AUTOMATIC DIFFERENTIATION METHODS IN COMPUTATIONAL DYNAMICAL SYSTEMS: INVARIANT MANIFOLDS AND NORMAL FORMS OF VECTOR FIELDS AT FIXED POINTS

A. HARO

Abstract. We present a methodology to compute power series expansions of parameterizations of invariant manifolds and of normal forms of vector fields near an equilibrium. The described seminumerical algorithms combine solving functional equations and automatic differentiation techniques.

As an application, we compute a high order approximation of the 4D center manifold of the L_1 Lagrangian equilibrium point of the Earth-Moon system. We explore the range of validity of this asymptotic expansions. We study the dynamics on the center manifold.

Mathematics Subject Classification: 34C20,34C30,34C30,34C30,65Pxx,68W30

1. Introduction

Poincaré's program for the global analysis of a dynamical system starts by considering its fixed points and their invariant manifolds, and normal form reduction is a fruitful technique. There is a vast theoretical and applied literature dealing with questions about existence and regularity of invariant manifolds and normal forms [9, 10, 37, 50]. Low order expansions are adequate for local studies, including bifurcation analysis [16] and some globalization algorithms of invariant manifolds [47], but may be inadequate for accurate computations of center and weak stable manifolds. High order expansions provide semilocal approximations of invariant manifolds, reliable in a large neighborhood of the steady state [56]. On this issue, there is a long tradition in Celestial Mechanics, the field that inspired Poincaré, in developing specific software for manipulating multivariate series expansions [24]. Notable recent examples of algebraic manipulators are `TRIP` [1, 29], Piranha [2, 18], and Jorba's public domain software [3]. Since coefficients of

A.H. acknowledges the support of Spanish Grants MTM2006-11265 and MTM2009-09723 and Catalan grants 2005-SGR-01028 and 2009-SGR-67.

power series expansions are (normalized) derivatives, it is natural to consider methods of Automatic Differentiation (AD). AD is a set of techniques based on the mechanical application of the chain rule to obtain derivatives of a function given as a computer program [4] (see [36] for a survey of the state-of-the-art of AD.) A number of AD tools [4] are being used in studies of dynamical systems, such as `ADOL-C`, [5, 34, 38, 48], `COSY INFINITY` [6, 15, 49], or `C-XSC` [7]. A main point is how to use and extend these techniques for problems involving high order derivatives.

In this paper, we present a new methodology to numerically compute the coefficients of high order power series expansions of parameterizations of invariant manifolds and normal form transformations around a fixed point of an *elementary* vector field, i.e. built from a finite number of arithmetic operations and basic functions such as constants, powers, exponentials, logarithms, trigonometric functions and their inverses, or elliptic functions. Most of what is explained in this paper can also be applied to the discrete time case, that is, for maps near a fixed point.

The methodology is rooted in the parameterization method of Cabré, Fontich and de la Llave [23] and the semi-numerical algorithms proposed by Simó [59], whose strategies go back to Poincaré. Under the umbrella of the parameterization method we place different strategies or *styles* [50] of parameterizations of invariant manifolds and their corresponding dynamics, including normal forms and the graph transform. The implementation of the method is based on *on-line* algorithms for manipulating power series, computing the terms of the series in increasing order.

Our approach to AD consists in a new and very simple generalization to multivariate power series of the well-known recurrence formulas for composition of univariate power series with elementary functions [46]. Several extensions to the multivariate case have been considered in the literature, see e.g. the review in [53]. Apart from its simplicity, our formulation matches with invariant manifolds and normal form computations [40].

We have developed our own `C` software to manipulate (truncated) multivariate dense power series, implementing AD with the on-line algorithms. The implementation benefits from the property of recursiveness, making the software potentially applicable for any number of variables and any degree of the polynomials. We note that general purpose manipulators, like `Mathematica` or `Maple`, are not adequate for high order computations with multivariate power series or for the kind of problems involving massive computations addressed in this paper.

As an application, we compute a high order polynomial approximation of the center manifold of the L_1 Lagrange equilibrium point of the Earth-Moon system, in the context of the (spatial) Restricted Three Body Problem (RTBP). This manifold has been computed in several works using partial normal forms with the Lie series method [43, 56], and has applications in spacecraft dynamics [31, 32] and in molecular dynamics [27, 63]. Instead of using partial normal forms that involve 6-variate power series, we compute the center manifold using 4-variate power series, an approach that has been used recently in a very similar model in [26] (see [39] for center manifold computations in maps). The reduction of the dimension of the problem, the use of AD methods with on-line recurrence formulas and an efficient implementation of the algebraic manipulator reduces both execution time and memory space by several orders of magnitude compared to standard methods. We also reach high order approximations (e.g. order 60) without difficulty in a few minutes. These power series are asymptotic expansions that provide quite sharp approximations of the center manifold in a relatively large neighborhood around the origin. The growth of the coefficients of the expansions give insight into the range of validity of the approximations [58]. Following [30, 31, 44, 60] we analyze the dynamics on the center manifold for different energy levels.

Organization of the paper. In Section 2 we present the AD methods based on on-line algorithms of manipulation of multivariate power series. Section 3 is devoted to the algorithms used to solve the functional equations to compute invariant manifolds and normal forms. We also evaluate their cost. The illustrative example, the computation of the invariant manifolds in the RTBP, is documented in Section 4. Some details of the implementation and benchmarks of the software package are described in Appendix A.

Benchmarks. We have compiled and run our codes on several machines, including three Macs, several Linux PC's, and a cluster system of 46 Sun Fire V20z servers (each of them with two 2.2Ghz AMD Opteron processors, 4 GB of RAM) running under Linux. The results reported in this paper correspond to three Mac machines, that we call Mac I (laptop MacBook Pro 15"; Processor: 2.16 GHz Intel Core Duo, 2 MB L2 Cache; Memory: 2 GB 667 MHz DDR2 SDRAM), Mac II (desktop iMac; 2 GHz Intel Core Due, 4MB L2 Cache; Memory: 1 GB 667 MHz DDR2 SDRAM) and Mac III (desktop iMac; 2.8 GHz Intel Core 2 Due, 6 MB L2 Cache; Memory: 4 GB 800 MHz DDR2 SDRAM). Mac I and Mac II run under MAC OS X 10.4.1 (Tiger), and

the codes are compiled with `gcc 4.0.1`. Mac III run under Mac OS X 10.6.2 (Leopard), and the codes are compiled with `gcc 4.2.1`.

Acknowledgements. I am very grateful to Rafael de la Llave and to Carles Simó for their interest and encouragement, and for their valuable comments in reviewing several versions of the paper. Special thanks to M. Gastineau and Carles Simó for benchmarking their codes and to Àngel Jorba for allowing me to use his public domain software package [3] for benchmarking, which helped me to improve my own codes. Thanks to the Department of Mathematics of the University of Texas at Austin for providing me with access to their computer facilities, in particular for the 46 node AMD opteron cluster system.

Many thanks to the organizers of the course "Advanced School on Specific Algebraic Manipulators", in Barcelona (September 12-15, 2007), for their invitation to explain the methodologies discussed in the first version of the paper, which were the basis of my lectures [40]. I extended the notes after my lectures on the course "Invariant manifolds" that I gave in the "i-Math DocCourse: Computational Methods in Dynamical Systems and Applications", also in Barcelona (autumn of 2010). The comments of my companion Josep Maria Mondelo, who gave the practical part of the course, were also very valuable. The enthusiasm and interest of the participants on the courses, and especially Luis Benet, were also very stimulating.

Thanks to IMA (University of Minnesota) and to the organizers of the course "New Directions Short Course: Invariant Objects in Dynamical Systems and their Applications" (June 20 to July 1, 2011), and very especially to Rafael de la Llave, for giving me the opportunity to lecture on this subject. I am sure that the comments of the participants will encourage me to improve the present version of these notes.

2. Multivariate power series

In this section we establish the notation and definitions related with multivariate power series. We introduce on-line formulas to compute compositions of power series with elementary functions. These formulas are among the keys for designing general purpose algorithms for computing power series expansions as solutions of functional equations, described in Section 3.

2.1. **The algebra of power series.** For $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$, a (real or complex) power series in the variables $x = (x_1, \ldots, x_d)$ with coefficients in $\mathbb{K}$ is an element of the commutative algebra $\mathbb{K}[[x]] = \mathbb{K}[[x_1, \ldots, x_d]]$, whose elements are of the form

$$f(x) = \sum_{k=0}^{\infty} f_k(x) \ , \tag{1}$$

where each $f_k(x)$ is a homogeneous polynomial of order k,

$$f_k(x) = \sum_{m_1+\cdots+m_d=k} f_{m_1,\ldots,m_d} x_1^{m_1} \ldots x_d^{m_d} = \sum_{|m|=k} f_m x^m \ ,$$

where in the latter we use the standard multi-index notation and $|m|$ denotes the ℓ_1-norm of the multi-index $m \in \mathbb{N}^d$.

The algebraic operations in $\mathbb{K}[[x]]$ are defined as follows. The scalar multiplication and addition in $\mathbb{K}[[x]]$ are defined coefficient-wise. The product $p(x) = f(x)g(x)$ of two power series is provided by the convolution formula

$$p_k(x) = \sum_{j=0}^{k} f_j(x) g_{k-j}(x) \ .$$

Assuming that g_0 is invertible, the division of the two power series, $d(x) = a(x)/b(x)$, is performed recursively by using the iteration

$$d_k(x) = \frac{1}{b_0} \left(a_k(x) - \sum_{j=0}^{k-1} d_j(x) b_{k-j}(x) \right) \ .$$

For $k \geq 0$, the kth truncation of $f(x)$ is the polynomial $f_{\leq k}(x) = \sum_{i=0}^{k} f_i(x)$. We will also use the obvious notations $f_{<k}(x)$ and $f_{>k}(x)$. Algebraic operations extend naturally to the set of kth truncated power series, which constitute a quotient algebra of finite dimension

$$\mathrm{n}_d(k) = \binom{d+k}{d} \sim \frac{1}{d!} k^d \ , \tag{2}$$

where $a_k \sim b_k$ means that $\lim_{k\to\infty} a_k b_k^{-1} = 1$.

2.2. **A working definition of algorithmic complexity.** The truncated product of two power series using the convolution formula involves products of homogeneous polynomials. In computing the multiplication of each pair of monomials we perform the following three-step operation: address the product of monomials, multiply the two numerical coefficients, add the result to the current address. The cost of an algorithm is its number of operations.

In particular, the cost of the kth truncated product of two d-variate series, using (naively) the convolution formula, is

$$\mathrm{p}_d(k) = \binom{2d+k}{2d} \sim \frac{1}{(2d)!}k^{2d} \sim \frac{d!^2}{(2d)!}\mathrm{n}_d(k)^2\,. \tag{3}$$

Many functions for manipulating power series are built in the convolution. Other operations, such as scalar multiplication or addition, have negligible cost compared with the cost of the convolution. We consider the following definition of complexity.

Definition 2.1. *Given an algorithm that computes kth truncated d-variate power series, let $\mathrm{c}_d(k)$ be its* cost*, i.e. the number of operations. We say that the algorithm is* efficient *if $\mathrm{c}_d(k) \sim C\mathrm{p}_d(k)$ for some constant C independent of k, where $\mathrm{p}_d(k)$ denotes the cost of the kth truncated product. We refer to C as the* complexity *of the algorithm.*

Remark 2.2. We have not explored the application of fast convolution algorithms such us the Karatsuba and Toom-Cook methods [21, 25, 45, 55, 62], or FFT methods [22, 46, 54, 64, 65], including multi-evaluation and interpolation techniques and reduction to univariate series [19, 35, 52]. These fast methods seem to be advantageous from very high orders (say, higher than 1000), which are not usual in dynamical systems computations. Moreover, their numerical stability is unclear. As far as we know, the influence of the choice of the interpolation nodes on round-off errors has not been studied to date.

