LBID-2030
UC-405

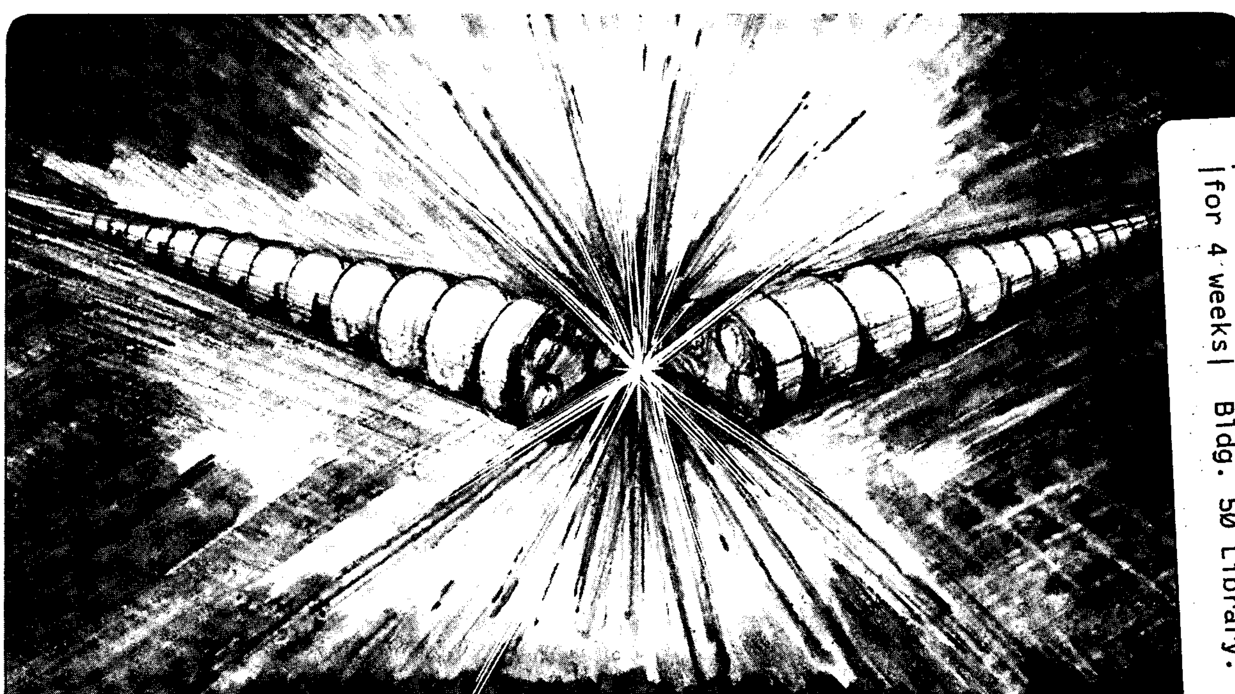
Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Accelerator & Fusion Research Division

Formal Calculus for Lie Methods in Hamiltonian Mechanics

P.-V. Koseleff
(Ph.D. Thesis)

June 1994



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

DISCLAIMER

DISCLAIMER

ÉCOLE POLYTECHNIQUE

THESIS

for the title of

DOCTOR IN SCIENCES
SPECIALITY: MATHEMATICS

by

Pierre-Vincent KOSELEFF

Subject: *Formal Calculus for Lie Methods in Hamiltonian Mechanics*

Translated by D. Boucher
Technical Assistance É. Forest.

The translation was supported by Department of Energy, Director's Office of Energy Research, contracts DE-AC03-76SF00098, DE-AC03-76SF00515, DOE grants DE-FG05-92ER40748, and DE-FG02-86ER40302.

ÉCOLE POLYTECHNIQUE

THÈSE

présentée pour obtenir le titre de

DOCTEUR EN SCIENCES
SPÉCIALITÉ : MATHÉMATIQUES

par

Pierre-Vincent KOSELEFF

Sujet : *Calcul formel pour les méthodes de Lie en mécanique hamiltonienne*

Soutenue le 29 janvier 1993 devant la commission d'examen :

P. CARTIER, Président

A. GIORGILLI,
J. MICHEL,
D. PINCHON, Rapporteurs

M. DEMAZURE,
M. GIUSTI,
D. KROB,
J. LASKAR, Examinateurs

Contents

FOREWORD TO THE TRANSLATION

The usage of Lie methods in classical mechanics is widespread. The reasons for this are fundamental: they arise naturally in the study of differential equations today just as they did in Sophus Lie's time. Indeed, it is the natural (and historical) place to introduce Lie brackets.

In beam dynamics, which is my field of study, Lie methods show their usefulness in at least two different ways.

First of all, in rings, where questions of stability are important, there are absolutely fundamental to an efficient theoretical framework. In fact, they are important for the hamiltonian problem of a hadron ring as well as in the case of an electron ring. In electron rings they enter in the study of the "radiative" non-hamiltonian map of the ring as well as in the stochastic map used to compute the equilibrium beam sizes. In all these cases, Lie transforms appear in some sort of normalized version of the one-turn map.

In addition, Lie methods often help in the design of certain systems by easing the exploitation of certain symmetries. This is true in electron microscopes, and in the final focusing part of linacs. Recently, John Irwin of the Stanford Linear Accelerator has used Lie methods extensively to design the interaction regions of linacs and colliders. This work relies on formal manipulations of Lie algebras which is the topic of this thesis. In fact, most analytical calculations in rings use a combination of techniques based on automatic differentiation and formal Lie algebra manipulations.

There is little literature on formal Lie algebras which connects with concepts such as the Deprit method, the Dragt-Finn factorization, explicit symplectic integrators and normalization methods. The thesis of Pierre-Vincent Koseleff covers all these concepts. Of course, it was not written for accelerator physicists. Indeed, while the thesis goes into the "rarefied atmosphere" of mathematics, its discussions of practical applications often deal with simple Hamiltonians by the standard of ring physicists. Nevertheless this thesis fills a gap in the formal Lie methods literature. For this reason, the Center for Beam Physics at the Lawrence Berkeley Laboratory, lead by Dr. Swapan Chattopadhyay, and the Accelerator Theory and Special Projects Group of the Stanford Linear Accelerator, lead by Dr. Ronald Ruth, sponsored its translation. In addition Profs. Alex Dragt of Maryland and John Cary of Colorado contributed financially to this endeavor. Dr. Leo Michelotti of the Fermi National Laboratory initially suggested this insane enterprise. And, finally, Dr. David Sutter of the United States Department of Energy gave me the encouragement to go ahead with the project.

The person most responsible for the bulk of the translation is Mrs. Danièle Boucher, a lecturer of French at the University of California at Berkeley. I would like to thank Mrs. Joy Kono for working out the financial and bureaucratic details which are always present in such an enterprise. And, finally, I would like to thank Dr. Pierre-Vincent Koseleff for permitting this translation to proceed and for proof-reading the entire document.

Étienne Forest
Berkeley, California, May 1994.

Introduction

Among the perturbation methods used in mechanics, Lie Methods play a major role in such diverse fields as celestial mechanics, geometrical optics, plasma physics, particle accelerator theory, neutron transport or electricity. Their development took off rapidly after the publication of two articles, one by celestial mechanicist Andre Deprit [11] in 1969 and another by two physicists, Alex Dragt and John Finn [14] in 1976.

Whenever possible, Lie Methods provide explicit change of variables (contrary to the von-Zeipel method, for example) in which the system's equations are "simpler". These methods provide formal invariants in the form of Lie series. Series do not generally converge, but computation of their first terms may provide information about the system's stability in the vicinity of points of equilibrium.

Lie methods are primarily based on the use of Lie transformations in order to achieve symplectic change of variables which preserve the system's integrity. When I first became interested in Lie methods in hamiltonian mechanics, and more specifically in celestial mechanics, there turned out to be two schools which used *a priori* distinct transformations. Celestial mechanics usually consider canonical transformations as Lie transformations, that is, as a mapping produced by a non-autonomous Hamiltonian for one unit of time. For others, canonical transformations are represented as compositions of maps which represent the flow of autonomous Hamiltonians for one unit of time. The use of one transformation or another induces perturbation methods that are very close in their formalism. Therefore, I was interested in comparing them.

Their comparison consisted, on the one hand, of showing that these two transformations where two ways of representing the same object, and on the other hand, studying the application costs of these two methods. Finally, since these methods produce series whose qualitative study can provide information about the system's stability, I deemed indispensable to propose convergence estimations for the Dragt-Finn method, in accordance with what had been done for the Deprit transformation.

That the two transformations be equivalent is a known fact, but I sought to express how they interrelate with each other, and to give some schemes to compute them effectively. These schemes rest essentially on the possibility of expressing identities between exponentials, in a free Lie algebra.

Several articles also proposed to compare both methods, either experimentally, based on examples, or by counting the number of terms in the expansion series. Here, algorithms are supplied for the computation of transformations, and the number of evaluations necessary for every order is compared.

Estimations for the Deprit transformation where offered, and yielded stability results. Following the same model, I propose some estimations for the Dragt-Finn method. These results are no better than for the Deprit transformation, but roughly of the same order.

Considering a hamiltonian system in appropriate form, here I will primarily focus on two kinds of problems. The first one, quantitative in nature, is to be able to provide, whenever possible, rough estimates of the trajectories for a given category of initial conditions. The second one, more qualitative (although effective bonds are nevertheless searched for), is to be able to estimate the system's stability in the vicinity of points of equilibrium.

In both problems, Lie methods are effective: for both problems, Lie methods yield results, because the obtained forms provide either approximate solutions, or approximate invariants, which gives us information about the system's stability.

The formal manipulation of Lie transformations is the first step. This will be achieved through a free Lie algebra in which methods for the computation of relationships between transformations will be given, allowing us to identify both perturbation methods. The methods to compute these relations are made possible by using the Lyndon basis rather than the Hall basis. This will give us the opportunity to discuss the specific properties of the Lyndon basis that allow the computation of such identities, particularly the Baker-Campbell-Hausdorff formula.

The algorithms developed in the first part will also enable us to effectively compute formal Lie transformations. The search of certain algebraic properties of Lie transformations' coefficients will provide symplectic integrators. Symplectic integrators are numerical schemes to compute, over very extensive periods, the approximate solutions for hamiltonian systems. hamiltonian systems characteristically maintain energy surfaces that are similar to the energy of the system to be studied.

***

The present thesis is divided into five distinct major sections.

The first section deals with free Lie algebras. Here, methods to compute exponential type relationships will be presented. I will also show how one can go from the exponential to the Lie transformation, and from the Lie transformation to the Dragt-Finn transformation. These relations will be expressed in one of the bases made available to us. The Hall basis, as well as the Lyndon basis will be considered, and I will show that polynomials can be manipulated in either. I originally focused on the Hall basis, in which decomposition algorithms are introduced. The Hall basis fortunately enabled the computation of relations between the generators of the Dragt-Finn transformation, and the Deprit transformation. I only just recently learned of a possible generalization for the Lyndon basis. The Lyndon basis made the computation of exponential type relations possible within the associative algebra.

However we shall discuss another alternative, which consists of using a factorization property of the Lyndon basis. This will in turn allow us to generalize the computation of relationships between transformations generators of the group of automorphisms of formal series of free Lie algebra, while remaining in free Lie algebra. This property essentially states that adjoint operator $[a,]$ defines a module isomorphism which can be represented by the identity in the Lyndon basis.

The second section deals with hamiltonian systems. After succinctly presenting the basic concepts of hamiltonian formalism, I will then focus on the utilization of the Dragt-Finn and Deprit transformations. The various perturbation methods induced by these transformations will be discussed. I will demonstrate the formal equivalence of the Deprit and Dragt-Finn transformations, and compare their respective costs. I shall also present various randomly compiled methods to solve the well known perturbation equation, keystone of the normal form construction.

In the third section, I will give estimations on the convergence radius of the series which come up in perturbation methods. In particular, and in the fashion of previous works published on the Deprit transformation, I shall also give the solution estimates related to the use of the Dragt-Finn transformation. However, since these estimates are unfortunately not better than those obtained for the Deprit transformation, we shall simply recall the deductions that can be drawn from them in the calculation of the diffusion or stability of hamiltonian systems in the vicinity of an elliptical point. It is worth mentioning that these deductions do not, in effect, depend on the transformation used, but simply on properties related to the existence of formal invariants.

In the fourth section, we shall discuss how one gets symplectic integrators, numerical integration schemes for hamiltonian systems, as an application of the results obtained in the first part. These integrators have recently appeared, and I will show how they can be constructed by using results from the first part. An exhaustive list of previously known results will be compiled for low orders. I will show that the various methods used for computing them are algebraically equivalent. I will propose some improvement in the case where kinetic energy is a second degree polynomial, and I will give a few integrators for the problem of perturbed Kepler motion. One of these integrators will soon be tested by the Bureau des Longitudes.

Finally, in the last, but not least copious section, I shall describe the various implementations achieved in AXIOM or in SCRATCHPAD II from algorithms of the preceding parts. After a brief overview of the basic concepts of the Formal Computation AXIOM system, I shall thoroughly describe the implementation of free Lie algebras. Specifically, I shall compare a Lie monomials representation by way of lists or trees (which seems more natural), and it will become patent that in the first representation, an algorithms derecursification is possible, and in particular, that the Hall order, that is to say a total order on the magma, is a lexicographical order.

I shall explain how the packages enabling the computation of relations between generators of Lie transformations were written, which will allow us to generate the equations to be solved for the search of symplectic integrators. These packages are achieved by finding the actual solutions to a system of polynomial equations. For this purpose, existing routines in AXIOM and in MACAULAY will be used, that is to say, the computation of the Gröbner basis.

I shall also describe various implementations of perturbation methods mentioned in previous sections. Various domains will have been created for the purpose of this discussion, including that of the Poisson series, which are convenient for working with action-angle variables, and for some non-autonomous cases.

In the appendix, I shall give a few examples implemented on an IBM-RS/6000-550 computer, using the obtained implementations. I shall also indicate the computation times, as well as the number of terms in the relations between the generators of the various transformations.

Chapter I
Free Lie Algebras

In this chapter, our goal is to provide schemes for the explicit computation of identities between exponentials such as, for example, the Baker-Campbell-Hausdorff formula.

These identities will first be proved within Lie series algebra on a well-balanced alphabet. In particular, we study the subgroup within the group of Lie automorphisms approximating the identity generated by exponentials.

We describe the tools and results necessary for the implementation of a manipulator of polynomials and formal Lie series in a free Lie algebra.

We use, in particular, two of the free Lie algebra bases: the Hall basis and the Lyndon basis. For both, we will give a construction algorithm, and we will recall a method of decomposition of Lie polynomials onto these bases.

A classic method (cf. [42, 57]) to compute the Campbell-Hausdorff and other exponential identities consists of using the bijection between the Lyndon basis and the Lyndon words. This bijection extends itself into an isomorphism of modules in which the triangular matrix is easily reversible. This technique requires the introduction of the associative algebra.

Here, we propose to use a property of the Lyndon basis (cf. lemma 18, page 23) which will enable us to factorize a Lie polynomial by an inner derivative of a monomial of the basis.

Hence we shall deduce general methods in order to express the Baker-Campbell-Hausdorff formula, as well as the exponential of a Lie series in the form of an increasing infinite product of homogeneous Lie polynomials, or as a Lie transformation, while remaining in the free Lie algebra.

These general methods will later be used to represent the canonical transformations in hamiltonian mechanics.

1. Definitions

In all that follows, R will denote a commutative ring, and X, an alphabet, that is, a totally ordered set whose elements will be called letters.

1.1. Free Monoid, Free Magma

We denote by X^* the free monoid over X, the free structure generated by X, relative to concatenation, which is an associative composition law, and which possesses a neutral element. The elements of X^* will be called the words. The composition law onto X^* admits a neutral element, which is the empty word. Given $w = uv$ a word of X^*. We will say that u and v are factors of w, and proper factors, if they are distinct from w. We will say that u is the left factor, and v, the right factor of w.

X^* is a set ordered by lexicographical order. One definition is:

— if u is a proper left factor of $v \in X^*$, thus $u < v$,

— for any u, v in X^*, $u < v$ if there is $r, a, b, t, s \in X^*$ with $u = ras, v = rbt$ and $a < b$.

This total order also satisfies:

— for any $w, u, v \in X^*$, we have $u < v$ if and only if $wu < wv$,

— if v is not a left factor of u, thus for any $w, z \in X^*$, we have $uw < vz$ if $u < v$.

For any u, v of X^*, we shall say that the words $w_1 = uv$ and $w_2 = vu$ are conjugates, and proper conjugates if u and v are not the empty word. The conjugation is an equivalence relation. We denote by $|u|$ the length u, that is to say the number of letters forming the word, and by X_n the set of words of X^* of length n.

$M(X)$ is the free magma generated by X. It is the set of parenthesized words provided with the composition law:

$$\begin{array}{ccc} M \times M & \rightarrow & M \\ (u, v) & \mapsto & (u, v). \end{array}$$

We inductively define the magma $M(X)$ by considering the family:

— $M_1(X) = X$,

— $M_n(X) = \bigsqcup_{p+q=n} M_p(X) \times M_q(X)$ for $n \geq 2$ where $\bigsqcup$ denotes the set union or fusion,

and by defining $M(X) = \bigsqcup_{n \geq 1} M_n(X)$. Likewise, we define the length over $M(X)$, as the only $n \geq 0$, such that $u \in M_n(X)$.

We denote by δ the canonical mapping of $M(X)$ over X^* given by unparenthesizing. For example, if $x, y \in X$

$$\delta\left(x, \left(\left(\left(x, y\right), \left(x, \left(x, y\right)\right)\right), y\right)\right) = xxyxxyy = x^2yx^2y^2.$$

1.2. Associative Algebra, Free Lie Algebra

$\mathcal{A}(X, R)$ denotes the free associative algebra, on ring R, generated by X. It is the algebra of non commutative polynomials in variables X. It is obviously a free R-module with basis X^*, totally graded in length. Barring exceptions, we use $\mathcal{A}(X)$ for $\mathcal{A}(X, R)$.

The free algebra $A(X, R)$ is an R-algebra of the free magma $M(X)$. Multiplication over $A(X, R)$ is an extension of the composition law of $M(X)$.

A $\mathfrak{L}$ Lie algebra is an algebra in which the multiplication law –or Lie bracket– is bilinear, antisymmetric, and satisfies the Jacobi identity. For any $a, b, c \in \mathfrak{L}$, we have

— $[a, a] = 0$

— $[a, [b, c]] + [b, [c, a]] + [c, [a, b]] = 0.$

The algebra on R with Lie bracket $[u,v] = uv - uv$, where $u, v, uv \in \mathcal{A}(X,R)$, is a Lie algebra.

$L(X,R)$ will denote the free Lie algebra on R, generated by X. It will be defined as the quotient of $A(X,R)$ by the ideal generated by the elements of form (u,u) and $(u,(v,w)) + (v,(w,u)) + (w,(u,v))$. Given the canonical map of $M(X) \to L(X,R)$, a magma morphism for the Lie bracket, the image of an element of $M(X)$ will be called a Lie alternate or monomial. Lie monomials of length n generate a free submodule $L_n(X,R)$ of $L(X,R)$ of elements of length n. We use, barring exceptions, $L(X)$ and $L_n(X)$ instead of $L(X,R)$ and $L_n(X,R)$. Thus $L(X,R)$ appears, graded by degree, but it is not the only possible gradation.

If X is finite of cardinality q, we have (cf. [6, 43, 44]) the Witt formula:

$$\sum_{d|n} d \dim(L_d(X)) = q^n. \tag{1.1}$$

Elements of $L(X)$ will be called Lie polynomials.

We define a canonical injection of R-module $\phi : L(X,R) \to \mathcal{A}(X,R)$, by extension of the inclusion of $X \subset \mathcal{A}(X,R)$:

— if $u \in X$ thus $\phi(u) = u$

— if $u = [v,w]$ thus $\phi(u) = \phi(v)\phi(w) - \phi(w)\phi(v)$.

2. Basic families of free Lie algebra

Let us consider the bases of $L(X)$, made-up by a set of Lie monomials.

2.1. Hall Basis

A Hall set is a family $\mathcal{H}$ of $M(X)$ with a total order $<$ satisfying:

— if $u, v \in \mathcal{H}$ and $|u| < |v|$ then $u < v$,

— $X \subset \mathcal{H}$ and $\mathcal{H} \cap M_2(X) = \{(x,y); x < y \in X\}$,

— $h = (u,v) \in \mathcal{H}$ and $|h| \geq 3$ if and only if $u, v \in \mathcal{H}$, $u < v$ and $v = (v_1, v_2)$ with $v_1 \leq u$.

