Smart: A program to automatically compute accelerations and variational equations

D. D. Carpintero[a,b,*], N. P. Maffione[c,d], F. A. Gómez[e,f]

[a]*Facultad de Ciencias Astronómicas y Geofísicas, Universidad Nacional de La Plata, La Plata, Argentina*
[b]*Instituto de Astrofísica de La Plata, UNLP–Conicet, La Plata, Argentina*
[c]*Universidad Nacional de Río Negro, Río Negro, Argentina*
[d]*Consejo Nacional de Investigaciones Científicas y Técnicas, Buenos Aires, Argentina*
[e]*Instituto de Investigación Multidisciplinar en Ciencia y Tecnología, Universidad de La Serena, Raúl Bitrán 1305, La Serena, Chile*
[f]*Departamento de Astronomía, Universidad de La Serena, La Serena, Chile*

Abstract

Modern astronomical potentials modeling galaxies or stellar systems can be rather involved, and deriving their first derivatives (accelerations) and second derivatives (variational equations) in order to compute orbits and their chaoticity may be a formidable task. We present here a fully automated routine, dubbed Smart, with which the accelerations and the variational equations of an arbitrary potential that has been written in the Fortran 77 language can be computed. Almost any Fortran 77 statement is admitted in the potential, and the output are standard Fortran 77 routines ready to use. We validate our algorithm with a set of potentials including time-dependent, velocity-dependent and very complex potentials that even involve auxiliary routines. We also describe with some detail a realistic seven-component Galactic potential, MilkyWayHydra, which yields very involved derivatives, thus being a good test bed for Smart.

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

---

*Corresponding author: Observatorio Astronómico - Paseo del Bosque s/n, 1900 La Plata, Buenos Aires, Argentina. E-mail address: ddc@fcaglp.unlp.edu.ar

1. Introduction

The chaotic behaviour of an orbit in a given dynamical system can be numerically established through algorithms that fall within two broad categories: those that analise the frequencies of the trajectory (e.g. Binney and Spergel, 1982; Laskar, 1990; Šidlichovský and Nesvorný, 1996; Carpintero and Aguilar, 1998; Papaphilippou and Laskar, 1998), and those based on the evolution of deviation vectors, called variational indicators (e.g. Benettin et al., 1976, 1980a,b; Voglis and Contopoulos, 1994; Contopoulos and Voglis, 1996; Froeschlé et al., 1997; Voglis et al., 1999; Cincotta and Simó, 2000; Sándor et al., 2000; Skokos, 2001; Lega and Froeschlé, 2001; Fouchard et al., 2002; Cincotta et al., 2003; Sándor et al., 2004; Skokos et al., 2007; Maffione et al., 2011; Darriba et al., 2012b,a; Maffione et al., 2013; Carpintero et al., 2014). For the sake of completeness, let us also mention the Poincaré surfaces of section (e.g. Hénon and Heiles, 1964), with which a qualitative study of chaos may be done.

To compute any of the variational indicators, the so-called variational equations should be integrated along with the equations of the motion. Whereas the latter includes the first derivatives of the potential with respect to the positions (accelerations), the former need its second derivatives. In cases of potentials that depend on velocities, the second derivatives with respect to them are also needed. As long as the potential is simple, these derivatives are usually tractable, in the sense that they can be readily computed by hand and coded in a program. But modern galactic potentials can be very involved, summing up several complex components, so the task of computing and programming the derivatives can be formidable. In this article, we present an automated algorithm to generate derivatives, the program SMART, which can be applied directly on a FORTRAN 77 (hereafter F77) coded potential, and that outputs F77 subroutines for its accelerations and its variational equations.

Despite the fact that the program presented here can be used on its own, our main motivation was to develop this code as part of a larger software package dedicated to the computation of the variational indicators. In that sense, SMART was conceived as the second element of the so-called LP-VISUITE (La Plata Variational Indicators Suite), an open-source software specifically designed to compute a plethora of chaos indicators based on the evolution of deviation vectors. The suite is composed of three basic elements, namely (a) the LP-VICODE (current version 2.0.1; Carpintero et al. 2014),

which is the kernel code and includes a library with more than ten of the most worldwide used variational indicators, (b) the automatic differentiation pre-processing slave program presented here, SMART (current version 1.2.1), and (c) a ready-to-use fully modifiable and realistic seven-component Galactic potential, MILKYWAYHYDRA (current version 2.0) that we also use here to validate SMART, and was previously used in other studies such as Maffione et al. (2018) and Gómez et al. (2021). Further details about the suite and its components can be found at:

http://lp-vicode.fcaglp.unlp.edu.ar/

The software package is freely available at the website.

2. Mathematical preliminaries

Let $\dot{\mathbf{w}} = \mathbf{F}(\mathbf{w})$ be the equations of the motion of a test particle in a dynamical system, where $\mathbf{w} = (\mathbf{x}, \mathbf{v})$ with $\mathbf{x}$ the position of the particle, $\mathbf{v}$ its velocity, and $\mathbf{F}$ the vector field defining the dynamical system. Let n be the dimension of the configuration space, so that $\mathbf{w}$ has $2n$ components.

If the motion is driven exclusively by a potential $\Phi(\mathbf{w}, t)$, the equations of the motion can be split as

$$\dot{\mathbf{x}} = \mathbf{v}, \tag{1}$$
$$\dot{\mathbf{v}} = -\nabla_{\mathbf{x}}\Phi(\mathbf{w}, t), \tag{2}$$

where the operator $\nabla_{\mathbf{x}}$ indicates that the derivatives should be computed only with respect to $\mathbf{x}$. In writing a program that computes the trajectories, Eqs. (1), being independent of the potential, can always be coded directly, whereas for the Eqs. (2) the first derivatives of the potential with respect to the positions are needed. The program SMART generates a source code that, given Φ, computes Eq. (2).

On the other hand, whether an orbit is of a regular or chaotic nature can be determined with the aid of the so-called variational indicators. These algorithms seek to determine whether two orbits infinitesimally apart will diverge exponentially or not. To this end, the equations of motion are solved along with the variational equations, that is, the first variations of the equations of the motion:

$$\frac{\mathrm{d}(\delta\mathbf{w})}{\mathrm{dt}} = \left.\frac{\partial\mathbf{F}}{\partial\mathbf{w}}\right|_{\mathbf{w}}\delta\mathbf{w}, \tag{3}$$

3

where $\delta\mathbf{w}$ is the phase-space deviation between the orbits, and $\partial\mathbf{F}/\partial\mathbf{w}$ is the $2n \times 2n$ matrix of derivatives of the components of $\mathbf{F}$ with respect to the components of $\mathbf{w}$. These equations can also be split as

$$\frac{\mathrm{d}(\delta\mathbf{x})}{\mathrm{dt}} = \delta\mathbf{v}, \tag{4}$$

$$\frac{\mathrm{d}(\delta\mathrm{v_j})}{\mathrm{dt}} = \sum_{i=1}^{n} -\frac{\partial^2\Phi}{\partial x_j \partial w_i}\delta w_i, \quad j = 1,\ldots,n. \tag{5}$$

Again, Eqs. (4) are independent of the potential, so they can be wired into any code, whereas Eqs. (5) are the ones that SMART generates as a routine given the potential Φ.

3. Description of the code

The program SMART is written in standard F77, except for the common non-standard extensions `DO−ENDDO`, `INCLUDE`, `DOWHILE`, lowercase characters, inline comments, and names longer than 6 characters. All real variables are `DOUBLE PRECISION`.

The program, after reading the name of the file where the potential is coded as a `FUNCTION` with three dummy arguments (time, position and velocity), starts scanning it in order to find all the derivable and no derivable variables.[1] Since the derivatives that are sought are with respect to the positions or the velocities, the derivable variables are those that are the result of some mathematical computation involving positions or velocities, or any of their descendants. If no variable is found that depended on the velocities, SMART sets a flag in order to avoid the future computation of derivatives with respect to velocities.

A second scan is then initiated, in which any statement not involving a derivable variable is copied as is to the output file. If, instead, there are one or more mathematical operations acting on a derivable variable, each operation is sent in turn to a routine that computes the corresponding code of its derivative. The resulting statement, for each operation, is dumped to the output file. When there are more than one mathematical operation on the

---

[1]In the standard F77 nomenclature, there is a distinction between (scalar) 'variables' and 'arrays'. In this article, we will use 'variable' to refer to both kinds of symbolic names, reserving 'array' for those cases in which a non-scalar variable is specifically meant.

same statement, the derivative is computed recursively, using the expression of the derivative of a composition of functions:

$$\frac{\mathrm{d}}{\mathrm{d}x}(f \circ g)(x) \equiv \frac{\mathrm{d}}{\mathrm{d}x}f(g(x)) = \frac{\mathrm{d}f}{\mathrm{d}g}\frac{\mathrm{d}g}{\mathrm{d}x}. \tag{6}$$

The file with the derivatives is preambled and ended with a set of statements that converts it in a ready-to-use F77 subroutine that computes the negative of the gradient of the potential (i.e., the accelerations).

