`LP-VIcode`: a program to compute a suite of variational chaos indicators

D. D. Carpintero[a,b], N. Maffione[a,b], L. Darriba[a,b]

[a]*Facultad de Ciencias Astronómicas y Geofísicas, Universidad Nacional de La Plata, Argentina*
[b]*Instituto de Astrofísica de La Plata, UNLP-Conicet, Argentina*

Abstract

An important point in analysing the dynamics of a given stellar or planetary system is the reliable identification of the chaotic or regular behaviour of its orbits. We introduce here the program `LP-VIcode`, a fully operational code which efficiently computes a suite of ten variational chaos indicators for dynamical systems in any number of dimensions. The user may choose to simultaneously compute any number of chaos indicators among the following: the Lyapunov Exponents, the Mean Exponential Growth factor of Nearby Orbits, the Slope Estimation of the largest Lyapunov Characteristic Exponent, the Smaller ALignment Index, the Generalized ALignment Index, the Fast Lyapunov Indicator, the Othogonal Fast Lyapunov Indicator, the dynamical Spectra of Stretching Numbers, the Spectral Distance, and the Relative Lyapunov Indicator. They are combined in an efficient way, allowing the sharing of differential equations whenever this is possible, and the individual stopping of their computation when any of them saturates.

Keywords:
Stellar systems, Planetary systems, Chaos, Numerical algorithms

1. Introduction

The structure and the dynamics of a self-consistent stellar system are related to each other; in particular, the orbits of the stars determine the mass distribution of the system. Thus, in order to model stellar systems, it is of interest to characterize the orbits supported by the astrophysical potential giving rise to the model. Schwarzschild (1979, 1982), for example, was able to construct steady state distribution functions of stellar sys-

tems starting from a given potential and a well chosen set of regular orbits supported by the former. Since these pioneer works, his method has been used by many authors (e.g. Merritt and Fridman, 1996; van den Bosch et al., 2008; Deibel et al., 2011, among many others). This led to the now common claim that regular orbits constitute a dynamical skeleton for stellar systems. Moreover, those regular orbits that turn out to be resonant and stable are the most important ones, because they spawn entire regular families around them; thus, they constitute the backbone of the system. On the other hand, the existence of chaotic orbits in realistic models is nowadays beyond doubt (e.g., Valluri and Merritt, 1998; Voglis et al., 2002; Muzzio et al., 2005; Deibel et al., 2011; Zorzi and Muzzio, 2012). They are important to the dynamical evolution of a stellar system because their diffusion through their allowed phase space may impact the system as a whole (e.g. Mahon et al., 1995; Kandrup and Siopis, 2003); even a new version of the Jeans theorem was advanced regarding the role of chaotic orbits that the traditional version ignores completely (Kandrup, 1998). Another interesting point regarding the relationship between the dynamics of a system and its chaotic orbits is that, among the latter, those obeying only one isolating integral of motion (dubbed fully chaotic) and those obeying two such integrals (partially chaotic) occupy quite different spatial regions (Muzzio and Mosquera, 2004; Zorzi and Muzzio, 2012). This suggests different dynamical roles for each type of chaoticity, though what the roles may be is still unknown. Although for different reasons, the dynamics of planetary systems also strongly depends on the chaoticity of their orbits, in particular, in terms of the stability of those systems (Laskar, 1990).

Thus, an important aspect of the dynamical study of a given stellar or planetary system consists in identifying the chaotic or regular behaviour of its orbits. Since the early work of Hénon and Heiles (1964), the number of indicators of chaos has steadily grown as time went by. Aside from the qualitative method of the Poincaré surfaces of section (e.g. Binney and Tremaine, 2008), the tools for such analyses are based either on the study of the fundamental frequencies of the trajectories (e.g. Binney and Spergel, 1982; Laskar, 1990; Šidlichovský and Nesvorný, 1996; Carpintero and Aguilar, 1998; Papaphilippou and Laskar, 1998), or else on the study of the evolution of deviation vectors, the so-called variational chaos indicators (CIs hereinafter, see bibliography below). However, among the available plethora, only a few of them are usually employed by most dynamicists. Therefore, it would prove useful to have a tool with which one can compute several CIs in an easy and fast way. This is the main

motivation of the present `LP-VIcode`, which stands for *La Plata Variational Indicators code.*

The alpha version was first introduced in Darriba et al. (2012a). Here, we present a significantly improved code which constitutes the first stable version of the program. The main achievement of the code is its speed: neither the orbit nor any of the sets of variational equations are computed more than once in each time step, even when they may be requested by more than one CI. The library of CIs in the present version include the following: the numerically computed Lyapunov Exponents, known as Lyapunov Indicators, Lyapunov Characteristic Exponents, Lyapunov Characteristic Numbers, or even Finite-Time Lyapunov Characteristic Numbers (LIs, Benettin et al., 1976, 1980a,b); the Mean Exponential Growth factor of Nearby Orbits (MEGNO, Cincotta and Simó, 2000; Cincotta et al., 2003); the Slope Estimation of the largest Lyapunov Characteristic Exponent (SElLCE, Cincotta et al., 2003); the Smaller ALignment Index (SALI, Skokos, 2001); the Generalized ALignment Index (GALI, Skokos et al., 2007); the Fast Lyapunov Indicator (FLI, Froeschlé et al., 1997; Lega and Froeschlé, 2001); the Orthogonal Fast Lyapunov Indicator (OFLI, Fouchard et al., 2002); the Spectral Distance (SD, Voglis et al., 1999); the dynamical Spectra of Stretching Numbers (SSNs, Voglis and Contopoulos, 1994; Contopoulos and Voglis, 1996); and the Relative Lyapunov Indicator (RLI, Sándor et al., 2000, 2004).[1] The potential of the dynamical system, which must be supplied by the user together with the corresponding accelerations and variational equations, may have any number of dimensions (or degrees of freedom). The user may then choose to compute simultaneoulsy any subset of the abovementioned CIs.

As already stated, the main goal of the program is to compute, for one or more orbits in a given potential, the set of chosen CIs at the same time and in an efficient way. This means that whenever two or more CIs need the same differential equations, the last would be integrated only once. Thus, at each run of the program the set of equations adaptes to the set of CIs the user has chosen to compute.

---

[1]A minimal package of CIs for analysing a general Hamiltonian system is studied in Maffione et al. (2011); Darriba et al. (2012b); Maffione et al. (2013).

2. Structure of the code

The program is written in standard Fortran77[2], except for the common nonstandard extensions `DO-ENDDO`, `INCLUDE`, `DOWHILE`, lowercase characters, inline comments, names longer than 6 characters, and variable names containing the nonstandard character "_". All real variables are `DOUBLE PRECISION`.

The program reads initial conditions (hereinafter, i.c.) for one or more orbits, and integrates them, computing at the same time the set of CIs chosen by the user. The integrator is a standard Bulirsch-Stoer routine. The fixed time step asked by the user is internally split whenever either the absolute or relative error of the step is greater than the allowed tolerance, set to 10^{-13} for our experiments except for the MEGNO, for which the tolerance was set to 10^{-12}. This difference turns out to be not important at all for the MEGNO, but avoids the slowing down of the code when this CI is combined with any other.