By abuse of notation, let us also denote by $\mathcal{H}$, the range of $\mathcal{H}$ in $L(X)$. Let us pose, for any n, $\mathcal{H}_n = \mathcal{H} \cap M_n(X)$.

Theorem 1 ([6]). — *$\mathcal{H}_n$ is a basis of $L_n(X)$.*

We shall prove that $\mathcal{H}_n$ is a generative family, by giving a multiplication algorithm of two Lie polynomials. With respect to linear independence, see [6], where a direct proof will be given.

2.2. Lyndon Basis

Let us consider, in the monoid X^*, conjugation classes, and let us take, to represent each class, the smallest element according to its lexicographical order. We have thus defined the set $\mathbf{L}$ of the Lyndon words. We have the following properties:

— any word of X^* is uniquely decomposed into a descending product of Lyndon words (lemma of Široov, [57]).

— $w \in \mathbf{L}$ if there exists $l < m \in \mathbf{L}$ such that $w = lm$.

However, a Lyndon word may be decomposed into Lyndon words in several ways. For example, we have $aabb = a(abb) = (aab)b$. The (l, m) pair, where m is of maximum length, is the standard factorization of lm which we will denote $\sigma(lm)$. Therefore, for the previous example, we have $\sigma(aabb) = (a, abb)$.

We can thus define a mapping Λ reciprocal of $\delta : \mathbf{L} \to M(X)$ as follows : $\Lambda(a) = a$ if $a \in X$, else $\Lambda(n) = (\Lambda(l), \Lambda(m))$ if $\sigma(n) = (l, m)$.

We denote $\mathcal{L} = \Lambda(\mathbf{L})$. In addition, we denote the Lyndon basis by $\mathcal{L}$, as well as its range in $L(X)$. We also express $\mathcal{L}_n = \mathcal{L} \cap M_n(X)$.

THEOREM 2 ([44]). — *$\mathcal{L}_n(X)$ is a basis of $L_n(X)$.*

Remark. — It appears that Široov might be the source of this family, usually called the Lyndon basis.

Because of another property, the Lyndon basis may be defined similarly to the Hall basis: it is a set $\mathcal{L}$ of $M(X)$, with a total order (the lexicographical order on the corresponding Lyndon words) satisfying:

— $X \subset \mathcal{L}$ and $\mathcal{L} \cap M_2(X) = \{(x, y); x < y\}$,

— $h = (u, v) \in \mathcal{L}$ and $|h| \geq 3$ if and only if $u, v \in \mathcal{L}$, $u < v$ and $|u| = 1$ or $u = (u_1, u_2)$ with $v \leq u_2$.

2.3. Construction of the Hall Basis

The recursive construction of the Hall basis is not as hard as one might have thought, since if $u \in \mathcal{H}_p(X)$ and $v \in \mathcal{H}_q(X)$, thus $[u, v] \in \mathcal{H}_{p+q}(X)$ provided that $u < v$ and that, if $v = [v_1, v_2]$ we have $v_1 \leq u$. Conversely, if $w \in \mathcal{H}_n(X)$, thus there exists p, q with $p + q = n$, $u \in \mathcal{H}_p(X)$ and $v \in \mathcal{H}_q(X)$ such that $w = [u, v]$.

We shall consider the following construction:

$$\begin{aligned}
\mathcal{H}_1(X) &= X, \\
\mathcal{H}_2(X) &= \{[x, y]; x < y \in \mathcal{H}_1(X)\}, \\
\mathcal{H}_n(X) &= \bigcup_{p=1}^{n-1} \{[x, y]; x \in \mathcal{H}_p(X), y = [y_1, y_2] \in \mathcal{H}_{n-p}(X), y_1 \leq x < [y_1, y_2]\} \\
&= \bigcup_{p+q=n, p \leq q} \{[x, y]; x \in \mathcal{H}_p(X), y = [y_1, y_2] \in \mathcal{H}_q(X), y_1 \leq x\}.
\end{aligned}$$

We notice that the above are disjoint reunions, which makes the construction easier, since we do not have to eliminate doubles. The underlying algorithm thus returns the sorted list of the elements of the Hall basis.

2.4. Construction of the Lyndon Basis

The construction of the Lyndon basis is similar to that of the Hall basis, with only a minor difficulty, since if $u, v \in \mathbf{L}$ and $u < v$ for the lexicographical order, thus $uv \in \mathbf{L}$, but $[\Lambda(u), \Lambda(v)]$ does not necessarily belong to $\mathcal{L}$, if (u, v) is not the standard factorization of uv. This inconvenience is definitely related to the fact that the lexicographical order is not a total order over $M(X)$. Nevertheless, by constructing successively $\bigcup_{x\in\mathcal{L}_p(X)}\left\{\bigcup_{y\in\mathcal{L}_q(X);x<y}\{[x,y]\}\right\}$ for increasing x and y, we shall directly construct the Lyndon bracket which corresponds to the standard factorization. Thus the following algorithm is applied:

$$\begin{aligned}
\mathcal{L}_1(X) &= X \\
\mathcal{L}_2(X) &= \{[x,y]; x < y \in \mathcal{L}_1(X)\} \\
\mathcal{L}_n(X) &= \bigsqcup_{x\in\mathcal{L}_1(X)} \{[x,y]; y \in \mathcal{L}_{n-1}(X), x < y\} \\
&\quad \bigsqcup \bigsqcup_{p=2}^{n-1} \left\{ \bigsqcup_{x\in\mathcal{L}_p(X)} \{[x,y]; y \in \mathcal{L}_{n-p}(X), [x_1,x_2] = x < y \le x_2\} \right\}
\end{aligned}$$

where $A \bigsqcup B$ denotes the fusion of both sorted lists A and B for the lexicographical order over the unparenthesized words.

2.5. Examples

For $X = \{x, y\}$ we have, as the elements of the basis to the sixth order:
Hall Basis:
$\{x, y, [x,y], [x,[x,y]], [y,[x,y]], [x,[x,[x,y]]], [y,[x,[x,y]]], [y,[y,[x,y]]], [x,[x,[x,[x,y]]]],$
$[y,[x,[x,[x,y]]]], [y,[y,[x,[x,y]]]], [y,[y,[y,[x,y]]]], [[x,y],[x,[x,y]]], [[x,y],[y,[x,y]]],$
$[x,[x,[x,[x,[x,y]]]]], [y,[x,[x,[x,[x,y]]]]], [y,[y,[x,[x,[x,y]]]]], [y,[y,[y,[x,[x,y]]]]],$
$[y,[y,[y,[y,[x,y]]]]], [[x,y],[x,[x,[x,y]]]], [[x,y],[y,[x,[x,y]]]], [[x,y],[y,[y,[x,y]]]],$
$[[x,[x,y]],[y,[x,y]]]\}$

Lyndon Basis:
$\{x, y, [x,y], [x,[x,y]], [[x,y],y], [x,[x,[x,y]]], [x,[[x,y],y]], [[[x,y],y],y], [x,[x,[x,[x,y]]]],$
$[x,[x,[[x,y],y]]], [[x,[x,y]],[x,y]], [x,[[[x,y],y],y]], [[x,y],[[x,y],y]], [[[[x,y],y],y],y],$
$[x,[x,[x,[x,[x,y]]]]], [x,[x,[x,[[x,y],y]]]], [x,[[x,[x,y]],[x,y]]], [x,[x,[[[x,y],y],y]]],$
$[x,[[x,y],[[x,y],y]]], [[x,[[x,y],y]],[x,y]], [x,[[[[x,y],y],y],y]], [[x,y],[[[x,y],y],y]],$
$[[[[[x,y],y],y],y],y]\}$

3. Decomposition onto the Bases

Let us now focus on the process of Lie polynomial decomposition onto one of the two previously defined bases. The question is to know whether common operations such as adding or multiplying can be done on this algebra.

If we consider Lie polynomials as linear combinations of monomials of the Hall basis or the Lyndon basis, operations related to the addition group structure follow naturally.

The main step in the multiplication (here multiplication refers to the Lie bracket) of two Lie polynomials is the multiplication of two monomials of the basis. In the case of the Hall basis as in that of the Lyndon basis, this operation can be inductively implemented without knowing any element in the basis.

Here, I shall give two decomposition algorithms of the product of two monomials of the basis. Since the $L(X)$ Lie algebra is graded by length, this will also provide the opportunity to show that the Hall basis and the Lyndon basis are generative families of $L(X)$.

In the last section, we propose an implementation of these algorithms.

3.1. Hall Basis

For u and v, two elements of the Hall basis, let us denote $\mathtt{mult}(u,v)$ the decomposition of $[u,v]$ onto the Hall basis.

The multiplication of two Lie polynomials: $p = \sum p_i.u_i$ and $q = \sum q_i.v_i$ is thus defined by $p * q = \sum p_i q_j.\mathtt{mult}(u_i, v_j)$. Given the properties of the Lie bracket and the Hall basis, `mult` may be defined as follows:

if $u = v$ **then** $\mathtt{mult}(u,v) := 0$
else if $v < u$ **then** $\mathtt{mult}(u,v) := -\mathtt{mult}(v,u)$;
if $|v| = 1$ **then** $\mathtt{mult}(u,v) := 1.[u,v]$
else $v = [v_1, v_2]$; **if** $v_1 \le u$ **then** $\mathtt{mult}(u,v) := 1.[u,v]$
else $\mathtt{mult}(u,v) := 1.v_1 * \mathtt{mult}(u,v_2) - 1.v_2 * \mathtt{mult}(u,v_1)$.

In order to show the convergence of this algorithm, one merely needs to show that any monomial in the form $[a,[b,c]]$, with $a < b < c < [b,c] \in \mathcal{H}$, is decomposed by $\mathtt{mult}(a,[b,c])$.

Let us speculate on the existence of monomials that cannot decompose. Let us specifically consider monomials of minimum length (which are in finite number if the alphabet is finite), and among these monomials, the one where a is maximum for the order on the Hall basis.

Since $a < b$, we have to compute $1.b * \mathtt{mult}(a,c) - 1.c * \mathtt{mult}(a,b)$. But, by hypothesis on triplet (a,b,c), $\mathtt{mult}(a,c)$ and $\mathtt{mult}(a,b)$ decompose respectively into $\sum \beta_i[a_i,c_i]$ and $\sum \gamma_i[a_i,b_i]$. Therefore, we are bound to compute terms $\mathtt{mult}(b,[a_i,c_i])$ and $\mathtt{mult}(c,[a_i,b_i])$.

Since $b < c$, we have $|b| \le |c| < |[a,c]| = |[a_i,c_i]|$, therefore, $b < [a_i,c_i]$ and $\mathtt{mult}(b,[a_i,c_i])$ decompose, either because $a_i \le b$, or because $a < b$ and a was presumed maximal.

If $c < [a_i,b_i]$, then either $a_i \le c$ and $\mathtt{mult}(c,[a_i,b_i]) = [c,[a_i,b_i]] \in \mathcal{H}$, or $a < c < a_i$ and, by assumption on a, $\mathtt{mult}(c,[a_i,b_i])$ decomposes. If $[a_i,b_i] < c$, then $|a| < |[a_i,b_i]|$ and $\mathtt{mult}(c,[a_i,b_i]) = -\mathtt{mult}([a_i,b_i],c)$ by assumption decomposes onto a, since $a < [a_i,b_i]$.

In any case, the existence of maximal a is contradicted. On the other hand, the evidence that the multiplication algorithm of two monomials of the Hall basis comes to an end in a finite number of steps proves that the Hall basis is in fact a generative family of $L(X)$, since $L_1(X)$ is generated by $X \subset M(X)$.

It is worth mentioning that it is not necessary to test whether or not a Lie monomial belongs to the Hall basis, but that all one needs to do is compare two Lie monomials. So the comparison of two monomials is the main phase of the algorithm, and consequently, it is essential to optimize it.

3.2. Lyndon Basis

The same considerations apply to the Lyndon basis, and one just needs to know how to multiply two monomials of the Lyndon basis. In the same fashion, (cf. [45]), we define `mult` as follows:

if $u = v$ **then** $\texttt{mult}(u, v) := 0$
else if $v < u$ **then** $\texttt{mult}(u, v) := -\texttt{mult}(v, u)$;
if $|u| = 1$ **then** $\texttt{mult}(u, v) := 1.[u, v]$
else $u = [u_1, u_2]$; **if** $v \leq u_2$ **then** $\texttt{mult}(u, v) := 1.[u, v]$
else $\texttt{mult}(u, v) := 1.u_1 * \texttt{mult}(u_2, v) + \texttt{mult}(u_1, v) * 1.u_2$.

in order to show the convergence of this algorithm, we pose

LEMMA 3 (Perrin [44]). — *For any $a < b \in \mathcal{L}$, $\texttt{mult}(a, b)$ is a finite linear combination of Lyndon monomials $[a_i, b_i]$ which satisfies*

$$\delta([a_i, b_i]) \geq \delta([a, b]) \quad or \quad \delta([a_i, b_i]) = \delta([a, b]) \quad and \quad a_i < a. \tag{3.1}$$

Since the case of length 1 or 2 brackets is at hand, as well as that of $[a, b] < c \leq b \in \mathcal{L}$, we will simply have to show that any monomial of type $[[a, b], c]$, with $a < [a, b] < b < c \in \mathcal{L}$, decomposes, by $\texttt{mult}([a, b], c)$, into the previous form.

Let us suppose the existence of monomials that cannot decompose according to (3.1). Among these, let us consider those of minimal length (they will be in finite number if the alphabet is finite), and among these, the monomial where $\delta(a)\delta(b)\delta(c)$ is maximal for the order on the Lyndon basis, and $\delta(ab)$ is minimal.

One must now compute $1.a * \texttt{mult}(b, c) + \texttt{mult}(a, c) * 1.b$. But by assumption on the lengths, $\texttt{mult}(b, c) = \sum \alpha_i [b_i, c_i]$ with $\delta([b_i, c_i]) \geq \delta(b)\delta(c)$ and $\texttt{mult}(a, c) = \sum \beta_i [a_i, c_i]$ with $\delta([a_i, c_i]) \geq \delta(a)\delta(c)$. We are led to compute terms like $\texttt{mult}([a_i, c_i], b)$ and $\texttt{mult}(a, [b_i, c_i])$.

$\delta(a)\delta([b_i, c_i]) \geq \delta(a)\delta(b)\delta(c)$ but this time $a < [a, b]$, and by assumption on a, b, c, we deduce from this that $\texttt{mult}(a, [b_i, c_i])$ decomposes in the desired form, that is, into a sum of monomials $[u_j, v_j]$, which satisfies $\delta(u_j)\delta(v_j) \geq \delta(a)\delta(b_i)\delta(c_i) \geq \delta(a)\delta(b)\delta(c) = \delta([[a, b], c])$.

Since $b < c$, we have $\delta(b)\delta(c) < \delta(c)\delta(b)$, therefore, from $\delta([a_i, c_i]) \geq \delta(a)\delta(c)$, we deduce $\delta([a_i, c_i])\delta(b) \geq \delta(a)\delta(c)\delta(b) \geq \delta(a)\delta(b)\delta(c)$, which is maximal by assumption. We conclude that $\texttt{mult}([a_i, c_i], b)$ decomposes into Lyndon monomials. $[u_j, v_j]$ which satisfies $\delta([u_j, v_j]) \geq \delta([a_i, c_i])\delta(b) \geq \delta(a)\delta(c)\delta(b) \geq \delta(a)\delta(b)\delta(c) = \delta([a, b])\delta(c)$.

From this we thus deduce that, for any pair of Lyndon monomials (a, b), $\texttt{mult}(a, b)$ is a linear combination of Lyndon monomials which satisfy, besides being greater for the lexicographical order, than $\delta(a)\delta(b)$. This proves that the Lyndon basis is a generative family of the free Lie algebra $L(X)$. In order to show that $\mathcal{L}$ is a free family of $L(X)$, we will show

LEMMA 4 (Viennot [56]). — *For any $u \in \mathcal{L}$, $\phi(u) = 1.\delta(u) + v$ where v is a linear combination of words that are strictly superior to u.*

So this lemma states that there is a bijection between Lyndon words and Lyndon brackets, and that module isomorphism occurs between $L(X,R)$ and the projection of $\mathcal{A}(X,R)$ onto the submodule generated by those Lyndon words which constitute a basis.

3.3. Examples

Let us take $X = \{a,b,c\}$ as an alphabet.

- $m = [[a,\ b],\ c]$ decomposes onto the Lyndon basis into $[a,\ [b,\ c]] + [[a,\ c],\ b]$, and on the Hall basis into $-[c,\ [a,\ b]]$
- $m = [[[[[c,\ b],\ a],\ a],\ a],\ a],\ a]$ decomposes onto the Hall basis into :

  $$[b,\ [a,\ [a,\ [a,\ [a,\ [a,\ c]]]]]] - [c,\ [a,\ [a,\ [a,\ [a,\ [a,\ b]]]]]] + 5\ [[a,\ b],\ [a,\ [a,\ [a,\ [a,\ c]]]]] \\ - 5\ [[a,\ c],\ [a,\ [a,\ [a,\ [a,\ b]]]]] + 10\ [[a,\ [a,\ b]],\ [a,\ [a,\ [a,\ c]]]] \\ + 10\ [[a,\ [a,\ c]],\ [a,\ [a,\ [a,\ b]]]]$$

  and onto the Lyndon basis into: $-[a,\ [a,\ [a,\ [a,\ [a,\ [b,\ c]]]]]]$
- $m = [[[[[a,\ b],\ c],\ c],\ c],\ c],\ c]$ decomposes onto the Lyndon basis into:

  $$[a,\ [[[[b,\ c],\ c],\ c],\ c],\ c]] + 5\ [[a,\ c],\ [[[[b,\ c],\ c],\ c],\ c]] + 10\ [[[a,\ c],\ c],\ [[[b,\ c],\ c],\ c]] \\ + 10\ [[[[a,\ c],\ c],\ c],\ [[b,\ c],\ c]] + 5\ [[[[[a,\ c],\ c],\ c],\ c],\ [b,\ c]] \\ + [[[[[[a,\ c],\ c],\ c],\ c],\ c],\ b]$$

  and onto the Hall basis into: $-[c,\ [c,\ [c,\ [c,\ [c,\ [a,\ b]]]]]]$.

4. Lie Polynomials and non commutative polynomials

We just saw that manipulation of Lie polynomials, considered as linear combinations of Hall basis or Lyndon basis monomials, is possible if one is able to implement the Lie bracket of two basic monomials. An implementation of such manipulations will be proposed in the last section.

Another problem is to decide whether or not a polynomial in non commutative variables is the developed form of a Lie polynomial. Several methods help to solve this problem. These methods are related to the manner in which we consider polynomials in non commutative variables.

A polynomial in non commutative variables may be considered as an element of the associative algebra $\mathcal{A}(X,R)$, that is, as a linear combination of words, with coefficients in R.

The associative algebra $\mathcal{A}(X,R)$ may be considered as the enveloping algebra of $L(X,R)$ (cf. [6, 42, 50, 57]), and the Poincaré-Birkhoff-Witt theorem provides a basis for this algebra from a basis of $L(X)$. More specifically, we have the theorem :

THEOREM 5 (Poincaré-Birkhoff-Witt). — *If $L(X)$ admits as its basis $\{P_1 < P_2 < \cdots < P_n \cdots\}$ thus its enveloping algebra admits as its basis the elements of form $P_{i_1} P_{i_2} \cdots P_{i_n}$ with $P_{i_1} \geq P_{i_2} \geq \cdots \geq P_{i_n}$.*

This theorem, and multiplication algorithms of some elements of the Poincaré-Birkhoff-Witt (cf. Petitot [45]) basis have enabled us to implement polynomials in non commutative variables as elements of the enveloping algebra.

For example, these algorithms allow us to test whether a non commutative polynomial belongs to the free Lie algebra, to compute the Baker-Campbell-Hausdorff formula, as well as other exponential identities ([42, 45, 56]).

Another way of proceeding, if we already know that the polynomial is a Lie polynomial, is to consider the property inductively developed by lemma 4, which leads us to state that

THEOREM 6 (Michel [43]). — *There is an isomorphism of modules between $L(X)$ and the submodule of $\mathcal{A}(X)$ generated by Lyndon words. Its matrix in the Lyndon basis is triangular, with 1's in its diagonal.*

Thus we use the following algorithm, in order to write a polynomial p into a sum of Lyndon monomials.

Let $p = \sum \alpha_i u_i$; $r := 0$
while $p \neq 0$ **do**
  $r := r + \alpha_1 \Lambda(u_1)$
  $p := p - \alpha_1 \tilde{\phi}(u_1)$
return r

The function Λ is the parenthesizing function on Lyndon words. If we know the Lyndon basis ahead of time, we do not need to compute $\Lambda(u_1)$, because the lexicographical order is a total order on the Lyndon brackets. Otherwise, we may apply a standard factorization algorithm, (cf. [44, 57]) which uniquely factorizes any word into a product of descending Lyndon words.

