Hamiltonian dynamical systems: symbolical, numerical and graphical study

Setsuo Takato and José A Vallejo

Abstract. Hamiltonian dynamical systems can be studied from a variety of viewpoints. Our intention in this paper is to show some examples of usage of two Maxima packages for symbolical and numerical analysis (`pdynamics` and `poincare`, respectively), along with the set of scripts KETCindy for obtaining the LaTeX code corresponding to graphical representations of Poincaré sections, including animation movies.

Mathematics Subject Classification (2010). Primary 97N80 ; Secondary 65P10.

Keywords. Hamiltonian systems, Poincaré sections, Mathematical Software.

1. Introduction

For simplicity, we will consider Hamiltonians defined on the symplectic manifold $\mathbb{R}^{2n}$, with coordinates (q^j, p_j) $(1 \leq j \leq n)$, endowed with the canonical form $w = \mathrm{d}p_j \wedge \mathrm{d}q^j$, and the induced Poisson bracket on $\mathcal{C}^\infty(\mathbb{R}^{2n})$

$$\{f, g\} = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial p_i} \frac{\partial f}{\partial q^i} - \frac{\partial f}{\partial q^i} \frac{\partial f}{\partial p_i} \right) ,$$

although all the results remain valid for an arbitrary symplectic manifold. For background on Hamiltonian systems, see [8].

Given a Hamiltonian system defined by the Hamiltonian function $H \in \mathcal{C}^\infty(\mathbb{R}^{2n})$ and the Poisson bracket $\{\cdot, \cdot\}$,

$$\begin{aligned} \dot{q}^j &= \frac{\partial H}{\partial p_j} \\ \dot{p}_j &= -\frac{\partial H}{\partial q^j} , \end{aligned} \tag{1.1}$$

two of the main goals in the theory of dynamical systems are the determination of possible closed, stable orbits, and the computation of adiabatic invariants (of course, taking for granted the impossibility of solving (1.1) explicitly). Of particular interest is the case in which the Hamiltonian H is a perturbation of an integrable one, say, $H = H_0 + \sum_{j=1}^{n} \varepsilon^j H_j$. A widely used procedure to study it, consists in writing the Hamiltonian in the so-called *normal form*, that is, as a formal series

$$H = \sum_{j=0}^{\infty} \varepsilon^j N_j$$

Work partially supported by grants KAKENHI 15K01037 (ST, Japan) and CONACyT CB-2012 179115 (JAV, México).

where $N_0 = H_0$, and each N_j commutes with the unperturbed Hamiltonian,

$$\{H_0, N_j\} = 0\,.$$

Let us recall that given an integral curve, that is, a $c(t) = (q(t), p(t))$ satisfying (1.1), the evolution of any observable $f \in \mathcal{C}^\infty(\mathbb{R}^{2n})$ along c is determined by

$$\dot{f}(t) = \{H, f\}\,.$$

Any smooth function such that $\{H, f\} = 0$ is thus a constant of motion, also called a first integral. Indeed, given enough first integrals one can solve the motion of the system, as the physical trajectories are determined by the intersection of their level hypersurfaces. Unfortunately, determining first integrals is a very difficult problem, and there are quite a few systems for which enough first integrals exist (roughly, these are the so-called integrable systems).

Notice that transforming to the normal form introduces a (possibly infinite) family of first integrals N_j which might not been present in the original system. These additional, spurious symmetries must be removed in order to have a system equivalent to the original one, and this is usually done by restricting the system to a reduced phase space through symplectic (singular) reduction. The basic idea is to restrict the system to a particular level hypersurface and to consider its evolution there. A number of well-known techniques are available to do this, for instance the ones based on Moser's theorem [10]: If M_h denotes the hypersurface $H_0 = h$, suppose that the orbits of the Hamiltonian flow $\mathrm{Fl}^t_{X_{H_0}}$ are all periodic with period T and let S be the quotient with respect to the induced $U(1)-$action on M_h. Then, to every non-degenerate critical point $\overline{p} \in S$ of the restricted averaged perturbation $N_1|_S = \langle H_1 \rangle|_S$ corresponds a *periodic* trajectory of the full Hamiltonian vector field X_H, that branches off from the orbit represented by $\overline{p}$ and has period close to 2π. When the critical points are degenerate, one can resort to the second-order normal form to decide the stability of orbits. An example of this situation is given by the Hénon-Heiles Hamiltonian [7]. These results illustrate the importance of being able to compute efficiently the normal form of a Hamiltonian system. In Section 2 we show how to do this using the Computer Algebra System (CAS) Maxima.

Another aspect related to the study of existence and stability of closed orbits is the construction of Poincaré sections. They provide a direct and very intuitive way for detecting these orbits, but their computation in closed form is usually impossible, so numerical methods are needed. The traditional method used for this task has been the fourth-order Runge-Kutta, but more recently methods based on symplectic integrators (such as symplectic Euler, Störmer-Verlet, symplectic RK, etc.) are also intensively used, see [3] for a recent review. The choice of one method or another depends very much on the properties of the system under consideration, in Section 3 we will use the RK method, but symplectic methods can be included by substituting the `rkfun` command in the code of `poincare` with `symplectic_ode`, recently included in Maxima (from version 5.39.1 onward). In any case, one of the main goals is to obtain a clean picture of the phase-space portrait of the system, something that can be challenging for CASs, whose graphical output is not very sophisticate in many cases. To deal with this issue we present in Section 4 the set of CindyScript macros called KETCindy, which parse the output of Maxima through the Dynamical Geometry Software (DGS) Cinderella and return the LaTeX code of the corresponding graphics, that can be included in any document even in the form of an animation. The data for these graphics are actually codes of TPIC specials for LaTeX, or pict2e commands in the case of pdfLaTeX, so there they can be inserted in scientific documentation with great flexibility (see [14, 15, 16, 17] for installation and examples of use).

The Maxima packages `pdynamics` (short for 'Poisson Dynamics') and `poincare` can be downloaded from `http://galia.fc.uaslp.mx/~jvallejo/pdynamics.zip` and `http:`

`//galia.fc.uaslp.mx/~jvallejo/poincare.mac` respectively[1]. The KETCindy package is available at `http://ketpic.com/?lang=english`.