After the foregoing operations are finished, a new scan computes the corresponding second derivatives, dumping the resulting statements to another file that also is converted to a subroutine that computes the time derivatives of the velocity components of the deviation vector. This pass is done only if the user chooses to do so. For economy of writing, hereinafter we will sometimes call 'pass 1' to refer to the computation of the accelerations, and 'pass 2' to refer to the computation of the derivatives of the deviation vector.

3.1. Input

The input parameters, read from the input files `smart.in` and `smart.par`, are only three: (a) the name of the file where the potential is coded (first record of `smart.in`), (b) a digit that indicates whether only the accelerations (digit = 1) or also the variational equations (digit = 2) are to be computed (second record of `smart.in`), and (c) the dimension n of the potential (first `PARAMETER` of `smart.par`).

3.1.1. Restrictions on the code of the potential

The user-provided F77 code to compute the potential should be written in `DOUBLE PRECISION`. Therefore, no `REAL*4` variables are expected to appear. For the same reason, no intrinsic function that converts from and to single precision is expected to be present in the code (`REAL`, `FLOAT`, `SNGL`, `DBLE`, `DPROD`). Complex variables and their corresponding intrinsics `CMPLX`, `AIMAG`, and `CONJG` are not expected in the code either. If the program SMART finds any of these functions, it stops with a warning message.

On the other hand, since the potential should be derivable in order to have a physical meaning, the non-derivable numerical intrinsics are also expected to be absent in the code (truncations: `INT`, `IFIX`, `IDINT`, `AINT`, `DINT`, `ANINT`, `DNINT`, `NINT`, `IDNINT`, transfer of sign: `SIGN`, `ISIGN`, `DSIGN`, positive difference: `DIM`, `IDIM`, `DDIM`, remaindering: `MOD`, `AMOD`, `DMOD`, and maxima/minima

(`MAX`, `MAXO`, `AMAX1`, `DMAX1`, `AMAXO`, `MAX1`, `MINO`, `AMIN1`, `DMIN1`, `AMINO`, `MIN1`).
If the program SMART finds any of these functions, it stops with a message.

The code of the potential routine should begin thus (aside from comment lines or lowercase/uppercase spelling):

```
FUNCTION pot(t,x,n)
INTEGER n
DOUBLE PRECISION pot,t,x(n)
```

That is, the routine should be a `FUNCTION` and its name should be `pot`. The three dummy arguments should be named and given in the order as shown. They are: the time coordinate `t`, that must be present even if the potential is time independent; an array `x` containing the Cartesian coordinates $\mathbf{x}, \dot{\mathbf{x}}$ (in that order) of the point of the phase space at which the potential is to be evaluated, and the dimension of the phase space `n`. Note that if the potential is independent of the velocities, the passing-by-address feature of F77 allows sending only the first half of `x`. The `DOUBLE PRECISION` declaration may be replaced by the non-standard `REAL*8` if the compiler accepts it. Any additional parameter should be passed to the routine by `COMMON`.

After the foregoing statements, the rest of the routine may be any valid F77 code, including slave subprograms and the non-standard extensions `DO-ENDDO`, `INCLUDE`, `DOWHILE`, lowercase characters, inline comments, and names of variables longer than 6 characters, except for the following seven limitations:

1. All the variables, with their dimensions if any, should be declared with a specification statement specifying their type, as if `IMPLICIT NONE` were in force. Therefore, neither the `DIMENSION` nor the `IMPLICIT` statements should be used. Any number of variables can be specified in a given specification statement.

2. The names of variables, including the declaration of their dimensions if any, should be less than 40 characters long.

3. The names of variables must not begin with the letter `o` (lowercase or uppercase). This initial letter is reserved for variables generated by SMART.

4. Arithmetic and logical `IF`s are not allowed. They should be replaced with block `IF`s, i.e., the keyword `THEN` should always be present.

5. Statement function statements are not allowed. They should be coded as normal statements in the body of the routine.

6. The dummy array `x` should not appear with a variable index (e.g. `x(k)`), but always with an explicit number (e.g. `x(1)`). This is by far the most restrictive limitation, but it is always achievable with a little bit of additional code.

7. No auxiliary (slave) subprogram should use derivable variables.

The SMART program detects whether any of these restrictions are violated, and in that event it stops with a warning message.

3.2. Output

The output consists of a file containing an F77 subprogram for the computation of the accelerations corresponding to the input potential, and an optional second file containing an F77 subprogram for the computation of the respective variational equations.

The initial statements of the first subprogram are:

```
****************************************************************
      SUBROUTINE acelera(t,x,n,acc)
      DOUBLE PRECISION acc(n/2)
      INTEGER n
      DOUBLE PRECISION pot,t,x(n)
****************************************************************
```

The meaning of the `t`, `x`, and `n` input dummy arguments is the same as in the potential subprogram. The `acc` output dummy argument returns to the calling program the components of the acceleration vector.

The first statements of the second subprogram are:

```
****************************************************************
      SUBROUTINE variac(t,x,dx,n,dax)
      DOUBLE PRECISION dx(n),dax(n/2),var(n/2,n)
      DOUBLE PRECISION acc(6/2)
      INTEGER n
      DOUBLE PRECISION pot,t,x(n)
****************************************************************
```

Here `t`, `x`, and `n` are again input dummy arguments with the same meaning as in the potential subprogram. The additional input array `dx` contains the components of the deviation vector. The output array `dax` returns the

7

derivatives with respect to the time of the deviations of the velocity. Note that the array `acc` is not a dummy argument, and therefore the standard F77 forbids the use of `n` in defining its dimension. That is why the number `6` appears in this example, which was taken from the output corresponding to a potential with a 6D phase space. Other (even) dimensions can appear according to the input potential.

Both output routines are complete F77 subprograms, ready for use.

3.3. Details of the algorithm of derivation

The lexical analysis of each F77 statement is always done taking into account that blank spaces can be freely interspersed, even inside a keyword, and that lowercase and uppercase letters are equivalent. Hereinafter, a 'real variable' is understood to mean a 'double precision real variable'.

3.3.1. Derivable variables

The first task of SMART is to load a list of variables used in the computation of the potential, with their respective dimensions if any. Since all variables should be declared, it suffices to scan the code only until the first executable statement is reached. This also allows to classify from the outset whether a variable is real, integer, logical or character; only the first ones are of interest, for only them can be derivable variables. After this first step, the rest of the code is scanned to determine whether each real variable is derivable or not. In this respect, let us emphasize that to compute the derivative of an F77 `FUNCTION`, the only statements that have to be processed are the assignments to a real variable. Any other statement, be it a declaration, a structural statement (e.g. `DO`, `IF`, etc.), a logical or integer assignment, etc., can be ignored for derivation purposes. Therefore, each statement is read in turn, with its continuation lines if any, to determine whether it corresponds to a real assignment. To this end, it is first scanned to detect the presence of an equal sign. If it is present, the statement is further analized to detect if it is any of the non-assignment F77 statements that may contain that sign: `OPEN`, `CLOSE`, `INQUIRE`, `READ`, `WRITE`, `PRINT`, `BACKSPACE`, `ENDFILE`, `REWIND`, `PARAMETER`, `DATA`, and `DO`. If the statement is not any of the above, it is additionally scrutinized to verify whether its left hand side is a real variable. Any statement that satisfies all these conditions is passed to the next algorithm; otherwise, it is discarded and the next one is read.

Once a real assignment statement is detected, its real variables are classified according to the following criteria:

8

Table 1: Atomic elements that can appear on a real assignment in F77.

Type	element
algebraic	`+ - * / ** =`
intrinsic	`ABS SQRT EXP LOG LOG10 SIN COS TAN ASIN`
	`ACOS ATAN SINH COSH TANH ATAN2`
constant	integer or real numbers
name	names of integer or real constants (`PARAMETER`s)
	names of integer or real variables
other	`, ( ) !` newline label

- `pot` and `x` are stored as derivable from the outset; `t` is stored as non-derivable.

- If the first appearance of a new variable is in the right hand side (rhs) of an assignment, it only can be a non-derivable variable, for its value should have already be set (that is, it can only be a variable that has come through a `COMMON`).

- If the first appearance of a new variable is in the left hand side (lhs) of an assignment, it is derivable if there are derivable variables in the rhs; otherwise, it is non-derivable.

- Subsequent appearances of any variable in a rhs or a derivable variable in a lhs do not change their condition.

- Subsequent appearances of a non-derivable variable in a lhs make it change to derivable if there are derivable variables in the corresponding rhs.

3.3.2. Lexical analysis of the statements

Once the derivable variables were identified, the potential routine is scanned again, from the first executable statement, in order to dissect each real assignment into its atomic constituents (Table 1).

Each atom is loaded on a stack (array) right away, except for: (a) a minus sign, which is previously classified as binary or unary, and transformed to a -1 factor in the latter case if it is not preceding a number, (b) a specific name of an intrinsic function (i.e., an intrinsic with an `A`, `D` or `C` prefixed to

its generic name), from which the first letter is removed to keep only the generic name,[2] and (c) a statement label, which is immediately output as a `CONTINUE` statement with the same label and is no further processed. A newline or a `!` character (i.e. the start of an inline comment) indicates the end of the statement.