The code is organized as a main program which controls the flux of the computation, plus a set of routines to accomplish the different tasks. These routines are grouped into four categories: those which take care of the input, initialization and output, those which compute the CIs themselves, those which are mathematical tools, and those which deal with the dynamics of the system. This last category includes the routines the user must supply, i.e., those that correspond to the potential being studied.

2.1. The main program

The main program begins with a call to the input routine, after which the main loop sweeps the different orbits. Inside this loop, the differential equations are adjusted to the set of CIs to be computed. Then, any needed deviation vector (DV) is generated, and the equations of motion as well as the variational equations are integrated. At each step, only one call to the integration routine is done, irrespective of the CIs being computed. After that, the chosen CIs are computed. If any of the CIs saturates during the integration of an orbit, the last value of the former is output, and the corresponding equations are deleted if none of the other CIs is using them. This continues until the integration loop is finished, in which case a new set of i.c.

---

[2]There is also a version written in Fortran90 for parallel computing, though in a developing stage.

is read, the differential equations are reset for the new orbit, and the whole process is repeated.

2.2. Input, initialization, and output

The user provides the input parameters (time step, length of the integration, choice of the CIs, frequency of the output, where to find the i.c., etc.) through a file. Although not strictly an input from the user, he/she should also provide the routines with which the potential, the accelerations and the variational equations should be computed. The syntax of these routines, as well as the structure of the input file and of the possible outputs, are described in detail in the User's Guide provided with the program.

The initialization is a three-step process. First, the DVs are generated. Second, the initial phase space values and the initial deviation vectors (hereinafter IDVs) are stored in a matrix which therefore holds all the dependent variables of the problem. Finally, according to the chosen CIs, the bookkeeping of equations and DVs to be used is done (more on this below).

2.3. Indicators

We briefly introduce here the corresponding definitions and algorithms of the indicators used in the code. Consider the Hamiltonian $H(\mathbf{w})$, with $\mathbf{w} = (\mathbf{p}, \mathbf{q})$ a phase-space vector, $\mathbf{q}$ the position of the orbit and $\mathbf{p}$ its momentum. Introducing the function $\mathbf{F}$:

$$\mathbf{F}(\mathbf{w}) = (\partial H/\partial \mathbf{q}, \ -\partial H/\partial \mathbf{q}), \tag{1}$$

the equations of motion can be written as

$$\dot{\mathbf{w}} = \mathbf{F}(\mathbf{w}). \tag{2}$$

Solving this system of first order ordinary differential equations with i.c. $\mathbf{w}_0$, we obtain the solution

$$\mathbf{w}(t) = \mathbf{\Phi}^t \mathbf{w}_0, \tag{3}$$

where $\mathbf{\Phi}^t$ is the operator evolution. Now, taking the first variation of Eqs. (3), we obtain the so-called variational equations

$$\delta \mathbf{w}(t) = \mathrm{d}_{\mathbf{w}} \mathbf{\Phi}^t \delta \mathbf{w}_0, \tag{4}$$

where $\delta \mathbf{w}$ is a DV and $\mathrm{d}_{\mathbf{w}}$ stands for the operator derivative with respect to the components of $\mathbf{w}$.

We can gain fundamental information about the Hamiltonian flow in the neighborhood of any orbit through the so-called LIs (Benettin et al., 1976, 1980a,b):

$$\text{LI}_j(t) = \frac{1}{t} \ln \frac{\|\delta\mathbf{w}_j(t)\|}{\|\delta\mathbf{w}_{j0}\|}, \tag{5}$$

where $\|\cdot\|$ is some norm (usually the Cartesian one), and the subindex j indicates that the calculation is to be performed using the jth deviation vector. For a system of n dimensions, there will be $2n$ LIs, n of which will be positive and the other n will be their respective opposites. Since, before the actual computation, the deviation vectors should be orthonormalized (Benettin et al., 1980a,b), we do this in the code by means of a Modified Gram-Schmidt procedure. It is worth noticing that, having tried both the discrete and continuous Eckmann and Ruelle algorithms of orthogonalisation (Eckmann and Ruelle, 1985), we have decided to discard them because, in the first case (discrete), the integration time within which the results were reliable turned out to be limited, whereas in the second case (continuous) the computing time was about 250 per cent that of the Gram-Schmidt procedure.

Among the set of LIs, the largest one is the most important, since it suffices to determine whether an orbit is regular or chaotic: if it tends to a positive value, then the orbit is chaotic, regardless of the behaviour of the rest of the LIs. On the other hand, if the largest LI tends to zero, the rest of LIs will also tend to zero, and the orbit in this case is regular. Hereinafter, we will refer to the largest LI just as LI.

The LI is the truncated (in time) value of the largest Lyapunov Characteristic Exponent σ_1 that can be defined in an integral form as:

$$\sigma_1 = \lim_{t\to\infty} \frac{1}{t} \int_0^t \frac{\|\dot{\delta\mathbf{w}}(t')\|}{\|\delta\mathbf{w}(t')\|} dt', \tag{6}$$

where $\dot{\delta\mathbf{w}}$ is a shorthand for $d(\delta\mathbf{w})/dt'$. Both σ_1 and the LI tend to a positive value for chaotic orbits, and tend to 0 for regular orbits.

Now we introduce the MEGNO Y (Cincotta and Simó, 2000; Cincotta et al., 2003) through the expression:

$$Y(t) = \frac{2}{t} \int_0^t \frac{\|\dot{\delta\mathbf{w}}(t')\|}{\|\delta\mathbf{w}(t')\|} t' dt'. \tag{7}$$

The actual expression of the indicator used in the code is its time average:

$$\overline{Y}(t) = \frac{1}{t} \int_0^t Y(t') \, dt'.$$ (8)

The asymptotic behavior of $\overline{Y}$ can be described as $\overline{Y}(t) \to a \cdot t + b$, where $a = \sigma_1/2$ and $b = 0$ for irregular, stochastic motion, while $a = 0$ and $b = 2$ for quasi-periodic motion. It is also possible to estimate the LI of the orbit from $a \cdot t + b$ by applying a linear least-squares fitting on $\overline{Y}(t)$. This estimation is the SElLCE indicator (Cincotta et al., 2003), in the computation of which only the last 80 per cent of the orbit is used, in order to avoid any initial transient.

2.3.2. The SALI and the GALI

Let $\delta\mathbf{w}_1$ and $\delta\mathbf{w}_2$ be two DVs belonging to the same orbit, linearly independent at $t = 0$, and let $\delta\hat{\mathbf{w}}_i(t) = \delta\mathbf{w}_i(t)/\|\delta\mathbf{w}_i(t)\|$, $i = 1, 2$ be the corresponding unit DVs. The parallel and antiparallel alignment indices are then defined as $d_- = \|\delta\hat{\mathbf{w}}_1 - \delta\hat{\mathbf{w}}_2\|$ and $d_+ = \|\delta\hat{\mathbf{w}}_1 + \delta\hat{\mathbf{w}}_2\|$, respectively (Skokos, 2001). Since, in a chaotic motion, two linearly independent DVs are expected to align with the same direction, we will have $d_- \to 0$ and $d_+ \to 2$ or $d_- \to 2$ and $d_+ \to 0$. On the other hand, if the motion is regular, d_- and d_+ will oscillate within the interval $(0, 2)$. The SALI (Skokos, 2001) is defined as the smaller of these two indices:

$$\mathrm{SALI}(t) = \min(d_+, d_-),$$ (9)

so that SALI $\to 0$ if the orbit is chaotic, whereas SALI $\nrightarrow 0$ if the orbit is regular.