Remark 2.3. In the complexity analysis of algorithms, besides counting the number of arithmetic operations among coefficients, one must also consider the addressing schemes to locate and save the coefficients in the data structure encoding polynomials. Efficient addressing schemes are crucial in the implementation of algebraic manipulators.

Remark 2.4. Efficient implementation of algorithms is an important issue. For instance, while the `Mathematica` implementation of an FFT method can take hours to compute the truncated exponential up to order 10 of a 10-variate power series on a PC [52], our C programs based on the naive algorithm take less than 0.20 seconds on a slightly old laptop (Mac I).

2.3. **Elementary functions of power series.** The composition of power series is the bottleneck in many algorithms. For instance, if $\varphi = \varphi(z)$ is a univariate function and $f(x)$ is a power series, one can compute the composition $\varphi(f(x))$ by Taylor expanding φ around $f_0 = f(0)$, computing the powers of $f(x)$ and adding up. However, if the function φ is *elementary*, then it is much better to derive specific formulae. Online algorithms for univariate power series go back to Euler [46], and here we generalize them for the multivariate case. A useful concept is that of radial derivative.

The radial derivative of a function $f(x)$ is defined by

$$Rf(x) = \mathrm{grad} f(x) \cdot x = \sum_{i=1}^{d} \frac{\partial f}{\partial x_i}(x)\ x_i\ .$$

For an homogeneous function of order k, f_k, $Rf_k(x) = kf(x)$, which is the famous Euler's identity. A simple observation is the following chain rule:

$$R(\varphi \circ f)(x) = \varphi'(f(x))\ Rf(x)\ . \tag{4}$$

Therefore, if φ satisfies an elementary differential equation, we can use (4) to obtain a recurrence for the homogeneous terms g_k of the power series $g = \varphi \circ f$, starting from the seed $g_0 = \varphi(f_0)$.

Example 2.5. Consider the case $\varphi(z) = z^\alpha$, the power function of exponent $\alpha \neq 0$ (in Section 4, $\alpha = -\frac{3}{2}$ or $\alpha = 2$). Since $z\varphi'(z) = \alpha\varphi(z)$, the composition $p(x) = (f(x))^\alpha$ satisfies the identity $f(x)Rp(x) = \alpha p(x)Rf(x)$. We then obtain the recurrence

$$p_k(x) = \frac{1}{k\ f_0} \sum_{j=0}^{k-1} (\alpha(k-j) - j)\, f_{k-j}(x) p_j(x)\ ,$$

starting with $p_0 = f_0^\alpha$ (we assume $f_0 \neq 0$). Observe that the complexity of the computation is 1. Special cases are $\alpha = 2$ and $\alpha = \frac{1}{2}$, whose complexities are $\sim \frac{1}{2}$.

Extending the arguments, one can derive efficient formulas for the composition of *multivariate* power series with any univariate elementary function φ that satisfies an elementary differential equation $\varphi'(z) = F(\varphi(z))$ (in this recursive definition we start with the constants and arithmetic operations as the most elementary functions). Table 1 summarizes several examples, indicating the corresponding complexities (C). Analogous formulas for *univariate* power series have appeared in many places, see e.g. [20].

Remark 2.6. From the existing extensions of the composition formulas to the multivariate case, see e.g. [53] for a review and [8] for some `C-XSC` implementations in the 2-variate case, our approach is closer to the formulation in [14, 51]. In a way, these recurrence algorithms are derived from the chain rule $\nabla(\varphi \circ f)(x) = \varphi'(f(x))\nabla f(x)$ involving the gradient operator, while our recurrence algorithms are derived from the chain rule $R(\varphi \circ f)(x) = \varphi'(f(x))Rf(x)$ for the radial derivative operator. Hence, instead of solving Pfaffian systems $\nabla g(x) = H(x)$, our method is based on solving first-order linear PDEs $Rg(x) = h(x)$ around the singularity 0. This is the key to the simplicity of our formulation.

2.4. **A working definition of complexity of a function.** Motivated by the on-line formulas introduced in the previous section, we consider the following working definition.

Definition 2.7. *A function is* elementary *if it can be evaluated through a sequence of finite* length *of steps, where one step is an expression involving either a constant, or an arithmetic operation or a univariate elementary function (see Table 1 for some examples). The* complexity *of the elementary function is the sum of the complexities of the steps.*

In the computation of the composition of an elementary function with power series, the computer memory space needed to store the intermediate steps has to do with its length, while the execution time is related to its complexity. Both definitions of the length and the complexity depend on how the equations of the model are written.

3. Computation of invariant manifolds and normal forms

In this section, under the umbrella of the parameterization method [23], we overview several algorithms for computing power series expansions of invariant manifolds and normal forms of fixed points of vector fields [9, 10, 17, 23, 37, 50, 59].

We estimate the complexity of the derived algorithms, specifically for elementary vector fields.

3.1. **The invariance equation.** Let $z_* \in \mathbb{R}^n$ be a fixed point of an n-dimensional vector field

$$\dot{z} = F(z) \ , \tag{5}$$

where $z = (z_1, \ldots, z_n)$. Let $E \subset \mathbb{R}^n$ be a d-dimensional subspace, invariant for the linearization $\dot{v} = \mathrm{D}F(z_*)v$ around z_*. The goal is

elementary function	$k \geq 1$	C
$p(x) = af(x) \pm bg(x)$	$p_k(x) = af_k(x) \pm bg_k(x)$	0
$p(x) = f(x)g(x)$	$p_k(x) = \sum_{j=0}^{k} f_j(x) g_{k-j}(x)$	1
$d(x) = f(x)/g(x)$	$d_k(x) = \frac{1}{g_0}\left(f_k(x) - \sum_{j=0}^{k-1} d_j(x) g_{k-j}(x)\right)$	1
$p(x) = (f(x))^\alpha$	$p_k(x) = \frac{1}{k\, f_0} \sum_{j=0}^{k-1} (\alpha(k-j) - j)\, f_{k-j}(x) p_j(x)$	1
$p(x) = (f(x))^2$	$p_k(x) = \begin{cases} 2 \sum_{j=0}^{(k-1)/2} f_j(x) f_{k-j}(x), & \text{if } k \text{ odd} \\ 2 \sum_{j=0}^{(k-2)/2} f_j(x) f_{k-j}(x) + (f_{k/2}(x))^2, & \text{if } k \text{ even} \end{cases}$	0.5
$p(x) = (f(x))^{\frac{1}{2}}$	$p_k(x) = \begin{cases} \frac{1}{2p_0}\left(f_k(x) - 2 \sum_{j=1}^{(k-1)/2} p_j(x) p_{k-j}(x)\right), & \text{if } k \text{ odd} \\ \frac{1}{2p_0}\left(f_k(x) - 2 \sum_{j=1}^{(k-2)/2} p_j(x) p_{k-j}(x) - (p_{k/2}(x))^2\right), & \text{if } k \text{ even} \end{cases}$	0.5
$e(x) = \exp(f(x))$	$e_k(x) = \frac{1}{k} \sum_{j=0}^{k-1} (k-j) f_{k-j}(x) e_j(x)$	1
$l(x) = \log(f(x))$	$l_k(x) = \frac{1}{f_0}\left(f_k(x) - \frac{1}{k} \sum_{j=1}^{k-1} j f_{k-j}(x) l_j(x)\right)$	1
$s(x) = \sin(f(x))$	$s_k(x) = \frac{1}{k} \sum_{j=0}^{k-1} (k-j) f_{k-j}(x) c_j(x)$	
$c(x) = \cos(f(x))$	$c_k(x) = -\frac{1}{k} \sum_{j=0}^{k-1} (k-j) f_{k-j}(x) s_j(x)$	2
$t(x) = \tan(f(x))$	$t_k(x) = f_k(x) + \frac{1}{k} \sum_{j=0}^{k-1} (k-j) f_{k-j}(x) p_j(x)$	
$p(x) = (t(x))^2$		1.5
$a(x) = \arcsin(f(x))$	$a_k(x) = \frac{1}{\sqrt{1 - f_0^2}}\left(f_k(x) - \frac{1}{k} \sum_{j=1}^{k-1} j r_{k-j}(x) a_j(x)\right)$	
$s(x) = (f(x))^2$ $r(x) = \sqrt{1 - s(x)}$		2
$a(x) = \arctan(f(x))$	$a_k(x) = \frac{1}{1 + f_0^2}\left(f_k(x) - \frac{1}{k} \sum_{j=1}^{k-1} j r_{k-j}(x) a_j(x)\right)$	
$r(x) = 1 + (f(x))^2$		1.5
$s(x) = \mathrm{sn}(f(x))$	$s_k(x) = \frac{1}{k} \sum_{j=0}^{k-1} b_{k-j}(x) d_j(x)$	
$c(x) = \mathrm{cn}(f(x))$	$c_k(x) = -\frac{1}{k} \sum_{j=0}^{k-1} a_{k-j}(x) d_j(x)$	
$d(x) = \mathrm{dn}(f(x))$	$d_k(x) = -\frac{\kappa^2}{k} \sum_{j=0}^{k-1} a_{k-j}(x) c_j(x)$	
elliptic modulus: κ		
$a(x) = s(x) Rf(x)$	$a_k(x) = \sum_{j=0}^{k-1} (k-j) f_{k-j}(x) s_j(x)$	
$b(x) = c(x) Rf(x)$	$b_k(x) = \sum_{j=0}^{k-1} (k-j) f_{k-j}(x) c_j(x)$	5

TABLE 1. Some elementary functions of power series, and their complexities (C).

to compute a high order approximation of a d-dimensional invariant manifold $\mathcal{W}$ for (5), tangent to E at z_*.

In the light of the parameterization method [23], we look for a parameterization $z = W(s)$ of the invariant manifold $\mathcal{W}$, where $s = (s_1, \ldots, s_d)$ are the coordinates and $W(0) = z_*$, and for the dynamics on the manifold, which is described by a vector field $\dot{s} = f(s)$, with $f(0) = 0$. The invariance equation is given by

$$F(W(s)) = \mathrm{D}W(s) f(s) \ . \tag{6}$$

Using an affine transformation, we may assume that $z_* = 0$ and $E = \{(x, 0) \in \mathbb{R}^n \mid x \in \mathbb{R}^d\}$. We split coordinates in $z = (x, y)$, with $x = (z_1, \ldots, z_d)$ and $y = (z_{d+1}, \ldots, z_n)$. Then, $F(0) = 0$ and the linearization is block triangular, i.e.

$$\mathrm{D}F(0) = \begin{pmatrix} A_1 & B \\ 0 & A_2 \end{pmatrix} \ .$$

In what follows, we use notations such as v^x or v^y to represent the projections of $v \in \mathbb{R}^n$ in x or y components respectively. Note that, since matrices A_1 and A_2 come from a real vector field (and a real invariant subspace E), then the corresponding spectra are invariant under complex conjugation.

We consider Equation (6) formally in terms of power series for the unknowns W and f. Hence, we split the parameterization W in x, y components, as

$$\begin{aligned} W^x(s) &= s + \sum_{k \geq 2} W_k^x(s) \ , \\ W^y(s) &= \sum_{k \geq 2} W_k^y(s) \ . \end{aligned} \tag{7}$$

The reduced vector field is

$$f(s) = A_1 s + \sum_{k \geq 2} f_k(s) \ . \tag{8}$$

First order terms of W and f are known, and our aim is to find higher order terms. The standard procedure to solve (formally) (6) is by substituting the expansions (7) and (8) in (6), and find their homogeneous terms in increasing order.