2. Symbolic study of Hamiltonian systems: normal forms

Given a smooth vector field in $\mathbb{R}^m$ (although what follows is valid in an arbitrary manifold) its flow is a mapping $\mathrm{Fl}_X : \mathbb{R}^m \to \mathbb{R}^m$ defined by

$$\mathrm{Fl}_X(t,p) \doteq \mathrm{Fl}_X^t(p) \doteq c_p(t)\,,$$

where c_p is the integral curve of X such that, for $t = 0$, passes through $p \in \mathrm{R}^m$ (i.e., $c_p(0) = p$). When $m = 2n$, there is a canonical symplectic form

$$\Omega = \mathrm{d}p_1 \wedge \mathrm{d}q_1 + \cdots + \mathrm{d}p_n \wedge \mathrm{d}q_n\,.$$

Any Hamiltonian $H \in \mathcal{C}^\infty(\mathbb{R}^{2n})$, has an associated vector field X_H defined by the condition $i_{X_H}\Omega = -\mathrm{d}H$. In local coordinates (q_i, p_i) it has the expression

$$X_H = \left(\frac{\partial H}{\partial p_1}, -\frac{\partial H}{\partial q_1}, \ldots, \frac{\partial H}{\partial p_n}, -\frac{\partial H}{\partial q_n}\right)\,.$$

Suppose now that X is the generator of an $\mathbb{S}^1-$action, so the flow Fl_X^t is periodic in the variable t. This property can be used to put H in normal form (for details, see [1]). To this end, it is essential to define two averaging operators acting on observables. The first one is denoted as $\langle\cdot\rangle$ and is given by integrating the pullback

$$\langle g\rangle \doteq \frac{1}{2\pi}\int_0^{2\pi} (\mathrm{Fl}_X^t)^* g\,\mathrm{d}t\,,$$

for any observable $g \in \mathcal{C}^\infty(\mathbb{R}^{2n})$. The second operator, denoted $\mathcal{S}$, is defined as

$$\mathcal{S}(g) \doteq \frac{1}{2\pi}\int_0^{2\pi} (t-\pi)(\mathrm{Fl}_X^t)^* g\,\mathrm{d}t\,.$$

In the particular case of a perturbed Hamiltonian, of the form $H = H_0 + \varepsilon H_1 + \frac{\varepsilon^2}{2}H_2 + \cdots$, if the non-perturbed part H_0 generates an $\mathbb{S}^1-$action in such a way that its flow is periodic with frequency function w, it can be proved (see [1]) that its second-order normal form is

$$N = H_0 + \varepsilon\,\langle H_1\rangle + \frac{\varepsilon^2}{2}\left(\langle H_2\rangle + \left\langle\left\{\mathcal{S}\left(\frac{H_1}{w}\right), H_1\right\}\right\rangle\right)\,.$$

There are other representations of the normal form (it must be stressed that it is *not* unique), but this one has the particular features that it is global (not depending on action-angle variables), and particularly well-suited for symbolic computation. Let us illustrate the use of the `pdynamics` Maxima package by considering the example of the Pais-Uhlenbeck oscillator. This system is a toy model of a field theory defined by a Lagrangian depending on higher-order derivatives. These Lagrangians are believed to lead to perturbatively renormalizable theories, where the infinities appearing in the perturbation series for the field equations can be cured through some well-defined regularization procedure. The corresponding Hamiltonian is constructed through a higher-order analog of the Legendre transformation, called the Ostrogadskii formalism [11]. After some suitable transformations

[1]There is a documentation file inside `pdynamics.zip`, and the documentation for `poincare.mac` can be found at `http://galia.fc.uaslp.mx/~jvallejo/PoincareDocumentation.pdf`. Both files contain detailed instructions about the installation.

(see [12]) the Hamiltonian can be expressed as the *difference* of two harmonic oscillators with respective frequencies w_1, w_2. Adding an interaction term in the form of a homogeneous polynomial results in the Hamiltonian

$$H = \frac{1}{2}(p_1^2 + w_1^2 q_1^2) - \frac{1}{2}(p_2^2 + w_2^2 q_2^2) + \frac{\lambda}{4}(q_1 + q_2)^4 \,. \tag{2.1}$$

This can be considered as a perturbed system of the form $H = H_0 + \lambda H_1$, let us study it symbolically. We would use the following sequence of commands in Maxima:

(% i1) `load(pdynamics)$`
(% i2) `declare(w1,integer)$`
(% i3) `assume(w1>0)$`
(% i4) `declare(w2,integer)$`
(% i5) `assume(w2>0)$`
(% i6) `H0(q1,p1,q2,p2):=(p1^2+w1^2*q1^2)/2-(p2^2+w2^2*q2^2)/2$`
(% i7) `H1(q1,p1,q2,p2):=(q1+q2)^4/4$`
(% i8) `H2(q1,p1,q2,p2):=0$`

Up to here, we have just defined the parameters of the system (the frequencies w_1, w_2, and the subhamiltonians H_i). Let us check that the Hamiltonian flow of the non-perturbed part H_0 is periodic by explicitly computing it (we have slightly edited the output of Maxima by writing it as a column matrix, to make it more readable):

(% i9) `phamflow(H0);`

$$\begin{pmatrix} \frac{p_1 \, \sin(t\,w_1)}{w_1} + q_1 \, \cos(t\,w_1) \\ p_1 \, \cos(t\,w_1) - q_1 \, w_1 \, \sin(t\,w_1) \\ q_2 \, \cos(t\,w_2) - \frac{p_2 \, \sin(t\,w_2)}{w_2} \\ q_2 \, w_2 \, \sin(t\,w_2) + p_2 \, \cos(t\,w_2) \end{pmatrix} \tag{\% o9}$$

It is clear that the flow is periodic with period $T = 2\pi w_1 w_2$. Thus, we define the frequency function as

(% i10) `u(q1,p1,q2,p2):=1/(w1*w2)$`

Finally, we can compute $N_1 = \langle H_1 \rangle$:

(% i11) `phamaverage(H1,H0,u(q1,p1,q2,p2));`

$$\begin{aligned} \frac{1}{32 w_1^4 \, w_2^4} \big[\big(\big(3q_2^4 + 12 q_1^2 \, q_2^2 + 3 q_1^4\big) \, w_1^4 + \big(12 p_1^2 \, q_2^2 + 6 p_1^2 \, q_1^2\big) \, w_1^2 + 3 p_1^4\big) \, w_2^4 \\ + \big(\big(6 p_2^2 \, q_2^2 + 12 p_2^2 \, q_1^2\big) \, w_1^4 + 12 p_1^2 \, p_2^2 \, w_1^2\big) \, w_2^2 + 3 p_2^4 \, w_1^4 \big] \end{aligned} \tag{\% o11}$$

The second-order normal form can be computed along similar lines but, as one can guess, the expressions become very cumbersome, and not much illuminating (see (% o15) below). Indeed, it is customary to simplify these expressions by rewriting them in terms of the so-called Hopf variables. The idea behind these variables is the following: in the case in which the normal subhamiltonians N_i are polynomials, the fact that they commute with $N_0 = H_0$ means that they are invariant under the smooth $\mathcal{S}^1-$action of X_{H_0}. The space of smooth invariant functions is finitely generated (this is a generalization to the smooth case of a classical result of Hilbert dealing with algebraic invariants, called the Schwarz theorem [13]), and a set of functional generators is precisely given by the Hopf polynomials, that can be considered as new variables. In other words, any smooth invariant function can be expressed as a smooth function of the Hopf variables. For the Pais-Uhlenbeck oscillator with

resonance $1:2$ (that is, when $w_1 = 1$ and $w_2 = 2$), these Hopf invariants can be readily computed [2] and they turn out to be

$$\begin{aligned}\rho_1 =& q_1^2 + p_1^2\\ \rho_2 =& 4q_2^2 + p_2^2\\ \rho_3 =& p_2(p_1^2 - q_1^2) - 4p_1 q_1 q_2\\ \rho_4 =& 2q_2(p_1^2 - q_1^2) + 2q_1 p_1 p_2\,.\end{aligned} \tag{2.2}$$

We can compute N_2 by extracting the coefficient of λ^2 in the second-order normal form of H. The following commands show how to study this resonance, defining a function `phopf6res12` (not contained in the `pdynamics` package) adapted to this case, whose purpose is to express everything in terms of the variables (2.2). First, we define the Hamiltonian:

(% i12) `K0(q1,p1,q2,p2):=(p1^2+q1^2)/2-(p2^2+4*q2^2)/2$`

(% i13) `K1(q1,p1,q2,p2):=(q1+q2)^4/4$`

(% i14) `K2(q1,p1,q2,p2):=0$`

and then compute the second-order normal form (here and below, the Maxima output has been slightly edited in order to fit the page):

(% i15) `pnormal2(K0,K1,K2,%lambda);`

$$\begin{aligned}&\frac{\lambda^2}{4587520}\left(255168q_2^6 + \left(1184256q_1^2 + 191376p_2^2 + 1184256p_1^2\right)\,q_2^4\right.\\ &+ \left(225792q_1^4 + \left(592128p_2^2 + 4580352p_1^2\right)\,q_1^2 + 47844p_2^4 + 592128p_1^2\,p_2^2 + 225792p_1^4\right)\,q_2^2\\ &+ \left(2064384p_1\,p_2\,q_1^3 - 2064384p_1^3\,p_2\,q_1\right)\,q_2 + 48384q_1^6 + \left(314496p_2^2 + 145152p_1^2\right)\,q_1^4\\ &+ \left(74016p_2^4 - 403200p_1^2\,p_2^2 + 145152p_1^4\right)\,q_1^2 + 3987p_2^6 + 74016p_1^2\,p_2^4 + 314496p_1^4\,p_2^2 + 48384p_1^6\Big)\\ &+ \frac{\lambda}{512}\left(48q_2^4 + \left(192q_1^2 + 24p_2^2 + 192p_1^2\right)\,q_2^2 + 48q_1^4 + \left(48p_2^2 + 96p_1^2\right)\,q_1^2 + 3p_2^4 + 48p_1^2\,p_2^2 + 48p_1^4\right)\\ &- \frac{4q_2^2 + p_2^2}{2} + \frac{q_1^2 + p_1^2}{2}\end{aligned} \tag{\% o15}$$

Now, the term N_2 can be easily extracted:

(% i16) `expand(coeff(%,%lambda,2));`

$$\begin{aligned}&\frac{3987{q_2}^6}{71680} + \frac{2313{q_1}^2\,{q_2}^4}{8960} + \frac{11961{p_2}^2\,{q_2}^4}{286720} + \frac{2313{p_1}^2\,{q_2}^4}{8960} + \frac{63{q_1}^4\,{q_2}^2}{1280} + \frac{2313{p_2}^2\,{q_1}^2\,{q_2}^2}{17920}\\ &+ \frac{639{p_1}^2\,{q_1}^2\,{q_2}^2}{640} + \frac{11961{p_2}^4\,{q_2}^2}{1146880} + \frac{2313{p_1}^2\,{p_2}^2\,{q_2}^2}{17920} + \frac{63{p_1}^4\,{q_2}^2}{1280} + \frac{9p_1\,p_2\,{q_1}^3\,q_2}{20}\\ &- \frac{9{p_1}^3\,p_2\,q_1\,q_2}{20} + \frac{27{q_1}^6}{2560} + \frac{351{p_2}^2\,{q_1}^4}{5120} + \frac{81{p_1}^2\,{q_1}^4}{2560} + \frac{2313{p_2}^4\,{q_1}^2}{143360} - \frac{45{p_1}^2\,{p_2}^2\,{q_1}^2}{512}\\ &+ \frac{81{p_1}^4\,{q_1}^2}{2560} + \frac{3987{p_2}^6}{4587520} + \frac{2313{p_1}^2\,{p_2}^4}{143360} + \frac{351{p_1}^4\,{p_2}^2}{5120} + \frac{27{p_1}^6}{2560}\end{aligned} \tag{\% o16}$$

(% i17) `define(N2(q1,p1,q2,p2),%)$`

And, finally, the reduction to Hopf variables can be achieved as follows:

(% i18)
```
phopf6res12(expr):=block(
[aux,list_coeff,eq,eqs,W,Wp,U,Up,a,l,
w:[q1^2+p1^2,4*q2^2+p2^2],
u:[-4*p1*q1*q2-p2*q1^2+p1^2*p2,
   -2*q1^2*q2+2*p1^2*q2+2*p1*p2*q1]],
W:makelist(w[1]^i*w[2]^(3-i),i,makelist(j,j,0,3)),
Wp:makelist(%rho[1]^i*%rho[2]^(3-i),i,makelist(j,j,0,3)),
U:makelist(u[1]^i*u[2]^(2-i),i,makelist(j,j,0,2)),
Up:makelist(%rho[3]^i*%rho[4]^(2-i),i,makelist(j,j,0,2)),
a:makelist(a[k],k,1,length(W)+length(U)),
aux:facsum(expandwrt(expr-sum(a[i]*W[i],i,1,length(W))-
                     sum(a[i+length(W)]*U[i],i,1,length(U)),
                     q1,p1,q2,p2),
           q1,p1,q2,p2),
list_coeff:coeffs(aux,q1,p1,q2,p2),
l:length(list_coeff),
for j:2 thru l do (k:j-1, eq[k]:first(list_coeff[j])),
eqs:makelist(eq[k],k,1,l-1),
subst(first(algsys(eqs,a)),
      sum(a[i]*Wp[i],i,1,length(Wp))
      +sum(a[i+length(W)]*Up[i],i,1,length(U)))
)$
```

(% i19)
```
phopf6res12(N2(q1,p1,q2,p2));
```

$$-\frac{\rho_4^2\,(5120\%r_1-63)}{5120}-\frac{\rho_3^2\,(5120\%r_1-351)}{5120}+\rho_1^2\,\rho_2\,\%r_1 \qquad (\%\ \text{o19})$$
$$+\frac{3987\rho_2^3}{4587520}+\frac{2313\rho_1\,\rho_2^2}{143360}+\frac{27\rho_1^3}{2560}$$

(% i20)
```
subst(%r1=0,expand(%));
```

$$\frac{63\rho_4^2}{5120}+\frac{351\rho_3^2}{5120}+\frac{3987\rho_2^3}{4587520}+\frac{2313\rho_1\,\rho_2^2}{143360}+\frac{27\rho_1^3}{2560} \qquad (\%\ \text{o20})$$

Of course, a similar analysis can be done for N_1. The resulting expression appears in [2] (see equation (16) in that paper), applied to the determination of the existence of closed, stable orbits in the Pais-Uhlenbeck oscillator.

3. Numerical study: Poincaré sections

Given a Hamiltonian $H \in \mathcal{C}^\infty(\mathbb{R}^{2n})$, the Maxima package `poincare` provides several functions to study its Poincaré sections. The functionality of this package, and even the syntax, is similar to the package `DEtools` in Maple™ but it offers two advantages: first, it uses free (both as in 'freedom' and as in 'free beer') software and, second, it is almost three times faster, thus being a serious competitor for long computations. Moreover, as we will see in Section 4, in conjunction with KETCindy animation movies describing the evolution of the system in phase space can be easily constructed.

It should be stressed that Maxima is a CAS, not a language intended for numerical computations such as Octave. Thus, speed in computations is not one of its goals, nor was it designed to achieve it. However, the fact that LISP is its underlying programming language, allows the possibility of writing specialized routines using declared variables, that can be then compiled. The package `poincare` uses a compiled version of the Runge-Kutta method, called `rkfun`, developed by Richard Fateman (`http://people.eecs.berkeley.edu/~fateman/lisp/rkfun.lisp`), and this is the ultimate reason for the gain in speed.

The function called `hameqs` constructs the Hamiltonian equations for a given Hamiltonian $H(q1, p1, \ldots, q2, p2)$. Any names can be used for the variables, but they must be given in pairs "coordinate, conjugate momentum". A good choice (used internally) is $(q1, p1, ..., qn, pn)$. A name must be provided for the components of the Hamiltonian vector field

$$X_H(q_1, p_1, ..., q_n, p_n) = \sum_{i=1}^{n} \left(\frac{\partial H}{\partial p_i} \frac{\partial}{\partial q_i} - \frac{\partial H}{\partial q_i} \frac{\partial}{\partial p_i} \right) .$$

Once a name, say XH, is chosen, the components of the Hamiltonian vector field will be globally defined functions XHj with $1 \leq j \leq 2n$, where n is the number of degrees of freedom, and will be available to Maxima. Notice that, for instance,

$$XH1(t, q_1, p_1, ..., q_n, p_n) = \frac{\partial H}{\partial p_1}$$

and

$$XH2(t, q_1, p_1, ..., q_n, p_n) = -\frac{\partial H}{\partial q_1} .$$

Although we will work with *autonomous* Hamiltonian systems, the components XHj returned by this command will have the set $(t, q_1, p_1, ..., q_n, p_n)$ as arguments. This is necessary to maintain consistency with the `rkfun` routine (implementing the Runge-Kutta method), which can work with both, autonomous and non-autonomous systems.

The function `poincare3d` constructs the projection of the Hamiltonian orbits along a certain coordinate which is given as an argument `coord`. Other arguments are: a list of initial conditions `inicond` $= [q_1(0), p_1(0), \ldots, q_n(0), p_n(0)]$, and a list characterizing the time domain `timestep` $= [t, t_{ini}, t_{fin}, step]$. Thus, the syntax is `poincare3d(H,name,inicond,timestep,coord)`. The package is loaded with

(%i1) `batch("poincare.mac")$`

As a simple example, let us construct the $3D-$surface of a couple of harmonic oscillators (this is based on Chapter 9 of [9], where a similar discussion using Maple™ is presented):

(%i22) `H(x,v,q,p):=w1*(x^2+v^2)/2+w2*(q^2+p^2)/2$`

The corresponding Hamiltonian equations are:

(%i23) `hameqs(H,XH);`

$$[v\ w1, -w1\,x, p\ w2, -q\ w2] \qquad (\%o7)$$

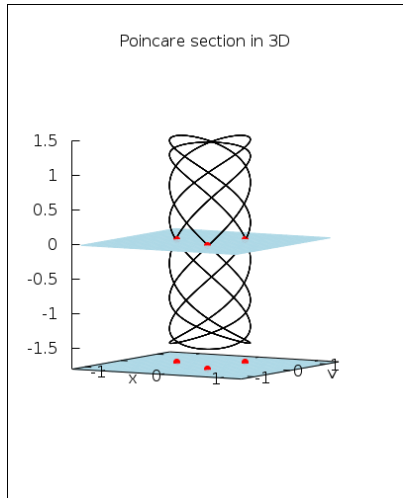
and we fix the values of the frequencies $w1 = 1$, $w2 = 3$ so they are commensurable:

(%i24) `[w1,w2]:[8,3]$`

Next we plot the $3D$ Poincaré surface by projecting along the p coordinate (thus, the resulting graphics has (x, v, q) coordinates):

(%i25) `data1:poincare3d(H,XH,[0.3,0.5,0,1.5],[t,0,40,0.01],p)$`

(%i26)
```
draw3d(title="Poincare section in 3D",
dimensions=[350,500],view=[85,30],
xlabel="x",ylabel="v",zlabel="q",
xtics=1,ytics=1,
surface_hide=true,color="light-blue",
    explicit(0,x,-1.35,1.35,y,-1.35,1.35),
point_size=0,points_joined=true,color=black,line_width=1,
    points(data1),
user_preamble="set xyplane at -1.8",color="light-blue",
    explicit(-1.8,u,-1.5,1.35,v,-1.35,1.35),
point_size=1,point_type=filled_circle,color=red,points_joined=false,
    points([[-0.56,0,-1.78],[0.28,-0.52,-1.78],[0.3,0.48,-1.78],
            [-0.56,0,0],[0.28,-0.52,0],[0.3,0.48,0]]));
```

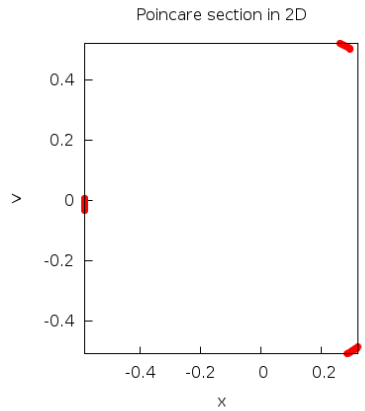


(%t26,%o26)

The function `poincare2d` constructs the surface of section selected by a list of arguments of the form `scene` $= [q0, c, qi, qj]$, that is, the surface $q0 = c$ in which coordinates $[qi, qj]$ are shown. The method used in the computation of the Poincaré surface is that described in the paper [5] we select a set of initial conditions, follow the corresponding orbit numerically, and detect where we have crossed the $q0 = c$ surface by looking at changes of sign in the list of values for this coordinate minus c. In the previous example, we plotted the $3D-$Poincaré surface of a couple of commensurable oscillators, and we included a $2D-$section (corresponding to $q = 0$) showing that the periodicity of the system reflects itself in the discrete character of the $2D-$Poincaré map (only three points appear in it). Now we can check this directly with `poincare2d` (notice the selection of the $q = 0$ section in the last argument, `[q,0,x,v]`):

(%i27)
```
data2:poincare2d(H,XH,[0.3,0.5,0,1.5],[t,0,40,0.01],[q,0,x,v])$
```

(%i28)
```
draw2d(title="Poincare section in 2D",
xlabel="x",ylabel="v",
xtics=0.2,
point_size=1,point_type=7,color=red,
points_joined=false,proportional_axes=xy,
    points(data2));
```



(%t28,%o28)

For a different example, let us consider the case of a elastic pendulum [4], with Hamiltonian

(%i29)
```
H(q1,p1,q2,p2):=(p1^2+p2^2)/2+(q1^2+q2^2)/2-0.75*q1^2*(1+q2)/2$
```

We can obtain an analytic expression for p_2 once the energy E, and the initial values of (q_1, p_1, q_2) are known. Here we work with $E = 0.00875$:

(%i30)
```
solve(H(q1,p1,q2,p2)=0.00875,p2);
```

$$[p_2 = -\frac{\sqrt{-4{q_2}^2 + 3{q_1}^2\,q_2 - {q_1}^2 - 4{p_1}^2 + 0.07}}{2}, p_2 = \frac{\sqrt{-4{q_2}^2 + 3{q_1}^2\,q_2 - {q_1}^2 - 4{p_1}^2 + 0.07}}{2}]$$

(%o30)

Let us define the corresponding functions:

(%i31)
```
define(f(q1,p1,q2),rhs(first(%)))$
```

(%i32)
```
define(g(q1,p1,q2),rhs(second(%th(2))))$
```

Now, we compute the $q_2 = 0$ surface of section, for enough initial conditions (q_1, p_1, q_2) (10 different sets) using the positive value of p_2, and joining all the resulting points in a big list of $2D-$coordinates called `points1`:

(%i33)
```
for j:1 thru 10 do data1[j]:poincare2d(H,XH,
[0.15,j/100,0.001,g(0.15,j/100,0.001)],[t,0,1000,0.01],[q2,0,q1,p1])$
```

For future reference, here is the time invested in the computation:

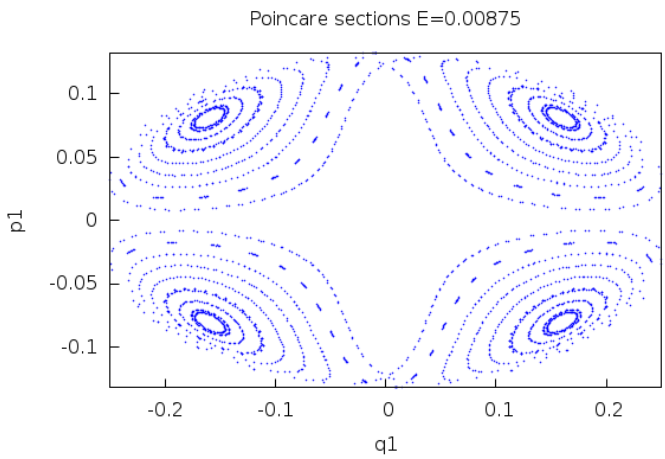
(%i34)
```
time(%);
```

$$[17.222] \qquad (\%o61)$$

(%i35)
```
points1:xreduce(append,create_list(data1[j],j,makelist(k,k,1,10)))$
```