The function $\tilde{\phi}$ is the composition of the projection π of $\mathcal{A}(X, R)$ on the submodule generated by the Lyndon words with the decomposition function ϕ of a Lie bracket into an $\mathcal{A}(X, R)$ polynomial. These techniques where used, for example, by [43] to compute the coefficients of the Baker-Campbell-Hausdorff formula. I also proposed an implementation in LE-LISP, described in [32].

5. FORMAL LIE SERIES

In this section, we suppose that $\mathbb{Q} \subset R$. Let us denote $\hat{L}(X, R) = \prod_{n \geq 0} L_n(X, R)$, the ring of formal Lie series. We denote $\hat{L}(X)$ for $\hat{L}(X, R)$ whenever there is no ambiguity. An element of $\hat{L}(X)$ is uniquely written as a sum of elements of $L_n(X)$, this ring containing the Lie bracket

$$\left[\sum_{n \geq 0} u_n, \sum_{n \geq 0} v_n\right] = \sum_{n \geq 0} \sum_{p+q=n} [u_p, v_q],$$

is a graded Lie algebra, separate and complete. In the same fashion, we define the ring of non commutative formal Lie series $\hat{\mathcal{A}}(X, R) = \prod_{n \geq 0} \mathcal{A}_n(X, R)$. Thus we note $\mathfrak{M}$, ideal of

$\hat{\mathcal{A}}(X)$ generated by X and we define maps exponential and logarithm by:

$$
\begin{array}{rccl}
\exp: & \mathfrak{M} & \longrightarrow & 1+\mathfrak{M} \\
& x & \mapsto & \sum_{n\geq 0}\dfrac{x^n}{n!}, \\
\log: & 1+\mathfrak{M} & \longrightarrow & \mathfrak{M} \\
& x & \mapsto & -\sum_{n\geq 1}\dfrac{(1-x)^n}{n}.
\end{array}
\tag{5.1}
$$

5.1. Baker-Campbell-Hausdorff Formula

We demonstrate that exp and log are reciprocal maps (cf. [6, 42]). Moreover, we have the Campbell-Hausdorff theorem:

THEOREM 7 (Campbell-Hausdorff). — *Given $X=\{x,y\}$, we have*

$$\log[\exp(x)\exp(y)] \in \hat{L}(X,\mathbb{Q}).$$

This result extends (cf. [6, Ch. II, §6]) to any graded free Lie algebra, separate and complete, and we have the more general theorem:

THEOREM 8. — *If $x,y\in\hat{L}(X,\mathbb{Q})$ then*

$$\log[\exp(x)\exp(y)] \in \hat{L}(X,\mathbb{Q}).$$

5.2. Weighted Lie Algebras

Given an alphabet X, finite or not, we attribute a weight $|x|_*$ to each x letter. Next, we grade $\mathcal{A}(X)$ and $L(X)$ by the weighted degree or the quasi-homogeneous multi-degree $||_*$ which we usually define as the sum of the weight of the letters which make up the Lie word or bracket. Of course, it is the only morphism of X^* (resp. $M(X)$) $\to \mathbb{N}$ which prolongs the weight on X. We can thus grade $L(X)$ and $\mathcal{A}(X)$ by the quasi-homogeneous multi-degree, and we denote $\tilde{L}_n(X)$ and $\tilde{\mathcal{A}}_n(X)$, submodules of $L(X)$ and $\mathcal{A}(X)$, generated by the elements of weight n.

Remark. — If all letters of the alphabet are of weight 1, we have $\hat{L}_n(X)=\tilde{L}_n(X)$ and $\hat{\mathcal{A}}_n(X)=\tilde{\mathcal{A}}_n(X)$. If none of the letters of the alphabet have a null weight, we have

$$\prod_{n\geq 1} L_n(X) \simeq \prod_{n\geq 1}\tilde{L}_n(X).$$

5.2.1. Examples

If $X=\{x_1,x_2,x_3,x_4,x_5\}$ and $|x_i|_*=i$, $\tilde{L}_5(X)$ admits as the Lyndon basis

$$\{[x_1,\ [x_1,\ [x_1,\ x_2]]],\ [x_1,\ [x_1,\ x_3]],\ [[x_1,\ x_2],\ x_2],\ [x_1,\ x_4],\ [x_2,\ x_3],\ x_5\}. \tag{5.2}$$

If $X=\{x_1,\ldots,x_n,\ldots\}$ where $|x_i|_*=i$, then we have $\dim\tilde{L}_1(X)=1$ and for any $n\geq 2$, $\dim\tilde{L}_n(X)=l_n$ where l_n is the dimension of $L_n(\{a,b\})$. Using the Witt formula, we then deduce

$$\sum_{d|n} d\dim L_n(X) = 2^n-1. \tag{5.3}$$

Proof. — Let us pose $Y = \{a, b\}$, and show that monotonically increasing mapping

$$\begin{array}{rccc} \eta: & X & \longrightarrow & Y^* \\ & x_k & \longmapsto & ab^{k-1} \end{array} \tag{5.4}$$

extends into a module isomorphism $\tilde{L}_n(X) \longrightarrow L_n(Y)$ for $n \geq 2$. More specifically, we show recurrently on n, that any Lyndon word of length n of Y^* is uniquely written, as any Lyndon word of X^* of weight n, and conversely.

If $n = 1$, then a is written as u_1.

Let us consider (u, v), the standard factorization of $w \in Y^*$, a Lyndon word of length n, different from b. If $v = b$, then $w = ab^{n-1}$ is written u_n, else u and v are written like Lyndon words of X^*. η being increasing, uv is then written like a Lyndon word.

Conversely, the image by η of any increasing product of two Lyndon words of X^* is an increasing product of two Lyndon words and is a Lyndon word of Y^*.

We then deduce that $\dim \tilde{L}_n(X) = \dim L_n(Y)$, for any $n \geq 2$, and the predicted result. There is a bijection for $\mathcal{L}_5(Y)$

$$\{[a, [a, [a, [a, b]]]], [a, [a, [[a, b], b]]], [[a, [a, b]], [a, b]], [a, [[[a, b], b], b]], [[a, b], [[a, b], b]], [[[[a, b], b], b], b]\}$$

⁂

Let us define

$$\tilde{L}(X, R) = \prod_{n\geq 0} \tilde{L}_n(X, R) \quad \text{and} \quad \tilde{A}(X, R) = \prod_{n\geq 0} \tilde{A}_n(X, R) \tag{5.5}$$

as formal Lie series or words. An element of $\tilde{L}(X)$ is uniquely written as a sum of elements of increasing weight. For $x \in \tilde{L}(X)$, we denote $(x)_n$ or x_n, whenever there is no ambiguity, the term of weight n of x. If $x, y \in \tilde{L}(X, R)$, we have

$$([x, y])_n = \sum_{p+q=n} [x_p, y_q]. \tag{5.6}$$

In this fashion $\tilde{L}(X, R)$ is a graded Lie algebra, separate and complete.

Let us define $\tilde{A}(X, R)^+$ the module generated by words of weight ≥ 1. In the same fashion, we define $\tilde{L}(X, R)^+$ which may be included in $\tilde{A}(X, R)^+$. We can then define the exponential and the logarithm by:

$$\begin{array}{rccc} \exp: & \tilde{A}(X, R)^+ & \longrightarrow & 1 + \tilde{A}(X, R)^+ \\ & x & \longmapsto & \sum_{n\geq 0} \frac{x^n}{n!}. \\ \log: & 1 + \tilde{A}(X, R)^+ & \longrightarrow & \tilde{A}(X, R)^+ \\ & x & \longmapsto & -\sum_{n\geq 1} \frac{(1-x)^n}{n}. \end{array}$$

The logarithm and the exponential are reciprocal maps, and we get an equivalent of the Campbell-Hausdorff theorem:

LEMMA 9. — *Given $x, y \in \tilde{L}(X, R)^+$, we then have*

— $\exp(x)\exp(y) = \exp(z)$, *where $z \in \tilde{L}(X, R)^+$,*

— $z = z_p + z_q + \sum_{n \geq p+q} z_n$ *and* $z_p + z_q = x_p + y_q$,

— *For any* $m \geq p+q$, $z_m \in \tilde{L}_m(x_p, \ldots, x_m, y_q, \ldots, y_m)$.

Given $x, y \in \tilde{L}(X,R)^+$, *we then have*

— $\exp(x)\exp(y) = \exp(z)$, *where* $z \in \tilde{L}(X,R)^+$,

— *For any* $n \in \mathbb{N}$, $z_n - x_n - y_n \in \tilde{L}_n(x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1})$.

Remark. — $\tilde{L}_m(x_p, \ldots, x_m, y_q, \ldots, y_m)$ is the submodule of elements of weight m of the subalgebra generated by $(x_p, \ldots, x_m, y_q, \ldots, y_m)$, or the submodule of elements of weight m of free Lie algebra on the alphabet $\{x_p, \ldots,_m, y_q, \ldots, y_m\}$, which ultimately amounts to the same thing.

The fact that $\exp(x)\exp(y) = \exp(z)$, where $z \in \tilde{L}(X,R)$, is a consequence of the Campbell-Hausdorff theorem, (theorem7) and is due to the fact that $\prod_{n\geq 1} L_n(X) \simeq \prod_{n\geq 1} \tilde{L}_n(X)$. We also know that in $\hat{L}(X,R)$

$$z = x + y + \sum_{n\geq 2} z_n \quad \text{where} \quad z_n \in L_n(X),$$

and for any $u \in L_n(x,y)$, we have $|u|_* \geq p+q$.

***

Based on this lemma, we can then state the following property.

PROPOSITION 10 (Factorization in Product of Exponentials, [53]).
For any $k \in \tilde{L}(X,R)^+$, *there exists a unique series* $g \in \tilde{L}(X,R)^+$ *such that*

$$\exp(k) = \cdots \exp(g_n) \cdots \exp(g_1). \tag{5.7}$$

Proof. — We construct series g term by term, and we show that for any $n \geq 1$, there exists a unique $k^{(n)} \in \prod_{p>n} \tilde{L}_p(X)$, such that

$$\exp(k)\exp(-g_1) \cdots \exp(-g_n) = \exp(k^{(n)}). \tag{5.8}$$

- If $n = 1$, as a consequence of lemma 9, for $g_1 = k_1$,

  $$\exp(k^{(1)}) = \exp(k)\exp(-g_1)$$

  with $k_1^{(1)} = k_1 - g_1 = 0$, therefore $k^{(1)} \in \prod_{n>1} \tilde{L}_n(X)$.

- Let us suppose property (5.8) demonstrated at rank p, we then get

  $$\exp(k)\exp(-g_1) \cdots \exp(-g_p) = \exp(k^{(p)}).$$

  Let us take $g_{p+1} = k_{p+1}^{(p)}$. As a consequence of lemma 9, we still have

  $$\exp(k^{(p)})\exp(-g_{p+1}) = \exp(k^{(p+1)}),$$

  where $k_{p+1}^{(p+1)} = k_{p+1}^{(p)} - g_{p+1} = 0$, which confirms property (5.8) at rank $p+1$.

From this, we deduce the result by going to the limit.

6. Lie transformations

Here, we focus on Lie automorphisms, that is to say on Lie algebra automorphisms of $\tilde{L}(X,R)$. More specifically, we focus on mappings near the identity, that is to say for any $x \in X$, we have:

$$Tx - x \in \prod_{n>|x|_*} \tilde{L}_n(X,R). \tag{6.1}$$

For $x \in M(X)$, let us denote $t_i x = (Tx)_{|x|_*+i}$. We then deduce from $[Tx, Ty] = T[x,y]$, that $t_0[x,y] = [x,y]$ and $x \mapsto t_1 x$ is a derivation:

$$[t_1 x, y] + [x, t_1 y] = t_1[x,y]. \tag{6.2}$$

The set of derivations from $\tilde{L}(X,R)$ is a Lie algebra. We have proposition (cf. [49]):

Theorem 11. — *If D is a derivation from $\tilde{L}(X)$, then*

- — $\exp(tD)$ *is a Lie automorphism, for any $t \in R$,*
- — *The Lie algebra of the group of Lie automorphisms is identified to the set of derivations.*

6.1. Adjoint Lie Algebra

We define $L(x) = L_x$, the inner derivation $y \mapsto [x,y]$, also called Lie operator. We have, in $\tilde{L}(X)$,

$$L(\textstyle\sum_{n\geq 0} f_n) = \sum_{n\geq 0} L(f_n). \tag{6.3}$$

The set of inner derivations is a Lie algebra, called the adjoint Lie algebra, for the Lie bracket $[L_f, L_g] = L_f L_g - L_g L_f$ and we have the following property, based on the Jacobi identity:

$$[L_f, L_g] = L_{[f,g]}. \tag{6.4}$$

We then define the group of Lie transformations or canonical transformations as the group of automorphisms of $\tilde{L}(X)$ near the identity, in the previously defined sense. The group of canonical transformations operates on adjoint Lie algebra by automorphisms. We have

$$TL_f T^{-1} g = T[f, T^{-1}g] = [Tf, g] \tag{6.5}$$

therefore

$$TL_f T^{-1} = L_{Tf}. \tag{6.6}$$

Some of the elements of the group of Lie automorphisms near the identity are

- the exponential of an inner derivation L_f defined by

$$\exp(L_f)g = \sum_{i\geq 0} \frac{L_f^i}{i!} g. \tag{6.7}$$

If $f \in \tilde{L}(X)^+$, $g \in \tilde{L}(X)$, $G = \exp(L_f)g$ is defined by

$$\left\{ \begin{array}{lcl} G_0 & = & g_0, \\ G_n & = & \sum_{p=0}^n G_{p,n-p}, \end{array} \right. \quad \text{where} \quad \left\{ \begin{array}{lcl} G_{0,n} & = & g_n, \\ G_{p,q} & = & \sum_{p_1+\cdots+p_k=p} \dfrac{L_{f_{p_1}} \cdots L_{f_{p_k}}}{k!} g_q. \end{array} \right. \tag{6.8}$$

Remark. — Here $\sum_{p_1+\cdots+p_k=p}$ designates $\sum_{k,p_1,\ldots,p_k;p_1+\cdots+p_k=p}$.

- Given $f \in \tilde{L}(X)^+$, we define the mapping T_f as follows: if $g \in \tilde{L}(X)$, $F = T_f g$ is defined by

$$F_n = \sum_{p=0}^n (T_f)_p g_{n-p} \quad \text{where} \quad (T_f)_0 = I, \quad (T_f)_n = -\sum_{p=1}^n \tfrac{p}{n} (T_f)_{n-p} L_{f_p}. \tag{6.9}$$

Its inverse (shown at a later stage) T_f^{-1}, is defined in similar fashion. For $G = T_f^{-1} g$, we have

$$G_n = \sum_{p=0}^n (T_f^{-1})_p g_{n-p} \quad \text{where} \quad (T_f^{-1})_0 = I, \quad (T_f^{-1})_n = \sum_{p=1}^n \tfrac{p}{n} L_{f_p} (T_f^{-1})_{n-p}. \tag{6.10}$$

The previous relations also enable us to write

$$\left\{ \begin{array}{lcl} G_0 & = & g_0, \\ G_n & = & \sum_{p=0}^n G_{p,n-p}, \end{array} \right. \quad \text{where} \quad \left\{ \begin{array}{lcl} G_{0,n} & = & g_n, \\ G_{p,q} & = & \sum_{k=1}^p \frac{k}{p} [f_k, G_{p-k,q}]. \end{array} \right. \tag{6.11}$$

We use the transformation T_f^{-1} rather than T_f, because it is simpler to define recursively.

⁂

Before verifying that these operators are Lie automorphisms, we shall give a characterization of the type T_f transformations, which will enable us to verify that T_f and T_f^{-1} are in fact inverses of each other.

Let us consider a smallness parameter ε and

$$f(\varepsilon) = \sum_{n\geq 0} \varepsilon^n f_n \in \tilde{L}(X, R)[\![\varepsilon]\!].$$

Let us pose $T_{f(\varepsilon)} = \sum_{n\geq 0} \varepsilon^n (T_f)_n$. By identifying its members term by term, one sees that $T_{f(\varepsilon)}$ satisfies differential equation

$$\frac{\partial T_{f(\varepsilon)}}{\partial \varepsilon} = -T_{f(\varepsilon)} L_{\frac{\partial f}{\partial \varepsilon}}. \tag{6.12}$$

Conversely, $T_{f(\varepsilon)}$ is the solution of (6.12) equal to the identity at $\varepsilon = 0$. We then deduce the following characterization

PROPOSITION 12. — *The transformations T_f are the solutions for $\varepsilon = 1$ of the differential equation*

$$\frac{\partial T_{f(\varepsilon)}}{\partial \varepsilon} = -T_\varepsilon L_{\frac{\partial f(\varepsilon)}{\partial \varepsilon}},\ T_{f(0)} = I. \tag{6.13}$$

In the same fashion, we characterize T_f^{-1} by the differential equation

$$\frac{\partial T^{-1}_{f(\varepsilon)}}{\partial \varepsilon} = L_{\frac{\partial f(\varepsilon)}{\partial \varepsilon}} T^{-1}_{f(\varepsilon)}. \tag{6.14}$$

We then deduce that

$$\frac{\partial}{\partial \varepsilon} T_{f(\varepsilon)} T^{-1}_{f(\varepsilon)} = -T_{f(\varepsilon)} L_{\frac{\partial f(\varepsilon)}{\partial \varepsilon}} T^{-1}_{f(\varepsilon)} + T_{f(\varepsilon)} L_{\frac{\partial f(\varepsilon)}{\partial \varepsilon}} T^{-1}_{f(\varepsilon)} = 0, \tag{6.15}$$

that is to say $T_f T_f^{-1} = T_{f(\varepsilon)} T^{-1}_{f(\varepsilon)} = I$.

