486200678996954e-01	4.6494454724444878e-01	4.8785033947562140e-01

Table 6: Positive argument of the eigenvalues of the Floquet matrix of the tori found for the multi-circular model.

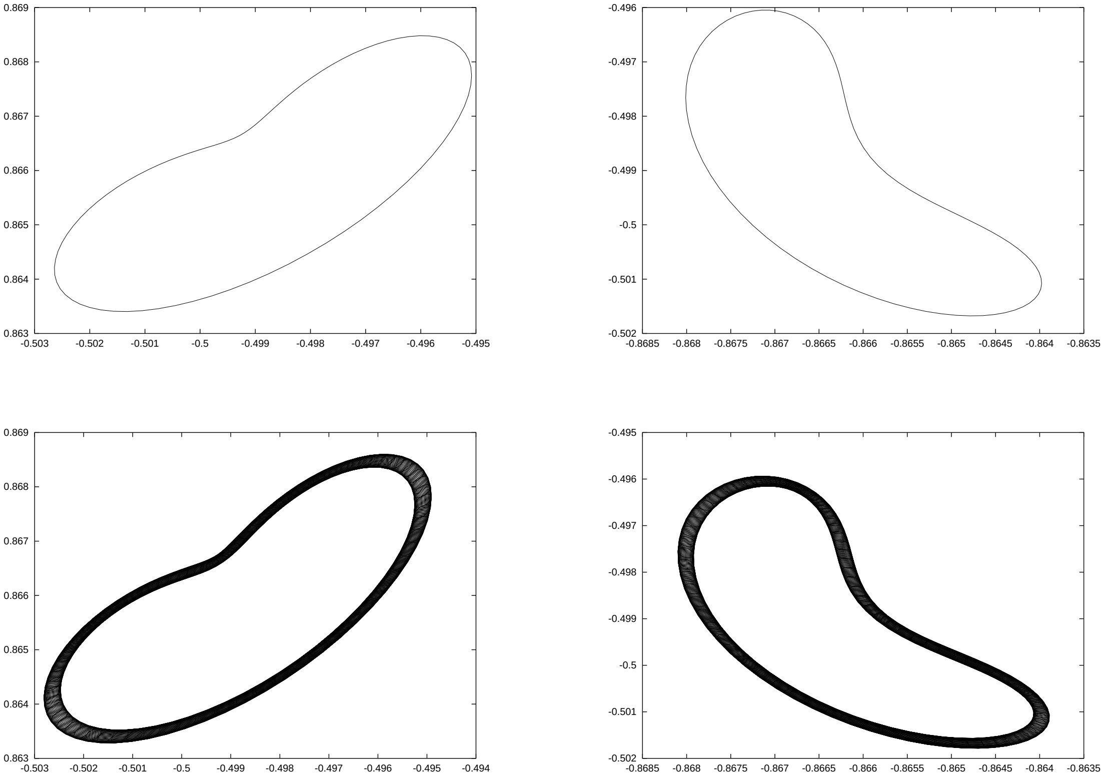

Figure 5: Tori of the multi-circular model for $d = 1$ (first row) and $d = 2$ (second row). First column: (x, y) projection; second column: (p_x, p_y) projection.

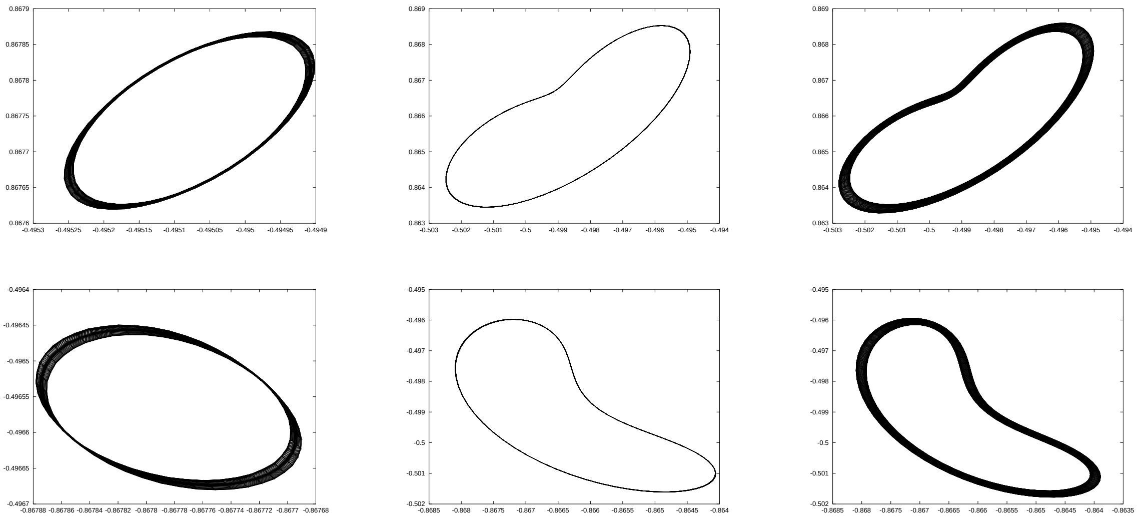

Figure 6: Tori of the multi-circular model for $d = 3$. First row: (x, y) projection; second row: (p_x, p_y) projection. The three columns refer to the slices $\theta_1 = 0$, $\theta_2 = 0$ and $\theta_3 = 0$.

