Computation of quasi-periodic solutions in a parallel computer

Àngel Jorba & Estrella Olmedo
Universitat de Barcelona

Sevilla, May 2004

Purpose:

To compute quasi-periodic solutions in a parallel computer.

Contents:

1) Introduction and concepts (I).

Consider a dynamical system given by:

$$\begin{aligned} \bar{x} &= f(x, \theta), \\ \bar{\theta} &= \omega + \theta, \end{aligned} \qquad (1)$$

with $\theta \in \mathbb{T}^d$, $x \in \mathbb{R}^n$, $\omega \in \mathbb{R}^d$ is known and satisfying that if $(k, \omega) = 2\pi\mathbb{Z}^d$ with $k \in \mathbb{Z}^d$ then $k = 0$.

Definition 1 *(1) has a quasi-periodic solution with rotation number ω when there exists a continuous parametrisation*

$$x : \mathbb{T}^d \to \mathbb{R}^n$$

such that:

$$f(x(\theta), \theta) = x(\theta + \omega), \qquad \forall \theta \in \mathbb{T}^d.$$

1) Introduction and concepts (II).

Definition 2 *A quasi-periodic solution* $x(\theta)$ *with frequency* ω *is* **reducible** *if there exists a change of variables* $x = C(\theta)y$ *(***Floquet transformation***) such that transforms the linear system:*

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

to:

$$\begin{aligned} \bar{y} &= By, \\ \bar{\theta} &= \omega + \theta, \end{aligned}$$

where B *is a matrix that does not depend on* θ *(***Floquet matrix***).*

2) Characteristics of the method.

- Method to compute a **reducible quasi-periodic solution**, its **Floquet matrix** and **Floquet transformation**.

- It is an iterative method with quadratic convergence.

- We need an initial approximation to the solution, the Floquet matrix and Floquet transformation.

- Based on the analytic proof of the paper "On Quasi-periodic Perturbations of Elliptic Equilibrium Points.", by À. Jorba, C. Simó, (1996).

2) The idea of the method.

Initial approximations:

$$y_0 = \bar{x}_0 - f(x_0, \theta) \qquad \|y_0\| \approx \varepsilon$$

$$\left.\begin{array}{rcc} \bar{x} & = & A_0(\theta)x \\ \bar{\theta} & = & \omega + \theta \end{array}\right\} \xrightarrow{x=C_0(\theta)y} \left\{\begin{array}{rcc} \bar{y} & = & (B_0 + Q_0(\theta))y \\ \bar{\theta} & = & \omega + \theta \end{array}\right.$$

$$\|Q_0(\theta)\| \approx \varepsilon, \qquad A_0(\theta) = D_x f(x_0(\theta), \theta)$$

$$\downarrow$$

After one step:

$$y_1 = \bar{x}_1 - f(x_1, \theta), \qquad \|y_1\| \approx \varepsilon^2$$

$$\left.\begin{array}{rcc} \bar{x} & = & A_1(\theta)x \\ \bar{\theta} & = & \omega + \theta \end{array}\right\} \xrightarrow{x=C_1(\theta)y} \left\{\begin{array}{rcc} \bar{y} & = & (B_1 + Q_1(\theta))y \\ \bar{\theta} & = & \omega + \theta \end{array}\right.$$

$$\|Q_1(\theta)\| \approx \varepsilon^2, \qquad A_1(\theta) = D_x f(x_1(\theta), \theta)$$

2) Improving the solution:

We want:

$$\boxed{x_1(\theta) = x_0(\theta) + h(\theta) \text{ with } h(\theta) \text{ ``small''.}}$$

$$\downarrow$$

$h(\theta)$ must satisfy:

$$\boxed{h(\theta + \omega) = A_0(\theta)h(\theta) - y_0(\theta).}$$

$$\downarrow$$

We apply:

$$\boxed{h(\theta) = C_0(\theta)u(\theta)}$$

$$\downarrow$$

We obtain:

$$\boxed{u(\theta + \omega) = B_0 u(\theta) + r(\theta)}$$

$$\downarrow$$

We solve:

$$\boxed{2N + 1 \text{ independent linear systems } n \times n}$$

2) Improving the Floquet matrix:

We want:

$$\boxed{C_1(\theta) = C_0(\theta)(I + H(\theta)) \text{ with } H(\theta) \text{ ``small''}.}$$

$$\downarrow$$

We want to reduce until order ε^2:

$$\boxed{\bar{x} = A_1(\theta)x, \ \bar{\theta} = \omega + \theta.}$$

$$\downarrow$$

We apply:

$$\boxed{x(\theta) = C_0(\theta)(I + H(\theta))z(\theta)}$$

$$\downarrow$$

We obtain:

$$\boxed{H(\theta + \omega)B_1 = B_1 H(\theta) + R(\theta)}$$

$$\downarrow$$

We solve:

$$\boxed{N \text{ independent linear systems } 2n^2 \times 2n^2}$$

3.1) Parallelism on the implementation:

- **Master-Slave Model:** The *Master* distributes the tasks to the *Slaves* and the *Slaves* make the hardest work.

- The program has three kinds of *Slaves*:
  1.) Solving the linear systems to find the new approximation of the solution.
  2.) Solving the linear systems to find the new approximation of the Floquet matrix.
  3.) Evaluating the map and its differential.

- The program runs on a cluster of PCs.

- We have used the PVM library for the communications.

3.2) Table of values and Fourier coefficients.

Two discrete representations of $x(\theta)$:

a) Evaluations over a discrete set of values $\{x(\theta_i)\}_{\theta_i \in \mathcal{A}}$ with

$$\mathcal{A} = \{\theta_i \in \mathbb{T}^d : \theta_i = (\theta^j_{i_j}),\, \theta^j_{i_j} = \frac{i_j}{2K_j + 1} 2\pi\}$$

with $0 \leq i_j \leq 2K_j$, $1 \leq j \leq d$.

b) Truncation of a Fourier series

$$x(\theta) = \frac{\alpha_0}{2} + \sum_{k \in \mathcal{Z}} \alpha_k \cos(k\theta) + \beta_k \sin(k\theta),$$

with

$$\mathcal{Z} = \{k \in \tilde{\mathbb{Z}}^d :\, k = (k_j),\, |k_j| \leq K_j\}$$

with

$$\tilde{\mathbb{Z}}^d = \{\mathbb{Z}^{d-1} \times \mathbb{N}\} \cup \{\mathbb{Z}^{d-2} \times \mathbb{N} \times \{0\}\} \cup \cdots \cup \{\mathbb{N} \times \{0\}^{d-1}\}$$

and $1 \leq j \leq d$.

3.3) Other remarks and comments about the implementation.

1. The algorithm uses both representations.

2. $\#\{\theta_i \in \mathcal{A}\} = \#\{\frac{\alpha_0}{2}, \alpha_k, \beta_k, k \in \mathcal{Z}\} = N$ with $N = \prod_{i=1}^{d}(2K_j + 1)$.

3. Defining an order over $\mathcal{A}$ and $\mathcal{Z}$, to store the dates in a array.

4. Using *fftw3.h* in order to make the Fourier Transform.

4) Control of the error.

Given a tolerance (TOL), the results pass two different tests before being accepted as good solution of the problem:

1. First test: Controlling the norm of the highest order harmonics in each direction. If this norm is greater than TOL we will increase the number of harmonics taken in the worse approximated direction. Otherwise, the approximation passes the first test.

2. Second test: Evaluating the solution in a different set of angles $\tilde{\mathcal{A}}$. Consider $\tilde{\mathcal{A}}$ as the displacement of a half of the distance between the points of $\mathcal{A}$ in each direction of the angle. If the evaluations over $\tilde{\mathcal{A}}$ give results with accuracy less than TOL we will increase the number of harmonics in every direction of the angle. Otherwise the approximation is accepted as good solution of the problem.

5) Application: A perturbation of the pendulum.

Consider

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= -\alpha \sin(x) + \frac{\varepsilon}{d + 2 + \sum_{i=0}^{d} \cos(\theta_i)}, \end{aligned}$$

with $\alpha = 0.8$, $d = 1, \ldots, 4$, $\theta_i = \omega_i t + \theta_i^0$, θ_i^0 the initial phases and

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

How to apply the method:

We consider the Poincare section $\theta_0 \equiv 0$ and take the diffeomorphism given by the evaluation of the flow at time $2\pi/\omega_0$.