The loading of atoms is temporarily stopped if one of these conditions is met: (a) the loaded atom is an entire operation (`+` or binary `-`) and the previous operation loaded in the stack is an algebraic operation (`+`, binary `-`, `*`, `/` or `**`)[3]; (b) the loaded atom is a rational operation (`*` or `/`) and the previous operation loaded in the stack is a power or a rational operation; (c) the loaded atom is a right bracket and the previous operation loaded in the stack is an algebraic operation or an intrinsic. In all these cases a mathematical operation is being closed, and the derivation process is applied to it before a new atom is loaded. Once the derivation process is ended, the operation and its operands are removed from the stack and replaced by their result, the loading of atoms is resumed, and the foregoing algorithm is repeated until there are no more atoms to read in the statement. At this point, any pending mathematical operation is resolved (i.e., derived), and the next statement is read. The whole process is repeated until the `END` statement is reached, and then SMART simply copies the remaining lines (i.e. slave routines), if any, into the output. Notice that if the routine of the potential and that of the accelerations and/or variational equations are to be present in the same program, these duplicates of the slave routines should be deleted.

3.3.3. Derivation procedure

Upon reception of a mathematical operation and its operands by the derivation routine, it first reconstructs the operation in the form of an F77 statement, since its result is usually needed in the derivative. If the operation is the last one of a statement of the potential routine, the lhs of the statement of this partial result is the original variable into which the result is stored. Otherwise, the lhs is defined as the array element $o(x)$ (pass 1) or $oq(x)$ (pass 2), where x are consecutive numbers; this array element is of course

---

[2]A widespread belief asserts that F77 intrinsic functions operating on double precision objects should be used with their specific names beginning with `D`. It is not true.

[3]By a slight abuse of language, we call the power an algebraic operation even when the exponent is not an integer.

Table 2: Results and arrays for the derivatives. The `//` operation symbolizes the concatenation of strings.

the derivative of...	generates...
x	1 or 0
pot	acc
acc	var
name of derivable variable (pass 1)	o$//name$
name of derivable variable (pass 2)	oq$//name$
any other	0

considered a derivable variable from there on. In either case, the statement is output and the atomic operation and its operands are then subjected to the derivative process.

For both, the two operands and the left member, a name of an array which will contain their derivatives with respect to positions (and possibly to velocitites) is generated. Table 2 summarizes the different cases. Note that the entry 'any other' in the Table cannot happen in the case of the left member. Note also that the derivative of an o(x) or an oq(x) variable will be prefixed with an additional o or an additional oq, respectively. When performing this step, the argument of an intrinsic function is considered as the left operand, and a right operand with zero derivative is assumed, except in the case of `ATAN2`, which takes two operands. Then, the mathematical derivation is performed on the operation itself. This is explicitly coded for each kind of operation and intrinsic by using the rules of derivation in each case.

A simple example will help to understand the procedure. For the sake of brevity, let us suppose that the potential depends only on the Cartesian coordinates (x, y). If the following statements are to be processed:

```
a = b*x(1) + x(2)
c = SIN(a)
```

where b is a non-derivable variable, then the first atomic operation will produce the following code:

```
o(1) = b*x(1)
```

and its derivatives, using the rules of derivation and the results of Table 2, would be coded

```
oo(1,1) = b*1 + 0*x(1)
oo(1,2) = b*0 + 0*x(1)
```

Then, the second atomic operation of the first statement will produce

```
a = o(1) + x(2)
```

with derivatives

```
oa(1) = oo(1,1) + 0
oa(2) = oo(1,2) + 1
```

Since the second statement is an atomic operation in itself, it will not generate any new $o(x)$ variable, and it will be copied as is. Its derivatives will be:

```
oc(1) = COS(a)*oa(1)
oc(2) = COS(a)*oa(2)
```

As already said, once each elemental derivative is completed the stack is updated by replacing the entire operation by its result. The derivation process continues until the END statement of the potential file is reached.

To reduce useless code, each statement is simplified before its output in the following cases: (a) any term containing a zero factor is deleted, (b) any unit factor in a product is deleted, (c) any unit exponent is eliminated, leaving only the base.

3.3.4. Final statements

In pass 1, the routine generated by the foregoing algorithm is ended with the lines

```
DO oi=1,n/2
   acc(oi)=-acc(oi)
ENDDO
```

in order to return to the calling program the negative of the gradient of the potential, i.e., the accelerations. For pass 2, the routine is ended with the lines

```
DO oi=1,n/2
    dax(oi)=0d0
    DO oj=1,n
        dax(oi)=dax(oi)+var(oi,oj)*dx(oj)
    ENDDO
ENDDO
```

which compute the temporal derivative of the velocity components of the deviation vector from the second derivatives of the potential. If the potential does not depend on the velocities, the inner loop goes only up to n/2.

4. Validation of the code

4.1. Validation of the code using known potentials

The acceleration part of the code was validated with an assorted set of potentials (Table 3), ranging from the most elemental (harmonic, Keplerian) to the most involved (Hernquist & Ostriker, perfect ellipsoid, see references on Table 3). The latter require extra routines and even numerical integrations of functions to compute the potential. A time dependent potential (logarithmic with a variable flattening) and a velocity dependent potential (global bifurcation) were also included. For each potential, various initial conditions were chosen with which orbits were computed with the LP-VICode, both with the accelerations computed and coded by hand, and with the accelerations obtained with SMART. In all regular cases the orbits thus obtained were the same, aside from the expected numerical differences when the same algorithm is coded in two different ways. Chaotic orbits, on the other hand, remained the same only until the exponential divergence makes the orbits differ. This is, again, the expected behaviour when the same trajectory in the chaotic region of a potential is integrated with different codes. We also probed sticky orbits, i.e. orbits which behave as regular for a certain time, but then they show themselves as chaotic. As expected, the differences between the manual coding and SMART were similar to the regular case when the orbits behaved regularly, and to the chaotic case thereafter. Fig. 1 shows the distance in phase space between orbits of the Binney potential computed with accelerations coded by hand and by SMART, for a regular, a chaotic, and a sticky case.

No rotating potential was considered in the validation. This is due to the fact that, in non-inertial frames, there are non-inertial ('fictitious') forces

13

Table 3: Potentials with which the acceleration was validated. Symbols in the expressions, aside from the coordinates x, y, and z, are parameters of the corresponding potential.

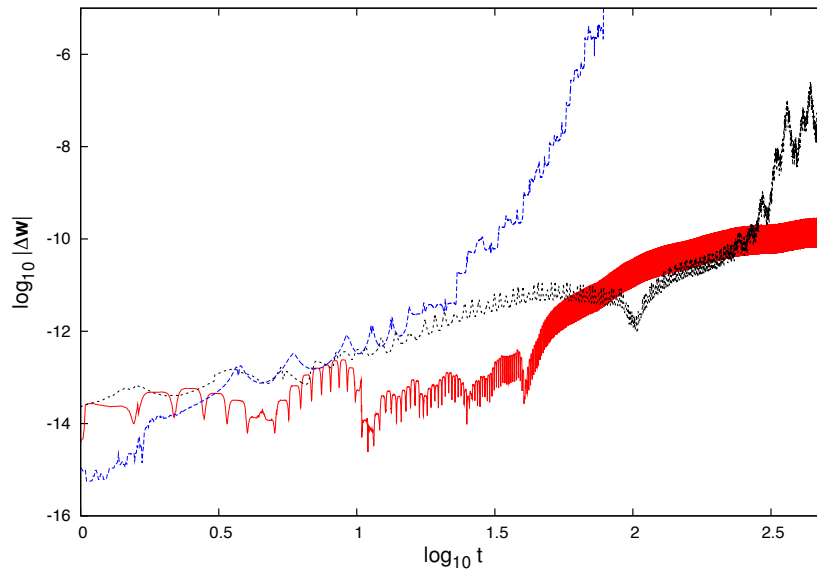| Name	Dim	Reference or expression
Binney	2D	Binney (1982)
Global bifurcation	1D	Jackson (1989), p. 300
Harmonic	1D	$\frac{1}{2}\omega^2 x^2$
	3D	$\frac{1}{2}(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2)$
Quartic	1D	$\frac{1}{2}x^2 + \frac{1}{3}x^3 + \frac{1}{4}\epsilon x^4$
Hénon-Heiles	2D	Hénon and Heiles (1964)
Hernquist	3D	Hernquist (1990)
Hernquist-Ostriker	3D	Hernquist and Ostriker (1992)
Keplerian	3D	$-GM/\sqrt{x^2 + y^2 + z^2}$
Logarithmic	2D	$\frac{1}{2}v_0^2 \ln(x^2 + y^2/q^2 + R_c^2)$
Logarithmic, time dependent	2D	same as above, with
		$q(t) = A + B\exp(-(t - C)^2/2)$
Logarithmic quadrupolar	3D	Carpintero and Wachlin (2006)
Merritt-Fridman	3D	Merritt and Fridman (1996)
Miyamoto-Nagai (MN)	3D	Miyamoto and Nagai (1975)
MN + Hernquist + NFW	3D	Gómez et al. (2010)
MN + Plummer	3D	see individual entries
MN + Plummer, merid. plane	2D	Zotos and Carpintero (2013)
NFW, triaxial	3D	Vogelsberger et al. (2008)
Perfect ellipsoid	3D	de Zeeuw (1985)
Plummer	3D	Plummer (1911)
Rational quadrupolar	3D	Muzzio et al. (2005)
Satoh, prolate	3D	Carpintero et al. (1999)
Schwarzschild	3D	Schwarzschild (1993)