⁂

Let us now verify that these transformations are Lie transformations. We use lemma ([53])

LEMMA 13. — *For any $k \geq 0$: $L_f^k[g,h] = \sum_{i=0}^k \binom{k}{i} \left[L_f^i g, L_f^{k-i} h\right]$.*

We demonstrate this property recursively, by using the Jacobi identity:

$$[f,[u,v]] = -[u,[v,f]] - [v,[f,u]] = [u,[f,v]] + [[f,u],v]. \tag{6.16}$$

At rank $k+1$, we describe

$$\begin{aligned}
L_f^{k+1}[g,h] &= L_f L_f^k [g,h] \\
&= L_f \sum_{i=0}^{k} \binom{k}{i} \left\{ \left[L_f^i g, L_f^{k-i} h\right] + \left[L_f^i g, L_f^{k-i} h\right] \right\} \\
&= \sum_{i=0}^{k} \binom{k}{i} \left\{ \left[L_f^{i+1} g, L_f^{k-i} h\right] + \left[L_f^i g, L_f^{k+1-i} h\right] \right\} \\
&= \sum_{i=0}^{k+1} \binom{k+1}{i} \left[L_f^i g, L_f^{k+1-i} h\right].
\end{aligned} \tag{6.17}$$

⁂

We then deduce, for the exponential

$$\begin{aligned}
\exp(L_f)[g,h] &= \sum_{n\geq 0} \frac{L_f^n}{n!}[g,h] && (6.18)\\
&= \sum_{n\geq 0} \frac{1}{n!} \sum_{p=0}^{n} \binom{n}{p} \left[L_f^p g, L_f^{n-p} h\right] && (6.19)\\
&= \sum_{p+q\geq 0} \frac{1}{p+q!} \frac{p+q!}{p!q!} \left[L_f^p g, L_f^q h\right] && (6.20)\\
&= [\sum_{p\geq 0} \tfrac{1}{p!} L_f^p g, \sum_{q\geq 0} \tfrac{1}{q!} L_f^q h] && (6.21)\\
&= [\exp(L_f) g, \exp(L_f) h]. && (6.22)
\end{aligned}$$

⁂

For the transformation T_f^{-1}, posing $G = T_f^{-1}g$, $H = T_f^{-1}h$ and $F = T_f^{-1}[g,h]$, we demonstrate (cf. [24]) that for any $k \geq 1$, we have:

$$F_k = \sum_{i=0}^{k} [G_i, H_{k-i}] = [G, H]_k. \tag{6.23}$$

This is shown recursively by considering

$$\begin{aligned}
F_n &= \sum_{k=1}^{n} \tfrac{k}{n}[f_k, F_{n-k}] && (6.24)\\
&= \sum_{k=1}^{n} \tfrac{k}{n}\Big[f_k, [G,H]_{n-k}\Big] && (6.25)\\
&= \sum_{k=1}^{n}\sum_{i=0}^{n-k} \tfrac{k}{n}[f_k, [G_i, H_{n-k-i}]] && (6.26)\\
&= \sum_{k=1}^{n}\sum_{i=0}^{n-k} \tfrac{k}{n}\{[G_i, [f_k, H_{n-k-i}]] + [[f_k, G_i], H_{n-k-i}]\} && (6.27)\\
&= \sum_{i=0}^{n-1}\sum_{k=0}^{n-i} \tfrac{k}{n}\{[G_i, [f_k, H_{n-k-i}]] + [[f_k, G_{n-k-i}], H_i]\} && (6.28)\\
&= \sum_{i=0}^{n-1} \tfrac{n-i}{n}[G_i, H_{n-i}] + \sum_{i=1}^{n} \tfrac{i}{n}[G_i, H_{n-i}] && (6.29)\\
&= \sum_{i=1}^{n} [G_i, H_{n-i}]. && (6.30)
\end{aligned}$$

⁂

These two transformations are noteworthy in that they are entirely determined by a Lie series called the associate generative series. Thus in the group of canonical transformations we have at least three types of transformations:

— The exponentials of inner derivations, $\exp(L_k)$, and their inverses, which are also exponentials

— Type T_f transformations and their inverses T_f^{-1}

— Type M_g or M_g^{-1} transformations, also called Dragt-Finn transformations. To $g \in \tilde{L}(X)$ we associate the product of exponentials

$$M_g^{-1} = \cdots \exp(L_{g_n}) \cdots \exp(L_{g_1}). \tag{6.31}$$

These three transformations may be defined as formal series of Lie operators $\sum_{n\geq 0} T_n$, so as to have for any $f \in \tilde{L}(X)$

$$Tf = \sum_{n\geq 0}\sum_{p=0}^{n} T_p f_{n-p}. \tag{6.32}$$

— for the exponential $\exp(L_k)$,

$$T_0 = I, \quad T_n = \sum_{p_1+\cdots+p_m=n} \frac{L_{k_{p_1}} \cdots L_{k_{p_m}}}{m!}, \tag{6.33}$$

— for the transformation T_f^{-1},

$$T_0 = I, \quad T_n = \sum_{k=1}^{n} \tfrac{k}{n} L_{w_k} T_{n-k}, \tag{6.34}$$

and, consequently, for the transformation T_f,

$$T_0 = I, \quad T_n = -\sum_{k=1}^{n} \tfrac{k}{n} T_{n-k} L_{w_k}. \tag{6.35}$$

The developed formula of T_n is given by

$$T_n = \sum_{k_1+\cdots+k_j=n} \tfrac{k_1}{k_1} \tfrac{k_2}{k_1+k_2} \cdots \tfrac{k_j}{k_1+\cdots+k_j} L_{w_{k_1}} \cdots L_{w_{k_j}}. \tag{6.36}$$

— for the product of exponentials (inverse Dragt-Finn transformation),

$$T_0 = I, \quad T_n = \sum_{\wp(1,n;n)} \frac{L_{g_n}^{m_n} \cdots L_{g_1}^{m_1}}{m_n! \cdots m_1!}, \tag{6.37}$$

and as its inverse (that is to say the Dragt-Finn transformation)

$$T_0 = I, \quad T_n = \sum_{\wp(1,n;n)} (-1)^{m_1+\cdots+m_p} \frac{L_{g_1}^{m_1} \cdots L_{g_n}^{m_n}}{m_1! \cdots m_n!}. \tag{6.38}$$

Here $\wp(p, q; n)$ represents partitions of n, made [1]out from p and q,

$$\wp(p, q; n) = \{(m_p, \ldots, m_q); pm_p + \cdots + qm_q = n\}. \tag{6.39}$$

6.2. Relations between Transformations

Let us show that these three transformations are but three different ways of representing the same transformation. More specifically, we demonstrate by steps

PROPOSITION 14. — *Given $w, g, k \in \tilde{L}(X, \mathbb{Q})^+$, there exist*

— *$g' \in \tilde{L}(X)^+$, such that $T_w = M_{g'}$, and for any $n \geq 1$, $g'_n - w_n \in L(w_1, \ldots, w_{n-1})$,*

— *$k' \in \tilde{L}(X)^+$, such that $M_g = \exp(L_{k'})$, and for any $n \geq 1$, $k'_n - g_n \in L(g_1, \ldots, g_{n-1})$,*

— *$w' \in \tilde{L}(X)^+$, such that $\exp(L_k) = T_{w'}$, and for any $n \geq 1$, $w'_n - k_n \in L(k_1, \ldots, k_{n-1})$.*

From proposition 9 and relation $[L_f, L_g] = L_{[f,g]}$, we deduce that the exponential of an inner derivation may be written as a product of inner derivations of exponentials, that is to say, given $k \in \tilde{L}(X)^+$, there exists $g \in \tilde{L}(X)^+$ such that

$$\exp(L_k) = \cdots \exp(L_{g_n}) \cdots \exp(L_{g_1}), \tag{6.40}$$

and for any $n \geq 1$, $g_n \in L(k_1, \cdots, k_n)$ and $g_n - k_n \in L(k_1, \cdots, k_{n-1})$, and conversely.

[1] "Sommants" in the French original.

$\star^\star\star$

Here, we show how a product of exponentials (Dragt-Finn transformation) may be written as a type T_f transformation, and conversely. This result was partially demonstrated by Finn [19], but he does not explicitly give the relations between generators.

PROPOSITION 15. — *Given a generating series $g(\varepsilon) = \sum_{n\geq 1} \varepsilon^n g_n$, and corresponding Dragt-Finn transformation*

$$M_{g(\varepsilon)} = \exp(-\varepsilon L_{g_1}) \cdots \exp(-\varepsilon^n L_{g_n}) \cdots,$$

there exists a unique series $w(\varepsilon) = \sum_{n\geq 1} \varepsilon^n w_n$ such that

$$T_{w(\varepsilon)} = M_{g(\varepsilon)}.$$

We have, for any n,

$$w_n = \sum_{1\leq k\leq n} \frac{k}{n} \sum_{\wp(k+1,n-k;n-k)} \frac{L_{g_{n-k}}^{m_{n-k}} \cdots L_{g_{k+1}}^{m_{k+1}}}{m_{k+1}! \cdots m_{n-k}!} g_k.$$

Proof. — By differentiating $M_{g(\varepsilon)}$ in relation to ε, we get

$$\begin{aligned}
\frac{\partial M_{g(\varepsilon)}}{\partial \varepsilon} &= \sum_{n\geq 1} [\exp(-\varepsilon L_{g_1}) \cdots \exp(-\varepsilon^{n-1} L_{g_{n-1}})] \left[\frac{\partial}{\partial \varepsilon} [\exp(-\varepsilon^n L_{g_n})]\right] [\exp(-\varepsilon^{n+1} L_{g_{n+1}}) \cdots] \\
&= \sum_{n\geq 1} [\exp(-\varepsilon L_{g_1}) \cdots \exp(-\varepsilon^{n-1} L_{g_{n-1}})] \left[-n\varepsilon^{n-1} L_{g_n}\right] [\exp(-\varepsilon^n L_{g_n}) \cdots] \\
&= M_{g(\varepsilon)} \sum_{n\geq 1} M_{g(\varepsilon)}^{-1} [\exp(-\varepsilon L_{g_1}) \cdots \exp(-\varepsilon^{n-1} L_{g_{n-1}})] \left[-n\varepsilon^{n-1} L_{g_n}\right] [\exp(-\varepsilon^n L_{g_n}) \cdots] \\
&= M_{g(\varepsilon)} \sum_{n\geq 1} [\cdots \exp(\varepsilon^n L_{g_n})] \left[-n\varepsilon^{n-1} L_{g_n}\right] [\exp(-\varepsilon^n L_{g_n}) \cdots] \\
&= M_{g(\varepsilon)} L \left[\sum_{n\geq 1} -n\varepsilon^{n-1} [\cdots \exp(\varepsilon^n L_{g_n})]\, g_n\right]. \qquad (6.41)
\end{aligned}$$

Thus we deduce that

$$\frac{\partial M_{g(\varepsilon)}}{\partial \varepsilon} = -M_{g(\varepsilon)} L(S),$$

where

$$\begin{aligned}
S &= \sum_{n\geq 1} n\varepsilon^{n-1} \left[\cdots \exp(\varepsilon^{n-1} L_{g_{n-1}})\right] g_n \qquad (6.42) \\
&= \sum_{n\geq 1} \varepsilon^{n-1} \left[\sum_{k=1}^{n} k \sum_{\wp(k+1,n-k;n-k)} \frac{L_{g_{n-k}}^{m_{n-k}} \cdots L_{g_{k+1}}^{m_{k+1}}}{m_{k+1}! \cdots m_{n-k}!} g_k\right], \qquad (6.43)
\end{aligned}$$

and $\wp$ is defined in (6.39). On the other hand, according to (6.12), and initial conditions $T_0 = M_0 = I$, a necessary and sufficient condition to get $T_{w(\varepsilon)} = M_{g(\varepsilon)}$ is that

$$S = \frac{\partial w(\varepsilon)}{\partial \varepsilon},$$

that is to say, considering each term of the formal series,

$$\begin{aligned} w_n &= \sum_{1\le k\le n} \frac{k}{n} \sum_{\wp(k+1,n-k;n-k)} \frac{L_{g_{n-k}}^{m_{n-k}} \cdots L_{g_{k+1}}^{m_{k+1}}}{m_{k+1}! \cdots m_{n-k}!} g_k \\ &= g_n + \sum_{1\le k\le n-1} \frac{k}{n} \sum_{\wp(k+1,n-k;n-k)} \frac{L_{g_{n-k}}^{m_{n-k}} \cdots L_{g_{k+1}}^{m_{k+1}}}{m_{k+1}! \cdots m_{n-k}!} g_k. \end{aligned} \tag{6.44}$$

These explicit relations enable us to prove that, for any n, $w_n \in L(g_1, \ldots, g_n)$, and $w_n - g_n \in L(g_1, \ldots, g_{n-1})$. The triangular form of the relations thus leads us to assert that, for any n, $g_n \in L(w_1, \ldots, w_n)$ and $g_n - w_n \in L(w_1, \ldots, w_{n-1})$.

Taking $\varepsilon = 1$, we deduce the desired result for the Lie series.

***

The relation between the exponential and type T_f^{-1} transformation is deduced from the two previous relations.

6.3. Composition of Transformations

By considering the family of T_f Lie transformations and the differential equation which characterizes it (proposition 12), we show that the composition of two T_f transformations is obviously a T_f transformation, and we find the relations between generators.

Let us consider the two families of Lie transformations $T_{u(\varepsilon)}$ and $T_{v(\varepsilon)}$. From (6.12), we have

$$\frac{\partial T_{u(\varepsilon)}}{\partial \varepsilon} = -T_{u(\varepsilon)} L_{\frac{\partial u(\varepsilon)}{\partial \varepsilon}} \quad \text{and} \quad \frac{\partial T_{v(\varepsilon)}}{\partial \varepsilon} = -T_{v(\varepsilon)} L_{\frac{\partial v(\varepsilon)}{\partial \varepsilon}}, \tag{6.45}$$

whence we deduce that

$$\begin{aligned} \frac{\partial}{\partial \varepsilon} T_{u(\varepsilon)} T_{v(\varepsilon)} &= -T_{u(\varepsilon)} L_{\frac{\partial u(\varepsilon)}{\partial \varepsilon}} T_{v(\varepsilon)} - T_{u(\varepsilon)} T_{v(\varepsilon)} L_{\frac{\partial v(\varepsilon)}{\partial \varepsilon}} \\ &= -T_{u(\varepsilon)} T_{v(\varepsilon)} \left[T_{v(\varepsilon)}^{-1} L_{\frac{\partial u(\varepsilon)}{\partial \varepsilon}} T_{v(\varepsilon)} + L_{\frac{\partial v(\varepsilon)}{\partial \varepsilon}} \right] \\ &= -T_{u(\varepsilon)} T_{v(\varepsilon)} L \left[T_{v(\varepsilon)}^{-1} \tfrac{\partial u(\varepsilon)}{\partial \varepsilon} + \tfrac{\partial v(\varepsilon)}{\partial \varepsilon} \right], \end{aligned} \tag{6.46}$$

which proves that $T_{u(\varepsilon)} T_{v(\varepsilon)}$ is the Lie transformation associated with $w(\varepsilon)$ where

$$\tfrac{\partial w(\varepsilon)}{\partial \varepsilon} = T_{v(\varepsilon)}^{-1} \tfrac{\partial u(\varepsilon)}{\partial \varepsilon} + \tfrac{\partial - v(\varepsilon)}{\partial \varepsilon}, \tag{6.47}$$

which uniquely determines w, since $w_0 = 0$. More specifically, (6.47) is written term by term:

$$\begin{aligned} n w_n &= \sum_{p=0}^{n-1} (T_v)_{n-p-1}^{-1} ((p+1) u_{p+1}) + n v_n & (6.48) \\ &= \sum_{p=1}^{n} (T_v)_{n-p}^{-1} p u_p + n v_n. & (6.49) \end{aligned}$$

From this property and from the relations between the various representations of canonical transformations, we show that the composition of two exponentials is an exponential (which we already knew), that the composition of two T_f transformations is a T_f transformation, that the composition of two M_g transformations is an M_g transformation. Hence we deduce the following result:

PROPOSITION 16. — *The set of exponentials of Lie operators, the set of T_f transformations, and the set of M_g transformations, are the same subgroup of the set of Lie automorphisms near the identity.*

Remark. — We can dispense with T_f^{-1} or M_g^{-1} inverses by using proposition 14.

We denote by $\mathbf{G}$, the subgroup of Lie automorphisms near the identity generated by exponentials of Lie operators. We then have the following proposition:

PROPOSITION 17. — *If any derivation of $L(X)$ is an inner derivation, then $\mathbf{G}$ is the group of Lie automorphisms near the identity.*

Remark. — The proof will also be implemented within the framework of a Poisson bracket algebra where a less strong property is satisfied (cf. proposition 25, page 43).

Given T, a Lie automorphism near the identity. We will show that there exists $g \in \tilde{L}(X)^+$, such that $T = M_g$, and such that, for any $p \geq 0$ and any $x \in M(X)$, posing $T^{(0)} = T$ and $T^{(p)} = \exp(-L_{g_p}) \cdots \exp(-L_{g_1})T$, we get:

$$T^{(p)}x - x \in \prod_{n > p+|x|_*} \tilde{L}_n(X). \tag{6.50}$$

• For $p = 0$, (6.50) is verified for T by assumption.
• Let us suppose $(g_1, \ldots, g_p) \in \tilde{L}_1(X) \times \cdots \times \tilde{L}_p(X)$ are constructed and satisfy (6.50). $T^{(p)}$ is a Lie automorphism near the identity, as a product of Lie automorphisms near the identity and, consequently, for any $x, y \in M(X) \subset \tilde{L}(X)$, we have

$$T^{(p)}[x, y] = [T^{(p)}x, T^{(p)}y]. \tag{6.51}$$

Hence we deduce that

$$\begin{aligned}
(T^{(p)}[x,y])_{|[x,y]|_*} &= [(T^{(p)}x)_{|x|_*}, (T^{(p)}y)_{|y|_*}] = [x,y], && (6.52)\\
(T^{(p)}[x,y])_{|[x,y]|_*+k} &= 0, \quad 1 \leq k \leq p, && (6.53)\\
(T^{(p)}[x,y])_{|[x,y]|_*+p+1} &= [(T^{(p)}x)_{|x|_*+p+1}, y] + [x, (T^{(p)}y)_{|y|_*+p+1}]. && (6.54)
\end{aligned}$$

Let us define $t_p : L(X) \to L(X)$, by linear prolongation

$$x \in M(X) \mapsto (T^{(p)}x)_{|x|_*+p+1} \in \tilde{L}_{|x|_*+p+1}(X) \subset L(X). \tag{6.55}$$

We thus deduce that, for any $x, y \in M(X)$, and for any $x, y \in L(X)$, that t_p is a derivation, therefore, an inner derivation. Since $t_p x \in \tilde{L}_{|x|_*+p+1}$, for $x \in M(X)$, we then deduce that there exists $g_{p+1} \in \tilde{L}_{p+1}(X)$, such that $t_p = -L_{g_{p+1}}$. Thus composing to the left by $\exp(L_{g_{p+1}})$, we deduce, for any $x \in M(X)$, that

$$T^{(p+1)}x - x \in \prod_{n > |x|_*+p+1} \tilde{L}_n(X). \tag{6.56}$$

In the limit, we deduce that $T = M_g^{-1}$, therefore that $T \in \mathbf{G}$.

$\circ^{\circ}\circ$

In this section, we have essentially established the fact that the set of exponentials of Lie operators, the set of T_f transformations, the set of M_g transformations, are the same subgroup of the group of Lie automorphisms near the identity.

We based our argument on the fact that it is possible to write an exponential as a T_f or M_g transformation, and conversely, and that the sum of exponentials of Lie operators form a set, which is a consequence of the Campbell-Hausdorff theorem.

We have given an explicit formula between the generator of the M_g transformation, and the generator of the T_f transformation, which enabled us to link these two transformations (an approximated result had been demonstrated by Finn [19], but he had failed to show that the generators belonged to $\tilde{L}(X)$).

7. Computation of the Relations

Here, we give a method which is general enough to enable us to write an element of $\mathbf{G}$ in the form of one of the three transformations previously studied. These techniques thus allow us to retrieve the Baker-Campbell-Hausdorff formulas by using manipulations of Lie polynomials only, without resorting to the associative algebra.

$\star^{\star}\star$

Originally, I had been able to find a means of computing the relations between the generators of the T_f^{-1}, and the M_g^{-1} transformations, without knowing general form (6.44), and by using the Hall basis, since I only learned of the algorithms for the Lyndon basis later.

The idea was to find the relations between w and g through direct identification of $(T_w^{-1})_n h_0$ and $(M_g^{-1})_n h_0$. Because of the form of the expansion, we can write L_{w_n} in relation to the L_{g_i}'s, and conversely. On the other hand, in order to find the relations between the w_i and the g_i, one has to be able to recognize Lie brackets in an expanded expression, that is to say, it is equivalent to the general problem of recognizing Lie polynomials in enveloping algebra.

For example, from $[L_f, L_g] = L_f L_g - L_g L_f = L_{[f,g]}$, the equalities $(M_g)_i^{-1} = (T_w)_i^{-1}$ induce

$$L_{g_1} = L_{w_1}, L_{g_2} = L_{w_2}, L_{g_3} = L_{w_3} + \tfrac{1}{3}L_{w_1}L_{w_2} - \tfrac{1}{3}L_{w_2}L_{w_1},$$
$$L_{g_4} = L_{w_4} + \tfrac{1}{4}L_{w_1}L_{w_3} - \tfrac{1}{4}L_{w_3}L_{w_1} + \tfrac{1}{12}L_{w_1}^2 L_{w_2} + \tfrac{1}{12}L_{w_2}L_{w_1}^2 - \tfrac{1}{6}L_{w_1}L_{w_2}L_{w_1},$$

that is to say,

$$g_3 = w_3 + \tfrac{1}{3}[w_1, w_2] \quad \text{and} \quad g_4 = w_4 + \tfrac{1}{4}[w_1, w_3] + \tfrac{1}{12}[w_1, [w_1, w_2]], \tag{7.1}$$

and so on and so forth, identification becoming increasingly difficult.

In the special case of the computation of the Lie transformation generator as a function of a product of exponentials, we can use $\tilde{L}(X)$ where $X = \{h_0, g_1, \ldots, g_n\}$, and decompose $(T_w^{-1})_n h_0 = (M_g^{-1})_n h_0$ onto the Hall basis.