The initial approximation:
As approximation of the solution $\to$ the fixed point,
As approximation of the Floquet matrix $\to$ the evaluation of the differential of the Poincare map of the unperturbed system in the fixed point at time $2\pi/\omega_0$,
As change of variables $\to$ the identity.

Number of harmonics of the solution in function of the desired accuracy.

d	Accuracy	Initial	Final
1	1.e-12	21	29
1	1.e-14	21	33
2	1.e-12	441	1 225
2	1.e-14	441	1 591
3	1.e-12	9261	53 235
4	1.e-6	390 625	531 441
4	1.e-8	390 625	1 185 921
4	1.e-10	390 625	1 975 467

Linear stability for the different cases $d = 1, 2, 3, 4$.

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

Computing time (in seconds) varying the number of processors (n) in order to find the solution corresponding to:

$d = 2$ with accuracy 1.e-14.

n	Total	Master	System
1	35.183	1.700	0.100
2	19.683	1.630	0.110
3	16.279	1.700	0.250

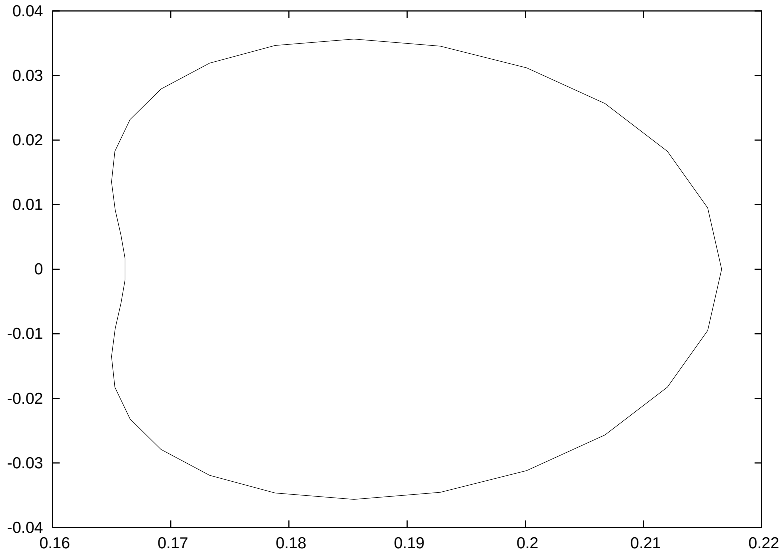
$d = 3$ with accuracy of 1.e-12.

n	Total	Master	System
1	1224.672	33.510	3.650
2	662.825	35.310	5.750
3	458.339	36.380	6.460
4	364.559	37.050	6.660
5	309.761	38.200	8.690
6	274.825	39.280	9.900
7	250.926	40.880	11.510

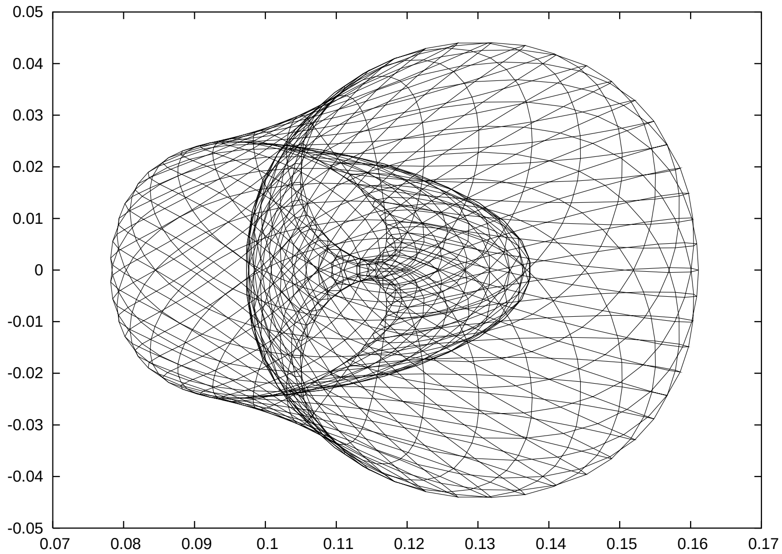
(Each processor is a 266 MHz Pentium II).