The following figure is the plot of these points on the Poincaré surface:

(%i36)
```
draw2d(title="Poincare sections E=0.00875",
xlabel="q1",ylabel="p1",
point_type=7,point_size=0.1,
    points(points1)
);
```



(%t36,%o36)

In order to complete the section, we must select another set of initial conditions, whose orbits pass through the empty region at the center. This time we use negative values of the momentum p_2:

(%i37)
```
for j:1 thru 10 do data2[j]:poincare2d(H,XH,
[2*j/100,0,j/100+0.0025,f(2*j/100,0,j/100+0.0025)],[t,0,1000,0.01],
[q2,0,q1,p1])$
```

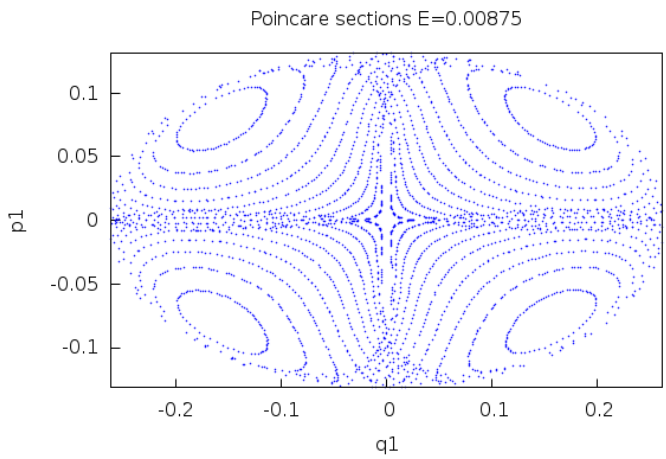
(%i38) `time(%);`

[17.369] (%o38)

The $2D-$dimensional coordinates of the corresponding points are stored in the list `points2` and then plotted with the aid of the `draw2d` command, which admits lots of optional arguments to fine tuning the appearance of the figure. Here, we specify that points be represented by filled circles (`point_type=7`) with a given radius size (`point_size=0.1`):

(%i39) `points2:xreduce(append,create_list(data2[j],j,makelist(k,k,1,10)))$`

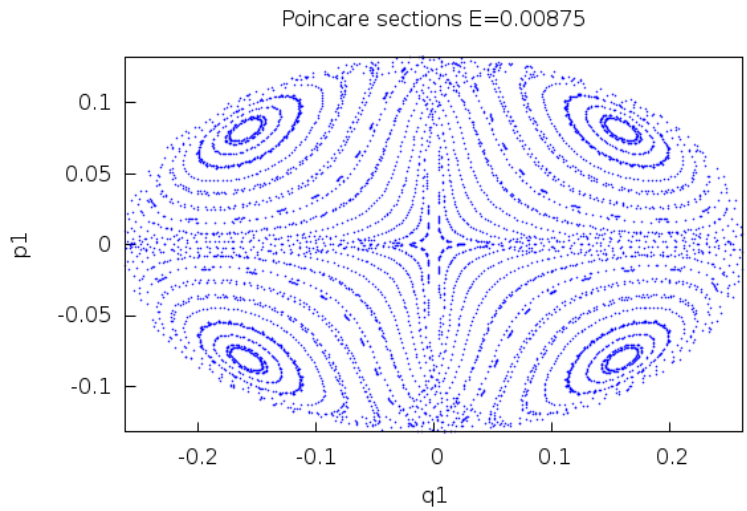
(%i40)
```
draw2d(title="Poincare sections E=0.00875",
xlabel="q1",ylabel="p1",
point_type=7,point_size=0.1,
    points(points2)
);
```



(%t40,%o40)

The full Poincaré section is obtained by joining both sets of points. The Maxima command `append` does exactly that when two lists are given:

```
(%i41) draw2d(title="Poincare sections E=0.00875",
       xlabel="q1",ylabel="p1",
       point_type=7,point_size=0.1,
           points(append(points1,points2))
       );
```



(%t41)

We have done our computations with a fixed value for the energy $E = 0.00875$. The same steps can be followed to consider other values. The collection of Poincaré sections so obtained is a valuable tool to visualize the dynamics of the system. In Section 4 we will see how to put all these sections together in the form of a movie animating the evolution in phase space.

4. Graphical study with KeTCindy

KETpic is a macro package involving several mathematical software for producing high-quality figures to be inserted into LaTeX documents, developed by one of the authors (ST). It can use the DGS Cinderella as a graphical interface through KETCindy, another set of macros which acts as a interface between them. The reason for choosing Cinderella is that it has its own scripting language, CindyScript, featuring a simple to understand syntax. Using CindyScript, we have added a layer to KETCindy to call other software such as Maxima, **R**, Fricas, Risa/Asir or C. We refer to previous works for more details on the CindyScript syntax [14, 15, 16, 17]; here we proceed in a more direct way, explaining how KETCindy calls to Maxima by way of an example devoted to find the indefinite and definite integrals of a function. For this, the following code must be inserted into the script editor of Cinderella:

```
cmdL=[
 "f(x):=sin(x)+cos(2*x)",[],
 "ans1:integrate",["f(x)","x"],
 "ans2:integrate",["f(x)","x",0,"%pi/3"],
 "ans1::ans2",[]
];
CalcbyM("ans",cmdL,[""]);
```

Here `cmdL` is a list of Maxima commands which are parsed sequentially. By executing `CalcbyM` in the next line, a file called `simpleexampleans.txt` in the directory `ketwork` with the contents given below will be created:

```
writefile("ketwork/simpleexampleans.txt")$/*##*/
powerdisp:false$/*##*/
display2d:false$/*##*/
linel:1000$/*##*/
f(x):=sin(x)+cos(2*x)$/*##*/
ans1:integrate(f(x),x)$/*##*/
ans2:integrate(f(x),x,0,%pi/3)$/*##*/
disp(ans1)$/*##*/
disp(ans2)$/*##*/
closefile()$/*##*/
quit()$/*##*/
```