For $i_{n-q} \neq 0$ and $i_{k+p} \neq 0$ in (6.44), we get the following decomposition onto the Hall basis:

$$L_{g_{n-q}}^{i_{n-q}} \cdots L_{g_{k+p}}^{i_{k+p}} g_k = \overbrace{[g_{n-q}, [\cdots,}^{i_{n-q}} [\cdots, \overbrace{[g_{k+p}, [\cdots,}^{i_{k+p}} g_k] \cdots] \tag{7.2}$$

$$= -[\overbrace{g_{n-q}, [\cdots, [}^{i_{n-q}}\cdots, [\overbrace{g_{k+p}, [\cdots,}^{i_{k+p}-1} [g_k, g_{k+p}]\cdots]. \quad (7.3)$$

the general term in (6.44), thus decomposes onto the Hall basis into

$$L\left[L_{g_{n-q}}^{i_{n-q}}\cdots L_{g_{k+p}}^{i_{k+p}}g_k\right]h_0 = \left[L_{g_{n-q}}^{i_{n-q}}\cdots L_{g_{k+p}}^{i_{k+p}}g_k, h_0\right] \quad (7.4)$$

$$= [h_0, [\overbrace{g_{n-q}, [\cdots, [}^{i_{n-q}}\cdots, [\overbrace{g_{k+p}, [\cdots,}^{i_{k+p}-1} [g_k, g_{k+p}]\cdots]. \quad (7.5)$$

We deduce that, if $\sum \alpha_i G_{i,n}$ is the decomposition onto the Hall basis of $w_n - g_n = G_n$, thus the decomposition onto the Hall basis of $[w_n, h_0] - [g_n, h_0]$ is $= -\sum \alpha_i [h_0, G_{i,n}]$. At this point, all we need to do is solve term by term in relation to the g_i :

$$\begin{aligned}
0 &= (T_w^{-1})_n h_0 - (M_g^{-1})_n h_0 && (7.6)\\
&= [w_n, h_0] + \sum_{p=1}^{n-1} \tfrac{p}{n} L_{w_p}(T_w^{-1})_{n-p}h_0 - [g_n, h_0] - \sum_{\wp(1,n-1;n)} \frac{L_{g_{n+1}}^{m_{n-1}}\cdots L_{g_1}^{m_1}}{m_1!\cdots m_{n-1}!}h_0 && (7.7)\\
&= [w_n, h_0] - [g_n, h_0] + \sum_{p=1}^{n-1} \tfrac{p}{n} L_{w_p}(M_g^{-1})_{n-p}h_0 - \sum_{\wp(1,n-1;n)} \frac{L_{g_{n+1}}^{m_{n-1}}\cdots L_{g_1}^{m_1}}{m_1!\cdots m_{n-1}!}h_0 && (7.8)\\
&= [w_n, h_0] - [g_n, h_0] + \sum_i \alpha_i [h_0, G_{i,n}]. && (7.9)
\end{aligned}$$

We have posed $w_p = g_p + G_p$ for $1 \le p \le n$, thus $T_p^{-1} = M_p^{-1}$, and the right hand side of the previous formula is necessarily $[h_0, G_n]$. Furthermore, it is unnecessary to compute $(M_g^{-1})_n h_0$, because the decomposition

$$[g_n, h_0] + \sum_{\wp(1,n-1;n)} \frac{[g_{n+1}, [\cdots [g_1, h_0]]]}{m_1!\cdots m_{n-1}!}, \quad (7.10)$$

does not provide terms for the form $[h_0,]$.

So step by step, we get the generating series of the Lie transformation as a function of the ascending product of exponentials. However, the success of this technique rests on the most peculiar form of the G_n. To confirm this, we observe that the same method to compute the same relations –though in reverse–, stumbles on the term of order 7. In the Hall basis, we have

$$\begin{aligned}
g_7 = w_7 &+ \tfrac{1}{7}[w_1, w_6] + \tfrac{2}{7}[w_2, w_5] + \tfrac{3}{7}[w_3, w_4] + \tfrac{1}{42}[w_1, [w_1, w_5]] + \tfrac{11}{105}[w_2, [w_1, w_4]]\\
&+ \tfrac{4}{35}[w_2, [w_2, w_3]] + \tfrac{5}{28}[w_3, [w_1, w_3]] - \tfrac{1}{21}[w_4, [w_1, w_2]] + \tfrac{1}{210}[w_1, [w_1, [w_1, w_4]]]\\
&+ \tfrac{1}{28}[w_2, [w_1, [w_1, w_3]]] + \tfrac{19}{210}[w_2, [w_2, [w_1, w_2]]] + \tfrac{31}{420}[w_3, [w_1, [w_1, w_2]]]\\
&+ \tfrac{13}{420}\underline{[[w_1, w_2], [w_1, w_3]]} + \tfrac{1}{840}[w_1, [w_1, [w_1, [w_1, w_3]]]] + \tfrac{1}{70}[w_2, [w_1, [w_1, [w_1, w_2]]]]\\
&+ \tfrac{11}{630}\underline{[[w_1, w_2], [w_1, [w_1, w_2]]]} + \tfrac{1}{2520}[w_1, [w_1, [w_1, [w_1, [w_1, w_2]]]]].
\end{aligned}$$

g_7 contains two terms in which the Lie bracket with z decomposes (the first one) into

$$[z, [[w_1, w_2], [w_1, w_3]]] = [[w_1, w_2], [z, [w_1, w_3]]] - [[w_1, w_3], [z, [w_1, w_2]]] \quad (7.11)$$

Here, we took $h = z > w$. The problem originates in the use of the Hall basis, which, contrary to the Lyndon basis, does not permit a direct factorization of the previous expressions.

7.1. Pseudo-Factorization and Triangular Systems

The Lyndon basis has a property which allows us to circumvent the afore-mentioned drawback:

LEMMA 18. — *Given $X = \{a\} \bigcup X_1$ an alphabet and $a < b$ for any $b \in X_1$. Then for any $x = \sum_i \alpha_i x_i \in L(X_1)$, the decomposition of $[a, x]$ onto the Lyndon basis $\mathcal{L}(X)$ is*

$$[a, x] = \sum_i \alpha_i [a, x_i].$$

We simply need to show that for any $x_i \in \mathcal{L}(X_1)$, we have $[a, x_i] \in \mathcal{L}(X)$, which, according to the construction of the Lyndon basis, is verified, since $a < x_i$, for the lexicographical order on Lyndon words.

***

The previous lemma enables us to state

PROPOSITION 19. — *Given $X = \{a\} \bigcup X_1$ an alphabet and $a < b$ for any $b \in X_1$. The injection*

$$\begin{array}{ccc} \mathcal{L}(X_1) & \rightarrow & \mathcal{L}(X) \\ x & \mapsto & [a, x] \end{array} \tag{7.12}$$

extends itself into an isomorphism of R-module $L_a : L(X_1) \rightarrow L_a L(X_1) \subset L(X)$. Its matrix being the identity, in the corresponding Lyndon bases.

Generally, if X_1 and X_2 are two alphabets satisfying $X_2 \setminus X_1 = x_2 \in X_2$, $L_{x_2} : L(X_1) \rightarrow L(X_2)$ is an injective morphism, and proposition 19 states that the matrix is the identity matrix, by using the Lyndon bases.

Remark. — If we attempt to compare proposition 19 and proposition 6, we reach the following conclusions.

If X_1 is an alphabet of cardinality q, the submodule of $\mathcal{L}_{q+1}(X)$, generated by elements of partial degree 1 in a, has dimension 2^q. Indeed, if we use the isomorphism between the Lyndon words and the Lyndon basis of this submodule, a word w of X_{q+1}^* with partial degree 1 in a, is a Lyndon word if it is smaller than its own conjugates, that is to say if it spells $w = au$ where $u \in X_{1\,q}^*$. Thus the number of such words is 2^q. The range of $L_a : L_q(X_1) \rightarrow L_{q+1}(X)$ is a submodule of dimension l_q in a module of dimension 2^q.

In proposition 6, we set up a submodule isomorphism of dimension l_q, and the range is considered in the homogeneous part of degree q of the associative algebra, whose dimension is also 2^q.

***

Here, we show how a triangular type system can be solved, with or without factorization. A triangular system has the following form

$$\left\{ \begin{array}{rcl} f_p & = & R_p \\ f_{p+1} + F_{p+1} & = & R_{p+1} \\ \vdots & \vdots & \vdots \\ f_{p+n} + F_{p+n} & = & R_{p+n} \end{array} \right. , \tag{7.13}$$

where the unknown quantities are the f_i. For any $p + 1 \leq i \leq n$, $F_i \in L_R(f_p, \ldots, f_{i-1})$ and $R_i \in L_R(X)$ for a given ring R, and a given alphabet X.

PROPOSITION 20. — *Any triangular system can be solved by successive evaluations.*

We immediately get $f_p = R_p$, and it is possible to modify system (7.13) into

$$\left\{\begin{array}{lcl} f_p & = & R_p \\ f_{p+1} & = & R_{p+1} - F_{p+1} \\ \vdots & \vdots & \vdots \\ f_{p+n} & = & R_{p+n} - F_{p+n} \end{array}\right. , \qquad (7.14)$$

then we deduce that $f_{p+1} = R_{p+1} - F_{p+1} = R'_{p+1}$ in which we have evaluated F_{p+1} in $f_p = R_p$ (here f_p is considered as a Lie letter or monomial). After applying this recursive process, we eventually get $f_{p+n} = R_{p+n} - F_{p+n}$ in which we have evaluated F_{p+n} in $(f_p, \dots, f_{p+n}) = (R'_p, \dots, R'_{p+n})$.

We use this algorithm to invert the relations we find, for example, when we compute a generating function in terms of another one.

Let us now suppose that we know that $(f_p, \dots, f_{p+n}) = (R'_p, \dots, R'_{p+n})$ satisfies a factorizable triangular system —which is the case, for example, when one looks for Baker-Campbell-Hausdorff relations— with form

$$\left\{\begin{array}{rcl} [a, f_p] & = & R_p \\ [a, f_{p+1}] + F_{p+1} & = & R_{p+1} \\ \vdots & \vdots & \vdots \\ [a, f_{p+n}] + F_{p+n} & = & R_{p+n} \end{array}\right. , \qquad (7.15)$$

We have $[a, f_p] = R_p = \sum_i \alpha_i^{(p)}[a, t_i^{(p)}]$, which yields $f_p = \sum_i \alpha_i^{(p)} t_i^{(p)}$. Similarly, from $(f_p, \dots, f_{p+n-1}) = (R'_p, \dots, R'_{p+n-1})$, we deduce that $[a, f_{p+n}] = R''_{p+n}$ which we evaluate according to the previously found values, and must transform itself into $[a, f_{p+n}] = \sum_i \alpha_i^{(p+n)}[a, t_i^{(p+n)}]$. Finally, we end up with $f_{p+n} = \sum_i \alpha_i^{(p+n)} t_i^{(p+n)}$.

If one of the equations cannot be factorized, it is because system (7.15) does not have solutions.

7.2. Representation of Canonical Transformations

Given $\mathbf{G}$ the set of canonical transformations generated by the exponentials (or T_f or M_g transformations).

LEMMA 21. — *Given $T = \exp(L_k)$ (resp. $T = T_w^{-1}$, $T = M_g^{-1}$) and $A = Ta = \sum_n T_n a$. In the decomposition onto the Lyndon basis, we have: $A_n = -[a, k_n] + K_n$ (resp. $A_n = -[a, w_n] + W_n$, $A_n = -[a, g_n] + G_n$) where $K_n \in L(a, k_1, \dots k_{n-1})$ (resp. $W_n \in L(a, w_1, \dots w_{n-1})$, $G_n \in L(a, g_1, \dots g_{n-1})$).*

We merely need to write $Ta = \sum_{n\geq 0}(Ta)_n$, and consider the expanded form of $(T_a)_n$ as needed.

$\star\star\star$

The previous lemma may be generalized for $T \in \mathbf{G}$. These considerations thus allow us to write any transformation of $\mathbf{G}$ as one of the three types of transformations.

PROPOSITION 22. — *Given $T \in \mathbf{G}$. T may be written as an exponential, a T_f^{-1} transformation, or an infinite product of M_g^{-1} exponentials. Generating series are computed by solving a factorizable triangular system.*

Let us consider $T \in \mathbf{G}$, and the expansion of $A = Ta$ in the Lyndon basis of the free Lie algebra on an alphabet $X = \{a\} \bigcup X_1$ which contains all the requisite letters. Then we have

$$A_1 = T_1 a = \sum_i \alpha_i [a, t_i], \tag{7.16}$$

where $t_i \in L(X_1)$, (the list may in some cases be empty). If we wish to write T as an exponential, or as a Lie transformation Lie, or as a Dragt-Finn transformation, we necessarily have k_1 (resp. g_1, w_1) $= -\sum_i \alpha_i t_i$.

Let us suppose, in order to establish the notations, that we wish to write T as exponential $\exp(L_k)$. Given $Ta = \sum_{n \geq 0} A_n$, let us pose $\exp(L_k)a = \sum_{n \geq 0} E_n$. The successive k_n are computed by solving the triangular system

$$\left\{ \begin{array}{lllll} A_1 & = & E_1 & = & -[a, k_1] \\ A_2 & = & E_2 & = & -[a, k_2] + R_2 \\ \vdots & & \vdots & & \vdots \\ A_n & = & E_n & = & -[a, k_n] + R_n \end{array} \right. \tag{7.17}$$

which, according to the previous section, has a solution.

7.3. Examples

Formula (6.44) yields the expression of w as a function of g. This is typically the case of a triangular system.

$$\begin{aligned}
w_1 &= g_1 \\
w_2 &= g_2 \\
w_3 &= g_3 - \tfrac{1}{3}\,[g_1, g_2] \\
w_4 &= g_4 - \tfrac{1}{4}\,[g_1, g_3] \\
w_5 &= g_5 - \tfrac{1}{5}\,[g_1, g_4] - \tfrac{2}{5}\,[g_2, g_3] + \tfrac{1}{10}\,[[g_1, g_2], g_2] \\
w_6 &= g_6 - \tfrac{1}{6}\,[g_1, g_5] - \tfrac{1}{3}\,[g_2, g_4] + \tfrac{1}{6}\,[g_1, [g_2, g_3]] + \tfrac{1}{6}\,[[g_1, g_3], g_2]
\end{aligned}$$

This system may be inverted by successive evaluations, and for the first 6 terms of generating series for g, we find

$$\begin{aligned}
g_1 &= w_1 \\
g_2 &= w_2 \\
g_3 &= w_3 + \tfrac{1}{3}\,[w_1, w_2] \\
g_4 &= w_4 + \tfrac{1}{4}\,[w_1, w_3] + \tfrac{1}{12}\,[w_1, [w_1, w_2]] \\
g_5 &= w_5 + \tfrac{1}{5}\,[w_1, w_4] + \tfrac{2}{5}\,[w_2, w_3] + \tfrac{1}{20}\,[w_1, [w_1, w_3]] - \tfrac{7}{30}\,[[w_1, w_2], w_2] \\
&\quad + \tfrac{1}{60}\,[w_1, [w_1, [w_1, w_2]]] \\
g_6 &= w_6 + \tfrac{1}{6}\,[w_1, w_5] + \tfrac{1}{3}\,[w_2, w_4] + \tfrac{1}{30}\,[w_1, [w_1, w_4]] - \tfrac{1}{10}\,[w_1, [w_2, w_3]] - \tfrac{1}{4}\,[[w_1, w_3], w_2] \\
&\quad + \tfrac{1}{120}\,[w_1, [w_1, [w_1, w_3]]] - \tfrac{1}{15}\,[w_1, [[w_1, w_2], w_2]] + \tfrac{1}{360}\,[w_1, [w_1, [w_1, [w_1, w_2]]]]
\end{aligned}$$

Let us compute, to order 5, $\exp(\sum_{p \geq 1} \varepsilon^p L_{k_p}) \varepsilon^0 a$ in which the weight of k_i is i, and the weight of a is 0. We get, by expanding in terms of ε, $\sum_{n \geq 0} \varepsilon^n A_n$ in which

$$
\begin{aligned}
A_0 &= a\\
A_1 &= -[a,\ k_1]\\
A_2 &= -[a,\ k_2] + \tfrac{1}{2}\,[[a,\ k_1],\ k_1]\\
A_3 &= -[a,\ k_3] + \tfrac{1}{2}\,[a,\ [k_1,\ k_2]] + [[a,\ k_2],\ k_1] - \tfrac{1}{6}\,[[[a,\ k_1],\ k_1],\ k_1]\\
A_4 &= -[a,\ k_4] + \tfrac{1}{2}\,[a,\ [k_1,\ k_3]] + \tfrac{1}{2}\,[[a,\ k_2],\ k_2] + [[a,\ k_3],\ k_1] - \tfrac{1}{6}\,[a,\ [k_1,\ [k_1,\ k_2]]]\\
&\quad - \tfrac{1}{2}\,[[a,\ [k_1,\ k_2]],\ k_1] - \tfrac{1}{2}\,[[[a,\ k_2],\ k_1],\ k_1] + \tfrac{1}{24}\,[[[[a,\ k_1],\ k_1],\ k_1],\ k_1]\\
A_5 &= -[a,\ k_5] + \tfrac{1}{2}\,[a,\ [k_1,\ k_4]] + \tfrac{1}{2}\,[a,\ [k_2,\ k_3]] + [[a,\ k_3],\ k_2] + [[a,\ k_4],\ k_1]\\
&\quad - \tfrac{1}{6}\,[a,\ [k_1,\ [k_1,\ k_3]]] - \tfrac{1}{6}\,[a,\ [[k_1,\ k_2],\ k_2]] - \tfrac{1}{2}\,[[a,\ k_2],\ [k_1,\ k_2]] - \tfrac{1}{2}\,[[a,\ [k_1,\ k_3]],\ k_1]\\
&\quad - \tfrac{1}{2}\,[[[a,\ k_2],\ k_2],\ k_1] - \tfrac{1}{2}\,[[[a,\ k_3],\ k_1],\ k_1] + \tfrac{1}{24}\,[a,\ [k_1,\ [k_1,\ [k_1,\ k_2]]]]\\
&\quad + \tfrac{1}{6}\,[[a,\ [k_1,\ [k_1,\ k_2]]],\ k_1] + \tfrac{1}{4}\,[[[a,\ [k_1,\ k_2]],\ k_1],\ k_1] + \tfrac{1}{6}\,[[[[a,\ k_2],\ k_1],\ k_1],\ k_1]\\
&\quad - \tfrac{1}{120}\,[[[[[a,\ k_1],\ k_1],\ k_1],\ k_1],\ k_1]
\end{aligned}
$$