Some pictures:

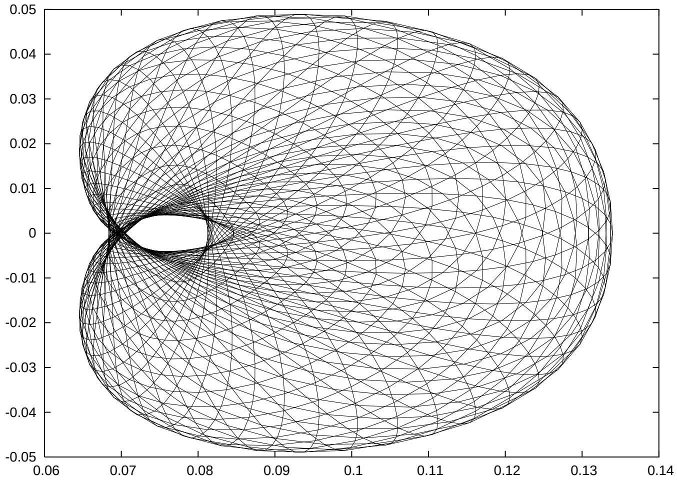
Solution with $d = 1$ over the Poincare section:

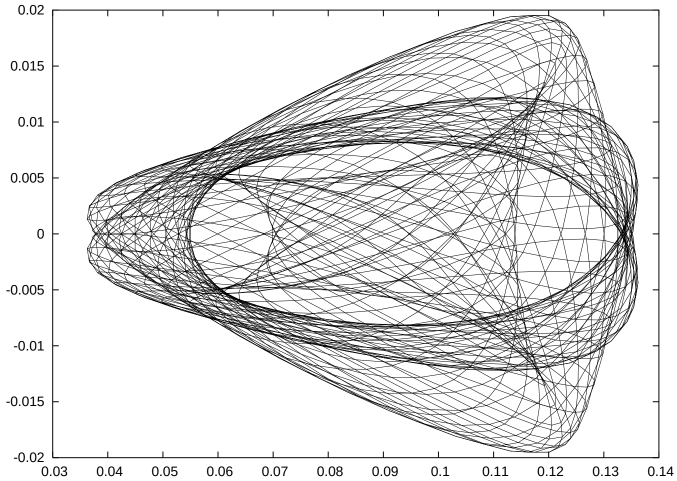


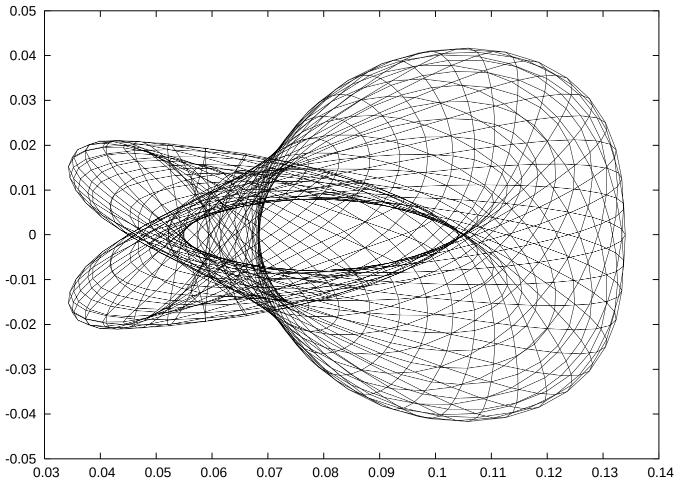
Solution with $d = 2$ over the Poincare section:



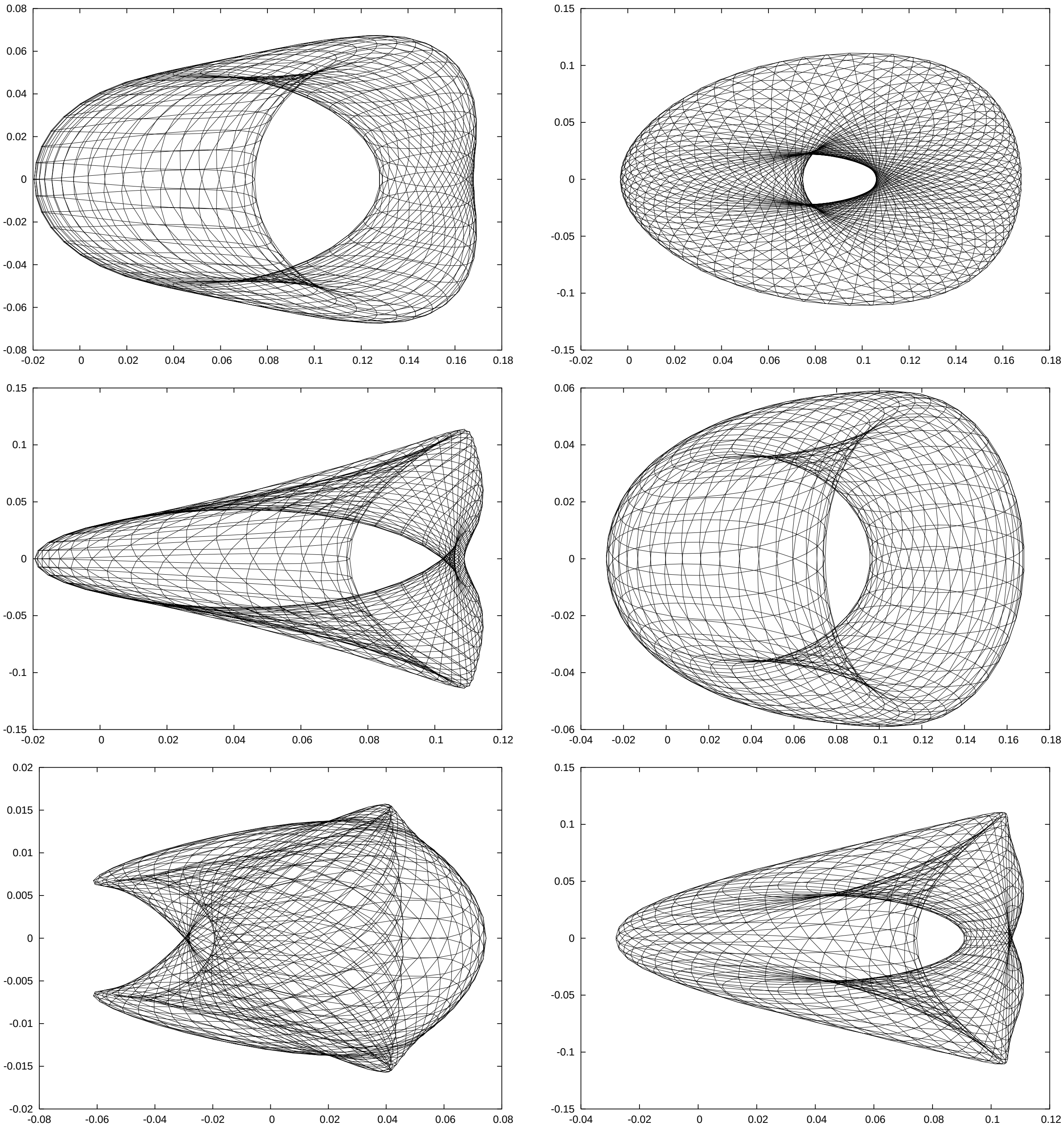
Projections of the solution with $d = 3$







Projections of the solution with $d = 4$



6) Conclusions.

- We have developed a method for the computation of invariant tori of dimension higher than two, in the case in which the flow on the torus is quasi-periodic and its linear normal flow is reducible.

- The algorithm has a high degree of parallelism that is exploited by our implementation.

- The process of the control of the error gives the solution with the needed number of harmonics to obtain the desired solution.

- The method seems very suitable when the solution needs a large number of harmonics.