3.2. **The homological equations.** At each step $k > 1$, the goal is to compute $W_k(s)$ and $f_k(s)$, assuming that $W_{<k}(s)$ and $f_{<k}(s)$ have already been computed in previous steps. Moreover, the left hand side of (6) up to order $k - 1$, that is $[F(W_{<k}(s))]_{<k}$, is also known. We

first compute $[F(W_{<k}(s))]_k$. Then, the order-k terms in the invariance equation (6) lead us to the order-k *homological equation* for W_k and f_k,

(9)
$$\mathrm{D}W_k(s)A_1 s - AW_k(s) + \mathrm{D}W_1(s)f_k(s) = [F(W_{<k}(s))]_k - [\mathrm{D}W_{<k}(s)f_{<k}(s)]_k \,,$$

which is equivalent to

$$\mathrm{D}W_k^x(s)A_1 s - A_1 W_k^x(s) + f_k(s) = R_k^x(s) + BW_k^y(s) \,, \tag{10}$$
$$\mathrm{D}W_k^y(s)A_1 s - A_2 W_k^y(s) = R_k^y(s) \,, \tag{11}$$

where $R_k(s)$ denotes the right hand side of (9).

After solving (9) (see below), we compute $[F(W_{\leq k}(s))]_{\leq k}$ just adding $AW_k(s)$ to $[F(W_{<k}(s))]_{\leq k}$. This is necessary to start the next step of the on-line method.

Solution of the homological equation depends on the eigenvalues of A_1 (which we refer to as *internal eigenvalues*) and the eigenvalues of A_2. For the sake of simplicity, we analyze the solution of (9) under the generic assumption that the block-diagonal matrices A_1 and A_2 can be reduced to diagonal form, possibly with complex entries: $A_1 = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ and $A_2 = \mathrm{diag}(\lambda_{d+1}, \ldots, \lambda_n)$. In the following, $\mathrm{Re}\, w$ denotes the real part of a complex number w.

3.2.1. *Cross resonances.* Equation (11) is written, for $i = d+1, \ldots, n$:

$$\lambda_1 \frac{\partial W_k^i}{\partial s_1} s_1 + \cdots + \lambda_d \frac{\partial W_k^i}{\partial s_d} s_d - \lambda_i W_k^i(s) = R_k^i(s) \,.$$

These equations are diagonal in the coefficients W_m^i of the homogeneous polynomials $W_k^i(s)$. In particular, for $i = d+1, \ldots, n$, $|m| = k$:

$$(\lambda^x \cdot m - \lambda_i) W_m^i = R_m^i. \tag{12}$$

The pairs $(m, i) \in \mathbb{N}^d \times \{d+1, \ldots, n\}$ with $|m| \geq 2$ such that $\lambda^x \cdot m - \lambda_i = 0$ are *cross resonances* [9], and are obstructions to solve the homological equation (11) (and to the existence of the invariant manifold). Hence, if there are no cross resonances, for $i = d+1, \ldots, n$, $|m| = k$:

$$W_m^i = \frac{R_m^i}{\lambda^x \cdot m - \lambda_i}.$$

The non-existence of crossing resonances depends on the way the eigenvalues are grouped. Notable examples are the following groupings (where $1 \leq i \leq d$ and $d+1 \leq j \leq n$): $\mathrm{Re}\lambda_i < 0, \mathrm{Re}\lambda_j \geq 0$ (stable manifold); $\mathrm{Re}\lambda_i > 0, \mathrm{Re}\lambda_j \leq 0$ (unstable manifold); $\mathrm{Re}\lambda_i = 0, \mathrm{Re}\lambda_j \neq 0$ (center manifold); $\mathrm{Re}\lambda_i \leq 0, \mathrm{Re}\lambda_j > 0$ (center-stable manifold); $\mathrm{Re}\lambda_i \geq 0, \mathrm{Re}\lambda_j < 0$ (center-unstable manifold).

3.2.2. *Internal resonances.* Equation (10) is written, for $i = 1, \dots, d$:

$$\lambda_1 \frac{\partial W_k^i}{\partial s_1} s_1 + \cdots + \lambda_d \frac{\partial W_k^i}{\partial s_d} s_d - \lambda_i W_k^i(s) + f_k^i(s) = \hat{R}_k^i(s) \ , \tag{13}$$

where $\hat{R}_k^x(s) = R_k^x(s) + BW_k^y(s)$. This is equivalent to split the equations for $i = 1, \dots, d$, $|m| = k$:

$$(\lambda^x \cdot m - \lambda_i) W_m^i + f_m^i = \hat{R}_m^i. \tag{14}$$

The pairs $(m, i) \in \mathbb{N}^d \times \{1, \dots, d\}$ with $|m| \geq 2$ such that $\lambda^x \cdot m - \lambda_i = 0$ are *internal resonances*, and are obstructions to the linearization of the dynamics on the manifold. A main difference of (10) with respect to (11) is that, even in presence of resonances, Equation (14) can be solved by adjusting f_m^i.

3.3. **Styles of parameterizations.** Since solutions to (10) are not unique, one can adapt the *style* of the parameterization to the problem at hand. In the following, we consider several examples.

3.3.1. *The normal form style.* This style consists in simplifying the equations of the dynamics on the manifold, finding a normal form for f. To this end, for $i = 1, \dots, d$, $|m| = k$:

$$\begin{cases} f_m^i = 0 \ , \ W_m^i = \dfrac{\hat{R}_m^i}{\lambda^x \cdot m - \lambda_i}, & \text{if } \lambda^x \cdot m - \lambda_i \neq 0; \\ f_m^i = \hat{R}_m^i \ , \ W_m^i = 0, & \text{if } \lambda^x \cdot m - \lambda_i = 0. \end{cases} \tag{15}$$

The case $d = n$ corresponds to computing the normal form f of the vector field F.

The normal form style is especially well suited for cases in which there is a finite number of internal resonances, and hence dynamics on the manifold can be described by a polynomial vector field. This happens when the d internal eigenvalues belong to the Poincaré domain [11] (the internal eigenvalues all lie either in the left half-plane or in the right half-plane). This is the case of the stable and unstable manifolds, but one can also consider manifolds associated to subsets of stable eigenvalues (with negative real part). A special case is the strong or fast stable manifold, which is associated with eigenvalues of "most negative" real part. An important case is the weak or slow stable manifold (if there are no cross resonances), which corresponds to internal eigenvalues of "less negative" real part, which dominates the dynamics inside the stable manifold. See [23, 41] for rigorous results on the existence, uniqueness and regularity of these invariant manifolds, and [33] for some numerical computations of stable, fast stable and slow manifolds in maps.

3.3.2. *The graph style.* This style consists in simplifying the parameterization of the manifold rather than its dynamics, by taking $W_k^x(s) = 0$ and $f_k(s) = \hat{R}_k^x(s)$. Hence, the (local) invariant manifold is a graph $y = \psi(x)$ (i.e. $W^x(s) = s$ and $W^y(s) = \psi(s)$, and the dynamics is given by $\dot{x} = f(x) = F(x, \psi(x))$. Hence, for $i = 1, \ldots, d$, $|m| = k$:

$$f_m^i = \hat{R}_m^i \ , \ W_m^i = 0. \tag{16}$$

The graph style is feasible for all the types of invariant manifolds, but it is not adapted to the shape of the manifold (that could be folded). It is adequate for parameterizing a center manifold, since there are infinitely many internal resonances. Although power series expansions of center manifolds may be divergent [9], they may provide good local approximations. See Section 4.

3.3.3. *The subgraph style.* In this style, we solve the homological equations to reveal the existence of an invariant submanifold of dimension $\hat{d} < d$, represented as a graph, inside the manifold of dimension d. Let us denote $\hat{\lambda}^x = (\lambda_1, \ldots, \lambda_{\hat{d}})$, and $\hat{m} \in \mathbb{N}^{\hat{d}}$. We assume there are no cross resonances of the $\hat{d}$ eigenvalues: for $i = \hat{d}+1, \ldots, d$ and $|\hat{m}| \geq 2$, $\hat{\lambda}^x \cdot \hat{m} - \lambda_i \neq 0$.

Then, a suitable choice for the parameterization is, for $i = 1, \ldots, d$, $|m| = k$:

$$\begin{cases} f_m^i = 0 \ , \ W_m^i = \dfrac{\hat{R}_m^i}{\hat{\lambda}^x \cdot \hat{m} - \lambda_i}, & \text{if } i > \hat{d},\ m = (\hat{m}, 0); \\ f_m^i = \hat{R}_m^i \ , \ W_m^i = 0, & \text{otherwise.} \end{cases} \tag{17}$$

This style is adequate, for instance, for the parameterization of a center stable manifold, in which the center-manifold is a subgraph.

3.3.4. *Further comments.* Besides the styles described above, there are many other possibilities. For instance, one can consider using partial normal form styles to show the existence of invariant submanifolds by reducing to partial normal forms. The simplest example of this methodology is the mentioned subgraph style.

We emphasize that the lack of uniqueness of the solutions of the invariance equations is suitable for controlling the numerical stability of the computations. A first choice is the length of the eigenvectors generating the linear approximation (which is equivalent to scale the parameterization), which control the growth of the coefficients in the expansions. In normal form styles, one can save the nearly resonant

coefficients to avoid small divisors when solving the homological equations.

If matrices A_1, A_2 are not diagonal the given non-resonance conditions are the same but the solution of the homological equations is harder [17, 50]. Using their triangular reduced forms (e.g. the Jordan normal form), the homological equations are also triangular.

A more refined normal form style can be designed for the case of a 1D parabolic manifold (a 1D center manifold associated to an eigenvalue 1), finding a polynomial normal form for the dynamics [12, 13].

3.4. **Complexity.** Let F be an n-dimensional elementary vector field F of complexity c. We estimate the computational cost of solving the invariance equation (6) up to order k, for a parameterization W of a d-dimensional manifold and the corresponding reduced d-dimensional vector field f.

The computational effort of the step k of the on-line algorithm is concentrated in obtaining the right hand side of the kth order equation (9), because the cost of solving the homological equations (especially in the diagonal case) is negligible. The computation of $[F(W_{<k}(s))]_k$ involves the application of the on-line composition formulas (at order k) of the elementary functions giving F, which has a fix complexity c. In order to avoid extra computations, one has to save all the intermediate steps in the compositions with F. Note that the complexity of computing $[\mathrm{D}W_{<k}(s)f_{<k}(s)]_k = \sum_{l=2}^{k-1} \mathrm{D}W_{k-l+1}(s)f_l(s)$ is related with the style of the parameterization.

Summarizing, the computational cost for solving (6) up to order k is

$$\mathrm{w}_{n,d}(k) \sim C\mathrm{p}_d(k),$$

where the complexity C depends only on the dimensions n, d, the complexity of the vector field, c, and the style of the parameterization. Note that $C \leq (c + nd)$. In the normal form style, with polynomial normal form, $C = c$. In the graph style, $C = c + (n - d)d$.