Figure 1: Distance $|\Delta \mathbf{w}|$ in the phase space, as a function of time t, between orbits computed with manually coded accelerations and with accelerations obtained with SMART. The orbits belong to the Binney potential with $q = 0.9, R_c = 0.14, v_0^2 = 1, R_e = 3$, with initial conditions $(x, y, \dot{x}, \dot{y}) = (0.1, 0, 0.5, 0.02)$ (regular orbit, red), $(x, y, \dot{x}, \dot{y}) = (0.1, 0.5, 0, 1)$ (chaotic orbit, blue), and $(x, y, \dot{x}, \dot{y}) = (0.077, 0, 0, 1.73203)$ (sticky orbit, black).
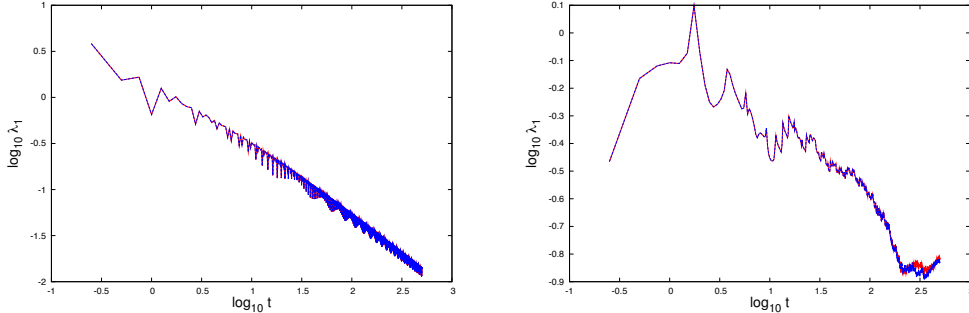
Figure 2: Left: The maximal Lyapunov exponent λ_1 (a chaos indicator) of the regular orbit of Fig. 1 (left). The indicator λ_1 should tend to 0 with time for a regular orbit; in a log-log plot it should be a straight line with negative slope. The blue and red curves correspond to manually coded variational equations and to these equations obtained with SMART, respectively. Right: The same but for the chaotic orbit of Fig. 1 (right). The curves split when the chaoticity begins to show. The chaotic nature of the orbit is nevertheless found in both cases.

that are not derived from the gradient of a potential. Whereas any centrifugal force can be masked as the gradient of a centrifugal potential, Coriolis forces cannot, and thus are not suitable for SMART. However, in these cases the user could add the corresponding accelerations by hand.

The variational equations, on the other hand, were validated with a more restricted set of potentials, due to the difficulty in writing down these equations for most potentials (which is, incidentaly, the main reason for developing SMART). The chosen potentials were: global bifurcation, Hénon-Heiles, logarithmic (with a null core radius), and NFW triaxial, all of which admit chaotic regions. We repeated the procedure described for the accelerations, plus the variational equations coded by hand and coded by SMART. With them, we computed several chaos indicators with LP-VICODE for each orbit. The results were similar to those obtained in the case of accelerations: no differences were found between codes for the regular orbits; the chaotic ones showed slight differences, especially from the moment the indicators showed their chaotic condition, and sticky orbits showed both regimes. In any case, the regular or chaotic quality was always preserved. Fig. 2 shows examples of a regular and a chaotic case.

In addition to the above, we run a number of experiments to test the

16

computing times of the manually coded accelerations and variational equations, against those generated with Smart, for several of the potentials used to validate the code. We used an Intel(R) Core(TM) i7-4790K CPU (4 cores, 8 threads) @ 4.00 GHz with 8 GB RAM, Windows 7 Ultimate Service Pack 1, 64 bits, running a virtual machine Oracle VM VirtualBox using Ubuntu 18.04 64 bits with 4 GB RAM. The codes were compiled with `gfortran`, with the optimization flag `O3` on.

Table 4 shows the outcome. The reported times correspond to the integration of 50 000 orbits on each potential until an arbitrary final time of 6.135 time units. For the accelerations, the manual coding is, on average, 15 per cent faster than that produced by the Smart coding, though individual entries range from 2 per cent to 46 per cent. However, there is no clear correlation between the complexity of the potential and the additional time took by Smart.

On the other hand, with the variational equations the manual coding is between 35 per cent and 60 per cent faster than Smart, except for the Hénon-Heiles case in which both codes took almost the same time.

In view of these figures, when orbits need to be computed on big samples of initial conditions, the handcrafted versions of the accelerations and variational equations are worth using over the automatic differentiation version provided by Smart. In all other cases, Smart is the right choice due to the almost nil effort from the user to set things up.

4.2. Validation of the code using MilkyWayHydra

MilkyWayHydra (MWH for short) is the third element of the LP-VIsuite: a ready-to-use fully modifiable and realistic multi-component 3D galactic potential. The MWH has been conceived mainly to represent late-type galaxies and, in particular, Milky Way-type galaxies. Nevertheless, it can be used to represent early-type galaxies as well, due to its straightforward programming style. The potential is written in standard F77 following the rules specified by the suite's latest version of the kernel code —the LP-VIcode—, and each of the galactic components are clearly identified by using separated blocks. The result is a set of subprograms in a file called `milkywayhydra.pav`, which includes the computation of the potential, the accelerations and the first variational equations (manually coded and checked with symbolic manipulators) ready to use with (exclusively) the LP-VIcode. Current version MWH 2.0 describes a galactic potential with the following

Table 4: Comparison of integration times between manual and Smart codings. EQM stands for equations of motion (accelerations); VEQ stands for variational equations (plus EQM). Times are given in minutes:seconds.

Potential	manual EQM	Smart EQM	%	manual VEQ	Smart VEQ	%
Binney	2:35	3:19	22.1			
Harmonic 3D	2:40	2:55	8.8			
Hénon-Heiles	2:03	2:07	2.9	14:05	14:16	1.2
Hernquist	2:45	3:07	12.1			
Keplerian 3D	3:18	4:00	17.3			
Logarithmic	4:20	6:18	31.1	52:30	81:17	35.4
Logarithmic, time dependent	2:18	2:45	16.3			
Logarithmic quad.	7:07	7:16	2.0			
Merritt-Fridman	5:44	9:51	45.8			
MN + Hernquist + NFW	6:20	6:28	2.1			
NFW triaxial	9:11	9:59	12.5	394:31	630:00	38.4
Perfect ellipsoid	13:38	17:55	23.9	243:03	582:22	58.3
Schwarzschild	2:46	2:50	2.6			

components: a nuclear region, a peanut-shaped bulge, spiral arms, a disk, and a dark matter halo.

In this last validation experiment we test SMART with the most demanding (by far) potential we have at hand. Therefore, we are using the default MWH 2.0 configuration, the so-called HYDRA 2.0 which is a realistic Milky Way-type galactic potential meant to provide an accurate representation of the Milky Way potential (for further details on the potential see Appendix A). In particular for the present set of tests, we are using two flavours of the HYDRA 2.0, i.e. the time independent and the full time dependent versions.

For both versions of the HYDRA 2.0 potential, 500 initial conditions sampling the phase-space distribution of the dark matter halo were randomly chosen with which orbits were computed with the LP-VICODE, both with the accelerations computed and coded already in HYDRA 2.0, and with the accelerations obtained with SMART. In the case of the time independent version, for all regular examples the orbits thus obtained were the same, aside from the expected numerical differences mentioned in the previous experiments. On the other hand, as previously discussed, chaotic orbits only remain the same until the exponential divergence phase begins to show. The left panel of Fig. 3 shows the distance in phase space between orbits computed with accelerations coded in HYDRA 2.0 and those coded by SMART, for a regular and a chaotic case (initial conditions are specified on Table 5). We repeated the above procedure for the variational equations coded in HYDRA 2.0 and coded by SMART. With them, we computed all the eleven available chaos indicators with LP-VICODE for each orbit. The results were similar to those obtained in the case of accelerations: no significant differences were found between codes for both the regular and chaotic orbits. Furthermore, the regular or chaotic quality was always preserved. For instance, the right panel of Fig. 3 shows the Orthogonal Fast Lyapunov Indicator, OFLI (a fast chaos indicator, Fouchard et al. (2002)) applied to the regular and the chaotic case presented in the panel on the left.

For the full time dependent version of the HYDRA 2.0, the left panel of Fig. 4 shows the distance in phase space between orbits computed with accelerations coded in HYDRA 2.0 and by SMART, for the same regular and chaotic orbits presented on Table 5. Once again, for all regular examples the orbits thus obtained were the same, aside from the expected numerical differences, and chaotic orbits remained the same only until the exponential divergence makes the orbits differ. On the right panel of Fig. 4 we show the OFLI for both regular and chaotic cases, where the accelerations plus the

Table 5: Initial conditions used with HYDRA 2.0 for the orbit showed in Fig. 3. Positions are in kpc and velocities in km s^{-1}.