On the other hand, let us compute to the same order $\exp(\varepsilon L_y)\exp(\varepsilon L_x)\varepsilon^0 a = \sum_{n\geq 0}\varepsilon^n B_n$

$$
\begin{aligned}
B_0 &= a\\
B_1 &= -[a,\ x] - [a,\ y]\\
B_2 &= +[a,\ [x,\ y]] + \tfrac{1}{2}\,[[a,\ x],\ x] + [[a,\ y],\ x] + \tfrac{1}{2}\,[[a,\ y],\ y]\\
B_3 &= -\tfrac{1}{2}\,[a,\ [x,\ [x,\ y]]] - \tfrac{1}{2}\,[a,\ [[x,\ y],\ y]] - [[a,\ y],\ [x,\ y]] - [[a,\ [x,\ y]],\ x]\\
&\quad - \tfrac{1}{6}\,[[[a,\ x],\ x],\ x] - \tfrac{1}{2}\,[[[a,\ y],\ x],\ x] - \tfrac{1}{2}\,[[[a,\ y],\ y],\ x] - \tfrac{1}{6}\,[[[a,\ y],\ y],\ y]\\
B_4 &= +\tfrac{1}{6}\,[a,\ [x,\ [x,\ [x,\ y]]]] + \tfrac{1}{4}\,[a,\ [x,\ [[x,\ y],\ y]]] + \tfrac{1}{6}\,[a,\ [[[x,\ y],\ y],\ y]]\\
&\quad + \tfrac{1}{2}\,[[a,\ y],\ [x,\ [x,\ y]]] + \tfrac{1}{2}\,[[a,\ y],\ [[x,\ y],\ y]] + \tfrac{1}{2}\,[[a,\ [x,\ y]],\ [x,\ y]]\\
&\quad + \tfrac{1}{2}\,[[[a,\ y],\ y],\ [x,\ y]] + \tfrac{1}{2}\,[[a,\ [x,\ [x,\ y]]],\ x] + \tfrac{1}{2}\,[[a,\ [[x,\ y],\ y]],\ x]\\
&\quad + [[[a,\ y],\ [x,\ y]],\ x] + \tfrac{1}{2}\,[[[a,\ [x,\ y]],\ x],\ x] + \tfrac{1}{24}\,[[[[a,\ x],\ x],\ x],\ x]\\
&\quad + \tfrac{1}{6}\,[[[[a,\ y],\ x],\ x],\ x] + \tfrac{1}{4}\,[[[[a,\ y],\ y],\ x],\ x] + \tfrac{1}{6}\,[[[[a,\ y],\ y],\ y],\ x]\\
&\quad + \tfrac{1}{24}\,[[[[a,\ y],\ y],\ y],\ y]\\
B_5 &= -\tfrac{1}{24}\,[a,\ [x,\ [x,\ [x,\ [x,\ y]]]]] - \tfrac{1}{12}\,[a,\ [x,\ [x,\ [[x,\ y],\ y]]]] - \tfrac{1}{12}\,[a,\ [x,\ [[[x,\ y],\ y],\ y]]]\\
&\quad - \tfrac{1}{4}\,[a,\ [[x,\ y],\ [[x,\ y],\ y]]] - \tfrac{1}{6}\,[a,\ [[x,\ [x,\ y]],\ [x,\ y]]] - \tfrac{1}{24}\,[a,\ [[[[x,\ y],\ y],\ y],\ y]]\\
&\quad - \tfrac{1}{6}\,[[a,\ y],\ [x,\ [x,\ [x,\ y]]]] - \tfrac{1}{4}\,[[a,\ y],\ [x,\ [[x,\ y],\ y]]] - \tfrac{1}{6}\,[[a,\ y],\ [[[x,\ y],\ y],\ y]]\\
&\quad - \tfrac{1}{2}\,[[a,\ [x,\ y]],\ [x,\ [x,\ y]]] - \tfrac{1}{4}\,[[[a,\ y],\ y],\ [x,\ [x,\ y]]] - \tfrac{1}{4}\,[[[a,\ y],\ y],\ [[x,\ y],\ y]]\\
&\quad - \tfrac{1}{2}\,[[a,\ [[x,\ y],\ y]],\ [x,\ y]] - \tfrac{1}{2}\,[[[a,\ y],\ [x,\ y]],\ [x,\ y]] - \tfrac{1}{6}\,[[[[a,\ y],\ y],\ y],\ [x,\ y]]\\
&\quad - \tfrac{1}{6}\,[[a,\ [x,\ [x,\ [x,\ y]]]],\ x] - \tfrac{1}{4}\,[[a,\ [x,\ [[x,\ y],\ y]]],\ x] - \tfrac{1}{6}\,[[a,\ [[[x,\ y],\ y],\ y]],\ x]\\
&\quad - \tfrac{1}{2}\,[[[a,\ y],\ [x,\ [x,\ y]]],\ x] - \tfrac{1}{2}\,[[[a,\ y],\ [[x,\ y],\ y]],\ x] - \tfrac{1}{2}\,[[[a,\ [x,\ y]],\ [x,\ y]],\ x]\\
&\quad - \tfrac{1}{2}\,[[[[a,\ y],\ y],\ [x,\ y]],\ x] - \tfrac{1}{4}\,[[[a,\ [x,\ [x,\ y]]],\ x],\ x] - \tfrac{1}{4}\,[[[a,\ [[x,\ y],\ y]],\ x],\ x]\\
&\quad - \tfrac{1}{2}\,[[[[a,\ y],\ [x,\ y]],\ x],\ x] - \tfrac{1}{6}\,[[[[a,\ [x,\ y]],\ x],\ x],\ x] - \tfrac{1}{120}\,[[[[[a,\ x],\ x],\ x],\ x],\ x]\\
&\quad - \tfrac{1}{24}\,[[[[[a,\ y],\ x],\ x],\ x],\ x] - \tfrac{1}{12}\,[[[[[a,\ y],\ y],\ x],\ x],\ x] - \tfrac{1}{12}\,[[[[[a,\ y],\ y],\ y],\ x],\ x]\\
&\quad - \tfrac{1}{24}\,[[[[[a,\ y],\ y],\ y],\ y],\ x] - \tfrac{1}{120}\,[[[[[a,\ y],\ y],\ y],\ y],\ y]
\end{aligned}
$$

We find a triangular system of equations, which can be factorized by a, and solved for the variables $k_1, k_2, k_3, k_4, k_5, k_6$,

$$
\begin{aligned}
k_1 &= x + y \\
k_2 &= -\tfrac{1}{2}\,[x,\, y] \\
k_3 &= \tfrac{1}{12}\,[x,\, [x,\, y]] + \tfrac{1}{12}\,[[x,\, y],\, y] \\
k_4 &= -\tfrac{1}{24}\,[x,\, [[x,\, y],\, y]] \\
k_5 &= -\tfrac{1}{720}\,[x,\, [x,\, [x,\, [x,\, y]]]] + \tfrac{1}{180}\,[x,\, [x,\, [[x,\, y],\, y]]] + \tfrac{1}{180}\,[x,\, [[[x,\, y],\, y],\, y]] \\
&\quad + \tfrac{1}{120}\,[[x,\, y],\, [[x,\, y],\, y]] + \tfrac{1}{360}\,[[x,\, [x,\, y]],\, [x,\, y]] - \tfrac{1}{720}\,[[[[x,\, y],\, y],\, y],\, y] \\
k_6 &= \tfrac{1}{1440}\,[x,\, [x,\, [x,\, [[x,\, y],\, y]]]] - \tfrac{1}{360}\,[x,\, [x,\, [[[x,\, y],\, y],\, y]]] - \tfrac{1}{240}\,[x,\, [[x,\, y],\, [[x,\, y],\, y]]] \\
&\quad - \tfrac{1}{720}\,[x,\, [[x,\, [x,\, y]],\, [x,\, y]]] + \tfrac{1}{1440}\,[x,\, [[[[x,\, y],\, y],\, y],\, y]]
\end{aligned}
$$

which is in effect the Campbell-Hausdorff formula to order 6.

Similarly, we find the generating series for the exponential in terms of the Lie transformation,

$$
\begin{aligned}
k_1 &= w_1 \\
k_2 &= w_2 \\
k_3 &= w_3 - \tfrac{1}{6}\,[w_1, w_2] \\
k_4 &= w_4 - \tfrac{1}{4}\,[w_1, w_3] \\
k_5 &= w_5 - \tfrac{3}{2}\,[w_1, w_4] - \tfrac{1}{2}\,[w_2, w_3] + \tfrac{1}{120}\,[w_1, [w_1, w_3]] + \tfrac{1}{60}\,[[w_1, w_2], w_2] \\
&\quad + \tfrac{1}{360}\,[w_1, [w_1, [w_1, w_2]]] \\
k_6 &= w_6 - \tfrac{1}{3}\,[w_1, w_5] - \tfrac{1}{6}\,[w_2, w_4] + \tfrac{1}{60}\,[w_1, [w_1, w_4]] + \tfrac{1}{30}\,[w_1, [w_2, w_3]] + \tfrac{1}{24}\,[[w_1, w_3], w_2] \\
&\quad + \tfrac{1}{240}\,[w_1, [w_1, [w_1, w_3]]] - \tfrac{1}{180}\,[w_1, [[w_1, w_2], w_2]] \\
k_7 &= w_7 - \tfrac{5}{14}\,[w_1,\, w_6] - \tfrac{3}{14}\,[w_2, w_5] - \tfrac{1}{14}\,[w_3, w_4] + \tfrac{1}{42}\,[w_1, [w_1, w_5]] + \tfrac{1}{21}\,[w_1, [w_2, w_4]] \\
&\quad - \tfrac{1}{420}\,[w_2, [w_2, w_3]] + \tfrac{5}{168}\,[[w_1, w_3], w_3] + \tfrac{3}{70}\,[[w_1, w_4], w_2] + \tfrac{1}{210}\,[w_1, [w_1, [w_1, w_4]]] \\
&\quad - \tfrac{11}{2520}\,[w_1, [w_1, [w_2, w_3]]] - \tfrac{19}{1260}\,[w_1, [[w_1, w_3], w_2]] - \tfrac{1}{315}\,[[w_1, w_2], [w_1, w_3]] \\
&\quad + \tfrac{1}{630}\,[[[w_1, w_2], w_2], w_2] - \tfrac{1}{5040}\,[w_1, [w_1, [w_1, [w_1, w_3]]]] - \tfrac{1}{2520}\,[w_1, [w_1, [[w_1, w_2], w_2]]] \\
&\quad + \tfrac{1}{1890}\,[[w_1, [w_1, w_2]], [w_1, w_2]] - \tfrac{1}{15120}\,[w_1, [w_1,\, [w_1, [w_1, [w_1, w_2]]]]]
\end{aligned}
$$

and as a function of the generating series of the exponential of the Dragt-Finn transformation,

$$
\begin{aligned}
k_1 &= g_1 \\
k_2 &= g_2 \\
k_3 &= g_3 - \tfrac{1}{2}\,[g_1, g_2] \\
k_4 &= g_4 - \tfrac{1}{2}\,[g_1, g_3] + \tfrac{1}{12}\,[g_1, [g_1, g_2]] \\
k_5 &= g_5 - \tfrac{1}{2}\,[g_1, g_4] - \tfrac{1}{2}\,[g_2, g_3] + \tfrac{1}{12}\,[g_1, [g_1, g_3]] + \tfrac{1}{12}\,[[g_1, g_2], g_2] \\
k_6 &= g_6 - \tfrac{1}{2}\,[g_1, g_5] - \tfrac{1}{2}\,[g_2,\, g_4] + \tfrac{1}{12}\,[g_1, [g_1, g_4]] + \tfrac{1}{3}\,[g_1, [g_2, g_3]] + \tfrac{1}{6}\,[[g_1, g_3], g_2]
\end{aligned}
$$

$$- \tfrac{1}{24}\,[g_1,[[g_1,g_2],g_2]] - \tfrac{1}{720}\,[g_1,[g_1,[g_1,[g_1,g_2]]]]$$

$$\begin{aligned}
k_7 = g_7 &- \tfrac{1}{2}\,[g_1,g_6] - \tfrac{1}{2}\,[g_2,g_5] - \tfrac{1}{2}\,[g_3,g_4] + \tfrac{1}{12}\,[g_1,[g_1,g_5]] + \tfrac{1}{3}\,[g_1,[g_2,g_4]] + \tfrac{1}{12}\,[g_2,[g_2,g_3]] \\
&+ \tfrac{1}{12}\,[[g_1,g_3],g_3] + \tfrac{1}{6}\,[[g_1,g_4],g_2] - \tfrac{1}{12}\,[g_1,[g_1,[g_2,g_3]]] - \tfrac{1}{12}\,[g_1,[[g_1,g_3],g_2]] \\
&- \tfrac{1}{720}\,[g_1,[g_1,[g_1,[g_1,g_3]]]] + \tfrac{1}{180}\,[g_1,[g_1,[[g_1,g_2],g_2]]] + \tfrac{1}{360}\,[[g_1,[g_1,g_2]],[g_1,g_2]]
\end{aligned}$$

The generating series of the Dragt-Finn transformation as a function of the generating series of the exponential,

$$\begin{aligned}
g_1 &= k_1 \\
g_2 &= k_2 \\
g_3 &= k_3 + \tfrac{1}{2}\,[k_1,k_2] \\
g_4 &= k_4 + \tfrac{1}{2}\,[k_1,k_3] + \tfrac{1}{6}\,[k_1,[k_1,k_2]] \\
g_5 &= k_5 + \tfrac{1}{2}\,[k_1,k_4] + \tfrac{1}{2}\,[k_2,k_3] + \tfrac{1}{6}\,[k_1,[k_1,k_3]] - \tfrac{1}{3}\,[[k_1,k_2],k_2] + \tfrac{1}{24}\,[k_1,[k_1,[k_1,k_2]]] \\
g_6 &= k_6 + \tfrac{1}{2}\,[k_1,k_5] + \tfrac{1}{2}\,[k_2,k_4] + \tfrac{1}{6}\,[k_1,[k_1,k_4]] - \tfrac{1}{12}\,[k_1,[k_2,k_3]] - \tfrac{5}{12}\,[[k_1,k_3],k_2] \\
&\quad + \tfrac{1}{24}\,[k_1,[k_1,[k_1,k_3]]] - \tfrac{1}{8}\,[k_1,[[k_1,k_2],k_2]] + \tfrac{1}{120}\,[k_1,[k_1,[k_1,[k_1,k_2]]]] \\
g_7 &= k_7 + \tfrac{1}{2}\,[k_1,k_6] + \tfrac{1}{2}\,[k_2,k_5] + \tfrac{1}{2}\,[k_3,k_4] + \tfrac{1}{6}\,[k_1,[k_1,k_5]] + \tfrac{1}{6}\,[k_1,[k_2,k_4]] \\
&\quad + \tfrac{1}{6}\,[k_2,[k_2,k_3]] - \tfrac{1}{3}\,[[k_1,k_3],k_3] - \tfrac{1}{6}\,[[k_1,k_4],k_2] + \tfrac{1}{24}\,[k_1,[k_1,[k_1,k_4]]] \\
&\quad - \tfrac{1}{8}\,[k_1,[k_1,[k_2,k_3]]] - \tfrac{1}{4}\,[k_1,[[k_1,k_3],k_2]] + \tfrac{1}{8}\,[[k_1,k_2],[k_1,k_3]] + \tfrac{1}{8}\,[[[k_1,k_2],k_2],k_2] \\
&\quad + \tfrac{1}{120}\,[k_1,[k_1,[k_1,[k_1,k_3]]]] - \tfrac{1}{30}\,[k_1,[k_1,[[k_1,k_2],k_2]]] - \tfrac{1}{60}\,[[k_1,[k_1,k_2]],[k_1,k_2]] \\
&\quad + \tfrac{1}{720}\,[k_1,[k_1,[k_1,[k_1,[k_1,k_2]]]]]
\end{aligned}$$

The generating series of the Lie transformation as a function of the generating series of the exponential,

$$\begin{aligned}
w_1 &= k_1 \\
w_2 &= k_2 \\
w_3 &= k_3 + \tfrac{1}{6}\,[k_1,k_2] \\
w_4 &= k_4 + \tfrac{1}{4}\,[k_1,k_3] + \tfrac{1}{24}\,[k_1,[k_1,k_2]] \\
w_5 &= k_5 + \tfrac{3}{2}\,[k_1,k_4] + \tfrac{1}{2}\,[k_2,k_3] + \tfrac{1}{15}\,[k_1,[k_1,k_3]] - \tfrac{1}{30}\,[[k_1,k_2],k_2] + \tfrac{1}{120}\,[k_1,[k_1,[k_1,k_2]]] \\
w_6 &= k_6 + \tfrac{1}{3}\,[k_1,k_5] + \tfrac{1}{6}\,[k_2,k_4] + \tfrac{1}{12}\,[k_1,[k_1,k_4]] - \tfrac{1}{12}\,[[k_1,k_3],k_2] + \tfrac{1}{72}\,[k_1,[k_1,[k_1,k_3]]] \\
&\quad - \tfrac{1}{72}\,[k_1,[[k_1,k_2],k_2]] + \tfrac{1}{720}\,[k_1,[k_1,[k_1,[k_1,k_2]]]] \\
w_7 &= k_7 + \tfrac{5}{14}\,[k_1,k_6] + \tfrac{3}{14}\,[k_2,k_5] + \tfrac{1}{14}\,[k_3,k_4] + \tfrac{2}{21}\,[k_1,[k_1,k_5]] + \tfrac{1}{42}\,[k_1,[k_2,k_4]] \\
&\quad + \tfrac{1}{42}\,[k_2,[k_2,k_3]] - \tfrac{1}{21}\,[[k_1,k_3],k_3] - \tfrac{2}{21}\,[[k_1,k_4],k_2] + \tfrac{1}{56}\,[k_1,[k_1,[k_1,k_4]]] \\
&\quad - \tfrac{1}{168}\,[k_1,[k_1,[k_2,k_3]]] - \tfrac{1}{28}\,[k_1,[[k_1,k_3],k_2]] - \tfrac{1}{168}\,[[k_1,k_2],[k_1,k_3]] \\
&\quad + \tfrac{1}{168}\,[[[k_1,k_2],k_2],k_2] + \tfrac{1}{420}\,[k_1,[k_1,[k_1,[k_1,k_3]]]] - \tfrac{1}{280}\,[k_1,[k_1,[[k_1,k_2],k_2]]] \\
&\quad + \tfrac{1}{840}\,[[k_1,[k_1,k_2]],[k_1,k_2]] + \tfrac{1}{5040}\,[k_1,[k_1,[k_1,[k_1,[k_1,k_2]]]]]
\end{aligned}$$

7.3.1. Composition of Transformations

The composition of the two inverse Lie transformations T_u^{-1} and T_v^{-1} is T_w^{-1}, where

$$
\begin{aligned}
w_1 &= u_1 + v_1 \\
w_2 &= u_2 + v_2 + \tfrac{1}{2}\,[u_1, v_1] \\
w_3 &= u_3 + v_3 + \tfrac{2}{3}\,[u_1, v_2] + \tfrac{1}{3}\,[u_2, v_1] + \tfrac{1}{6}\,[u_1, [u_1, v_1]] \\
w_4 &= u_4 + v_4 + \tfrac{3}{4}\,[u_1, v_3] + \tfrac{1}{2}\,[u_2, v_2] + \tfrac{1}{4}\,[u_3, v_1] + \tfrac{1}{4}\,[u_1, [u_1, v_2]] + \tfrac{1}{12}\,[u_1, [u_2, v_1]] \\
&\quad - \tfrac{1}{6}\,[[u_1, v_1], u_2] + \tfrac{1}{24}\,[u_1, [u_1, [u_1, v_1]]] \\
w_5 &= u_5 + v_5 + \tfrac{4}{5}\,[u_1, v_4] + \tfrac{3}{5}\,[u_2, v_3] + \tfrac{2}{5}\,[u_3, v_2] + \tfrac{1}{5}\,[u_4, v_1] + \tfrac{3}{2}\,[u_1, [u_1, v_3]] \\
&\quad + \tfrac{2}{15}\,[u_1, [u_2, v_2]] + \tfrac{1}{20}\,[u_1, [u_3, v_1]] + \tfrac{1}{2}\,[u_2, [u_2, v_1]] - \tfrac{3}{20}\,[[u_1, v_1], u_3] - \tfrac{4}{15}\,[[u_1, v_2], u_2] \\
&\quad + \tfrac{1}{15}\,[u_1, [u_1, [u_1, v_2]]] + \tfrac{1}{60}\,[u_1, [u_1, [u_2, v_1]]] - \tfrac{1}{12}\,[u_1, [[u_1, v_1], u_2]] \\
&\quad - \tfrac{1}{20}\,[[u_1, u_2], [u_1, v_1]] + \tfrac{1}{120}\,[u_1, [u_1, [u_1, [u_1, v_1]]]] \\
w_6 &= u_6 + v_6 + \tfrac{5}{6}\,[u_1, v_5] + \tfrac{2}{3}\,[u_2, v_4] + \tfrac{1}{2}\,[u_3, v_3] + \tfrac{1}{3}\,[u_4, v_2] + \tfrac{1}{6}\,[u_5, v_1] + \tfrac{1}{3}\,[u_1, [u_1, v_4]] \\
&\quad + \tfrac{1}{6}\,[u_1, [u_2, v_3]] + \tfrac{1}{12}\,[u_1, [u_3, v_2]] + \tfrac{1}{30}\,[u_1, [u_4, v_1]] + \tfrac{1}{6}\,[u_2, [u_2, v_2]] + \tfrac{1}{15}\,[u_2, [u_3, v_1]] \\
&\quad - \tfrac{2}{15}\,[[u_1, v_1], u_4] - \tfrac{1}{4}\,[[u_1, v_2], u_3] - \tfrac{1}{3}\,[[u_1, v_3], u_2] - \tfrac{1}{2}\,[[u_2, v_1], u_3] \\
&\quad + \tfrac{1}{12}\,[u_1, [u_1, [u_1, v_3]]] + \tfrac{1}{36}\,[u_1, [u_1, [u_2, v_2]]] + \tfrac{1}{120}\,[u_1, [u_1, [u_3, v_1]]] \\
&\quad + \tfrac{1}{60}\,[u_1, [u_2, [u_2, v_1]]] - \tfrac{3}{40}\,[u_1, [[u_1, v_1], u_3]] - \tfrac{5}{36}\,[u_1, [[u_1, v_2], u_2]] \\
&\quad - \tfrac{1}{12}\,[[u_1, u_2], [u_1, v_2]] - \tfrac{1}{20}\,[[u_1, u_3], [u_1, v_1]] - \tfrac{1}{45}\,[[u_1, [u_2, v_1]], u_2] \\
&\quad + \tfrac{2}{45}\,[[[u_1, v_1], u_2], u_2] + \tfrac{1}{72}\,[u_1, [u_1, [u_1, [u_1, v_2]]]] + \tfrac{1}{360}\,[u_1, [u_1, [u_1, [u_2, v_1]]]] \\
&\quad - \tfrac{1}{40}\,[u_1, [u_1, [[u_1, v_1], u_2]]] - \tfrac{7}{360}\,[u_1, [[u_1, u_2], [u_1, v_1]]] + \tfrac{1}{90}\,[[u_1, [u_1, v_1]], [u_1, u_2]] \\
&\quad + \tfrac{1}{720}\,[u_1, [u_1, [u_1, [u_1, [u_1, v_1]]]]] \\
w_7 &= u_7 + v_7 + \tfrac{6}{7}\,[u_1, v_6] + \tfrac{5}{7}\,[u_2, v_5] + \tfrac{4}{7}\,[u_3, v_4] + \tfrac{3}{7}\,[u_4, v_3] + \tfrac{2}{7}\,[u_5, v_2] + \tfrac{1}{7}\,[u_6, v_1] \\
&\quad + \tfrac{5}{14}\,[u_1, [u_1, v_5]] + \tfrac{4}{21}\,[u_1, [u_2, v_4]] + \tfrac{3}{28}\,[u_1, [u_3, v_3]] + \tfrac{2}{35}\,[u_1, [u_4, v_2]] \\
&\quad + \tfrac{1}{42}\,[u_1, [u_5, v_1]] + \tfrac{3}{14}\,[u_2, [u_2, v_3]] + \tfrac{4}{35}\,[u_2, [u_3, v_2]] + \tfrac{1}{21}\,[u_2, [u_4, v_1]] \\
&\quad + \tfrac{1}{14}\,[u_3, [u_3, v_1]] - \tfrac{5}{42}\,[[u_1, v_1], u_5] - \tfrac{8}{35}\,[[u_1, v_2], u_4] - \tfrac{9}{28}\,[[u_1, v_3], u_3] \\
&\quad - \tfrac{8}{21}\,[[u_1, v_4], u_2] - \tfrac{2}{21}\,[[u_2, v_1], u_4] - \tfrac{6}{35}\,[[u_2, v_2], u_3] + \tfrac{2}{21}\,[u_1, [u_1, [u_1, v_4]]] \\
&\quad + \tfrac{1}{28}\,[u_1, [u_1, [u_2, v_3]]] + \tfrac{1}{70}\,[u_1, [u_1, [u_3, v_2]]] + \tfrac{1}{210}\,[u_1, [u_1, [u_4, v_1]]] \\
&\quad + \tfrac{1}{35}\,[u_1, [u_2, [u_2, v_2]]] + \tfrac{1}{105}\,[u_1, [u_2, [u_3, v_1]]] + \tfrac{1}{42}\,[u_2, [u_2, [u_2, v_1]]] \\
&\quad - \tfrac{1}{15}\,[u_1, [[u_1, v_1], u_4]] - \tfrac{9}{70}\,[u_1, [[u_1, v_2], u_3]] - \tfrac{5}{28}\,[u_1, [[u_1, v_3], u_2]] \\
&\quad - \tfrac{4}{105}\,[u_1, [[u_2, v_1], u_3]] - \tfrac{3}{28}\,[[u_1, u_2], [u_1, v_3]] - \tfrac{3}{35}\,[[u_1, u_3], [u_1, v_2]] \\
&\quad - \tfrac{1}{42}\,[[u_1, u_3], [u_2, v_1]] - \tfrac{1}{21}\,[[u_1, u_4], [u_1, v_1]] + \tfrac{1}{21}\,[[u_1, v_1], [u_2, u_3]] \\
&\quad - \tfrac{4}{105}\,[[u_1, [u_2, v_2]], u_2] - \tfrac{1}{84}\,[[u_1, [u_3, v_1]], u_2] + \tfrac{1}{12}\,[[[u_1, v_1], u_3], u_2] \\
&\quad + \tfrac{8}{105}\,[[[u_1, v_2], u_2], u_2] + \tfrac{1}{56}\,[u_1, [u_1, [u_1, [u_1, v_3]]]] + \tfrac{1}{210}\,[u_1, [u_1, [u_1, [u_2, v_2]]]] \\
&\quad + \tfrac{1}{840}\,[u_1, [u_1, [u_1, [u_3, v_1]]]] + \tfrac{1}{420}\,[u_1, [u_1, [u_2, [u_2, v_1]]]] - \tfrac{19}{840}\,[u_1, [u_1, [[u_1, v_1], u_3]]] \\
&\quad - \tfrac{3}{70}\,[u_1, [u_1, [[u_1, v_2], u_2]]] - \tfrac{1}{30}\,[u_1, [[u_1, u_2], [u_1, v_2]]] - \tfrac{2}{105}\,[u_1, [[u_1, u_3], [u_1, v_1]]]
\end{aligned}
$$