3.5. **Error estimates.** After the computation of an (order k) approximation of the parameterization $z = W(s)$ of the invariant manifold of the vector field $\dot{z} = F(z)$, and the corresponding reduced vector field, $\dot{s} = f(s)$, numerical tests have to be performed in order to estimate the quality of the approximation. In particular, a *fundamental domain* in which the approximation is sufficiently accurate has to be determined. For invariant manifolds in which the dynamics is asymptotically stable (in forward or backward time), this is the first step to globalize the manifold, that is to extend the manifold far away from the equilibrium point [47], propagating the fundamental domain. But

for invariant manifolds in which the dynamics is "practically" stable (like the center manifolds in Hamiltonian dynamics), the dynamics remains in this fundamental domain and has to be analyzed. In order to numerically estimate a fundamental domain, several error estimates can be considered.

From an initial condition $z_0 = W(s_0)$ on the (approximate) invariant manifold, let $z(t)$ and $s(t)$ be the (numerical) solutions of the Cauchy problems $\dot{z} = F(z)$, $z(0) = z_0$ and $\dot{s} = f(s)$, $s(0) = s_0$, respectively. We define the *error in the invariance equation* as

$$e_I(t, s_0) = \|F(W(s(t))) - \mathrm{D}W(s(t))f(s(t))\|_\infty \ ,$$

and the *error in the orbit* as

$$e_O(t, s_0) = \|W(s(t)) - z(t)\|_\infty.$$

For a small lapse of time, and if s_0 is sufficiently close to 0, $e_I(t, s_0)$ and $e_O(t, s_0)$ should be proportional to $|s_0|^{k+1}$.

If the vector field $\dot{z} = F(z)$ has a first integral H, one can also consider the *error in the (reduced) first integral* $H \circ W$ as

$$e_H(t, s_0) = |H(W(s(t))) - H(W(s(0)))|.$$

Observe that testing the preservation of the first integral is a test for the numerical integrator as well as for the quality of the approximation of the invariant object.

4. An example: center manifolds in the Restricted Three Body Problem

In this section, we present an example of computation of invariant manifold: the center manifold of a collinear equilibrium point in the Restricted Three Body Problem (RTBP). The computation of this 4D manifold in a 6D phase space has been performed e.g. in [30, 31, 44, 43, 60]. We reach higher orders that let us to explore the asymptotic character of the expansions and their range of validity.

Our aim is also to show how our methodology can perform very efficient computations of invariant manifolds in this and many other non-trivial problems.

4.1. **A brief description of the RTBP.**

The RTBP deals with the motion of a massless body under the gravitational forces induced by two punctual masses, usually called primaries, which evolve in circular Keplerian motion around the center of mass. In a rotating frame, and in suitable non-dimensional units, the equations of the motion

of the massless body are determined by the three degrees of freedom Hamiltonian

$$H(x,y,z,p_x,p_y,p_z) = \frac{1}{2}(p_x^2+p_y^2+p_z^2) + y\ p_x - x\ p_y + V(x,y,z) \tag{18}$$

where V is the gravitational potential

$$V(x,y,z) = -\frac{1-\mu}{r_1} - \frac{\mu}{r_2}. \tag{19}$$

In (19), $r_1 = \sqrt{(x-\mu)^2+y^2+z^2}$ and $r_2 = \sqrt{(x-\mu+1)^2+y^2+z^2}$ are the distances of the body to the primaries, whose masses are $1-\mu$ and $\mu \in]0, \frac{1}{2}]$, respectively. Hence, the equations of motion are:

(20)

$$\begin{aligned}
\dot{x} &= p_x + y\ , & \dot{p}_x &= p_y - \frac{1-\mu}{r_1^3}(x-\mu) - \frac{\mu}{r_2^3}(x-\mu+1)\ ,\\
\dot{y} &= p_y - x\ , & \dot{p}_y &= -p_x - \frac{1-\mu}{r_1^3}y - \frac{\mu}{r_2^3}y\ ,\\
\dot{z} &= p_z\ , & \dot{p}_z &= -\frac{1-\mu}{r_1^3}z - \frac{\mu}{r_2^3}z\ .
\end{aligned}$$

The equations of the RTBP are elementary. In fact, the complexity of the vector field (20) is $c = 6.5$.

4.2. **Computation of center manifolds in the RTBP.** We have applied our algorithms to compute the expansions of the invariant manifolds of the equilibrium points of the RTBP. Once one has chosen the mass parameter μ, the equilibrium point L_p ($p = 1,2,3,4,5$), the dimension of the manifold d, the order of its expansion k, and the (complex) eigenvectors expanding the tangent space of the manifold at the equilibrium point (the natural choice is to take their lengths as the distance of the equilibrium point to the nearest primary), the computer program produces the coefficients of the (complex) power series expansions of the graph style parameterization of the manifold, W. If $(s_1,s_2,s_3,s_4,s_5,s_6)$ are coordinates in which the linearization around L_p takes diagonal form, then the parameterization $W(s_1,s_2,s_3,s_4)$ represents a graph of (s_5,s_6) with respect to (s_1,s_2,s_3,s_4).

The mass parameter we have chosen to illustrate the methodology is $\mu \simeq 0.0121506$, which corresponds to the Earth-Moon system. The collinear points L_1, L_2, L_3 are of center×center×saddle type, that is there are two couples of complex eigenvalues of modulus 1, and one couple of real eigenvalues. Hence, collinear points have attached 4D center manifolds, 5D center stable and center unstable manifolds and

	Mac I			Mac II			Mac III		
k	product	graph	ratio	product	graph	ratio	product	graph	ratio
10	4.352e–04	7.790e–03	17.90	3.841e–04	6.090e–03	15.86	2.617e–04	4.716e–03	18.02
20	2.533e–02	4.048e–01	15.98	2.054e–02	3.039e–01	14.80	1.428e–02	2.329e–01	16.30
30	3.582e–01	5.497e+00	15.34	2.650e–01	3.819e+00	14.41	1.888e–01	2.988e+00	15.83
40	2.590e+00	3.921e+01	15.14	1.813e+00	2.641e+01	14.57	1.308e+00	2.047e+01	15.66
50	1.259e+01	1.900e+02	15.09	8.500e+00	1.243e+02	14.62	6.192e+00	9.643e+01	15.57
60	4.708e+01	7.104e+02	15.08	3.104e+01	4.555e+02	14.67	2.274e+01	3.536e+02	15.55
70	1.460e+02	2.207e+03	15.12	9.481e+01	1.397e+03	14.73	6.974e+01	1.083e+03	15.53

TABLE 2. Benchmark of execution time (in seconds) for computing the graph style parameterization of the center manifold of the L_1 point of the RTBP of the Earth-Moon system.

k	Lie series	Graph transform	Graph style + AD
8	0m 0.085s	0m 0.057s	0m 0.005s
16	0m 3.876s	0m 2.943s	0m 0.084s
24	2m 10.251s	1m 13.965s	0m 0.790s
32	33m 22.000s	14m 35.475s	0m 4.764s

TABLE 3. Comparison of several methods for center manifold reduction. Results for Lie series and direct graph transform have been taken from [26], which were obtained with an Intel(R) Core(TM)2 Quad CPU at 2.83GHz, similar to Mac III.

1D stable and unstable manifolds. The triangular points are linearly stable, and will be not considered here.

From now on, we report the study of the center manifold of the L_1 equilibrium point, whose energy is $H_{L_1} \simeq -1.594171$. In Table 2 we show some benchmarks of execution time. We include the ratio between the total time of computation and the time of the truncated product of complex power series (up to order k), observing that in all cases this is relatively close to the theoretical (asymptotic) ratio $(6.5+2*4) = 14.5$.

The center manifold has been computed in the literature by using partial normal forms based on the Lie series method (see e.g. [31, 44, 43, 60]) and, more recently, using the Graph transform method [26]. We also consider automatic differentiation methods in the design of the algorithms. See Table 3 for a benchmark of the speed using different methodologies, for computations up to order 32.

4.3. **Growth of the coefficients of the center manifold.** Even though the center manifold is not analytic, the coefficients of its power

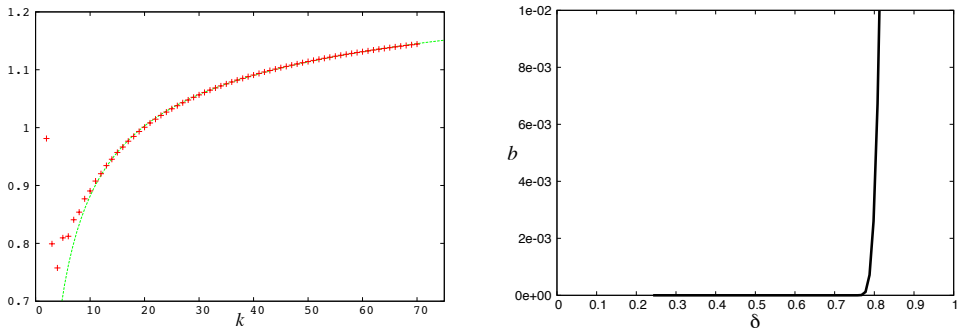


FIGURE 1. Mild growth of the coefficients (left) implies quite sharp behavior of the asymptotic expansion (right).

series expansions can grow in a mild way [42], less than Gevrey. For each $k > 0$, let $\ell_1(k)$ be the maximum of the ℓ_1 norms of the components of the order-k term W_k in the graph style parameterization. Figure 1 reveals a growth like

$$\ell_1(k) \sim \ell(k) = A\lambda^k (\log k)^{ck} \tag{21}$$

where $a = \log A$,$b = \log \lambda$ and c are estimated by

$$a = -1.25 \pm 0.05 \ , \ b = -0.212 \pm 0.008 \ , \ c = 0.252 \pm 0.005$$

(fitting the norms from $k = 10$ to $k = 70$). Similar behavior is observed in the growth of the coefficients of the expansions of the reduced vector field.

Assume that, for δ small enough, the expansion of $W(s)$ for $|s|_\infty \leq \delta$ is asymptotic, that is:

$$|W_{\leq k}(s) - W(s)|_\infty \leq \ell_1(k+1)\delta^{k+1} \sim \varepsilon(\delta, k) = \ell(k+1)\delta^{k+1} \ . \tag{22}$$

Following [58], the best bound $b(\delta)$ for the error in the approximation of $W(s)$ by $W_{\leq k}(s)$ in the box $|s|_\infty \leq \delta$ is obtained taking $k = k(\delta)$ minimizing $\varepsilon(\delta, k)$. In the present case,

$$\begin{aligned} k(\delta) &= \frac{1}{e} \exp\left((\lambda\delta)^{-\frac{1}{c}} \right) - 1 \ , \\ b(\delta) &= \varepsilon(\delta, k(\delta)) = A \exp\left(-c(\lambda\delta)^{\frac{1}{c}} (k(\delta) + 1) \right) \ . \end{aligned}$$

Hence, the mild growth of the coefficients of the expansions explains the behavior of the (best) error of the asymptotic approximation, observed in Figure 1, right. This behavior is much sharper than the typical exponentially small estimates arising from Gevrey expansions [58].

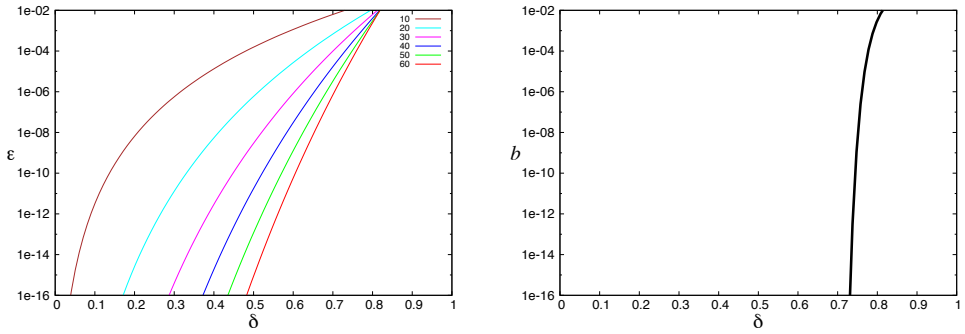


FIGURE 2. Error estimates as a function of δ for different order of approximations (left), and best error estimates (in log-scale).