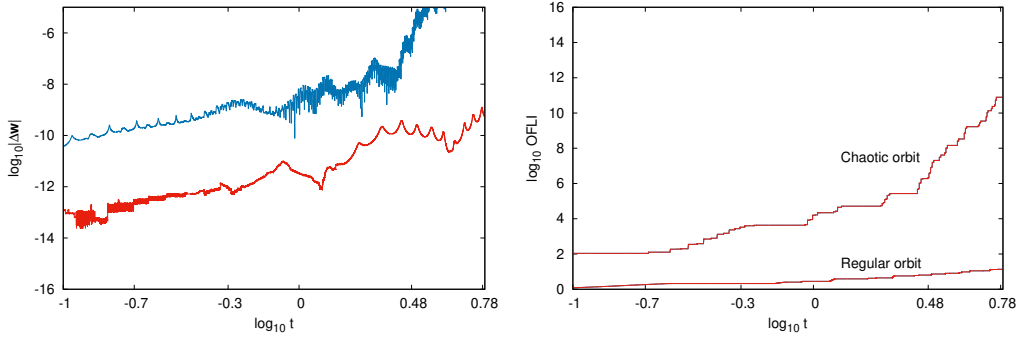|	Regular	Chaotic
x	0.31807616	0.73157591
y	6.7984977	-0.84585929
z	-20.295328	-0.21206708
$\dot{x}$	-92.1442	8.74071
$\dot{y}$	244.737	34.4792
$\dot{z}$	-86.8133	119.667

Figure 3: Left: Distance $|\Delta\mathbf{w}|$ in the phase space (in logarithmic scale), as a function of units of time: u.t. (1 u.t. $\sim$ 1 Gyr), between orbits computed with accelerations obtained from HYDRA 2.0 and with SMART. The orbits belong to the time independent flavour of the HYDRA 2.0 potential, with initial conditions given on Table 5 (regular orbit, red and chaotic orbit, blue). Right: The OFLI (a fast chaos indicator) of the regular and the chaotic orbits presented on the left panel. The indicator OFLI should grow linearly or exponentially with time for the regular or the chaotic orbit, respectively. The blue and red curves (almost superposed) correspond to variational equations obtained with HYDRA 2.0 and SMART, respectively. The agreement between both curves and for each orbit is evident.
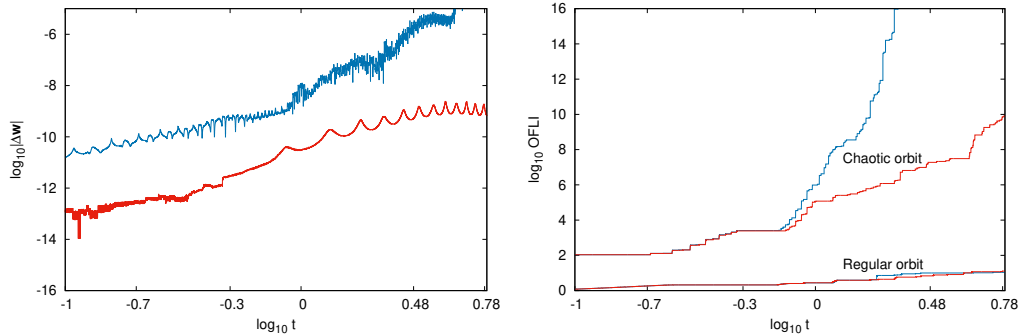
20

Figure 4: Idem Fig. 3 but for the full time dependent flavour of the Hydra 2.0 potential.

variational equations are those coded in Hydra 2.0 (blue) and by Smart (red). Both computations for the regular orbit follow very similar trajectories, which is not the case for the chaotic orbit. The separation for both chaotic orbits starts to become noticeable when the time dependent components of the Milky Way-like potential, i.e. its bar and spiral arms, start to gain strength relative to the background time independent potential. Final amplitudes of the bar and spiral arms are reached by 3.068 units of time, u.t. The results shown for the chaotic orbits are thus expected. In any case, the regular or chaotic quality was preserved also in this experiment.

With this set of experiments we have not only checked that Smart is working properly under very demanding circumstances, but also we have checked for a second time and with an independent program (Smart) that MWH 2.0 has its accelerations and variational equations correctly coded.

We also run for this potential a number of experiments to test the computing times both with MWH 2.0 (i.e., the accelerations and variational equations already coded by hand) and Smart. In all the experiments MWH 2.0 performed faster than Smart, an expected result due to the different goals of both codes.

On Table 6 we present the results. The computing times obtained are an average among the four threads running simultaneously for the experiments using big samples of initial conditions. In case of the experiments with only 500 initial conditions, the computing times are the actual times registered for one thread.

It is evident from the experiments that the subprogram MWH 2.0 speeds up the computation with respect to Smart, due to its more efficient coding.

21

Table 6: Computing times with MWH 2.0. The first column indicates whether the version of MWH 2.0 is time independent (TI) or the full time dependent version (TD). The second column shows whether the computation of the orbits were done with only the equations of motion (EQM) flag activated; or the EQM and the variational equations (VEQ) with only the MEGNO & SElLCE and FLI & OFLI flags activated; or the EQM, the VEQ for all chaos indicators and all other quantities programmed with the LP-VICODE (FULL). The third column presents the number of initial conditions integrated in the experiments. The last column shows the time reduction, in percentage, obtained by MWH 2.0 compared with SMART.

Flavour	Experiment type	Number of initial conditions	%
TI	EQM	1163956	7.04
TD	EQM	1163955	12
TD	EQM+VEQ	180336	19
TI	FULL	500	31.55
TD	FULL	500	37.11

However, such a difference in speed is large only when the computation of variational equations is involved: between 20 per cent and 40 per cent time reduction, depending on the number of chaos indicators computed.

Therefore, the performance of SMART with our Galactic potential is comparable to that obtained with the rest of the potentials. The conclusion is also similar: when chaos indicators on galactic potentials need to be computed on a huge number of orbits, the original version of MWH 2.0 is worth using over the automatic differentiation version provided by SMART. Otherwise, SMART is preferable due to its simplicity.

4.3. Comparison with other software

Automatic differentiation can be done in several ways. Perhaps the easiest way to get the expression of a derivative is through modern symbolic manipulators (e.g. MATHEMATICA, MAPLE, MACSYMA, etc.), though if the function to be derived is not simple enough, the resulting expression may be very involved and may require intensive handling to turn it into code form. Numerically, it is always possible to derive a function by using divided differences (e.g. Press et al., 1992), but this method is inexact and a precise handling of errors becomes necessary. Among the programs that perform the automatic differentiation of a preexisting code, the package TAYLOR (Jorba

and Zou, 2005) obtains the solution of ordinary differential equations (ODEs) by the Taylor method. By choosing a high enough order in the Taylor series an arbitrary precision can be obtained, but at the cost of increasing the computing time. Also, the ODEs should be written in a specific language to be understood by the program, and it requires a somewhat complex installation procedure. Another software, PCOMP (Dobmann et al., 1995) is similar to ours in facing the differentiation of a code, but, as in the case of TAYLOR, the routine to be derived should be rewritten in a special language. A very general automatic differentiation software, not in the public domain, is AD-IFOR 2.0 (Bischof et al., 1995), which takes an arbitrary function written in Fortran 77 code and produces its derivative, just as in our case. It overcomes our program in that it can handle simple precision and complex variables, statement functions, derivable variables in `COMMON` and `EQUIVALENCE` declarations, and exceptions with non-derivable intrinsic functions. However, in order to include these features a preprocessing step is needed. As already said, most of these features are not necessary in our case, due to the fact that our goal is to derive a double precision continuous potential. Also, the complete package consists of three programs, one of them in `C`, while our code is a simple program in Fortran. ~~We could not probe the installation procedure of ADIFOR 2.0 because it is not in the public domain, and our asking for a license had no answer; perhaps this is its main disadvantage. On the other hand, and unlike~~ We would have liked to be able to compare the performance of ADIFOR 2.0 with that of our program. Unfortunately, our asking for a license had no answer, so we assume ADIFOR 2.0 is not supported anymore. In contrast to the abovementioned programs, our code does not require any license, does not need installation nor preprocessing, and no new language has to be learned to use it. It is a single F77 program, and all it requires is to compile it and run it, yielding F77 routines ready to use for computing accelerations and variational equations.

5. Conclusions

We have developed a program that, given an arbitrary potential written in F77, automatically generates ready-to-use F77 routines to compute its corresponding accelerations (first derivatives) and variational equations (second derivatives). The code of the potential, aside from its heading, may be any valid F77 code except for seven slight limitations, all of which can be bypassed by rewriting the code appropriately. The code was validated with assorted

23

potentials, including time-dependent potentials, velocity-dependent potentials and very complex potentials that even need external routines to be computed. Last but not least, the program has been successfully tested against a realistic, seven-component time-dependent Galactic potential, MilkyWay-Hydra, which produces very involved derivatives and thus is a litmus test for our code.