These instructions will be processed by Maxima, and the results will be passed to KETCindy to generate either direct graphical output in Cinderella or a LaTeX file with the corresponding code to generate the graphics that can be inserted in another document. In more detail, `calcbyM` will sequentially do the following:

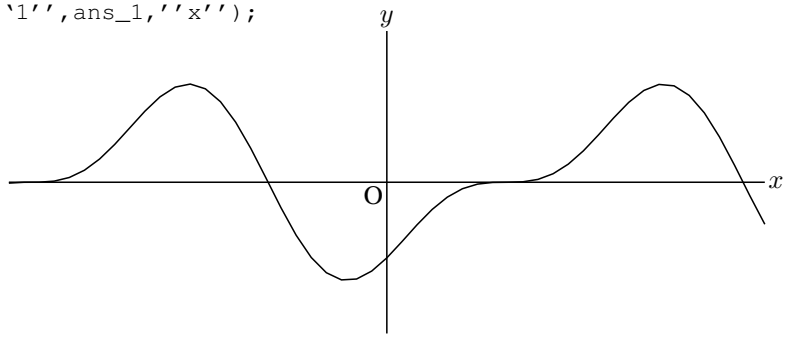
1. Create the `txt` file to be processed by Maxima.
2. Create a batch file `kc.sh(bat)` to call Maxima.
3. Call a Java program to execute the batch file above.
4. Hand the result from Maxima, parsed as strings, to KETCindy.

In the above example, the result `ans` is a list containing two strings:

```
[sin(2*x)/2-cos(x),(sqrt(3)+2)/4]
```

This result can be directly used in KETCindy; for example, we can draw the graph of the indefinite integral with the following command:

```
Plotdata(``1'',ans_1,''x'');
```



As mentioned, KETCindy can also produce a TEX animation. We illustrate this feature with the Poincaré sections of an elastic pendulum as an example. We begin by defining `Elist` as a list of increasing energies:

```
Elist=[0.00875,0.0125,0.01625,0.02,0.02375,0.0275,0.03125,0.035,0.03875];
```

Next, we generate the corresponding data for each energy using Maxima with the package `poincare`. The following list of Maxima commands is just the same as the one described in Section 3:

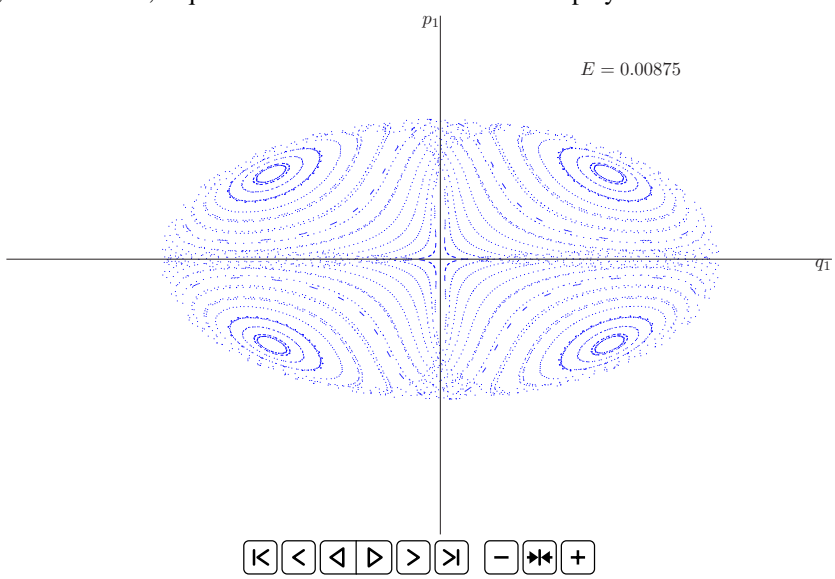
```
cmdL1=concat(Mxload("rkfun.lisp"),Mxbatch("pdynamics.mac"));
cmdL1=concat(cmdL1,Mxbatch("poincare.mac"));
cmdL1=concat(cmdL1,[
 "H(q1,p1,q2,p2):=(p1^2+p2^2)/2+(q1^2+q2^2)/2-0.75*q1^2*(1+q2)/2",[],
 "ans:solve(H(q1,p1,q2,p2)=E,p2)",[],
 "define(f(q1,p1,q2),rhs(first(%)))",[],
 "define(g(q1,p1,q2),rhs(second(%th(2))))",[]
]);
forall(1..9,nn,
```

```
  cmdL2=[
   "E:"+textformat(Elist_nn,6),[],
   "for j:1 thru 10 do data1[j]:poincare2d(H,XH,[0.15,j/100,0.001,
        g(0.15,j/100,0.001)],[t,0,1000,0.01],[q2,0,q1,p1])",[],
   "points1:xreduce(append,create_list(data1[j],j,makelist(k,k,1,10)))",[],
   "for j:1 thru 10 do data2[j]:poincare2d(H,XH,[2*j/100,0,j/100+0.0025,
        f(2*j/100,0,j/  100+0.0025)],[t,0,1000,0.01],[q2,0,q1,p1])",[],
   "points2:xreduce(append,create_list(data2[j],j,makelist(k,k,1,10)))",[],
   "points1::points2",[]
  ];
  cmdL=concat(cmdL1,cmdL2);
  CalcbyM("Points",cmdL,[mr,"Wait=40"]);
);
```

Each result is stored in a text file (with extension `txt`) with its name constructed appending a number sequentially to the prefix `ptdata`. Finally, we define a function `mf(nn)`, which describes the animation frame numbered `nn`.

```
mf(nn):=(
 regional(tmp,Points1,Points2);
 Com1st("ReadOutData('ptdata"+text(nn)+".txt')");
 Setcolor("red");
 Pointdata("1","Points",[red,"Size=0.4"]);
 Setcolor("black");
 Expr(D,"c","E="+textformat(Elist_(nn),6));
);
Setpara("poincare","mf(nn)",1..9,
   ["m","Frate=3","Scale=0.7","OpA=[loop]"]);
```

The animation is generated by putting together all the frames with the command `Mkanimation()`. The result, shown below, requires Adobe Acrobat Reader™ for playback[2]:



This graphical analysis illustrates several features common to any perturbed system of the form $H = H_0 + \varepsilon H_1$, where H_0 is an integrable Hamiltonian and $\varepsilon \sim 0$. Starting at low values of

[2]Currently, it is the only PDF reader capable of that. Some versions of the KDE reader Okular have been reported to be able of reproducing some animations, but we have not had success when using it.

the energy, many closed curves can be detected, corresponding to periodic motions of the system. Those closed curves are intersections of the tori determined by the non-perturbed part with the Poincaré surface. As the energy of the systems increases, the tori are destroyed and the trajectories initially confined to them start wandering all over the phase space. At a certain point, we can not distinguish any periodicity and the behavior is completely chaotic. This generic picture is the content of the famous KAM (for Kolgomorov, Arnold and Moser) theorem, although this theorem refers to increasing values of the perturbation parameter rather than the total energy (see [6] for the application to this case, along with some comments on the applicability of the KAM theorem, which is not immediate).