Skokos et al. (2007) generalize the SALI introducing an alternative way to compute it. They evaluate the wedge product $\|\delta\hat{\mathbf{w}}_1 \wedge \delta\hat{\mathbf{w}}_2\| \equiv (d_+ \cdot d_-)/2$, that is, the area of the parallelogram formed by the two DVs, which has the same behaviour as the SALI for regular and chaotic motions. Taking more independent IDVs, the wedge product can be generalized up to k factors, $2 < k \leq 2n$, with n the number of degrees of freedom, representing the volume of the parallelepiped formed by the k DVs. The GALI_k is defined as the volume of this k-parallelepiped:

$$\mathrm{GALI}_k(t) = \|\delta\hat{\mathbf{w}}_1(t) \wedge \delta\hat{\mathbf{w}}_2(t) \wedge \ldots \wedge \delta\hat{\mathbf{w}}_k(t)\|.$$ (10)

The implementation of Eq. (10) in the code is made through a singular value decomposition of the matrix of DVs (Skokos et al., 2008). As in the case of the SALI, the GALI_k for chaotic orbits tend to zero, although giving more information about the dynamics of the orbit than the SALI. On the other hand, if GALI_k tends to a non-zero value as t increases, the motion is regular.

2.3.3. The FLI and the OFLI

Given n linearly independent DVs in a $2n$ dimensional phase space, the FLI (Froeschlé et al., 1997; Lega and Froeschlé, 2001) at time t is defined as the greatest of the norms they had between $t = 0$ and the current t:

$$\text{FLI}(t) = \sup_t \left[\|\delta\mathbf{w}_1(t)\|, \|\delta\mathbf{w}_2(t)\|, \ldots, \|\delta\mathbf{w}_n(t)\| \right]. \tag{11}$$

It turns out that the FLI grows exponentially for chaotic motion and linearly for regular motion.

The OFLI (Fouchard et al., 2002), on the other hand, is computed like the FLI, but only the component orthogonal to the flow of each DV is taken into account. This modification makes the OFLI a CI that can easily distinguish periodicity of a regular motion: in this case, it oscillates around a constant value, while for chaotic and quasiperiodic motion it has the same behavior as the FLI.

2.3.4. The SSNs and the SD

The stretching number s_i (Voglis and Contopoulos, 1994; Contopoulos and Voglis, 1996) is defined as:

$$s_i = \frac{1}{\Delta t} \ln \frac{\|\delta\hat{\mathbf{w}}(t_i)\|}{\|\delta\hat{\mathbf{w}}(t_{i-1})\|}, \tag{12}$$

where Δt is the step of integration, and $t_i = i\Delta t$, $i \in \mathbb{N}$. The spectrum of the s_i, called SSN, is defined as the probability density of their values. If the s_i are binned into blocks of width Δs, then the SSN can be computed as:

$$\text{SSN}_j(t) = \frac{1}{Z} \frac{\Delta Z_j}{\Delta s}, \quad j = 1, \ldots, N, \tag{13}$$

where N is the number of blocks of the histogram, Z is the total number of s_i and ΔZ_j is the number of s_i in the j-th interval. However, when the

sample is very large, the analysis of the orbits by means of the SSN is no longer reliable. Thus, Voglis et al. (1999) introduce the SD:

$$\text{SD}^2(t) = \sum_{j=1}^{N} [\text{SSN}_{j,1} - \text{SSN}_{j,2}]^2 \cdot \Delta s, \tag{14}$$

where $\text{SSN}_{j,i}$ is the SSN_j corresponding to a DV $\delta \hat{\mathbf{w}}_i$. If the orbit is chaotic, the SD decreases towards zero. If, instead, the orbit is regular, the SD tends to a constant non-zero value.

2.3.5. The RLI

Let $\mathbf{w}(t)$ be an orbit with i.c. $\mathbf{w}_0$ (the *base orbit*), and $\mathbf{w}(t) + \Delta\mathbf{w}(t)$ be another orbit with i.c. $\mathbf{w}_0 + \Delta\mathbf{w}_0$ (the *shadow orbit*), where $\Delta\mathbf{w}_0$ is small. From the LIs of these orbits $\text{LI}_0(t)$ and $\text{LI}_1(t)$, respectively, the RLI (Sándor et al., 2000, 2004) is defined as:

$$\text{RLI}(t) = \|\text{LI}_1(t) - \text{LI}_0(t)\|. \tag{15}$$

Since the LI usually has fast fluctuations, it is better to smooth those out by averaging with respect to time, so the RLI is usually redefined and computed as

$$\overline{\text{RLI}}(t) = \frac{1}{t} \sum_{i=1}^{t/\Delta t} \text{RLI}(t_i), \tag{16}$$

where Δt is the step of integration and $t_i = i\Delta t$, $i \in \mathbb{N}$. The RLI values for chaotic motion exceed by several orders of magnitude those associated with regular motion.

2.4. Saturation

After the requested CIs are computed, the code takes care of any possible saturation: since there are CIs that tend to infinity when chaos is present, they are not further integrated after reaching a given threshold in order to avoid overflows (10^{16} in the case of the FLI and the OFLI, and 30 in the case of the MEGNO). Similarly, there are CIs (SALI and GALIs) that tend to 0 when chaos is present; for these CIs, we stop their computation when a value of 10^{-16} is reached, i.e., we take the machine precision as a numerical zero. In all cases, the differential equations are adapted to the new set of CIs by removing those which correspond to the saturated ones.

2.5. Mathematical and dynamical routines

The purely mathematical routines are standard ones; some of them are adaptations of those published in Press et al. (1992). The dynamical routines comprise two sets: those already written down, and those that must be provided by the user. In the first group, besides the computation of the energy, there is a routine which sets all the equations of motion and variational equations that are the same irrespective of the potential used. These equations are the following. Let $\mathbf{x}$ be the position vector in an n-dimensional space. The equations of motion corresponding to some potential Φ on this space are:

$$\ddot{\mathbf{x}} = -\nabla\Phi(\mathbf{x}), \tag{17}$$

which, for the sake of their numerical integration, are usually broken into the first-order differential equations

$$
\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{v}, & (18) \\
\dot{\mathbf{v}} &= -\nabla\Phi(\mathbf{x}), & (19)
\end{aligned}
$$

where $\mathbf{v}$ is the velocity vector. As can be seen, Eqs. (18) contain no references to the potential, so we have already coded them into our program. The same goes for the variational equations. Let $\mathbf{w} = (\mathbf{x}, \mathbf{v})$ be a phase-space vector, and let $\mathbf{F}$ be such that Eqs. (18) and (19) are written as in Eq. (2). Then, the variational equations are

$$\frac{\mathrm{d}(\delta\mathbf{w})}{\mathrm{dt}} = \left.\frac{\partial\mathbf{F}}{\partial\mathbf{w}}\right|_{\mathbf{w}} \delta\mathbf{w}, \tag{20}$$

which are the same as Eq. (4), but now written in terms of the function $\mathbf{F}$. Again, the first n equations, namely

$$\dot{\delta\mathbf{x}} = \delta\mathbf{v} \tag{21}$$

are independent of the potential, so we have wired them into the code.