Appendix A. The MilkyWayHydra potential

Current version MWH 2.0 describes a multi-component 3D Galactic potential: (i) a central region composed of a nuclear star cluster (NSC), described by a Plummer potential and a supermassive black hole (SMBH) placed in its centre; (ii) a peanut-shaped bulge composed of the bulge spheroid (described by a Hernquist profile) and a bar (described by a three-dimensional quadrupole); (iii) spiral arms, described by a N–armed spiral pattern outside of the bar region; (iv) two discs, thin and thick, which can be chosen from two options: a Miyamoto-Nagai (MN) profile or an exponential profile; and (v) a dark matter halo (DMH) that has also two options for choosing it: a bi-triaxial extension of the Navarro–Frenk–White (NFW) model or a modified logarithmic model.

Early versions of the MWH has been successfully used in the following works: Maffione et al. (2013, 2015, 2018); Simpson et al. (2019); Gómez et al. (2021). A fully comprehensible document about the MWH 2.0 can be found at the website of the LP-VIsuite (see Section 1).

In the following subsections we will describe each component of the MWH 2.0 potential. The default values of the parameters found on Table A.7 define the so-called Hydra 2.0 potential used for the validation experiment applied to Smart, Section 4.2. Fig. A.5 shows the circular velocities generated by the nucleus, the bulge, the discs, and the halo, along with the full curve of the complete model, both with and without the bar and the spiral arms. Table A.7 also includes the references from which the values of the parameters were taken; those parameters without a reference were chosen to reproduce the circular velocity profile of model Aq-C4/C5 from Marinacci et al. (2014).
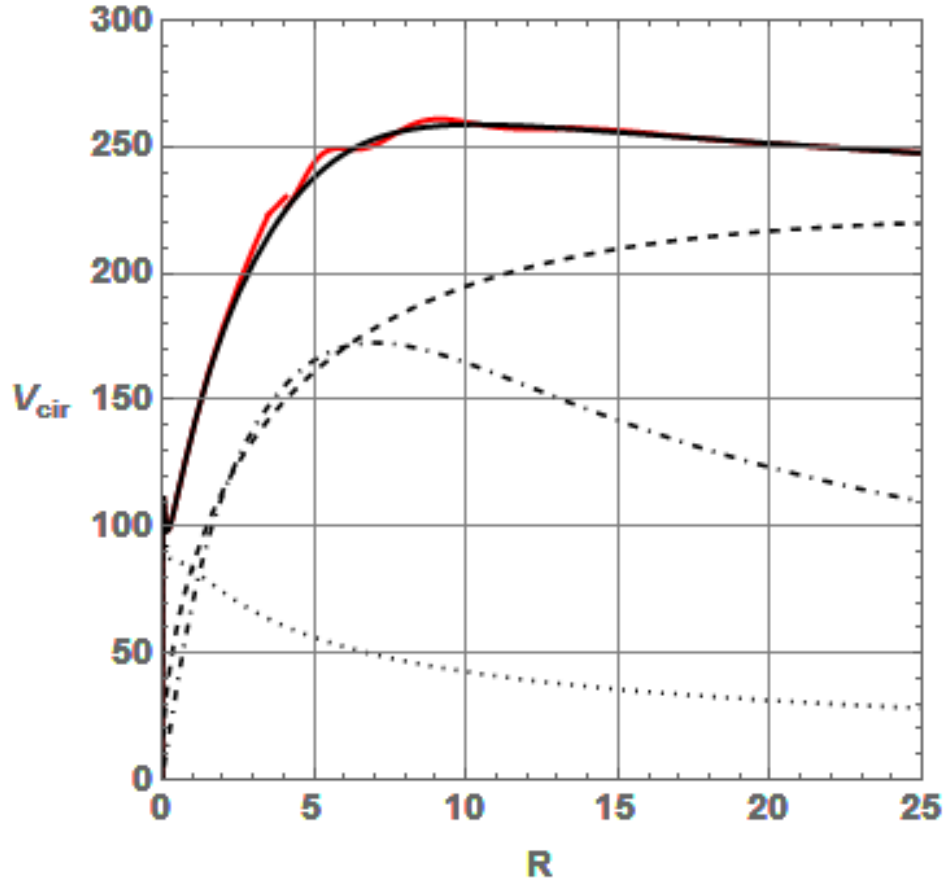
Figure A.5: Circular velocities of the HYDRA 2.0 model. Dotted line: nucleus plus spheroidal bulge; dot-dashed line: thin and thick discs; dashed line: dark matter halo; solid black line: full time-independent model (i.e., without bar or arms); solid red line: full model, including bar and arms. Distances expressed in kpc, velocities in km s^{-1}.

Table A.7: Default values of the parameters defining the HYDRA 2.0 potential. The parameters of the disc correspond to Option 2 (exponential disc), whereas those of the halo correspond to Option 1 (bi-triaxial NFW profile).

Component	Parameter	Default value	Reference
Nuclear region	M_{nuclear}	$2 \times 10^8 \, \mathrm{M_\odot}$	Maffione et al. (2018)
	$\epsilon_{\mathrm{nuclear}}$	$0.03 \, \mathrm{kpc}$	Launhardt et al. (2002)
Bulge spheroid	M_{bulge}	$4.74 \times 10^9 \, \mathrm{M_\odot}$	Maffione et al. (2018)
	$\epsilon_{\mathrm{bulge}}$	$0.835 \, \mathrm{kpc}$	Maffione et al. (2018)
Bar	α	0.01	Monari et al. (2016)
	v_0	$220.95 \, \mathrm{km \, s^{-1}}$	
	R_{b}	$3.5 \, \mathrm{kpc}$	Monari et al. (2016)
	R_0	$8 \, \mathrm{kpc}$	Monari et al. (2016)
	Ω_{b}	$-52.2 \, \mathrm{km \, s^{-1} kpc^{-1}}$	Monari et al. (2016)
	ϕ_{b}	5.65	Monari et al. (2016); Dehnen (2000)
Spiral arms	A	$2279 \, \mathrm{km^2 s^{-2}}$	
	R_{s}	$1 \, \mathrm{kpc}$	Monari et al. (2016)
	R_{sd}	$3.124 \, \mathrm{kpc}$	
	N	2	Monari et al. (2016)
	p	0.17	Monari et al. (2016); Banik and Bovy (2019)
	Ω_{s}	$-18.9 \, \mathrm{km \, s^{-1} kpc^{-1}}$	Monari et al. (2016); Antoja et al. (2014)
	ϕ_{s}	3.31	Monari et al. (2016)
	Corrotation radius	$4.08 \, \mathrm{kpc}$	
Thin disc	$M_{\mathrm{disc}}^{\mathrm{thin}}$	$5.276 \times 10^{10} \, \mathrm{M_\odot}$	
	R_d^{thin}	$3.124 \, \mathrm{kpc}$	Maffione et al. (2018)
	h_z^{thin}	$0.3 \, \mathrm{kpc}$	Maffione et al. (2018)
Thick disc	$M_{\mathrm{disc}}^{\mathrm{thick}}$	$0.686 \times 10^{10} \, \mathrm{M_\odot}$	
	R_d^{thick}	$3.124 \, \mathrm{kpc}$	Maffione et al. (2018)
	h_z^{thick}	$1 \, \mathrm{kpc}$	Monari et al. (2016)
DMH	M_{200}	$145.64 \times 10^{10} \, \mathrm{M_\odot}$	Maffione et al. (2018)
	c_{nfw}	16.0287081	Maffione et al. (2018)
	r_{s}	$14.63 \, \mathrm{kpc}$	Maffione et al. (2018)
	(a, b, c)	$(1.023, 1.010, 0.964)$	Maffione et al. (2018)
	(a', b', c')	$(1.069, 0.984, 0.941)$	Maffione et al. (2018)

Appendix A.1. The nuclear region

The nuclear region is described by a Plummer sphere:

$$\Phi_{\text{nuclear}} = -\frac{G\,M_{\text{nuclear}}}{\sqrt{x^2 + y^2 + z^2 + \epsilon^2_{\text{nuclear}}}}, \qquad (A.1)$$

where M_{nuclear} is the total mass of the nuclear region, which includes the NSC and the mass of the SMBH, and $\epsilon_{\text{nuclear}}$ is the scale length of the system, which defines the outer limit of the inner core.

Appendix A.2. The peanut-shaped bulge

The abovementioned nuclear region dominates the innermost parts of the Galaxy. The stellar bulge, on the other hand, dominates the region between 0.4 and 3 kpc, approximately Launhardt et al. (2002). The bulge spheroid is described by a Hernquist profile:

$$\Phi_{\text{bulge}} = -\frac{G\,M_{\text{bulge}}}{\sqrt{x^2 + y^2 + z^2} + \epsilon_{\text{bulge}}}, \qquad (A.2)$$

where M_{bulge} is the mass, and ϵ_{bulge} is the scale length.