5. Conclusions

The symbolic computation of second-order normal form for perturbed Hamiltonian systems can be quickly computed in closed form with the aid of the Maxima CAS, directly in terms of the Hopf invariants. The package `pdynamics` shows a practical implementation.

The Maxima package `poincare` can reproduce the results appearing in textbooks and research papers dealing with Hamiltonian systems. The graphical output quality is quite good, comparable (to say the least) to that of commercial software, but at no cost (for comparison, Maple™ in its student's version costs 1 000USD.) Regarding computation times, the Maxima version outperforms commercial competitors: the heaviest computation in this paper is executed in (%o38) while, for instance, the same takes 50 seconds in Maple™ [3] (as can be seen in the worksheet `http://galia.fc.uaslp.mx/~jvallejo/ElasticPendulum-MapleSession.pdf`, for which the same computer was used). Maxima only requires a third of this time.

On the other hand, KETCindy in combination with Maxima can produce TEX animations, ready for use in complex documents which require high-quality graphics, such as research papers or handouts to be used in teaching. The union of these features results in an easy-to-use, powerful integrated system particularly suitable for studying the dynamics of Hamiltonian systems.

References

[1] M. Avendaño–Camacho, J. A. Vallejo and and Yu. Vorobjev: *A simple global representation for second-order normal forms of Hamiltonian systems relative to periodic flows*. J. of Phys. A: Math. and Theor. **46** (2013) 395201.

[2] M. Avendaño–Camacho, J. A. Vallejo and and Yu. Vorobjev: *A perturbation theory approach to the stability of the Pais-Uhlenbeck oscillator*. J. of Math. Phys. **58** (2017) 093501.

[3] S. Blanes and F. Casas: *A Concise Introduction to Geometric Numerical Integration*. CRC Press, Boca Raton, FL, 2016.

[4] R. Carretero–González, H. N. Núñez–Yépez and A. L. Salas–Brito: *Regular and chaotic behaviour in an extensible pendulum*. Eur. J. Phys. **15** (1994) 139–148.

[5] E. S. Cheb–Terrab and H. P. Oliveira: *Poincaré sections of Hamiltonian Systems*. Comp. Phys. Comm. **95** 2–3 (1996) 171–189.

[6] R. Cuerno, A. F. Rañada and J. J. Ruiz-Lorenzo: *Deterministic chaos in the elastic pendulum: A simple laboratory for nonlinear dynamics*. Am. J. of Physics **60** 73 (1992) 73–79.

[7] R. H. Cushman: *Geometry of perturbation theory*. In "Deterministic chaos in General Relativity", Editors: D. Hobill et al. Springer Verlag 1994, 89–101.

[8] R. H. Cushman and L. Bates: *Global aspects of classical integrable systems*. Birkhauser, Basel, 1997.

[9] S. Lynch: *Dynamical Systems with Applications using Maple*. Birkhauser, Basel, 2001.

[3] Used here: Maple 2016:1a (build 1133417). Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario. Maxima version was 5.38.0.

[10] J. Moser: *Regularization of Kepler's problem and the averaging method on a manifold*. Comm. on Pure and Appl. Math. **XXIII** (1970) 609–636.

[11] M. Ostrogradsky: *Memoires sur les equations differentielles relatives au problème des isoperimètres*. Mem. Acad. St. Petersbourg, VI **4** (1850) 385517.

[12] M. Pavšič: *Stable self-interacting Pais-Uhlenbeck oscillator*. Mod. Phys. Letters **A28** 36 (2013) 1350165.

[13] G. Schwarz: *Smooth funtions invariant under the action of a compact Lie group*. Topology **14** (1975) 63–68.

[14] S. Takato: *What is and how to Use KETCindy–Linkage Between Dynamic Geometry Software and KETCindy Graphics Capabilities*. In "Mathematical Software –ICMS 2016", Editors: G-M. Greuel et al. Lecture Notes in Computer Science **9725**, Springer, Cham, 2016.

[15] S. Takato, A. McAndrew, J. A. Vallejo and M. Kaneko: *Collaborative Use of KETCindy and Free Computer Algebra Systems*. Mathematics in Computer Science **11** 3–4 (2017) 503–514.

[16] S. Takato: *Brachistochrone Problem as Teaching Material–Application of KETCindy with Maxima*. In "Computational Science and Its Applications–ICCSA 2017", Editors: O. Gervasi et al. Lecture Notes in Computer Science **10407**. Springer, Cham, 2017.

[17] S. Takato and J. A. Vallejo: *Interfacing Free Computer Algebra Systems and C with KETCindy*. In "Computer Algebra Systems in Teaching and Research", Siedlce University of Natural Sciences and Humanities Volume **6** (2017) 172–185.

Acknowledgment

The authors express their gratitude to Masataka Kaneko (Tōhō University) and Yasuyuki Nakamura (Nagoya Institute of Technology) for many fruitful discussions about the topics in this paper. Thanks are also due to Richard Fateman for developing `rkfun`.

Setsuo Takato
Faculty of Sciences
Tōhō University
Funabasi (Chiba) Japan
e-mail: takato@phar.toho-u.ac.jp

José A Vallejo
Faculty of Sciencies
State University of San Luis Potosí
San Luis Potosí (SLP) México
e-mail: jvallejo@fc.uaslp.mx