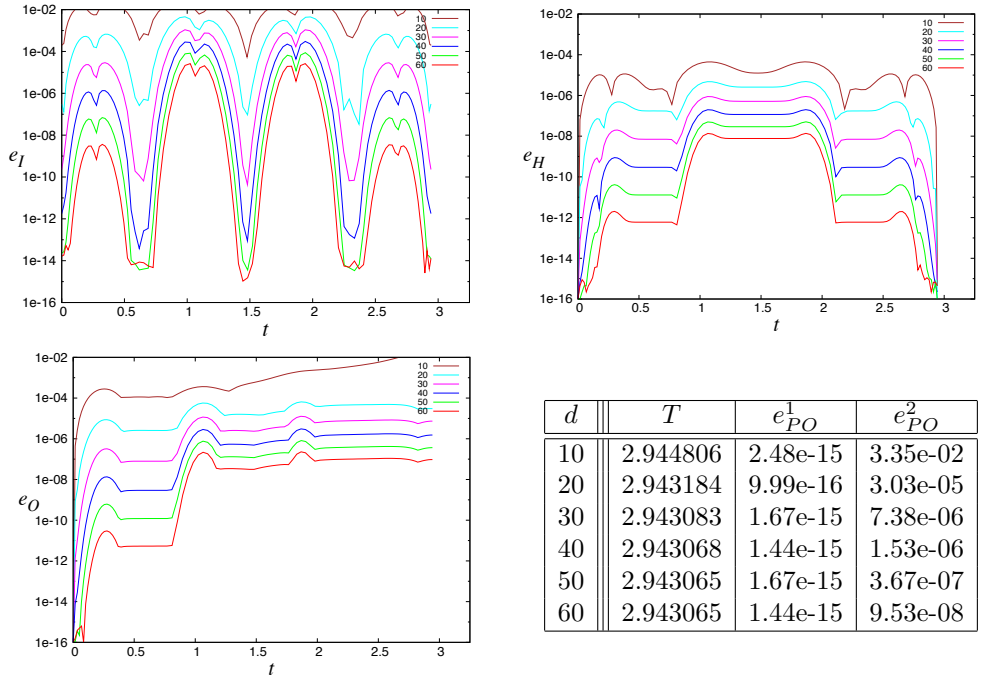
From the error bounds one can also give estimations of the validity domains of the expansions, depending on both the order of the approximations and the size of the domains. See Figure 2.

4.4. **Dynamics on an energy level in the center manifold.** Since the present example *is* Hamiltonian, the 4D reduced vector field on the center manifold, f, is Hamiltonian, with Hamiltonian $H \circ W$ (with respect to the push forward of the symplectic form, that is the symplectic form on the center manifold).

Since $H \circ W$ is a conserved quantity of the reduced vector field, the standard practice [30, 43, 44, 60] is to study its dynamics by using the Poincaré section trick on different Hamiltonian levels to obtain a collection of 2D phase portraits. The Poincaré section is $\{z = 0\}$, which corresponds to $s_4 = 0$ in the $s = (s_1, s_2, s_3, s_4)$ coordinates of the parameterization of the center manifold.

4.4.1. *The boundary of the center manifold.* The boundary of the intersection of the 4D center manifold with the 5D Poincaré section in a given 5D energy level $H > H_{L_1}$ in the 6D phase space is a closed curve tangent to the vector field: a planar Lyapunov periodic orbit. It is easy to continue planar Lyapunov orbits numerically with respect to the energy H.

The error estimates for a planar Lyapunov orbit provide upper bounds for the errors in the semi-local approximations for the points inside the center manifold for the corresponding energy level. This is useful to obtain estimates of the truncation order of the expansions to obtain accurate approximations of orbits in the center manifold for such an energy level.



d	T	e^1_{PO}	e^2_{PO}
10	2.944806	2.48e-15	3.35e-02
20	2.943184	9.99e-16	3.03e-05
30	2.943083	1.67e-15	7.38e-06
40	2.943068	1.44e-15	1.53e-06
50	2.943065	1.67e-15	3.67e-07
60	2.943065	1.44e-15	9.53e-08

FIGURE 3. Errors for the planar Lyapunov orbit ($H = -1.565$)

Figures 3 shows, in logarithmic scale, the error estimates e_I,e_O,e_H (see Section 3.5) for the planar Lyapunov orbit at the Hamiltonian level $H = -1.565$, which is relatively far from the Hamiltonian level $H_{L_1} \simeq -1.594171$ of the L_1 equilibrium point. The estimates have been produced for different orders of the expansions of the center manifold (from 10 to 60), in order to compare the quality of the expansions and the numerical computations. We note that errors e_I and e_H fluctuate with respect to time, and that orbital error e_O is essentially increasing in time, due to the hyperbolic directions, transversal to the center manifold. Figure 3 also shows the estimated period T of the periodic orbit, and the errors e^1_{PO} and e^2_{PO} in the return map on the manifold and on the whole phase space, respectively.

4.4.2. *Computation of Poincaré maps and their phase portraits.* One can obtain a global and synthetic view of the dynamics on the center manifold in a given energy level H by computing orbits of the Poincaré map. Two methods can be used: reduction and projection.

The method of reduction consists of computing the orbits along the center manifold by numerically integrating the reduced vector field f, using e.g. a Runge-Kutta method. We have used a Runge-Kutta method of order 7-8 with automatic step size control to obtain a local

error of 10^{-15}. At each step of the numerical integration method the *reduced* vector field has to be evaluated at 13 different 4D points, hence evaluating the power series giving f. These evaluations are computationally costly, and a single return map can be quite time-consuming, depending on the energy level (and, hence, of the order of the expansions). For instance, for $H = -1.580$, using expansions of order 20, each return map takes less than one second. For $H = -1.565$, using expansions of order 50, each return map takes around 45 s.

The method of projection is based on the fact that the computed manifold is (approximately) invariant. Hence, once one has taken a point in the Poincaré section, in the center manifold and in a given energy level, the return map is computed by integrating the *full* 6D vector field. For the numerical integration, we have used a Taylor method of order 18 with automatic step size control to obtain a local error of 10^{-15} [61]. Due to numerical instabilities produced by the hyperbolic directions, the error estimates tend to grow. Hence, at each return map we project the computed return point on the center manifold (and on the same energy level). With this method, one computes hundreds of iterations of the Poincaré map in a few seconds.

We use the method of reduction for a few orbits, in order to estimate error bounds and the quality of the approximations, and we use the method of projection in the simulations. If necessary, computing several iterations of the Poincaré map for a collection of points in the center manifold can be paralleled in a cluster of computers. Figures 4, 5 show the phase portraits of the Poincaré maps on different energy levels on the center manifold, both in center manifold coordinates (left) and synodic coordinates (right), in which we also scale the size of the Moon. The figures visualize the typical features of 2D area preserving maps: fixed points and periodic points (corresponding to periodic orbits), islands of invariant curves (corresponding to invariant tori), separatrizes, etc. The boundaries of the domains correspond to planar Lyapunov orbits, and the fixed point in the $s_2 = 0$ (or $y = 0$) axis correspond to vertical Lyapunov orbits.

In Figure 4, the energy levels are relatively close to H_{L_1}. For $H = -1.590$ the planar Lyapunov orbit is linearly stable (on the center manifold), but it has bifurcated into an unstable planar Lyapunov periodic orbit for $H = -1.580$, giving rise to two linearly stable halo orbits (on the center manifold). The invariant curves around the vertical Lyapunov orbit and the halo orbits correspond to the Lissajous orbits and the quasi-halo orbits respectively.

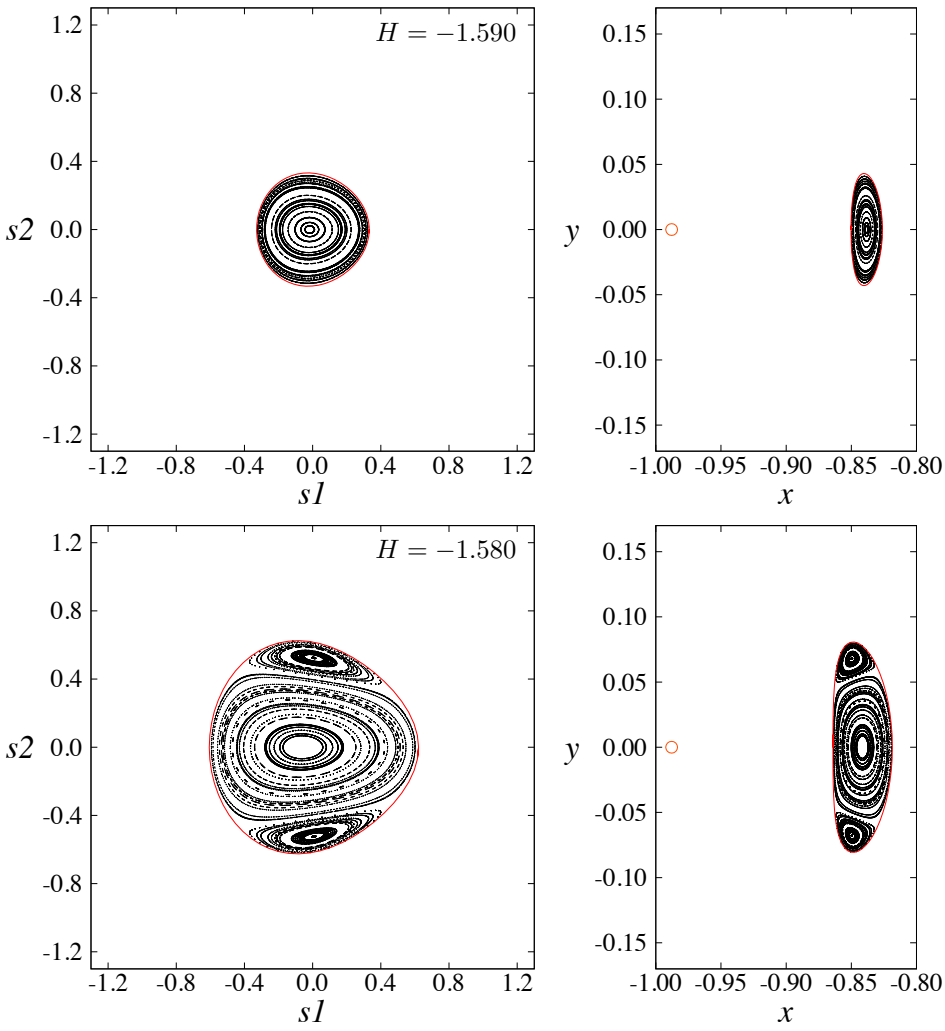


FIGURE 4. Dynamics on the center manifold, close to L_1

In Figure 5, the energy levels are close to the boundaries of the validity region of the expansions. See Figure 3 for estimates of the errors in the boundary of the center manifold, for $H = -1.565$.

4.4.3. *Numerical chaos.* In the examples described above, the measure of the chaotic sea (the complement of the regular motion) is very small. One can argue that the measure of the chaotic sea increases when the energy level increases. But then numerical errors also increase and the simulations are not reliable in some parts of the validity domain.