The bar potential is a 3D version taken from Monari et al. (2016) of the pure quadrupole model used by, e.g. Dehnen (2000):

$$\Phi_{\text{bar}}(R,\phi,z,t) = \alpha\frac{v_0^2}{3}\left(\frac{R_0}{R_{\text{b}}}\right)^3 U(r)\frac{R^2}{r^2}\cos(\gamma_{\text{b}}), \qquad (A.3)$$

where $R^2 = x^2 + y^2$ is the radial coordinate on the Galactic plane, $r^2 = R^2 + z^2$ is the spherical radius, R_{b} is the length of the bar, R_0 is the Galactocentric radius of the Sun, and v_0 is the circular velocity at R_0. The amplitude α is the ratio between the bar's and the axisymmetric contribution to the radial force along the bar's long axis at $(R,z) = (R_0,0)$. The parameter $\gamma_{\text{b}}(\phi,t) = 2\,(\phi - \phi_{\text{b}} - \Omega_{\text{b}}\,t)$, where ϕ_{b} and Ω_{b} are the initial phase and angular velocity of the bar pattern, respectively. For simulations that include the present time, it should be taken into account that $\phi_{\text{b,present}} = -25°$. Finally, $U(r)$ is given by:

$$U(r) = \begin{cases} -(r/R_{\text{b}})^{-3} & \text{for}\quad r \geq R_{\text{b}}, \\ (r/R_{\text{b}})^3 - 2 & \text{for}\quad r < R_{\text{b}}. \end{cases} \qquad (A.4)$$

Appendix A.3. The spiral arms

Following Monari et al. (2016), we describe the non-axisymmetric part of the Milky Way disc by a N-armed spiral pattern outside of the bar region, in the form originally proposed by Cox and Gómez (2002):

$$\Phi_{\text{spiral}}(R, \phi, z, t) = -\frac{A}{R_{\text{s}}KD} \cdot \text{e}^{-\frac{(R-R_{\text{s}})}{R_{\text{sd}}}} \cos(\gamma_{\text{s}}) \left[\text{sech}\left(\frac{Kz}{\beta}\right)\right]^{\beta}, \qquad \text{(A.5)}$$

where

$$\begin{aligned}
K(R) &= \frac{N}{R \sin p}, \\
\beta(R) &= K(R)h_{\text{s}}\left[1 + 0.4K(R)h_{\text{s}}\right], \\
D(R) &= \frac{1 + K(R)h_{\text{s}} + 0.3\left[K(R)h_{\text{s}}\right]^2}{1 + 0.3K(R)h_{\text{s}}}, \\
\gamma_{\text{s}}(R, \phi, t) &= N\left[\phi - \phi_{\text{s}} - \Omega_{\text{s}}t - \frac{\ln(R/R_{\text{s}})}{\tan p}\right].
\end{aligned} \qquad \text{(A.6)}$$

In Eqs. (A.5) and (A.6), p is the pitch angle, A the amplitude of the spiral potential, h_{s} controls the scale-height of the spiral, R_{s} is the reference radius for the angle of the spirals, R_{sd} is a length parameter that must be in agreement with the scale-length of the disc, $N = 2$ in order to have a two–armed spiral pattern (namely, the Scutum-Centaurus and Perseus arms), and ϕ_{s} and Ω_{s} are the initial phase and angular velocity of the spiral pattern, respectively (once again, those values are computed using a final integration time t_e). As in the case of the bar, the initial phase should be computed so that, at the present, the arms have their actual position.

The spiral arms representation is discarded inside the corrotation radius due to two main reasons: not only the model is poorly adequate for those inner locations (because of the multipolar expansion), but also it is a second-order perturbation that would go inside a region dominated by the bar.

A method to generalize the spiral pattern representation given by Eqs. (A.5) and (A.6) is explained in Cox and Gómez (2002). The user can add more modes by playing around with the values of n (number of modes) and N (number of arms) in the expression:

$$\Phi_{\text{spiral}}(R, \phi, z, t) = -\frac{A}{R_{\text{s}}} \cdot \text{e}^{-\frac{(R-R_{\text{s}})}{R_{\text{sd}}}} \sum_{i=1}^{n} \frac{C_i}{K_i D_i} \cos(\gamma_{\text{s},i}) \left[\text{sech}\left(\frac{K_i z}{\beta_i}\right)\right]^{\beta_i}, \qquad \text{(A.7)}$$

where

$$K_i(R) = \frac{iN}{R \sin p},$$
$$\beta_i(R) = K_i(R)h_{\mathrm{s}}\left[1 + 0.4K_i(R)h_{\mathrm{s}}\right],$$
$$D_i(R) = \frac{1 + K_i(R)h_{\mathrm{s}} + 0.3\left[K_i(R)h_{\mathrm{s}}\right]^2}{1 + 0.3K_i(R)h_{\mathrm{s}}} \qquad \text{(A.8)}$$
$$\gamma_{\mathrm{s},i}(R, \phi, t) = iN\left[\phi - \phi_{\mathrm{s}} - \Omega_{\mathrm{s}}t - \frac{\ln(R/R_{\mathrm{s}})}{\tan p}\right],$$

and C_i are coefficients that set the power of each mode. Thus, the series is a sum of terms, each one being similar to the two-armed potential of Eqs. (A.5) and (A.6). Then, since all the first and second derivatives are the same for all those terms (except for the constants), the user can build a general N-armed potential with its equations of motion and first variational equations by simply (but carefully) adding those expressions.

Appendix A.4. The disc

The MWH 2.0 offers two different options to represent the disc:

1. Option 1: a MN profile offers analytical and fairly simple expressions for the first and second derivatives of the disc potential in order to compute the equations of motion as well as the first variational equations;
2. Option 2: an exponential profile for the mass distribution had to be avoided, given that the corresponding potential includes Bessel functions that need to be integrated numerically. Instead, we have used a combination of MN models to build disc potentials that approximates such exponential mass profiles (following Smith et al. (2015)) and keeping at the same time the analytical and rather simple form of the first and second derivatives.

We first introduce the well-known MN disc potential that is used for both options coded in MWH 2.0 to build up the disc profiles:

$$\Phi_{\mathrm{disc}} = -\frac{G\,M_{\mathrm{disc}}}{\sqrt{x^2 + y^2 + \left(\epsilon_s + \sqrt{z^2 + \epsilon_h^2}\right)^2}}, \qquad \text{(A.9)}$$

where M_{disc} is its mass, and ϵ_s and ϵ_h are the scale length and scale height of the disc, respectively.

Option 1 can be turned into a two-component MN disc in a very easy way: to the default MN disc that is already coded in the subprogram `milkywayhydra.pav`, the user may simply add the blocks corresponding to the second disc by copying and pasting those of the first.

On the other hand, Option 2 uses a combination of MN discs to build a double exponential mass profile for each single disc. The density profile is given by:

$$\rho(R, z) = \rho_0 \exp(-R/R_\mathrm{d}) \exp(-|z|/h_z), \tag{A.10}$$

where the double-exponential is described by the parameters ρ_0, R_d, and h_z, which are the central density, the scale length and the scale height, respectively. Thin and thick exponential discs are already coded in the subprogram `milkywayhydra.pav`, following the work of Smith et al. (2015). The procedure is explained in detail in the MWH 2.0 descriptive memory that is freely available in the website of the suite (Section 1).

Appendix A.5. The dark matter halo

Finally, we introduce the hypothetical outermost galactic component: the DMH. The MWH 2.0 offers two different options to represent such component:

1. Option 1: a bi-triaxial extension of the NFW DMH used in Vogelsberger et al. (2008). They introduce a transition scale to have a triaxial profile (defined by an ellipsoidal radius) in the inner regions while leaving the outer regions round shaped as dark matter-only simulations have showed. In Maffione et al. (2018) we applied a second parameter to introduce triaxiality in the outer parts as well (this giving the bi-triaxial nature to the component). The idea is to have an oblate profile in the inner regions due to the impact of baryonic matter, while leaving the outer regions mildly triaxial as hydrodynamic simulations show, Marinacci et al. (2014); Grand et al. (2017);

2. Option 2: a modified logarithmic DMH formerly used in Vera-Ciro and Helmi (2013). The authors fitted the observations of the Sagittarius dwarf galaxy Law and Majewski (2010) guaranteeing at the same time a stable configuration for the disc. To this end, they also used a transition scale to obtain an oblate shape in the inner parts (as hydrodynamic simulations suggest, Marinacci et al. (2014)) and a triaxial shape in the outer parts because of the effect of the Sagittarius dwarf galaxy orbit.

Option 1 is described by the following expression:

$$\Phi_{\mathrm{DMH}} = -\frac{A_d}{r'} \ln\left[1 + \frac{r'}{r_{\mathrm{s}}}\right], \tag{A.11}$$

where

$$A_d = \frac{G\,M_{200}}{\ln(1 + c_{\mathrm{nfw}}) - \dfrac{c_{\mathrm{nfw}}}{1 + c_{\mathrm{nfw}}}}. \tag{A.12}$$

Here M_{200} is the halo virial mass (defined to be the mass inside a sphere in which the mean matter density is 200 times the critical density, $\rho_{cri} = 3H^2(z)/(8\pi G)$ with H the Hubble parameter), c_{nfw} is the concentration parameter, and $r' = r_{\mathrm{ie}} \cdot (r_{\mathrm{s}} + r_{\mathrm{oe}})/(r_{\mathrm{s}} + r_{\mathrm{ie}})$ is the modified scale radius adapted from Vogelsberger et al. (2008), with $r_{\mathrm{s}} = r_{200}/c_{\mathrm{nfw}}$ a scale radius and r_{200} the virial radius. Also, $r_{\mathrm{ie}}^2 = x^2/a^2 + y^2/b^2 + z^2/c^2$ is the square of the inner ellipsoidal radius with a, b, and c the inner semi-axes, while $r_{\mathrm{oe}}^2 = x^2/a'^2 + y^2/b'^2 + z^2/c'^2$ is its outer counterpart, with a', b', and c' the outer semi-axes. The ellipsoidal radii satisfy $a^2 + b^2 + c^2 = a'^2 + b'^2 + c'^2 = 3$.