$$
\begin{aligned}
&- \tfrac{1}{140}\,[u_1,[[u_1,[u_2,v_1]],u_2]] + \tfrac{11}{420}\,[u_1,[[[u_1,v_1],u_2],u_2]] - \tfrac{1}{252}\,[[u_1,u_2],[u_1,[u_2,v_1]]]\\
&+ \tfrac{2}{63}\,[[u_1,u_2],[[u_1,v_1],u_2]] + \tfrac{1}{84}\,[[u_1,[u_1,v_1]],[u_1,u_3]] + \tfrac{2}{105}\,[[u_1,[u_1,v_2]],[u_1,u_2]]\\
&+ \tfrac{1}{84}\,[[[u_1,u_2],u_2],[u_1,v_1]] + \tfrac{1}{420}\,[u_1,[u_1,[u_1,[u_1,[u_1,v_2]]]]]\\
&+ \tfrac{1}{2520}\,[u_1,[u_1,[u_1,[u_1,[u_2,v_1]]]]] - \tfrac{1}{180}\,[u_1,[u_1,[u_1,[[u_1,v_1],u_2]]]]\\
&- \tfrac{1}{210}\,[u_1,[u_1,[[u_1,u_2],[u_1,v_1]]]] + \tfrac{1}{180}\,[u_1,[[u_1,[u_1,v_1]],[u_1,u_2]]]\\
&+ \tfrac{1}{504}\,[[u_1,[u_1,u_2]],[u_1,[u_1,v_1]]] + \tfrac{1}{5040}\,[u_1,[u_1,[u_1,[u_1,[u_1,[u_1,v_1]]]]]]
\end{aligned}
$$

Hence we deduce the Lie transformation T_v in the form of an inverse Lie transformation $T_{v'}^{-1}$, by solving the previous system, where $w = 0$, in relation to the u_i

$$
\begin{aligned}
v'_1 &= -v_1\\
v'_2 &= -v_2\\
v'_3 &= -v_3 + \tfrac{1}{3}\,[v_1,v_2]\\
v'_4 &= -v_4 + \tfrac{1}{2}\,[v_1,v_3] - \tfrac{1}{12}\,[v_1,[v_1,v_2]]\\
v'_5 &= -v_5 + \tfrac{3}{5}\,[v_1,v_4] + \tfrac{1}{5}\,[v_2,v_3] - \tfrac{3}{20}\,[v_1,[v_1,v_3]] + \tfrac{1}{30}\,[[v_1,v_2],v_2] + \tfrac{1}{60}\,[v_1,[v_1,[v_1,v_2]]]\\
v'_6 &= -v_6 + \tfrac{2}{3}\,[v_1,v_5] + \tfrac{1}{3}\,[v_2,v_4] - \tfrac{1}{5}\,[v_1,[v_1,v_4]] - \tfrac{1}{15}\,[v_1,[v_2,v_3]] + \tfrac{1}{12}\,[[v_1,v_3],v_2]\\
&\quad + \tfrac{1}{30}\,[v_1,[v_1,[v_1,v_3]]] - \tfrac{1}{60}\,[v_1,[[v_1,v_2],v_2]] - \tfrac{1}{360}\,[v_1,[v_1,[v_1,[v_1,v_2]]]]\\
v'_7 &= -v_7 + \tfrac{5}{7}\,[v_1,v_6] + \tfrac{3}{7}\,[v_2,v_5] + \tfrac{1}{7}\,[v_3,v_4] - \tfrac{5}{21}\,[v_1,[v_1,v_5]] - \tfrac{1}{7}\,[v_1,[v_2,v_4]]\\
&\quad - \tfrac{3}{70}\,[v_2,[v_2,v_3]] + \tfrac{1}{28}\,[[v_1,v_3],v_3] + \tfrac{11}{105}\,[[v_1,v_4],v_2] + \tfrac{1}{21}\,[v_1,[v_1,[v_1,v_4]]]\\
&\quad + \tfrac{1}{84}\,[v_1,[v_1,[v_2,v_3]]] - \tfrac{19}{420}\,[v_1,[[v_1,v_3],v_2]] - \tfrac{1}{105}\,[[v_1,v_2],[v_1,v_3]]\\
&\quad + \tfrac{1}{210}\,[[[v_1,v_2],v_2],v_2] - \tfrac{1}{168}\,[v_1,[v_1,[v_1,[v_1,v_3]]]] + \tfrac{1}{210}\,[v_1,[v_1,[[v_1,v_2],v_2]]]\\
&\quad - \tfrac{1}{1260}\,[[v_1,[v_1,v_2]],[v_1,v_2]] + \tfrac{1}{2520}\,[v_1,[v_1,[v_1,[v_1,[v_1,v_2]]]]]
\end{aligned}
$$

Similarly, we get the inverse of the inverse Dragt-Finn transformation $M_{g'}^{-1} = M_g$, where

$$
\begin{aligned}
g'_1 &= -g_1\\
g'_2 &= -g_2\\
g'_3 &= -g_3 + [g_1,v_2]\\
g'_4 &= -g_4 + [g_1,g_3] - \tfrac{1}{2}\,[g_1,[g_1,v_2]]\\
g'_5 &= -g_5 + [g_1,g_4] + [g_2,g_3] - \tfrac{1}{2}\,[g_1,[g_1,g_3]] + \tfrac{1}{2}\,[[g_1,g_2],g_2] + \tfrac{1}{6}\,[g_1,[g_1,[g_1,g_2]]]\\
g'_6 &= -g_6 + [g_1,g_5] + [g_2,g_4] - \tfrac{1}{2}\,[g_1,[g_1,g_4]] - \tfrac{1}{2}\,[g_1,[g_2,v_3]] + \tfrac{1}{2}\,[[g_1,g_3],g_2]\\
&\quad + \tfrac{1}{6}\,[g_1,[g_1,[g_1,g_3]]] - \tfrac{1}{4}\,[g_1,[[g_1,g_2],g_2]] - \tfrac{1}{24}\,[g_1,[g_1,[g_1,[g_1,g_2]]]]\\
g'_7 &= -g_7 + [g_1,g_6] + [g_2,g_5] + [g_3,g_4] - \tfrac{1}{2}\,[g_1,[g_1,g_5]] - [g_1,[v_2,g_4]] - \tfrac{1}{2}\,[g_2,[g_2,g_3]]\\
&\quad + \tfrac{1}{2}\,[[g_1,g_3],g_3] + \tfrac{1}{6}\,[g_1,[g_1,[g_1,g_4]]] - \tfrac{1}{2}\,[g_1,[[g_1,g_3],g_2]] + \tfrac{1}{2}\,[[g_1,g_2],[g_1,g_3]]\\
&\quad + \tfrac{1}{6}\,[[[g_1,v_2],g_2],g_2] - \tfrac{1}{24}\,[g_1,[g_1,[g_1,[g_1,g_3]]]] + \tfrac{1}{12}\,[g_1,[g_1,[[g_1,g_2],g_2]]]\\
&\quad + \tfrac{1}{6}\,[[g_1,[v_1,g_2]],[g_1,g_2]] + \tfrac{1}{120}\,[g_1,[g_1,[g_1,[g_1,[v_1,g_2]]]]]
\end{aligned}
$$

Obviously, these formulas might have been obtained by using the bijection between the Lyndon words and the Lyndon basis, by using the expanded formulas of Lie automorphisms as functions of Lie operators. However, it would probably have been difficult to compose mappings, and it would have become necessary to combine several expanded formulas.

The convenient form of the injective morphism $L_a : x \mapsto [a, x]$ is the equivalent of the superior triangular form of the bijection between the Lyndon words and the Lyndon basis, written in this basis. In my opinion, the advantage lies in the fact that one can remain in the free Lie algebra, which, from the standpoint of implementation, is most useful.

This morphism has the same properties, regardless of the basis chosen, but it does not appear that its representation in the Hall basis would be simple.

On the other hand, the relations between the various representations of the elements of the group generated by the exponentials of Lie operators will later enable us to formally demonstrate the equivalence of perturbation methods, or the equivalence of methods for searching for symplectic integrators. The means of explicitly computing these relations will provide the means of explicitly finding symplectic integrators.

I had initially proposed an algorithm for Lie polynomial manipulation in the Hall basis (cf. [32]), without realizing that the same considerations applied for the Lyndon basis. Only recently, after reading Mr. Petitot's thesis, was I able to effectively transpose the algorithm. Both decompositions, in the Hall basis and in the Lyndon basis, will be implemented from the same representation of the basic principles, in the last section of this text.

Chapter II
Hamiltonian Formalism

In this chapter, we introduce the definitions and basic concepts of hamiltonian mechanics –which will be useful for the rest of this text– while setting them apart from actual perturbation methods. Hopefully, this section should provide the necessary tools to define the framework of perturbation methods, which are primarily relevant to computational issues.

First of all, we will recall hamiltonian formalism. All notions addressed here were primarily inspired by Arnold's two books, ([2, 3]) as well as Meyer & Hall's book ([41]).

Next, we focus on families of canonical transformations, specifically those which are written in formal series of a small parameter. Here my ideas were primarily inspired by original articles by Deprit [11] and Dragt & Finn [14] on the one hand, and by general articles like Cary's [7] on the other.

Finally, we compare the use of the Deprit transformation and that of the Dragt-Finn transformation as functions of the number of evaluations of Poisson brackets which they require.

8. Definitions and Notations

For this entire discussion, we are in a phase space E of $2n$ dimension, which we will eventually be able to regard as $\mathbb{R}^{2n}$. State variables can be represented either by the combination of coordinates and momenta

$$(q,p) = (q_1, \ldots, q_n,\ p_1, \ldots, p_n), \tag{8.1}$$

or by

$$z = (q,p) = (z_1, \ldots, z_{2n}), \tag{8.2}$$

according to the concision desired in the formulas. Naturally, this space is provided with an antisymmetric, bilinear form J

$$J = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix} \tag{8.3}$$

where I_n denotes the identity matrix of size n. The standard symplectic form is written as

$$\omega = p_1 \wedge q_1 + \cdots + p_n \wedge q_n. \tag{8.4}$$

A linear transformation T over the phase space is called symplectic if it maintains the symplectic form ω or if it belongs to the symplectic group

$$Sp(E, w) = \left\{T; T^T JT = J\right\}. \tag{8.5}$$

A linear operator H is formally defined as being hamiltonian if it is the derivative, at $t = 0$, of a 1-parameter (t) family of symplectic transformations, which takes the identity as its value, at $t = 0$. Thus for H, we get the condition

$$H^T J + JH = 0. \tag{8.6}$$

H being a hamiltonian operator, we define

$$h(x) = \tfrac{1}{2}\omega(x, Hx) \tag{8.7}$$

as the Hamiltonian of H. Conversely, to any quadratic form h, we associate a Hamiltonian H by

$$\omega(y, Hx) = h(x + y) - h(x) - h(y) \tag{8.8}$$

which defines an isomorphism between quadratic forms, and hamiltonian operators.

In matrix formalism, to any symmetrical matrix S, we associate a symplectic endomorphism T by

$$T = e^{JS} = \sum_{m \geq 0} \frac{(JS)^m}{m!}. \tag{8.9}$$

On the other hand, any hamiltonian operator H for which the matrix is

$$H = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{8.10}$$

satisfies conditions (8.6), that is to say

$$B^T = B, C^T = C, D^T = -A. \tag{8.11}$$

The corresponding Hamiltonian is a quadratic form h for which the matrix is a symmetrical matrix

$$\frac{1}{2} JH = \frac{1}{2} \begin{pmatrix} C & D \\ -A & -B \end{pmatrix}. \tag{8.12}$$

The Poisson bracket of two Hamiltonians is defined as the Hamiltonian associated with the hamiltonian operator, which is the commutator of the two respective hamiltonian operators. Thus we get, in canonical coordinates

$$\{h_1, h_2\} = \sum_{i=1}^{n} \frac{\partial h_1}{\partial p_i} \frac{\partial h_2}{\partial q_i} - \frac{\partial h_1}{\partial q_i} \frac{\partial h_2}{\partial p_i}. \tag{8.13}$$

From this definition

$$L_{\{h_1,h_2\}} = [L_{h_1}, L_{h_2}], \tag{8.14}$$

we deduce that hamiltonian operators form a Lie algebra with the commutator as the Lie bracket.

8.1. Canonical Transformations

Considering the 2-form $w = dp \wedge dq$, we define the Poisson bracket of two smooth functions f and g as

$$\{f,g\} = \omega(df, dg), \tag{8.15}$$

that is, in the Darboux coordinates

$$\{f,g\} = \sum_{i=1}^{n} \frac{\partial f}{\partial p_i}\frac{\partial g}{\partial q_i} - \frac{\partial f}{\partial q_i}\frac{\partial g}{\partial p_i}. \tag{8.16}$$

The Poisson bracket of two functions of phase space may be written in matrix form

$$\{f,g\} = \sum_{1\leq k,l\leq 2n} \frac{\partial f}{\partial z_k} J_{k,l} \frac{\partial g}{\partial z_l} = \nabla f^T J \nabla g \tag{8.17}$$

where

$$\nabla f = \left(\frac{\partial f}{\partial z_1}, \ldots, \frac{\partial f}{\partial z_{2n}}\right) \tag{8.18}$$

denotes the gradient of f.

In this fashion, the space of smooth functions on E is provided with a Lie algebra structure in relation to the Poisson bracket.

***

We denote by L_f or $L(f)$, the Lie operator $g \mapsto \{f,g\}$. The Jacobi identity thus enables us to write

$$[L_f, L_g] = L_{\{f,g\}}, \tag{8.19}$$

where $[A,B] = AB - BA$ denotes the commutator of A and B. This last property thus allows us to give Lie operators a Lie algebra structure.

By extension, we define a canonical transformation over phase space as a transformation which maintains the Poisson bracket. Thus, we must have for $Z = Tz$,

$$\{Z_i, Z_j\} = \{z_i, z_j\} = J_{i,j}, \ 1 \leq i,j \leq 2n. \tag{8.20}$$

that is to say, for any $1 \leq i,j \leq 2n$

$$J_{i,j} = \{Z_i, Z_j\} = \sum_{1\leq k,l\leq 2n} \frac{\partial Z_i}{\partial z_k} J_{k,l} \frac{\partial Z_j}{\partial z_l} = (M^T J M)_{i,j}, \tag{8.21}$$

where M denotes the Jacobian matrix of the transformation T. In other words, canonical transformations are those whose Jacobian belongs to the symplectic set. Literature indifferently uses the adjective canonical or symplectic to denote such transformations.

Canonical transformations of phase space is extended to the functions over phase space by posing

$$Tf(z) = f(Tz). \tag{8.22}$$

Since canonical transformations preserve the Poisson brackets, we have, for a canonical transformation T, and two functions f and g (denoting $Z = Tz$),

$$\begin{aligned}
\{f,g\}_z &= \sum_{1\le i,j\le 2n} \frac{\partial f}{\partial z_i} J_{i,j} \frac{\partial g}{\partial z_j} \\
&= \sum_{1\le i,j\le 2n} \left[\sum_{k=1}^{2n} \frac{\partial f}{\partial Z_k}\frac{\partial Z_k}{\partial z_i}\right] J_{i,j} \left[\sum_{l=1}^{2n} \frac{\partial g}{\partial Z_l}\frac{\partial Z_l}{\partial z_j}\right] \\
&= \sum_{1\le k,l\le 2n} \frac{\partial f}{\partial Z_k} \left[\sum_{i,j} \frac{\partial Z_k}{\partial z_i} J_{i,j} \frac{\partial Z_l}{\partial z_j}\right] \frac{\partial g}{\partial Z_l} \\
&= \sum_{1\le k,l\le 2n} \frac{\partial f}{\partial Z_k} J_{k,l} \frac{\partial g}{\partial Z_l} \\
&= \{f,g\}_Z.
\end{aligned} \tag{8.23}$$

which means that the Poisson bracket is unchanged, whether it is considered as a function of Z or z. Therefore, we have

$$\{f,g\}_z = \{f,g\}_Z \tag{8.24}$$

which, consequently, implies for any canonical transformation T,

$$T\{f,g\} = \{Tf,Tg\} \quad \text{and} \quad L_{Tf} = TL_fT^{-1}. \tag{8.25}$$

9. Hamiltonian Systems

Newton's laws for classical mechanics produce systems of differential equations of second order in $\mathbb{R}^n$. If the forces are derived from a potential, the equations of motion can be reduced to the hamiltonian system, that is to say to the study of a system of first order differential equations in a space of $2n$ dimension. Let us consider a material system for which Newton's laws of mechanics may be written

$$M\ddot{x} + \nabla F(x) = g(t) \tag{9.1}$$

where $x \in \mathbb{R}^n$, M is a symmetrical invertible $n\times n$ matrix, F is a smooth function with a real value defined over an open set of $\mathbb{R}^n$, and $g(t)$, a smooth function taking values in $\mathbb{R}^n$, defined on an open set of $\mathbb{R}$. By posing $p = M\dot{x}$, we have

$$\dot{x} = M^{-1}p, \quad \dot{p} = g(t) - \nabla F(x), \tag{9.2}$$

that is to say, by posing

$$h = \tfrac{1}{2}p^T M^{-1} p + F(x) - x^T g(t), \tag{9.3}$$

Newton's equations are reduced to a first order system

$$\dot{x} = \frac{\partial h}{\partial p}, \quad \dot{p} = -\frac{\partial h}{\partial x}. \tag{9.4}$$

h is called the Hamiltonian of the system (9.1). Generally, $T = \frac{1}{2}p^T M^{-1} p$ is the kinetic energy, and $U = -F(x) + x^T g(t)$ is the potential energy. If $g = 0$, the Hamiltonian is autonomous and the total energy $h = T + U$ is invariant.