For instance, Figure 6 shows the computed phase portraits of the dynamics for energy values $H = -1.555$ and $H = -1.545$. Computations of orbits close to vertical and halo orbits are quite reliable. But

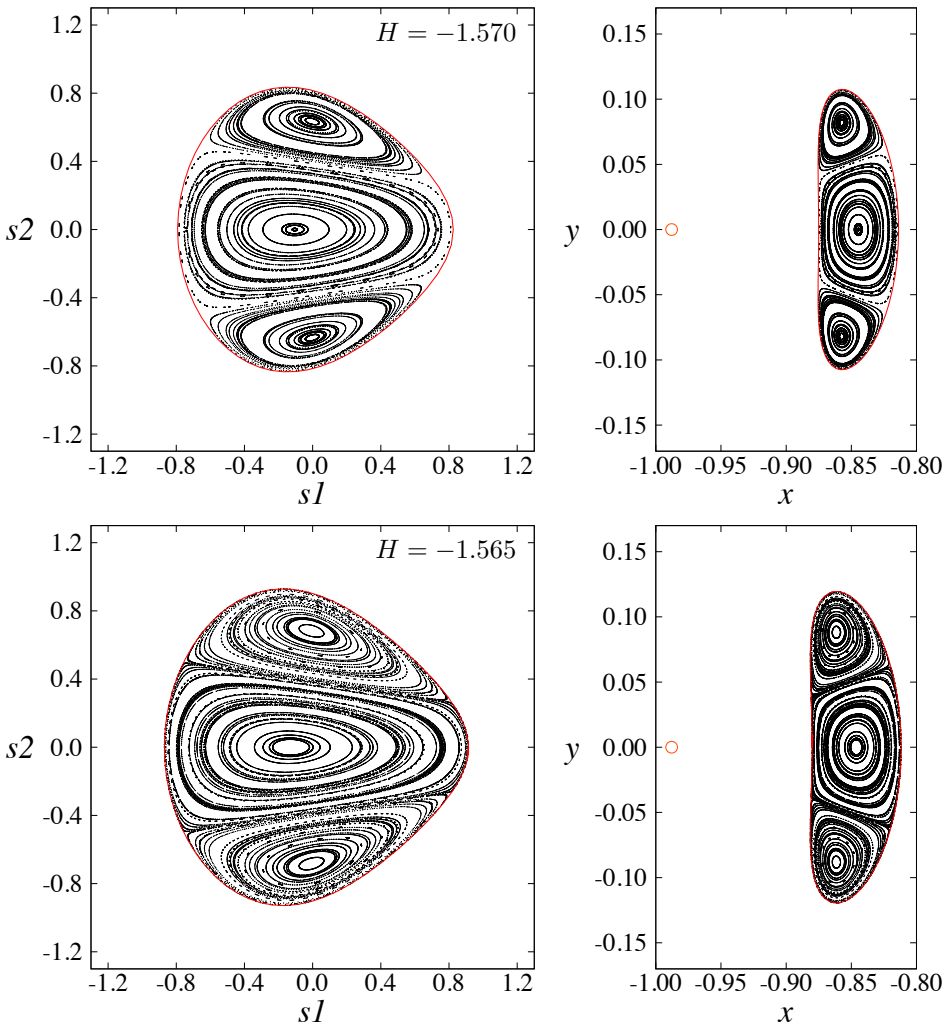


FIGURE 5. Dynamics on the center manifold, far to L_1

a non-reliable large chaotic sea is also observed. We note that orbits far away from the vertical and halo orbits travel relatively close to the planar Lyapunov orbit, the boundary of the center manifold in which the accuracy of the approximations is worst.

4.5. **Periodic orbits.** From the phase portraits of the Poincaré maps, one obtains insight on the dynamics on the center manifold. This information can be e.g. used to compute periodic orbits inside the center manifold. Here we only report some computations of periodic orbits for $H = -1.565$.

Figures 7,8 show, in logarithmic scale, the error estimates e_I,e_O,e_H for the vertical Lyapunov and halo orbits respectively (see Figure 3 for

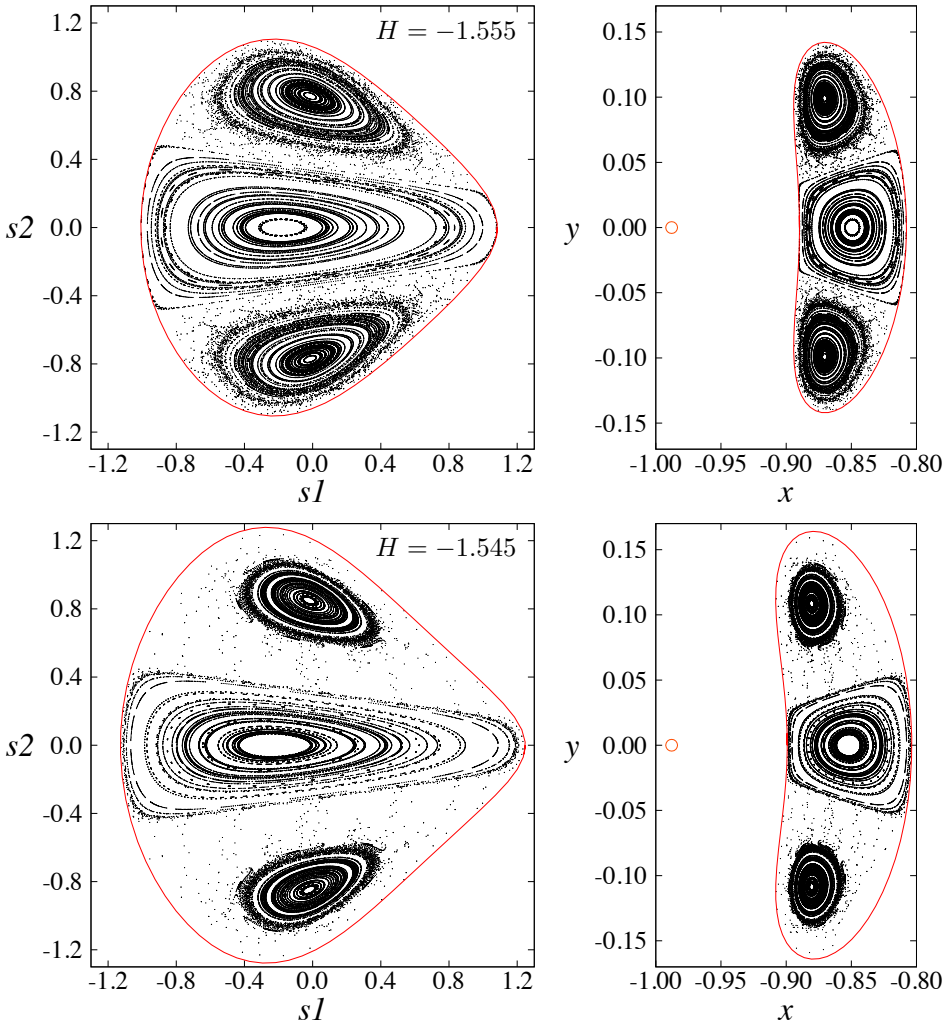
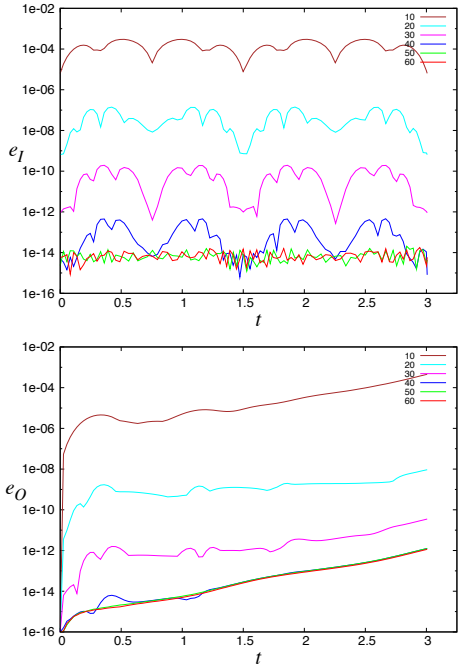


FIGURE 6. Numerical chaotic seas on the center manifold

the planar Lyapunov orbit). The initial conditions have been computed by using the Newton method from initial estimates obtained from the phase portrait in Figure 9, (a). These orbits are drawn in Figure 9, (b). One can also perform similar computations for higher periods, for periodic orbits around the vertical Lyapunov orbit, (see Figure 9, (c)) and around the halo orbit (see Figure 9, (d)).

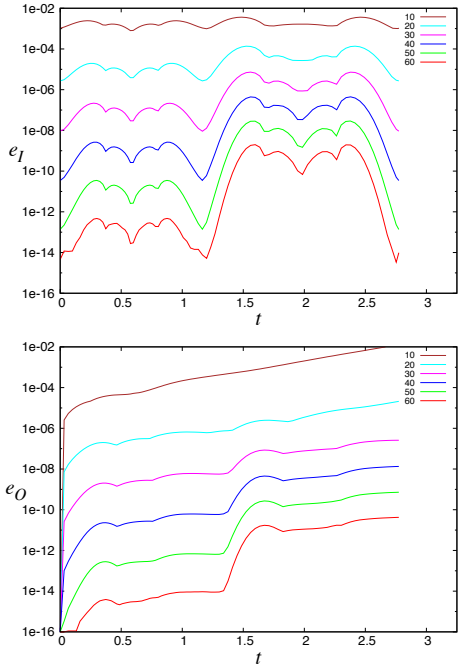
APPENDIX A. ON TWO IMPLEMENTATIONS OF ALGEBRAIC MANIPULATORS OF DENSE MULTIVARIATE POWER SERIES

In this section we briefly explain some details of our algebraic manipulator of *dense* multivariate power series using AD tools based on