The routines the user must supply should only contain, then, the n equations of motion (19) and the n variational equations (20) which do depend on the potential, plus a routine to compute the value of the potential itself. The User's Guide contains a complete example which can be used as a template.

3. Management of the differential equations

We took advantage of the passing-by-address paradigm of the Fortran77 language to manage the equations to be integrated according to the CIs being computed, i.e., those requested minus those that have saturated.

To this end, we first split the CIs into three groups, according to the type of DVs they use. These groups are:

(I) Normalized and orthogonalized DVs (LI).
(II) Normalized DVs (SALI, GALI, SD, SSN, RLI; the RLI needs also a second (shadow) orbit with its own normalized DV.)
(III) Unnormalized DVs (MEGNO, SElLCE, FLI, OFLI).

Then, we set up a $N \times (M+1)$ matrix (Fig. 3), where N is the dimension of the phase space (which depends on the potential being studied), and M is the total number of DVs to be integrated, which is computed from the requested CIs as follows. If the group (I) is to be computed, it requires N DVs. In group (II), one DV is needed if only the SD and/or the SSNs and/or the RLI are sought (the second orbit of the RLI is not counted in; an additional DV is integrated separately for the shadow orbit), two DVs if the SALI is present, or N DVs if the GALIs enter the computation. If there are a combination of these CIs to be computed, only the greater number of DVs are set up; the CIs requiring a number of DVs less than this maximum will share the DVs with the others. Finally, group (III) asks for only one DV, irrespective of the number of indicators present. But, if the MEGNO or the SElLCE are requested, an additional column is set up to hold the two additional dependent variables of the differential equations that are needed to compute those CIs (marked as `aux` in Fig. 3). Note that these last two dependent variables remain two in number irrespective of N. The remaining (first) column of the matrix holds the coordinates of the orbit itself.

If, for example, the potential is 2D (as in Fig. 3), and the GALI is not computed, the program eliminates the DVs that belongs exclusively to this CI (DV7 and DV8), leaving the other two of group (II) for the computation of the other CIs. When a CI which is being computed saturates, the program also eliminates its DVs, though only if they are not being used by another CI. In this way, the computing time is fully optimized.
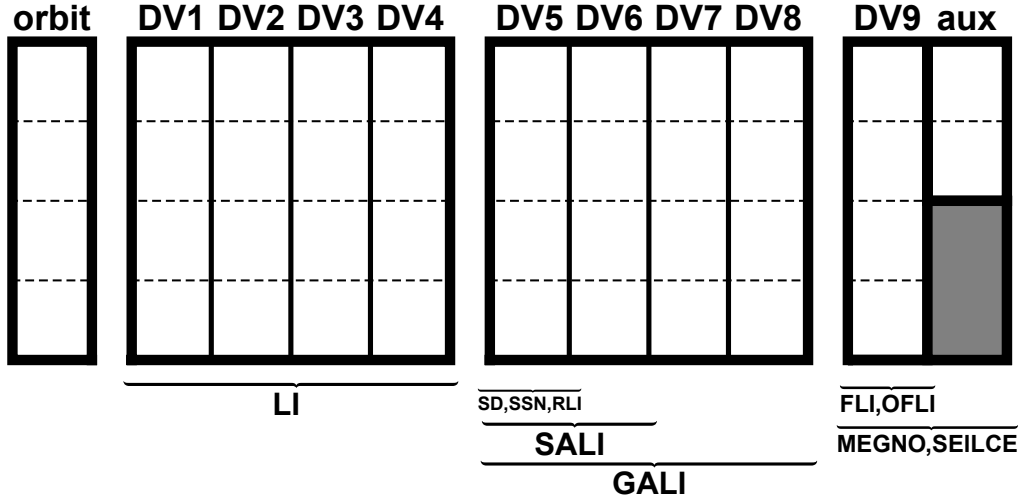
Figure 1: Matrix of dependent variables, for a 2D potential (i.e., 4D phase space). Rows are coordinates of the phase space, columns are the various sets of dependent variables, that is, orbit and DVs. The columns are separated into the three groups of DVs defined in the text. Here, the matrix corresponds to all the CIs switched on.

4. Results

We chose two astronomical potentials, the 2D Hénon-Heiles (Hénon and Heiles, 1964) and a 3D triaxial NFW profile (Vogelsberger et al., 2008), in order to show the performance of the code, as well as some additional characteristics. In all cases we used an Intel Core i5 with four cores, CPU at 2.67 GHz, 3 GB of RAM, an OS of 32 bits, and the `gfortran` compiler of `gcc` version 4.4.4, without any optimizations.

4.1. The Hénon-Heiles potential

For the Hénon-Heiles potential, we took three sets of i.c. For each of these sets, we recorded the elapsed times in computing all the indicators both together and individually, in order to assess the saved time.

The first set, dubbed H1, consisted of four orbits lying on the $x = p_y = 0$ line (Cincotta et al., 2003). Table 1 shows the i.c., which correspond to a stable periodic orbit sp, a quasi-periodic orbit qp, a quasi periodic orbit near an unstable periodic orbit up, and a chaotic orbit $c1$, all lying on the $E = 0.118$ energy surface.

Table 1: Initial conditions for the experiment H1.

Name	x	y	p_x	p_y
sp	0	0.295456	0.407308431	0
qp	0	0.483000	0.278980390	0
up	0	0.469120	0.291124891	0
$c1$	0	0.509000	0.254624859	0

Table 2: CPU times of the code. T_1 stands for the sum of all the previous rows; T_2 is the time elapsed when all the indicators were computed at the same time. All times in seconds, rounded to the first decimal.

	H1	H2	H3	N1	N2
orbit	0.4	—	—	—	—
LI	1.5	36.0	4534.9	137.2	—
SALI	0.9	21.8	2750.1	44.5	—
GALIs	2.5	64.5	8061.5	135.3	135.3
SD	0.9	22.5	2834.6	44.6	—
RLI	1.2	29.7	3753.4	50.9	50.9
MEGNO	0.8	19.7	2473.2	32.8	32.8
FLI	0.7	15.4	1947.2	28.0	28.0
T_1	8.9	209.6	26355.4	473.4	247.1
T_2	5.2	126.0	15770.5	370.5	204.6

We integrated the orbits until 1000 time units (t.u.), with a time step of 0.05 t.u.; the output to screen was enabled. The values of the indicators were dumped to files every 20 time steps. The IDVs were chosen at random and orthonormalized. Column H1 of Table 2 shows the outcome; as can be seen comparing T_1 (individual CIs) with T_2 (simultaneous CIs), a reduction of about 40 per cent in time was obtained by computing all the indicators at the same time. Taking into account that $GALI_2$ is the same as SALI (though they are computed through different algorithms), we repeated the above experiment but without the computation of the SALI, which in this case is superfluous. The resulting times ($T_1 = 8.0$, $T_2 = 5.1$) gave a reduction of 36 per cent.

```
Orbit #      1 10% 20% 30% 40% 50% 60% 70% 80% 90% DE = 2.16E-13
Orbit #      2 10% 20% 30% 40% 50% 60% 70% 80% 90% DE = 9.03E-13
Orbit #      3 10% 20% 30% 40% 50% 60% 70% 80% 90% DE = 5.88E-13
Orbit #      4 10% 20% 30% 40% 50% 60% 70% 80% 90% DE = 3.91E-13
Orbit #      5 10% 20% 30% 40% 50% 60% 70% 80% 90% DE = 1.18E-14
```

Figure 2: Typical output to screen: orbit number, progress of the integration (with percentages appearing on the screen as the integration progresses), and conservation of the energy for each orbit.