Option 2: this form for the DMH potential describes a triaxial ellipsoid with its intermediate and major axes rotated with respect to the coordinate axes about the galactic z axis. The expression of the potential is:

$$\Phi_{\mathrm{dmh}} = v_{\mathrm{halo}}^2 \ln[(r^{\star})^2 + d^2]. \tag{A.13}$$

Here v_{halo} is the halo mass renormalization parameter that should be calibrated in order to obtain the circular velocity of the Sun $v_{\mathrm{LSR}}(R_0)$. The radius $r^{\star}$ is defined as $r^{\star} = r_{\mathrm{ie}} \cdot (r_{\mathrm{s}} + r_{\mathrm{oe}})/(r_{\mathrm{s}} + r_{\mathrm{ie}})$, where now $r_{\mathrm{ie}}^2 = x^2 + y^2 + z^2/q_z^2$ and $r_{\mathrm{oe}}^2 = W x^2 + V y^2 + U xy + z^2/q_3^2$. Here the parameters q_z and q_3 are the axial flattenings perpendicular to the galactic disc, and $W = a_1^2/q_1^2 + a_2^2/q_2^2$, $V = a_1^2/q_2^2 + a_2^2/q_1^2$, $U = 2a_1 a_2(1/q_1^2 - 1/q_2^2)$, where q_1 and q_2 are the axial flattenings along the equatorial axes, $a_1 = \cos\phi$, $a_2 = \sin\phi$, and ϕ is the angle of rotation of the potential around the z axis. Finally, d is the core radius of the logarithmic potential.

Before leaving the subject of the MWH components, a comment on the stellar halo is worth mentioning. The stellar halo may be oblate, strengthening the chaotic effects given by the triaxiality of the DMH Deason et al. (2011). However, its mass is about 1 per cent of the stellar component of the galaxy, and thus for all except the most specialized studies it can be neglected. This is the main reason why the current version of MWH does not include such a stellar halo component.

References

Antoja, T., Helmi, A., Dehnen, W., Bienaymé, O., Bland-Hawthorn, J., Famaey, B., Freeman, K., Gibson, B.K., Gilmore, G., Grebel, E.K., Kordopatis, G., Kunder, A., Minchev, I., Munari, U., Navarro, J., Parker, Q., Reid, W.A., Seabroke, G., Siebert, A., Steinmetz, M., Watson, F., Wyse, R.F.G., Zwitter, T., 2014. Constraints on the Galactic bar from the Hercules stream as traced with RAVE across the Galaxy. A&A 563, A60. `1309.4272`.

Banik, N., Bovy, J., 2019. Effects of baryonic and dark matter substructure on the Pal 5 stream. MNRAS 484, 2009–2020. `1809.09640`.

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., 1982. Regular and irregular orbits in galactic bars. MNRAS 201, 1–14.

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

Bischof, C., Carle, A., Khademi, P., Mauer, A., 1995. The adifor 2.0 system for the automatic differentiation of fortran 77 programs .

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

Carpintero, D.D., Maffione, N., Darriba, L., 2014. LP-VIcode: A program to compute a suite of variational chaos indicators. Astronomy and Computing 5, 19–27. `1404.2152`.

Carpintero, D.D., Muzzio, J.C., Wachlin, F.C., 1999. Regular and Chaotic Motion in Globular Clusters. Celestial Mechanics and Dynamical Astronomy 73, 159–168. `astro-ph/9811392`.

Carpintero, D.D., Wachlin, F.C., 2006. Sensitivity of the orbital content of a model stellar system to the potential approximation used to describe it. Celestial Mechanics and Dynamical Astronomy 96, 129–136.

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.

Cox, D.P., Gómez, G.C., 2002. Analytical Expressions for Spiral Arm Gravitational Potential and Density. ApJS 142, 261–267. `astro-ph/0207635`.

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.

de Zeeuw, T., 1985. Elliptical galaxies with separable potentials. MNRAS 216, 273–334.

Deason, A.J., Belokurov, V., Evans, N.W., 2011. The Milky Way stellar halo out to 40 kpc: squashed, broken but smooth. MNRAS 416, 2903–2915. `1104.3220`.

Dehnen, W., 2000. The Effect of the Outer Lindblad Resonance of the Galactic Bar on the Local Stellar Velocity Distribution. AJ 119, 800–812. `astro-ph/9911161`.

Dobmann, M., Liepelt, M., Schittkowski, K., 1995. Algorithm 746: Pcomp: A fortran code for automatic differentiation. ACM Transactions on Mathematical Software (TOMS) 21, 233–266.

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.

Gómez, F.A., Helmi, A., Brown, A.G.A., Li, Y.S., 2010. On the identification of merger debris in the Gaia era. MNRAS 408, 935–946. `1004.4974`.

Gómez, F.A., Torres-Flores, S., Mora-Urrejola, C., Monachesi, A., White, S.D.M., Maffione, N.P., Grand, R.J.J., Marinacci, F., Pakmor, R., Springel, V., Frenk, C.S., Amram, P., Epinat, B., Mendes de Oliveira, C., 2021. A Tidally Induced Global Corrugation Pattern in an External Disk Galaxy Similar to the Milky Way. ApJ 908, 27.

Grand, R.J.J., Gómez, F.A., Marinacci, F., Pakmor, R., Springel, V., Campbell, D.J.R., Frenk, C.S., Jenkins, A., White, S.D.M., 2017. The Auriga Project: the properties and formation mechanisms of disc galaxies across cosmic time. MNRAS 467, 179–207. `1610.01159`.

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

Hernquist, L., 1990. An analytical model for spherical galaxies and bulges. ApJ 356, 359–364.

Hernquist, L., Ostriker, J.P., 1992. A Self-consistent Field Method for Galactic Dynamics. ApJ 386, 375.

Jackson, E.A., 1989. Perspectives of Nonlinear Dynamics. volume 1. Cambridge University Press.

Jorba, A., Zou, M., 2005. A software package for the numerical integration of odes by means of high-order taylor methods. Experimental Mathematics 14, 99–117.

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

Launhardt, R., Zylka, R., Mezger, P.G., 2002. The nuclear bulge of the Galaxy. III. Large-scale physical characteristics of stars and interstellar matter. A&A 384, 112–139. `astro-ph/0201294`.

Law, D.R., Majewski, S.R., 2010. The Sagittarius Dwarf Galaxy: A Model for Evolution in a Triaxial Milky Way Halo. ApJ 714, 229–254. `1003.1132`.

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`.

Maffione, N.P., Gómez, F.A., Cincotta, P.M., Giordano, C.M., Cooper, A.P., O'Shea, B.W., 2015. On the relevance of chaos for halo stars in the solar neighbourhood. MNRAS 453, 2830–2847. `1508.00579`.

Maffione, N.P., Gómez, F.A., Cincotta, P.M., Giordano, C.M., Grand, R.J.J., Marinacci, F., Pakmor, R., Simpson, C.M., Springel, V., Frenk, C.S., 2018. On the relevance of chaos for halo stars in the solar neighbourhood II. MNRAS 478, 4052–4067. `1801.03946`.

Marinacci, F., Pakmor, R., Springel, V., 2014. The formation of disc galaxies in high-resolution moving-mesh cosmological simulations. MNRAS 437, 1750–1775. `1305.5360`.

35

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

Miyamoto, M., Nagai, R., 1975. Three-dimensional models for the distribution of mass in galaxies. PASJ 27, 533–543.

Monari, G., Famaey, B., Siebert, A., Grand, R.J.J., Kawata, D., Boily, C., 2016. The effects of bar-spiral coupling on stellar kinematics in the Galaxy. MNRAS 461, 3835–3846. 1606.06785.

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.

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

Plummer, H.C., 1911. On the problem of distribution in globular star clusters. MNRAS 71, 460–470.

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., 1993. Self-consistent Models for Galactic Halos. ApJ 409, 563.

Simpson, C.M., Gargiulo, I., Gómez, F.A., Grand, R.J.J., Maffione, N., Cooper, A.P., Deason, A.J., Frenk, C., Helly, J., Marinacci, F., Pakmor, R., 2019. Simulating cosmological substructure in the solar neighbourhood. MNRAS 490, L32–L37. 1905.09842.

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.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.

Smith, R., Flynn, C., Candlish, G.N., Fellhauer, M., Gibson, B.K., 2015. Simple and accurate modelling of the gravitational potential produced by thick and thin exponential discs. MNRAS 448, 2934–2940. 1502.00627.

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

Vera-Ciro, C., Helmi, A., 2013. Constraints on the Shape of the Milky Way Dark Matter Halo from the Sagittarius Stream. ApJ 773, L4. 1304.4646.

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.

Zotos, E.E., Carpintero, D.D., 2013. Orbit classification in the meridional plane of a disk galaxy model with a spherical nucleus. Celestial Mechanics and Dynamical Astronomy 116, 417–438. 1307.4205.