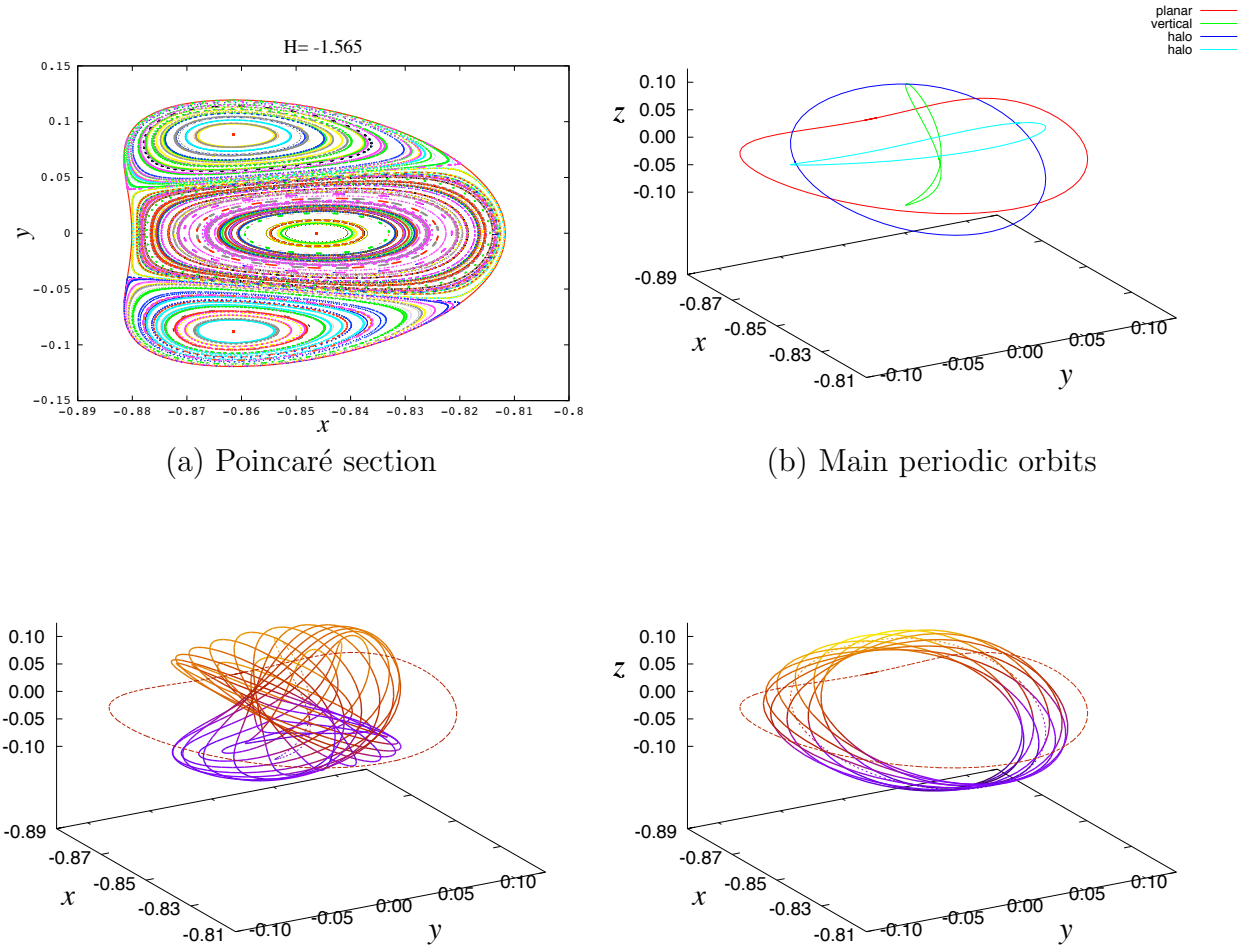
d	T	e^1_{PO}	e^2_{PO}
10	3.006224	9.99e-16	4.54e-04
20	3.006243	8.88e-16	9.33e-09
30	3.006243	7.77e-16	3.51e-11
40	3.006243	3.89e-16	1.25e-12
50	3.006243	4.37e-16	1.27e-12
60	3.006243	4.37e-16	1.14e-12

FIGURE 7. Errors for the vertical Lyapunov orbit ($H = -1.565$)



d	T	e^1_{PO}	e^2_{PO}
10	2.775478	1.22e-15	1.31e-02
20	2.774291	2.78e-16	2.11e-05
30	2.774274	2.54e-14	2.59e-07
40	2.774274	2.66e-15	1.34e-08
50	2.774274	1.33e-15	7.24e-10
60	2.774274	7.77e-16	4.23e-11

FIGURE 8. Errors for the halo orbits ($H = -1.565$)



(a) Poincaré section

(b) Main periodic orbits

(c) P.o. around the vertical orbit, of r.n. -1:18

(d) P.o. around a halo orbit, of r.n. 1:9

FIGURE 9. Periodic orbits on the energy level $H = -1.565$.

on-line formulas. This is the core of the programs to compute invariant manifolds and normal forms. We have implemented the algorithms using the `C` programming language.

A.1. **Data structures.** The basis of the computer algebra system is the data structure `homog` to encode homogeneous polynomials. The data structure `serie` for truncated power series is just the collection of homogeneous polynomials. The scalar coefficients are either double precision real numbers (`double` type, of 8 bytes) or double precision

complex numbers (`complex` type, of 16 bytes), but it is not difficult to change to other types.

We have implemented two versions of the computer algebra system, both of them based on the following recursive scheme: a homogeneous polynomial $f_k(x)$ of d variables $x = (x_1, \ldots, x_d)$ of order k is a combination of $(k+1)$ homogeneous polynomials of the first $(d-1)$ variables $\hat{x} = (x_1, \ldots, x_{d-1})$ of degrees $k, k-1, \ldots, 0$:

$$f_k(x) = f_k^d(\hat{x}) + f_{k-1}^d(\hat{x})x_d + \cdots + f_0^d(\hat{x})x_d^k \ . \tag{23}$$

In the *vector implementation*, the data structure `homog` is a tuple (or structure, in `C` terminology) whose fields are two bytes giving the number of variables (d) and the degree (k), respectively, and a pointer addressing the array storing the coefficients of the homogeneous polynomial. The coefficients are ordered in a graded reverse lexicographical ordering of the exponents of the monomials $(m_1, \ldots, m_d)$, that corresponds to the recursive scheme (23). For example, the 10 coefficients of a homogenous 3-variate polynomial of order 3 are ordered following the scheme: $x_1^3, x_1^2x_2, x_1x_2^2, x_2^3, x_1^2x_3, x_1x_2x_3, x_2^2x_3, x_1x_3^2, x_2x_3^2, x_3^3$.

In the *tree implementation*, the data structure `homog` includes as a field a pointer addressing an array whose components are of data type `homog`, corresponding to the $(k+1)$ homogeneous $(d-1)$-variate polynomials in (23). This approach simplifies the combinatorics of positioning the coefficients of a homogeneous polynomial that is used in the vector implementation, simplifying the codes. On the other side, the computer memory to store the coefficients increases in a factor about 5 with respect to the vector implementation.

Thanks to the recursive scheme, we avoid the use of indexing routines to locate coefficients, an approach that is used e.g. in [1, 57, 3], and the symbolic manipulator works for any number of variables and any truncation degree.

A.2. **Benchmark for the truncated product.** We report here the benchmarks for the truncated product of multivariate power series.

Table 4 shows several execution times in machine Mac I for computing the truncated product of polynomials of 4 and 6 variables, whose coefficients are `double` type. The speed is measured in megaflops, i.e. millions of operations per second. We define the overhead as the quotient between the execution time of the truncated product and the execution time of computing $p = \mathrm{p}_d(k)$ operations (multiplications and additions).

We observe that execution time is sublinear in the number $p = \mathrm{p}_d(k)$ of operations, or subquadratic in the number of coefficients $n = \mathrm{n}_d(k)$

$d=4$			vector			tree			TRIP	CS
k	n	p	time (s)	Mflops	over.	time (s)	Mflops	over.	time (s)	time (s)
10	1001	43758	2.740e−04	159.7	2.88	2.930e−04	149.3	3.05	NA	3.665e−04
20	10626	3108105	1.364e−02	227.9	2.50	1.364e−02	227.9	2.50	NA	1.906e−02
30	46376	48903492	1.700e−01	287.7	1.89	1.900e−01	257.4	2.38	2.000e−01	2.529e−01
40	135751	377348994	1.140e+00	331.0	1.75	1.120e+00	336.9	1.70	1.630e+00	1.767e+00
50	316251	1916797311	5.150e+00	372.2	1.57	5.050e+00	379.6	1.53	7.660e+00	1.032e+01
60	635376	7392009768	1.829e+01	404.2	1.45	1.793e+01	412.3	1.42	2.928e+01	5.371e+01
70	1150626	23446881315	5.456e+01	429.7	1.37	5.335e+01	439.5	1.34	9.257e+01	2.098e+02
80	1929501	64276915527	1.426e+02	450.8	1.31	1.393e+02	461.5	1.27	2.533e+02	6.623e+02
90	3049501	157366449604	3.361e+02	468.2	1.26	3.282e+02	479.5	1.23	6.229e+02	1.765e+03
100	4598126	352025629371	7.326e+02	480.5	1.23	7.093e+02	496.3	1.19	1.408e+03	4.195e+03

$d=6$			vector			tree			TRIP	AJ
k	n	p	time (s)	Mflops	over.	time (s)	Mflops	over.	time (s)	time (s)
10	8008	646646	4.421e−03	146.3	3.49	4.808e−03	134.5	4.17	NA	4.451e−02
20	230230	225792840	1.241e+00	181.9	3.17	1.310e+00	172.4	3.36	9.940e−01	1.722e+01
30	1947792	11058116888	4.915e+01	225.0	2.62	5.043e+01	219.3	2.69	4.408e+01	9.161e+02
40	9366819	206379406870	7.769e+02	265.7	2.23	7.808e+02	264.3	2.23	8.367e+02	1.926e+04
50	32468436	2160153123141	7.178e+03	301.0	1.96	7.183e+03	300.7	1.96	NA	NA

TABLE 4. Timings of 4-variate and 6-variate truncated products.

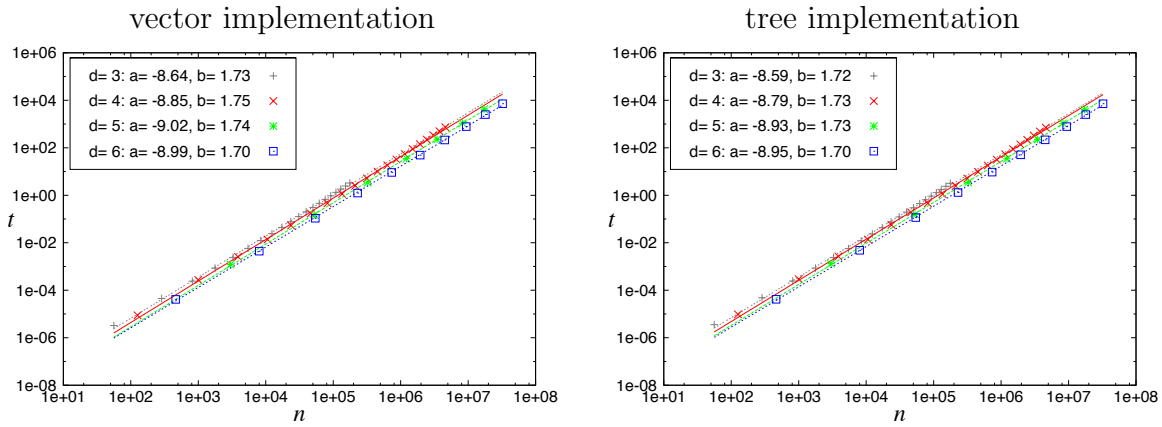


FIGURE 10. Execution time (in seconds) of the truncated product as a function of the number of coefficients per polynomialm, in log-log representation.

per polynomial. We have fit the execution times with functions of the form $t(n) = A\ n^b$ (where $a = \log_{10} A$), and the results are shown in Figure 10, using log-log scale. Observe that exponents b are around 1.72, rather than the theoretical estimate 2. This is mainly due to the fact that addressing schemes take advantage of the data structures encoding the power series, and in particular on the lexicographical order of the coefficients or the tree distribution of the coefficients.

The two recursive implementations produce similar results. Table 4 also includes the timings obtained with the package TRIP [1, 28], and two ad hoc implementations for 4 [57] and 6 [3] variables. Our results are similar to TRIP, which uses huge addressing tables to perform multiplication of polynomials, and whose codes were compiled with `icc` in a twin machine working under Mac OS X 10.4.1 (Tiger).