The second set, dubbed H2, was built with 1000 i.c. taken from the region $x = 0$, $y \in [-0.1, 0.1]$, $p_y = 0$, with the energy fixed at $E = 0.118$. The integration time was 100 t.u., the time step was set to 0.05 t.u., the IDVs were chosen as in experiment H1, and the output to screen was disabled. The indicators were dumped only at their last values. The orbits were not dumped to file, in the understanding that when there are lots of i.c., the user probably won't focus in studying the aspect of each orbit in the configuration space. Column H2 of Table 2 shows the elapsed times. Again, a reduction of about 40 per cent in time was obtained. We repeated the experiment without the SALI, obtaining $T_1 = 188.2$ and $T_2 = 125.8$, i.e., a reduction of 33 per cent.

The third set, dubbed H3, consisted of 125,751 i.c. from the region $x = 0$, $y \in [-0.1, 0.1]$, $p_y = \in [-0.05, 0.05]$, with the energy fixed at $E = 0.118$. The parameters were the same as in experiment H2. As can be seen from column H3 of Table 2, in this longer experiment the time saved was again about 40 percent. Without the SALI, we obtained $T_1 = 23604.7$ and $T_2 = 15734.1$, again, a reduction of 33 per cent.

In order to show some actual input/output of the code, we now compute for the Hénon-Heiles potential a subset of CIs (the MEGNO, the LI, the RLI, the GALI$_3$ and the OFLI) using the i.c. of experiment H1, plus a second chaotic orbit lying in a large chaotic sea, dubbed $c2$, with i.c. $x = 0$, $y = 0.56$, $p_y = 0.112$ and $E = 0.118$ (Cincotta et al., 2003). The integration time was 15,000 t.u. for each orbit, whereas the total cpu time was 55.5 s.

If the flag for visual control of the processing is enabled, the progress of the computation and the energy conservation of each orbit should appear on the screen, as is shown in Fig. 2.

One of the input parameters the user should give is an alphanumeric

```
1  E =   0.11799999984378129       DE =   2.16281795019983474E-013
2  E =   0.11800000000227605       DE =   9.03349899574294735E-013
3  E =   0.11799999987237160       DE =   5.88277073642108224E-013
4  E =   0.11799999974371828       DE =   3.90695010149895051E-013
5  E =   0.11799999989039134       DE =   1.17608371361909766E-014
```

Figure 3: Typical output to file `*.ene`: orbit number, energy and conservation of the energy for each orbit.

```
0.1499500000000000E+05 0.5138152865503478E+03 0.1340629863642921E+01
0.1500000000000000E+05 0.5139749984798356E+03 0.1340629863642921E+01
```

```
0.9700000000000000E+03 0.8812457069534003E+16 0.8635444202241726E+16
0.9704500000000000E+03 0.1000000000000000E+17 0.1000000000000000E+17
```

Figure 4: Typical output to file `*.fli` (only the two last rows for the first and last orbits are shown): time, FLI and OFLI.

prefix for the output files. Choosing for example `hh` for the prefix of the present run, the output files would be `hh.ene`, `hh.megno`, `hh.rli`, `hh.gali` and `hh.fli`. The first one will contain the energy and its conservation for each orbit, as shown in Fig. 3.

The files `hh.megno`, `hh.rli`, `hh.gali` and `hh.fli` will contain the values of the CIs. Fig. 4 shows the last two lines of the first and last orbits on the file `hh.fli` of our example. It can be seen that the first orbit reached the end of the integration (i.e., the FLI didn't saturate), whereas the FLI of the last one saturated at $t = 970.45$ t.u., from which time the integration of this CI stopped.

Figs. 5, 6 and 7 show the resulting CIs for the five orbits considered. The top left panel of Fig. 5 shows the behaviour of the MEGNO for the three regular orbits. We can see that the orbit sp tends to 2 from below and the orbit up tends to 2 from above, as expected from the dependence of the MEGNO on the stability of the orbit (Cincotta et al., 2003). On the other hand, the orbit qp quickly increases to 2, again as expected. The top right panel of Fig. 5 shows the behaviour of the MEGNO for the chaotic orbits.
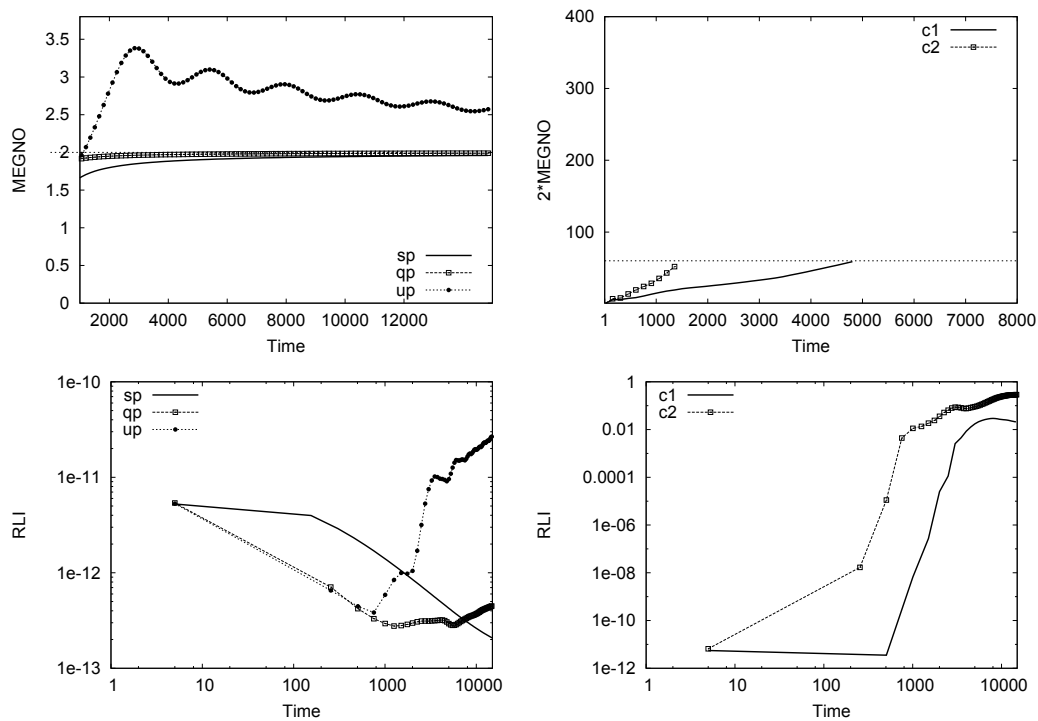
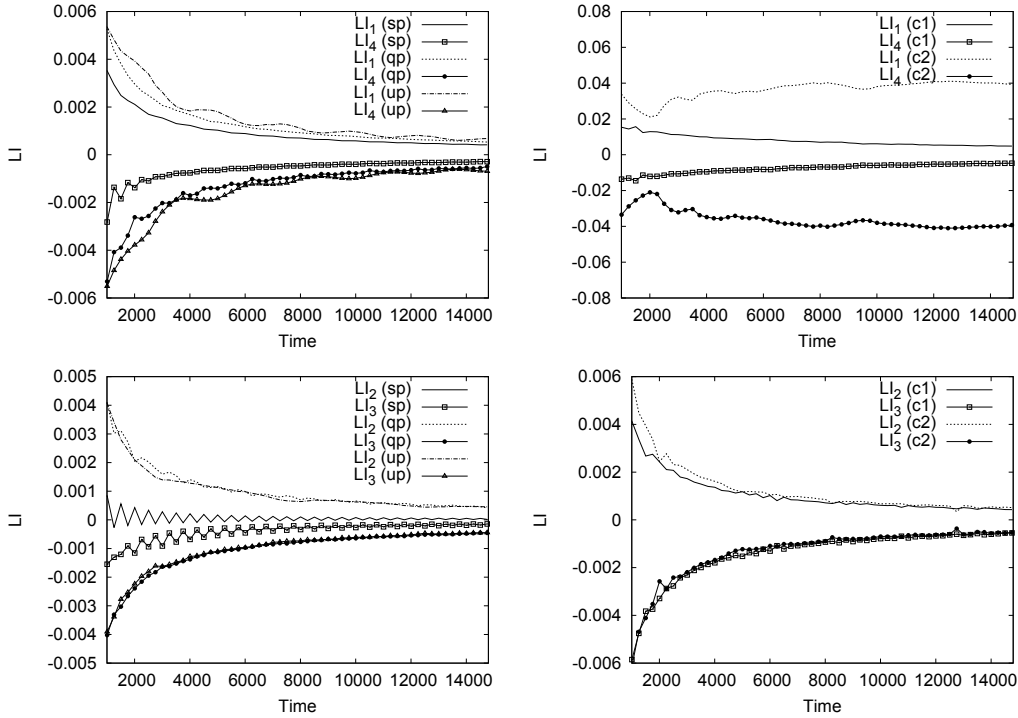Figure 5: Examples of the time evolution for the MEGNO (top panels) and the RLI (bottom panels).