Acknowledgements

E. Olmedo has been supported by a FI grant from the Generalitat de Catalunya. Moreover, this work has been supported by the MEC grant MTM2006-11265 and the CIRIT grant 2005SGR01028.

References

[AKL07] R.A. Adomaitis, I.G. Kevrekidis, and R. de la Llave. A computer-assisted study of global dynamic transitions for a noninvertible system. *Internat. J. Bifur. Chaos Appl. Sci. Engrg.*, 17(4):1305–1321, 2007.

[BHJ+03] H.W. Broer, H. Hanßmann, À. Jorba, J. Villanueva, and F.O.O. Wagener. Normal-internal resonances in quasiperiodically forced oscillators: a conservative approach. *Nonlinearity*, 16:1751–1791, 2003.

[BHS96] H.W. Broer, G.B. Huitema, and M.B. Sevryuk. *Quasi-Periodic Motions in Families of Dynamical Systems: Order amidst Chaos*, volume 1645 of *Lecture Notes in Math.* Springer, New York, 1996.

[BMS76] N. N. Bogoljubov, Ju. A. Mitropoliskii, and A. M. Samoĭlenko. *Methods of Accelerated Convergence in Nonlinear Mechanics.* Springer-Verlag, 1976.

[BS98] H.W. Broer and C. Simó. Hill's equation with quasi-periodic forcing: resonance tongues, instability pockets and global phenomena. *Bol. Soc. Bras. Mat.*, 29(2):253–293, 1998.

[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.

[CJ08] E. Castellà and À. Jorba. On stability regions near the triangular points of the real Earth-Moon system. Preprint, 2008.

[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.

[ERS00] K. D. Edoh, R. D. Russell, and W. Sun. Computation of invariant tori by orthogonal collocation. *Appl. Numer. Math.*, 32(3):273–289, 2000.

[FJ05] M. Frigo and S.G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

[GBD+94] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and S. Sunderam. *PVM: Parallel virtual machine: a users' guide and tutorial for networked parallel computing.* MIT Press, Cambridge, MA, USA, 1994.

[GJMS01a] G. Gómez, À. Jorba, J. Masdemont, and C. Simó. *Dynamics and mission design near libration points. Vol. III, Advanced methods for collinear 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, Advanced methods for triangular 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 triangular libration points*, volume 3 of *World Scientific Monograph Series in Mathematics.* World Scientific Publishing Co. Inc., 2001.

[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 (electronic), 2006.

[JO05] À. Jorba and E. Olmedo. A parallel method to compute quasi-periodic solutions. In *EQUADIFF 2003*, pages 181–183. World Sci. Publ., Hackensack, NJ, 2005.

[JO08] À. Jorba and E. Olmedo. Numerical computation of quasi-periodic motions for the FPU model. Preprint, 2008.

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

[JRV97] À. Jorba, R. Ramírez-Ros, and J. Villanueva. Effective reducibility of quasiperiodic linear equations close to constant coefficients. *SIAM J. Math. Anal.*, 28(1):178–188, 1997.

[JS92] À. Jorba and C. Simó. On the reducibility of linear differential equations with quasiperiodic coefficients. *J. Differential Equations*, 98:111–124, 1992.

[JS96] À. Jorba and C. Simó. On quasiperiodic perturbations of elliptic equilibrium points. *SIAM J. Math. Anal.*, 27(6):1704–1737, 1996.

[JV97] À. Jorba and J. Villanueva. On the persistence of lower dimensional invariant tori under quasi-periodic perturbations. *J. Nonlinear Sci.*, 7:427–473, 1997.

[Lan91] S. Lang. *Introduction to Diophantine Approximations.* Springer, New York, second edition, 1991.

[Lla01] R. de la Llave. A tutorial on KAM theory. In *Smooth ergodic theory and its applications (Seattle, WA, 1999)*, volume 69 of *Proc. Sympos. Pure Math.*, pages 175–292. Amer. Math. Soc., Providence, RI, 2001.

[Sev03] M.B. Sevryuk. The classical KAM theory at the dawn of the twenty-first century. *Moscow Math. J.*, 3(3):1113–1144, 2003.

[Sim94] C. Simó. Averaging under fast quasiperiodic forcing. In J. Seimenis, editor, *Hamiltonian Mechanics: Integrability and Chaotic Behaviour*, volume 331 of *NATO Adv. Sci. Inst. Ser. B Phys.*, pages 13–34. Held in Toruń, Polland, 28 June–2 July 1993. Plenum, New York, 1994.

[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.

[SOV05] F. Schilder, H.M. Osinga, and W. Vogt. Continuation of quasi-periodic invariant tori. *SIAM J. Appl. Dyn. Syst.*, 4(3):459–488 (electronic), 2005.

[SVSO06] F. Schilder, W. Vogt, S. Schreiber, and H.M. Osinga. Fourier methods for quasi-periodic oscillations. *Internat. J. Bifur. Chaos Appl. Sci. Engrg.*, 67(5):629–671, 2006.

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

[WM06] D.B. Wysham and J.D. Meiss. Iterative techniques for computing the linearized manifolds of quasiperiodic tori. *Chaos*, 16(2):023129, 14, 2006.