References

[1] `http://www.imcce.fr/Equipes/ASD/trip/trip.php`.
[2] `http://piranha.tuxfamily.org`.
[3] `http://www.maia.ub.es/~angel/soft.html`.
[4] `http://www.autodiff.org`.
[5] `http://www.coin-or.org/projects/ADOL-C.xml`.
[6] `http://www.bt.pa.msu.edu/index\_cosy.htm`.
[7] `http://www2.math.uni-wuppertal.de/~xsc/xsc/cxsc.html`.
[8] `http://www2.math.uni-wuppertal.de/~xsc/xsc/cxsc/software/taylor/ldim2taylor.cpp`.
[9] D. V. Anosov, S. Kh. Aranson, V. I. Arnold, I. U. Bronshtein, Yu. S. Il'yashenko, and V. Z. Grines. *Ordinary differential equations and smooth dynamical systems*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.
[10] V. I. Arnol′d. *Geometrical methods in the theory of ordinary differential equations*, volume 250 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Science]*. Springer-Verlag, New York, 1983. Translated from the Russian by Joseph Szücs, Translation edited by Mark Levi.
[11] V. I. Arnol′d. *Geometrical methods in the theory of ordinary differential equations*, volume 250 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Science]*. Springer-Verlag, New York, 1983. Translated from the Russian by Joseph Szücs, Translation edited by Mark Levi.
[12] I. Baldomá and A. Haro. One dimensional invariant manifolds of Gevrey type in real-analytic maps. *Discrete Contin. Dyn. Syst. Ser. B*, 10(2-3):295–322, 2008.
[13] Inmaculada Baldomá, Ernest Fontich, Rafael de la Llave, and Pau Martín. The parameterization method for one-dimensional invariant manifolds of higher dimensional parabolic fixed points. *Discrete Contin. Dyn. Syst.*, 17(4):835–865, 2007.
[14] Roberto Barrio. Sensitivity analysis of ODEs/DAEs using the Taylor series method. *SIAM J. Sci. Comput.*, 27(6):1929–1947 (electronic), 2006.
[15] Martin Berz. Algorithms for higher derivatives in many variables with applications to beam physics. In *Automatic differentiation of algorithms (Breckenridge, CO, 1991)*, pages 147–156. SIAM, Philadelphia, PA, 1991.
[16] Wolf-Jürgen Beyn, Alan Champneys, Eusebius Doedel, Willy Govaerts, Yuri A. Kuznetsov, and Björn Sandstede. Numerical continuation, and computation of normal forms. In *Handbook of dynamical systems, Vol. 2*, pages 149–219. North-Holland, Amsterdam, 2002.

[17] Wolf-Jürgen Beyn and Winfried Kleß. Numerical Taylor expansions of invariant manifolds in large dynamical systems. *Numer. Math.*, 80(1):1–38, 1998.

[18] Francesco Biscani. *Design and implementation of a modern algebraic manipulator for Celestial Mechanics*. PhD thesis, Dipartimento di Astronomia, Università di Padova, 2008. URL: `http://paduaresearch.cab.unipd.it/848/1/thesis.pdf`.

[19] C. Bischof, G. Corliss, and Griewank. A. Structured second- and higher-order derivatives through univariate taylor series. *Optim. Methods Software*, 2, 1993.

[20] F. Blomquist, W. Hofschuster, and W. Krämer. Real and complex taylor arithmetic in `c-xsc`. Preprint.

[21] Marco Bodrato. Towards optimal Toom-Cook multiplication for univariate and multivariate polynomials in characteristic 2 and 0. In Claude Carlet and Berk Sunar, editors, *WAIFI 2007 proceedings*, volume 4547 of *LNCS*, pages 116–133. Springer, June 2007. URL: `http://bodrato.it/papers/#WAIFI2007`.

[22] R. P. Brent and H. T. Kung. Fast algorithms for manipulating formal power series. *J. Assoc. Comput. Mach.*, 25(4):581–595, 1978.

[23] Xavier Cabré, Ernest Fontich, and Rafael de la Llave. The parameterization method for invariant manifolds. I. Manifolds associated to non-resonant subspaces. *Indiana Univ. Math. J.*, 52(2):283–328, 2003.

[24] Shannon Coffey, André Deprit, Etienne Deprit, Liam Healy, and Bruce R. Miller. A toolbox for nonlinear dynamics. In Kenneth R. Meyer and Dieter S. Schmidt, editors, *Computer aided proofs in analysis. (Cincinnati, OH, 1989)*, volume 28 of *IMA Vol. Math. Appl.*, pages 97–115. Springer-Verlag, New York, 1991.

[25] Stephen A. Cook. *On the minimum computation time of functions*. PhD thesis, Department of Mathematics, Harvard University, 1966. URL: `http://cr.yp.to/bib/entries.html#1966/cook`.

[26] Ariadna Farrés and Àngel Jorba. On the high order approximation of the centre manifold for odes. *Discrete Contin. Dyn. Syst. Ser. B*, 14(3):977–1000, 2010.

[27] Frederic Gabern, Wang S. Koon, Jerrold E. Marsden, and Shane D. Ross. Theory and computation of non-RRKM lifetime distributions and rates in chemical systems with three or more degrees of freedom. *Phys. D*, 211(3-4):391–406, 2005.

[28] Mickaël Gastineau. Private communication.

[29] Mickaël Gastineau and Jacques Laskar. Development of trip: Fast sparse multivariate polynomial multiplication using burst tries. In *International Conference on Computational Science (2)*, pages 446–453, 2006.

[30] G. Gómez, À. Jorba, J. Masdemont, and C. Simó. Study refinement of semi-analytical halo orbit theory. Technical report, European Space Agency, 1991.

[31] G. Gómez, À. Jorba, C. Simó, and J. Masdemont. *Dynamics and mission design near libration points. Vol. III*, volume 4 of *World Scientific Monograph Series in Mathematics*. World Scientific Publishing Co. Inc., River Edge, NJ, 2001. Advanced methods for collinear points.

[32] G. Gómez, W. S. Koon, M. W. Lo, J. E. Marsden, J. Masdemont, and S. D. Ross. Connecting orbits and invariant manifolds in the spatial restricted three-body problem. *Nonlinearity*, 17(5):1571–1606, 2004.

[33] Pere Gomis-Porqueras and Àlex Haro. A geometric description of a macroeconomic model with a center manifold. *J. Econom. Dynam. Control*, 33(6):1217–1235, 2009.

[34] Andreas Griewank, David Juedes, and Jean Utke. Algorithm 755: ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM Transactions on Mathematical Software*, 22(2):131–167, 1996.

[35] Andreas Griewank, Jean Utke, and Andrea Walther. Evaluating higher derivative tensors by forward propagation of univariate Taylor series. *Math. Comp.*, 69(231):1117–1130, 2000.

[36] Andreas Griewank and Andrea Walther. *Evaluating derivatives*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2008. Principles and techniques of algorithmic differentiation.

[37] John Guckenheimer and Philip Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1990. Revised and corrected reprint of the 1983 original.

[38] John Guckenheimer and Brian Meloon. Computing periodic orbits and their bifurcations with automatic differentiation. *SIAM J. Sci. Comput.*, 22(3):951–985 (electronic), 2000.

[39] A. Haro. Center and center-(un)stable manifolds of elliptic-hyperbolic fixed points of 4D-symplectic maps. An example: the Froeschlé map. In *Hamiltonian systems with three or more degrees of freedom (S'Agaró, 1995)*, volume 533 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 403–407. Kluwer Acad. Publ., Dordrecht, 1999.

[40] Àlex Haro. Automatic differentiation tools for the computation of invariant manifolds. `http://www.maia.ub.es/dsg/2008/0806haro.pdf`, 2008. Lectures in the Advanced School on Specific Algebraic Manipulators (Barcelona, September 12-15, 2007).

[41] Ale Jan Homburg. Invariant manifolds near hyperbolic fixed points. *J. Difference Equ. Appl.*, 12(10):1057–1068, 2006.

[42] A. Jorba and R. de la Llave. Regularity properties of center manifolds and applications. Manuscript.

[43] Àngel Jorba. A methodology for the numerical computation of normal forms, centre manifolds and first integrals of Hamiltonian systems. *Experiment. Math.*, 8(2):155–195, 1999.

[44] Àngel Jorba and Josep Masdemont. Dynamics in the center manifold of the collinear points of the restricted three body problem. *Phys. D*, 132(1-2):189–213, 1999.

[45] A. Karatsuba and Ofman Yu. Multiplication of many-digital numbers by automatic computers. *Dokl. Akad. Nauk SSSR*, 145:293–294, 1962. Translation in Physics-Doklady, 7 (1963), 595-596.

[46] Donald E. Knuth. *The art of computer programming. Vol. 2: Seminumerical algorithms*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont, third revised edition, 1997.

[47] B. Krauskopf, H. M. Osinga, E. J. Doedel, M. E. Henderson, J. Guckenheimer, A. Vladimirsky, M. Dellnitz, and O. Junge. A survey of methods for computing (un)stable manifolds of vector fields. *Internat. J. Bifur. Chaos Appl. Sci. Engrg.*, 15(3):763–791, 2005.

[48] Yu. A. Kuznetsov and H. G. E. Meijer. Numerical normal forms for codim 2 bifurcations of fixed points with at most two critical eigenvalues. *SIAM J. Sci. Comput.*, 26(6):1932–1954 (electronic), 2005.
[49] Kyoko Makino and Martin Berz. Taylor models and other validated functional inclusion methods. *Int. J. Pure Appl. Math.*, 4(4):379–456, 2003.
[50] James Murdock. *Normal forms and unfoldings for local dynamical systems*. Springer Monographs in Mathematics. Springer-Verlag, New York, 2003.
[51] Richard D. Neidinger. Computing multivariable taylor series to arbitrary order. *SIGAPL APL Quote Quad*, 25(4):134–144, 1995.
[52] Richard D. Neidinger. Directions for computing truncated multivariate Taylor series. *Math. Comp.*, 74(249):321–340 (electronic), 2005.
[53] Richard D. Neidinger. Introduction to automatic differentiation and matlab object-oriented programming. *SIAM Review*, 52(3):545–563, 2010.
[54] A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing (Arch. Elektron. Rechnen)*, 7:281–292, 1971.
[55] Éric Schost. Multivariate power series multiplication. In *ISSAC'05*, pages 293–300 (electronic). ACM, New York, 2005.
[56] C. Simó. Effective computations in celestial mechanics and astrodynamics. In *Modern methods of analytical mechanics and their applications (Udine, 1997)*, volume 387 of *CISM Courses and Lectures*, pages 55–102. Springer, Vienna, 1998.
[57] Carles Simó. Private communication.
[58] Carles Simó. "on the role of dynamical systems in celestial mechanics", howpublished="http://www.maia.ub.es/dsg/2010/1004simo.ps.gz",.
[59] Carles Simó. On the analytical and numerical approximation of invariant manifolds. In Daniel Benest and Claude Froeschlé, editors, *Les Méthodes Modernes de la Mecánique Céleste (Course given at Goutelas, France, 1989)*, pages 285–32, Paris, 1990. Editions Frontières.
[60] Carles Simó. Effective computations in Hamiltonian dynamics. In *Mécanique céleste*, volume 1996 of *SMF Journ. Annu.*, page 23. Soc. Math. France, Paris, 1996.
[61] Carles Simó. Global dynamics and fast indicators. In *Global analysis of dynamical systems*, pages 373–389. Inst. Phys., Bristol, 2001.
[62] A. L. Toom. The complexity of a scheme of functional elements simulating the multiplication of integers. *Dokl. Akad. Nauk SSSR*, 150:496–498, 1963.
[63] T. Uzer, Charles Jaffé, Jesús Palacián, Patricia Yanguas, and Stephen Wiggins. The geometry of reaction dynamics. *Nonlinearity*, 15(4):957–992, 2002.
[64] Joris van der Hoeven. Relax, but don't be too lazy. *J. Symbolic Comput.*, 34(6):479–542, 2002.
[65] Joris van der Hoeven. The truncated Fourier transform and applications. In *ISSAC 2004*, pages 290–296. ACM, New York, 2004.

Departament de Matemàtica Aplicada i Anàlisi.
Facultat de Matemàtiques, Universitat de Barcelona.
Gran Via de les Corts Catalanes 585, 08007 Barcelona (Spain).
E-mail address, Alex Haro: alex@maia.ub.es