16

Figure 6: Examples of the time evolution of the LI_1 and LI_4 (top panels), and of the LI_2 and LI_3 (bottom panels)

It is clearly seen that, once the MEGNO reached its saturation value, its computation was stopped. These plots should be compared with those of fig. 1(d) and (c) of Cincotta et al. (2003). The bottom panels of Fig. 5 show the RLI for the five orbits of the sample. As expected, the RLI maintains very small values for regular orbits, and reaches relatively high values for chaotic ones.

Fig. 6 shows the evolution of the different LIs for the three regular orbits (left panels) and for the two chaotic orbits (right panels). The top panels show the evolution of LI_1 and LI_4, whereas the bottom ones show the values of LI_2 and LI_3. These figures should be compared with figures 1(d) and 1(e) of Cincotta et al. (2003). In all cases, the LI behaves as expected. Though the LI_1 curve for the *c1* orbit seems to be tending to zero, it is really tending to a constant value greater than zero. This can be clearly seen in a logarithmic scale; we refrained to use that scale in order to be able to show all the LIs, both positive and negative.
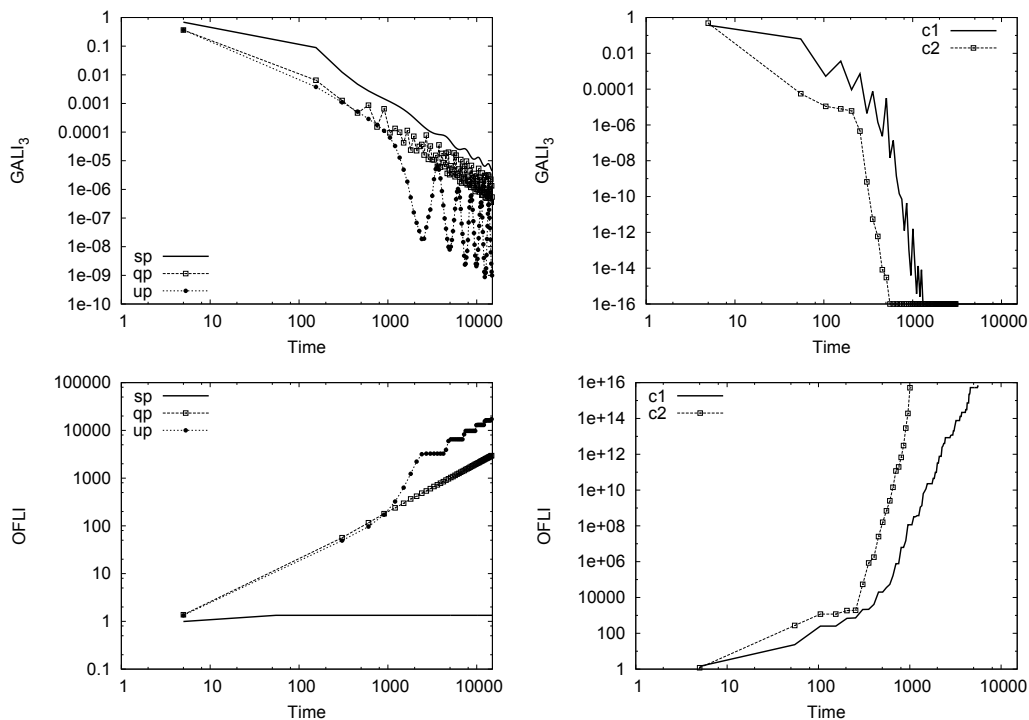
17

Figure 7: Examples of the time evolution for the GALI$_3$ (top panels) and the OFLI (bottom panels).

The top panels of Fig. 7 show the GALI$_3$ for our regular (left) and chaotic (right) orbits. The indicator behaves, again, as expected: a polynomial decrease in the case of regular orbits, and an exponential decrease in the chaotic case. The bottom panels of Fig. 7 present the OFLI for the five previous orbits. On the left panel, the CI shows a linear increment with time for the orbits *qp* and *up*, and a constant value for the orbit *sp*. The behaviour for the orbit *sp* reflects its proximity to a periodic orbit. For chaotic orbits, instead, the OFLI grows exponentially fast. It is seen that this CI reached the saturation value (10^{16}) for both chaotic orbits before the end of the integration, and their computation was therefore stopped.

Additionally, we have reproduced many other plots from papers in which different potentials and/or CIs were studied. We refrain from showing these plots because they are identical with the original ones. Suffice to say that we didn't find any instance of a different result from those of the literature.

Table 3: Constants used for the Φ_N potential.

A	4158670.1856267899
r_s	19.044494521343964
a	1.3258820840000000
b	0.86264540200000000
c	0.70560584600000000

4.2. The triaxial NFW potential

We also probed our code with a 3D potential, which is a triaxial extension of the NFW profile (Vogelsberger et al., 2008):

$$\Phi_N = -\frac{A}{r_p} \ln\left(1 + \frac{r_p}{r_s}\right), \tag{22}$$

where A and r_s are constants, and

$$r_p = \frac{(r_s + r)r_e}{r_s + r_e}, \tag{23}$$

with

$$r_e = \sqrt{\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2}, \tag{24}$$

a, b, c constants, and $r = \sqrt{x^2 + y^2 + z^2}$. The values of the constants we used are listed in Table 3. In this experiment, dubbed N1, in which we integrated 140 orbits, the total integration time was 13 t.u., which with our choice of constants corresponds to 13 Gyr. The time step was 0.005 t.u., and the rest of parameters were as in experiment H1. In column N1 of Table 2 we list the CPU times of this experiment. The saved time was, in this case, around 22 per cent. Avoiding the SALI as in the former cases, the runs lasted $T_1 = 428.9$ t.u. and $T_2 = 369.6$ t.u., a 14 per cent of saved time.

We repeated this last experiment but with only the GALIs, the RLI, the MEGNO and the OFLI enabled (experiment N2). Since the SALI and the DS share equations with the GALIs, we expect that the saved time will be less than before. This is confirmed in column N2 of Table 2, where we can see now that the percentage gained is about 17 per cent.

```
7.004d0   1.566d0   1.757d0   90.786d0  65.561d0  -43.003d0  130.0d0
6.903d0  -1.000d0   1.020d0  -37.876d0   8.507d0   41.721d0   13.0d0
5.743d0   1.221d0  -0.576d0   -8.451d0  10.923d0   21.592d0    1.3d0
```

Figure 8: Initial conditions for experiment N3. For each orbit (row) the given data are: x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$ and the desired time of integration.

Finally, we tested the implementation of the CIs that have not been used in the last experiment. In this new run, N3, we computed the LIs, the SALI and the FLI of three orbits for different intervals of integration. The values of the IDVs were taken random and orthonormal. The i.c. are listed in Fig. 8, just like it would appear in the i.c. file. Notice that, since we want the orbits to be integrated for different time intervals, the latter may be inserted at the end of each row of i.c.; in this way, the program automatically understands that the time of integration should be taken from these numbers.

Fig. 9 shows the results. The vertical dashed lines mark the three different times of integration used in the experiment. The top left panel shows the LI: it decreases for the three orbits, thus identifying them as regular. However, the SALI (top right panel) decreases exponentially fast by the end of the time interval for orbit (a), which means that the latter is chaotic. The SALI of the other two orbits, on the other hand, seem to oscillate around a finite value, that is, they have the behaviour corresponding to regular motion. The bottom panel shows the FLI. It grows exponentially for orbits (a) and (b), implying that both orbits are chaotic. Therefore, it is clear that 13 and 1.3 t.u. (the integration times used for orbits (b) and (c), respectively) are not enough to reliably classify them. Nevertheless, the FLI and the SALI prove to be faster indicators than the LI, as many papers in the literature stand for (e.g., Skokos, 2001; Sándor et al., 2004).

We have also probed the LP-VIcode with a variety of other astronomical potentials, both 2D and 3D, playing with different combinations of indicators, and obtained the expected results in all cases. For example, adding the computation of the FLI to an experiment in which the MEGNO is already enabled, does not involve any additional computational time. Or, in computing the LIs and the GALIs at the same time, the only gain is that the orbit is computed only once, due to the fact that these indicators don't share any variational equations.
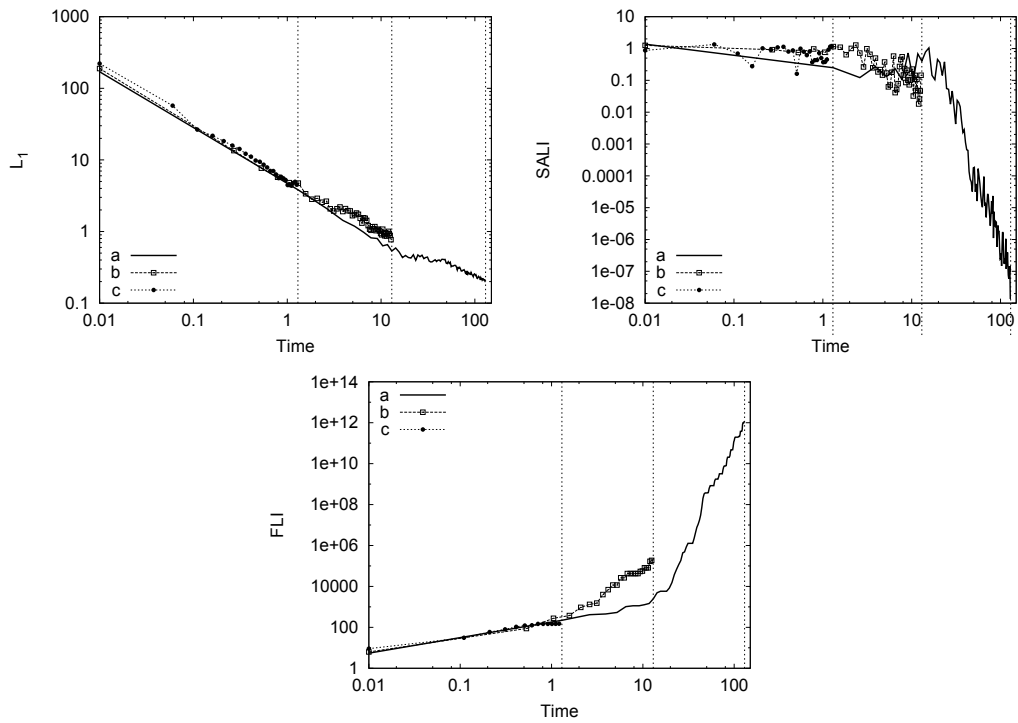
Figure 9: Examples of the time evolution for the LI (top left panel, labelled L_1), the SALI (top right panel) and the FLI (bottom panel) for the 3 orbits of experiment N3.

5. Conclusions

The main goal of the `LP-VIcode` is to cluster in a single, easy-to-use tool the plethora of CIs that are nowadays in the literature. The starting point is the present code, which is ready to use, except for the routines that compute the potential, the accelerations and the variational equations of the system, which should be provided by the user. The program can handle any number of dimensions, and is not limited to stellar or planetary systems, but it works with any dynamical system in which the abovementioned equations can be written down. The code is optimized to achieve the maximum speed, given a set of CIs to compute.

We expect researchers to collaborate with their own methods in developing newer versions of the code containing larger CIs' libraries. Also, we are open to people who might be interested in making the code more user-friendly, for example, changing the present command-driven interface to a menu-driven interface, etc.

The code, the User's Guide and complete examples are available at `www.fcaglp.unlp.edu.ar/LP-VIcode/`.

6. Acknowledgements

References

Benettin, G., Galgani, L., Giorgilli, A., Strelcyn, J.M., 1980a. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 1: Theory. Meccanica 15, 9–20.

Benettin, G., Galgani, L., Giorgilli, A., Strelcyn, J.M., 1980b. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 2: Numerical application. Meccanica 15, 21–30.

Benettin, G., Galgani, L., Strelcyn, J.M., 1976. Kolmogorov entropy and numerical experiments. Phys. Rev. A 14, 2338–2345.

Binney, J., Spergel, D., 1982. Spectral stellar dynamics. ApJ 252, 308–321.

Binney, J., Tremaine, S., 2008. Galactic Dynamics: (Second Edition). Princeton Series in Astrophysics, Princeton University Press.

Carpintero, D.D., Aguilar, L.A., 1998. Orbit classification in arbitrary 2D and 3D potentials. MNRAS 298, 1–21.

Cincotta, P.M., Giordano, C.M., Simó, C., 2003. Phase space structure of multi-dimensional systems by means of the mean exponential growth factor of nearby orbits. Physica D Nonlinear Phenomena 182, 151–178.

Cincotta, P.M., Simó, C., 2000. Simple tools to study global dynamics in non-axisymmetric galactic potentials - I. A&AS 147, 205–228.

Contopoulos, G., Voglis, N., 1996. Spectra of Stretching Numbers and Helicity Angles in Dynamical Systems. Celest. Mech. Dynam. Astron. 64, 1–20.

Darriba, L.A., Maffione, N.P., Cincotta, P.M., Giordano, C.M., 2012a. Chaos detection tools: the lp-vicode and its applications, in: P. M. Cincotta, C. M. Giordano, C.E. (Ed.), 3rd La Plata International School on Astronomy and Geophysics: Chaos, diffusion and non-integrability in Hamiltonian systems - Application to astronomy, Universidad Nacional de La Plata - Asociación Argentina de Astronomía. pp. 345–366.

Darriba, L.A., Maffione, N.P., Cincotta, P.M., Giordano, C.M., 2012b. Comparative study of variational chaos indicators and odes' numerical integrators. International Journal of Bifurcation and Chaos 22, 1230033.

Deibel, A.T., Valluri, M., Merritt, D., 2011. The Orbital Structure of Triaxial Galaxies with Figure Rotation. ApJ 728, 128. 1008.2753.

Eckmann, J.P., Ruelle, D., 1985. Ergodic theory of chaos and strange attractors. Reviews of Modern Physics 57, 617–656.

Fouchard, M., Lega, E., Froeschlé, C., Froeschlé, C., 2002. On the Relationship Between Fast Lyapunov Indicator and Periodic Orbits for Continuous Flows. Celest. Mech. Dynam. Astron. 83, 205–222.

Froeschlé, C., Gonczi, R., Lega, E., 1997. The fast Lyapunov indicator: a simple tool to detect weak chaos. Application to the structure of the main asteroidal belt. Planet. Space Sci. 45, 881–886.

Hénon, M., Heiles, C., 1964. The applicability of the third integral of motion: Some numerical experiments. AJ 69, 73.

Kandrup, H.E., 1998. Phase mixing in time-independent Hamiltonian systems. MNRAS 301, 960–974. `astro-ph/9809100`.

Kandrup, H.E., Siopis, C., 2003. Chaos and chaotic phase mixing in cuspy triaxial potentials. MNRAS 345, 727–742. `astro-ph/0305198`.

Laskar, J., 1990. The chaotic motion of the solar system - A numerical estimate of the size of the chaotic zones. Icarus 88, 266–291.

Lega, E., Froeschlé, C., 2001. On the relationship between fast lyapunov indicator and periodic orbits for symplectic mappings. Celest. Mech. Dynam. Astron. 81, 129–147.

Maffione, N.P., Darriba, L.A., Cincotta, P.M., Giordano, C.M., 2011. A comparison of different indicators of chaos based on the deviation vectors: application to symplectic mappings. Celest. Mech. Dynam. Astron. 111, 285–307. `1108.2196`.

Maffione, N.P., Darriba, L.A., Cincotta, P.M., Giordano, C.M., 2013. Chaos detection tools: application to a self-consistent triaxial model. MNRAS 429, 2700–2717. `1212.3175`.

Mahon, M.E., Abernathy, R.A., Bradley, B.O., Kandrup, H.E., 1995. Transient ensemble dynamics in time-independent galactic potentials. MNRAS 275, 443–453.

Merritt, D., Fridman, T., 1996. Triaxial Galaxies with Cusps. ApJ 460, 136. `arXiv:astro-ph/9511021`.

Muzzio, J.C., Carpintero, D.D., Wachlin, F.C., 2005. Spatial Structure of Regular and Chaotic Orbits in A Self-Consistent Triaxial Stellar System. Celest. Mech. Dynam. Astron. 91, 173–190.

Muzzio, J.C., Mosquera, M.E., 2004. Spatial Structure of Regular and Chaotic Orbits in Self-Consistent Models of Galactic Satellites. Celestial Mechanics and Dynamical Astronomy 88, 379–396.

Papaphilippou, Y., Laskar, J., 1998. Global dynamics of triaxial galactic models through frequency map analysis. A&A 329, 451–481.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes in Fortran 77: The Art of Scientific Computing. Cambridge University Press. 2nd edition.

Sándor, Z., Érdi, B., Efthymiopoulos, C., 2000. The Phase Space Structure Around L4 in the Restricted Three-Body Problem. Celest. Mech. Dynam. Astron. 78, 113–123.

Sándor, Z., Érdi, B., Széll, A., Funk, B., 2004. The Relative Lyapunov Indicator: An Efficient Method of Chaos Detection. Celest. Mech. Dynam. Astron. 90, 127–138.

Schwarzschild, M., 1979. A numerical model for a triaxial stellar system in dynamical equilibrium. ApJ 232, 236–247.

Schwarzschild, M., 1982. Triaxial equilibrium models for elliptical galaxies with slow figure rotation. ApJ 263, 599–610.

Skokos, C., 2001. Alignment indices: a new, simple method for determining the ordered or chaotic nature of orbits. Journal of Physics A: Mathematical and General 34, 10029–10043.

Skokos, C., Bountis, T., Antonopoulos, C., 2008. Detecting chaos, determining the dimensions of tori and predicting slow diffusion in Fermi-Pasta-Ulam lattices by the Generalized Alignment Index method. European Physical Journal Special Topics 165, 5–14. 0802.1646.

Skokos, C., Bountis, T.C., Antonopoulos, C., 2007. Geometrical properties of local dynamics in Hamiltonian systems: The Generalized Alignment Index (GALI) method. Physica D Nonlinear Phenomena 231, 30–54. 0704.3155.

Šidlichovský, M., Nesvorný, D., 1996. Frequency modified Fourier transform and its applications to asteroids. Celest. Mech. Dynam. Astron. 65, 137–148.

Valluri, M., Merritt, D., 1998. Regular and Chaotic Dynamics of Triaxial Stellar Systems. ApJ 506, 686–711. `arXiv:astro-ph/9801041`.

van den Bosch, R.C.E., van de Ven, G., Verolme, E.K., Cappellari, M., de Zeeuw, P.T., 2008. Triaxial orbit based galaxy models with an application to the (apparent) decoupled core galaxy NGC 4365. MNRAS 385, 647–666. `0712.0113`.

Vogelsberger, M., White, S.D.M., Helmi, A., Springel, V., 2008. The fine-grained phase-space structure of cold dark matter haloes. MNRAS 385, 236–254. `0711.1105`.

Voglis, N., Contopoulos, G., Efthymiopoulos, C., 1999. Detection of Ordered and Chaotic Motion Using the Dynamical Spectra. Celest. Mech. Dynam. Astron. 73, 211–220.

Voglis, N., Contopoulos, G.J., 1994. Invariant spectra of orbits in dynamical systems. Journal of Physics A: Mathematical and General 27, 4899–4909.

Voglis, N., Kalapotharakos, C., Stavropoulos, I., 2002. Mass components in ordered and in chaotic motion in galactic N-body models. MNRAS 337, 619–630.

Zorzi, A.F., Muzzio, J.C., 2012. Models of cuspy triaxial stellar systems - I. Stability and chaoticity. MNRAS 423, 1955–1963. `1204.5428`.