$\star^{\star}\star$

Another approach consists of considering the equations of Lagrangian mechanics. Given $L(x, \dot{x}, t) = T - U$ the Lagrangian of the system (9.1), the principle of least action of mechanics states that the system's evolution is defined by an extremal curve of

$$\int L(x, \dot{x}, t)dt, \tag{9.5}$$

which, according to Euler's equations, can be reached only if

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} = \frac{\partial L}{\partial x}. \tag{9.6}$$

Differentiating

$$h = \sum_{l=1}^{n} \dot{x}_l p_l - L(x, \dot{x}, t), \tag{9.7}$$

where p_l is a new variable, which we will define later, we get

$$\begin{aligned} dh &= \sum_{l=1}^{n} \frac{\partial h}{\partial x_l} dx_l + \sum_{l=1}^{n} \frac{\partial h}{\partial p_l} dp_l + \frac{\partial h}{\partial t} dt & (9.8) \\ &= \sum_{l=1}^{n} (p_l - \frac{\partial L}{\partial \dot{x}_l}) d\dot{x}_l + \sum_{l=1}^{n} \dot{x}_l dp_l - \sum_{l=1}^{n} \frac{\partial L}{\partial x_l} dx_l - \frac{\partial L}{\partial t} dt & (9.9) \\ &= \sum_{l=1}^{n} (p_l - \frac{\partial L}{\partial \dot{x}_l}) d\dot{x}_l + \sum_{l=1}^{n} \dot{x}_l dp_l - \sum_{l=1}^{n} \frac{d}{dt}[\frac{\partial L}{\partial \dot{x}_l}] dx_l - \frac{\partial L}{\partial t} dt. & (9.10) \end{aligned}$$

The previous equation is only satisfied if

$$\frac{\partial h}{\partial t} = -\frac{\partial L}{\partial t}, \quad p_l = \frac{\partial L}{\partial \dot{x}_l}, \quad \frac{\partial h}{\partial p_l} = \dot{x}_l, \quad \frac{\partial h}{\partial x_l} = -\frac{d}{dt}\frac{\partial L}{\partial \dot{x}_l}. \tag{9.11}$$

Posing $p_l = \frac{\partial L}{\partial \dot{x}_l}$, $q_l = x_l$, we get Hamilton's equations

$$\dot{p}_l = -\frac{\partial h}{\partial q_l}, \quad \dot{q}_l = \frac{\partial h}{\partial p_l}, \quad 1 \leq l \leq n. \tag{9.12}$$

$\star^{\star}\star$

A hamiltonian system consists of a phase space, and a Hamiltonian h. The study of a hamiltonian system is the study of the system of differential equations

$$\dot{p}_l = -\frac{\partial h}{\partial q_l}, \dot{q}_l = \frac{\partial h}{\partial p_l}, \quad 1 \leq l \leq n. \tag{9.13}$$

Such a system can also be written, by introducing the form J (8.3):

$$\dot{z} = J\nabla h = JDhz, \tag{9.14}$$

where Dh denotes the differential of h. Using the Poisson bracket, (9.14) can also be written

$$\dot{z} = \{z, h\}. \tag{9.15}$$

In the case of a linear hamiltonian operator, since h is a homogeneous polynomial of degree 2, we have:

$$J\nabla h = H(z) = JDh(z) = J\tilde{H}(z) \tag{9.16}$$

where $\tilde{H}$ is the Hessian of h:

$$\tilde{H} = \left(\frac{\partial^2 h}{\partial z_i \partial z_j}\right)_{1\leq i,j\leq 2n}. \tag{9.17}$$

In the space of vector fields, we will state that Jdh is the hamiltonian field of Hamiltonian h by analogy to the system of equations obtained with Jdh in canonical coordinates:

$$\dot{p}_l = -\frac{\partial h}{\partial q_l}, \dot{q}_l = \frac{\partial h}{\partial p_l}, \quad 1 \leq l \leq n. \tag{9.18}$$

The Hamiltonian h may in some cases depend on time, and in that case, we will state that the system is non-autonomous.

9.1. Integrable Systems

The integrability of a hamiltonian system relies on getting a certain number of invariants. We state that F is an invariant of the hamiltonian system given by h if and only if we have:

$$\{F, h\} = 0. \tag{9.19}$$

According to Liouville's theorem (cf. [3]), we state that a hamiltonian system is completely integrable, as implied by Liouville, if, and only if there exist n independent invariants in involution, that is to say if there exist $F_1, \ldots, F_n$ such that

$$\{F_i, h\} = 0 \quad \text{and} \quad \{F_i, F_j\} = 0 \quad 1 \leq i, j, \leq n. \tag{9.20}$$

For example,

— an autonomous hamiltonian system of 1 degree of freedom is completely integrable, since h is an invariant of the system.

— a linear hamiltonian system is completely integrable.

However, since Poincaré's contributions, we have known that generally, a hamiltonian system with several degrees of freedom is not integrable.

9.1.1. Examples

The study of a harmonic oscillator is the study of the system of differential equations of order 2:

$$\ddot{x} + \omega^2 x = 0, \tag{9.21}$$

where $\omega > 0$ is a constant. Posing $p = \frac{\dot{x}}{\omega}$ and $q = x$, we get the system of differential equations of order 1:

$$\dot{p} = \frac{\ddot{x}}{\omega} = -\omega x = -\omega q, \dot{q} = \dot{x} = \omega p, \tag{9.22}$$

that is to say, posing $h = \frac{\omega}{2}(p^2 + q^2)$, we have

$$\dot{p} = -\frac{\partial h}{\partial q}, \dot{q} = \frac{\partial h}{\partial p}. \tag{9.23}$$

The system has 1 degree of freedom, so it is integrable, which we verify with

$$\frac{d}{dt}\left(p^2 + q^2\right) = 0.$$

The solutions are given by

$$(q, p) = e^{-tH}(q_0, p_0)$$

where

$$H = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix}. \tag{9.24}$$

Therefore, we have

$$e^{-tH} = \begin{pmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{pmatrix} \tag{9.25}$$

and

$$\begin{cases} p &= & \cos(\omega t)p(0) & + & \sin(\omega t)q(0) \\ q &= & -\sin(\omega t)p(0) & + & \cos(\omega t)q(0) \end{cases}.$$

Similarly, the Hamiltonian of n uncoupled oscillators is written

$$h = \tfrac{1}{2}\sum_{l=1}^{n} \omega_l(p_l^2 + q_l^2), \tag{9.26}$$

in which the n invariants are actions

$$I_l = \tfrac{1}{2}(p_l^2 + q_l^2), \quad 1 \le l \le n. \tag{9.27}$$

The actions and their conjugate variables (angles) form a set of canonical variables defined by:

$$\begin{cases} p_l &= \sqrt{2I_l}\cos\phi_l \\ q_l &= \sqrt{2I_l}\sin\phi_l \end{cases}, \quad \phi_l(t) = \phi_l(0) + \omega_l t, \quad 1 \le l \le n. \tag{9.28}$$

⋆⋆⋆

We have Arnold-Liouville's theorem [2]:

THEOREM 23 (Arnold). — *If a system of n degrees of freedom is completely integrable, there exist n invariant tori for the hamiltonian system, that is to say a change in canonical action-angle variables. Furthermore, this change of variables can be carried out by quadrature.*

9.2. Time Evolution

In the case of an autonomous system, in the formalism of Lie operators, the time evolution of the system is given by

$$\dot{z} = \{z, h\} = -L_h z. \tag{9.29}$$

Thus, for any $n \geq 1$, we have

$$\frac{d^n z}{dt^n} = (-1)^n L_h^n z. \tag{9.30}$$

The formal solution $z(t)$ is given by its Taylor series

$$z(t) = \sum_{n\geq 0} (-1)^n \frac{L_h^n}{n!} z(0) = e^{-tL_h} z(0). \tag{9.31}$$

A family of transformations e^{-tL_h} is a one parameter family of canonical transformations, also called mapping at time t, of the hamiltonian flow. For any smooth function of the phase space, we have along the solution curves

$$\begin{aligned} \dot{f} &= \frac{\partial f}{\partial t} + \sum_{i=1}^{n} \left[\frac{\partial f}{\partial q_i} \dot{q}_i + \frac{\partial f}{\partial p_i} \dot{p}_i \right] \\ &= \frac{\partial f}{\partial t} + \sum_{i=1}^{n} \frac{\partial f}{\partial q_i} \frac{\partial h}{\partial p_i} - \frac{\partial f}{\partial p_i} \frac{\partial h}{\partial p_i} \\ &= \frac{\partial f}{\partial t} + \{f, h\}. \end{aligned} \tag{9.32}$$

In the general case ([7]), whenever h is time-dependent, we define, for an integrable system, the mapping in time t, as the mapping which, at one point of a phase space z, associates the point $z(t)$, position at time t of the solution, with a position z at $t = 0$. We denote this mapping $S_h(t)$.

Similarly, we define $S_h(t_1, t_2)$, the mapping which associates with z the point $z(t)$, position at time $t = t_2$ of the solution to the hamiltonian system, at position z and $t = t_1$.

***

The following theorem gives us a first family of canonical transformations.

THEOREM 24. — *$S_h(t_1, t)$, is a family of canonical transformations.*

In the linear case, that is to say whenever we study the hamiltonian system

$$\dot{z} = H(z), \tag{9.33}$$

where H is a hamiltonian operator, $S_h(t_1, t)$ is a linear mapping, and we have (cf. [41])

$$S_h(t_1, t_1) = Id \quad \text{and} \quad \dot{S}_h(t_1, t) = H S_h(t_1, t). \tag{9.34}$$

Let us pose then $U(t) = S_h(t_1, t)^T J S_h(t_1, t)$. We have, according to (8.6),

$$\dot{U}(t) = \dot{S}_h^T J S_h + S_h^T J \dot{S}_h = S_h^T (H^T J + JH) S_h = 0, \tag{9.35}$$

that is to say, for any t, $U(t) = U(0) = J$, and consequently, the linear operator S_h is a symplectic transformation.

In the general case, let us denote $Z(t_1, t, z) = S_h(t_1, t)z$ the general solution of (9.29). From

$$\frac{d}{dt} Z_i(t_1, t, z) = \{Z_i(t_1, t, z), h\} = J(Dh)Z_i(t_1, t, z), \tag{9.36}$$

we deduce, by differentiating, that

$$\frac{\partial}{\partial t_2} \Xi(t_1, t_2, z) = J\tilde{H}\Xi(t_1, t_2, z), \tag{9.37}$$

where $\Xi(t_1, t_2, z)$ is the Jacobian matrix of $Z(t_1, t_2, z)$. From $Z(t_1, t_1, z) = z$, we deduce that $\Xi(t_1, t_1, z) = I$, next that Ξ is the general solution of the linear hamiltonian system (9.37), and consequently, is symplectic according to (9.34). The time evolution transformation is thus canonical, according to definition (8.21).

⋆⋆⋆

These time evolution transformations form a group, since we have, for any t_1, t_2, t_3,

$$S_h(t_1, t_2)S_h(t_2, t_3) = S_h(t_1, t_3) \quad \text{and} \quad S_h^{-1}(t_1, t_2) = S_h(t_2, t_1).$$

$S_h(t_1, t_2)$ is derivable, and we have

$$\frac{\partial}{\partial t_1} S_h(t_1, t_2) = -S(t_1, t_2)L(h(t_1)) \quad \text{and} \quad \frac{\partial}{\partial t_2} S_h(t_1, t_2) = L(h(t_2))S(t_1, t_2).$$

If, for example, h no longer is time-dependent, we then have a group of parameter t, since $S_h(t_1, t_2) = S_h(0, t_2 - t_1)$. In this case

$$\frac{\partial S_h(t)}{\partial t} = -S_h(t)L_h = -L_h S_h(t), \tag{9.38}$$

and we are back to the equality $S_h L_h S_h^{-1}(t) = L_h$, that is to say that $S_h h = h$.

⋆⋆⋆

If we know the exact form of S_h, then we can solve the differential equation

$$\frac{\partial f}{\partial t} + \{f, h\} = g. \tag{9.39}$$

which is solved by noting that

$$\frac{\partial}{\partial t_1} S_h(t_1, t_2)f(t_1) = S_h(t_1, t_2)g(t_1),$$

which gives

$$S_h(t_1, t_2)f(t_1) = \int_{t_0}^{t_1} S_h(t, t_2)g(t)dt + S_h(t_0, t_2)f(t_0),$$

and consequently,

$$f(t) = S_h(t_0, t)f(t_0) + \int_{t_0}^{t} S_h(t_1, t)g(t_1)dt_1. \tag{9.40}$$

10. Families of Canonical Transformations

Given $w'(p,q,\theta)$, a family of functions of phase space of class C^1, let us consider the hamiltonian system given by

$$\frac{\partial p_l}{\partial \theta} = -\frac{\partial w'}{\partial q_l}, \quad \frac{\partial q_l}{\partial \theta} = \frac{\partial w'}{\partial p_l}, \quad 1 \le l \le n. \tag{10.1}$$

The mapping T_θ at time θ of the hamiltonian flow of w', verifies, for any z,

$$\begin{aligned} \frac{\partial}{\partial \theta} T_\theta z &= -\{w'(T_\theta z), T_\theta z\} \\ &= -\{T_\theta w'(z), T_\theta z\} \\ &= -T_\theta L_{w'} z, \end{aligned} \tag{10.2}$$

which solves the differential equation

$$\frac{\partial T_\theta}{\partial \theta} = -T_\theta L_{w'}. \tag{10.3}$$

Let us call w, the function whose derivative[2] with respect to θ is w', and vanishes at $\theta = 0$. If $L_{w'}$ commutes with L_w, that is to say if $[L_w, L_{w'}] = L_{\{w,w'\}} = 0$, then we have, for any $p \ge 0$,

$$\frac{\partial L_w^p}{\partial \theta} = p \frac{\partial L_w}{\partial \theta} L_w^{p-1} = p L_{w'} L_w^{p-1},$$

therefore

$$\frac{\partial}{\partial \theta} e^{-L_w} = -e^{-L_w} L_{w'} = -L_{w'} e^{-L_w}.$$

— If $w' = w'_0$ does not depend on θ, then $[L_{w'}, L_{\theta w'}] = L_{\{w',\theta w'\}} = 0$, $\frac{\partial}{\partial \theta}[\theta w'] = w'$ therefore, the solution of (10.3) is

$$T_\theta = e^{-\theta L_{w'}} = e^{-L_w}.$$

— If $w'(p,q,\theta) = a(\theta)\tilde{w}(p,q)$ then, calling $A(\theta) = \int_0^\theta a(\nu)d\nu$ the integral (see footnote) of a, which vanishes at 0, we have $w = A(\theta)\tilde{w}$ and $[A\tilde{w}, a\tilde{w}] = Aa[\tilde{w},\tilde{w}] = 0$. Similarly, the solution of (10.3) is given by

$$T_\theta = e^{-\int_0^\theta a(\nu)d\nu \tilde{w}} \tag{10.4}$$

— On the other hand, if $L_{w'}$ does not commute with L_w, the solution is no longer an exponential, but what we call the Lie or Deprit transform in relation to the exponential. Nevertheless, the exponential of a Lie operator is a canonical transformation (6.7). We can also show (cf. [42]), that the solution of (10.3) is an exponential.

[2] The integral of w: primitive in the French text.

10.1. General Form of Transformations

Let us now focus on the canonical transformations that are formal series

$$T = \sum_{n\geq 0} \varepsilon^n T_n \quad \text{and} \quad T^{-1} = \sum_{n\geq 0} \varepsilon^n T_n^{-1}.$$

The fact that T is a canonical transformation imposes some conditions on the various terms T_n. The following theorem indicates two possible representations of these transformations, which justifies, in a certain way, the use of either the Dragt-Finn transformation, or the Deprit transformation.

PROPOSITION 25. — *Given a transformation T, such that Tz_i is a formal series for any $1 \leq i \leq 2n$:*

$$Tz_i = z_i + \sum_{j\geq 1} \varepsilon^j a_{i,j}(z) = \sum_{j\geq 0} \varepsilon^j a_{i,j}(z), \tag{10.5}$$

where $a_{i,j}(z)$ is a function of $z = (z_1, \dots, z_{2n})$,

- *there exists a formal series $w = \sum_{n\geq 1} \varepsilon^n w_n$, such that T^{-1} is the mapping of the hamiltonian flow $\frac{\partial w}{\partial \varepsilon}$ integrated for at time ε,*
- *there exists a formal series $g = \sum_{n\geq 1} \varepsilon^n g_n$ such that T^{-1} is the infinite product of the mappings of the autonomous hamiltonian flows $\varepsilon^n g_n$ integrated for a unit of time.*

Remark. — We can consider the mappings of the autonomous hamiltonian flows $\varepsilon^n g_n$ integrated for one unit of time, as mappings at time ε^n of the autonomous hamiltonian flows g_n, or as the mappings at time ε of the non-autonomous hamiltonian flows $n\varepsilon^{n-1} g_n$.

The proof of this property may be found in similar form in [14], but there exist more general results for the form of the canonical transformations which continuously depend on a parameter. The proof only relies on algebraic considerations on the Poisson bracket, and more specifically on the following lemma, which states that any derivative is an inner derivative. The same proof may apply when showing proposition 17, page 20.

LEMMA 26 (Dragt-Finn [14]). — *Given the canonical coordinates $(z_1, \dots, z_{2n})$ and a set of $2n$ functions $f_1, \dots, f_{2n}$, there is equivalence between the two following properties:*

(i) for any $1 \leq i, j \leq 2n$, we have $\{z_i, f_j\} = \{z_j, f_i\}$,

(ii) there exists g such that for any $1 \leq i \leq 2n$, we have $f_i = \{z_i, g\}$.

Proof. — If we have for $1 \leq i \leq 2n$, $f_i = \{z_i, g\}$, then we deduce

$$\{z_i, f_j\} = \{z_i, \{z_j, g\}\} = -\{z_j, \{g, z_i\}\} - \{g, \{z_i, z_j\}\} = \{z_j, \{z_i, g\}\} = \{z_j, f_i\}$$

that is to say (i) is obtained using the Jacobi identity and the fact that $\{z_i, z_j\}$ is a constant.

Conversely, let us suppose that we have (i). Let us call $z^* = Jz$, then we have

$$\frac{\partial z_i^*}{\partial z_j} = J_{i,j}, \quad 1 \leq i \leq 2n. \tag{10.6}$$

For any f we then deduce

$$\{z_i, f\} = \sum_{1\leq j,k\leq 2n} \frac{\partial z_i}{\partial z_j} J_{j,k} \frac{\partial f}{\partial z_k} = \sum_{1\leq k\leq 2n} J_{i,k} \frac{\partial f}{\partial z_k} = \frac{\partial f}{\partial z_i^*}, \tag{10.7}$$

so that we have, for any $1 \leq i, j \leq 2n$,

$$\frac{\partial f_j}{\partial z_i^*} = \frac{\partial f_i}{\partial z_j^*}.$$

Hence we deduce $\sum f_i dz_i^*$ is an exact form, and that $g = -\int \sum f_i dz_i^*$ is indeed defined, and satisfies, for any $1 \leq i \leq 2n$, $\{z_i, g\} = f_i$.

⁂

In order to show proposition 25, we construct the g_k term by term. Let us show the following result:

LEMMA 27 (Dragt-Finn [14]). — *There exists $g_1, \ldots, g_p, \ldots$ such that for any $p \geq 1$, we have*

$$e^{\varepsilon^p L_{g_p}} \cdots e^{\varepsilon L_{g_1}} T z_i = z_i + \sum_{j>p} \varepsilon^j a_{i,j}^{(p)}, \quad 1 \leq i \leq 2n. \tag{10.8}$$

• If $p = 1$, T is a canonical transformation. If we expand $\{z_i, z_j\} = \{Tz_i, Tz_j\}$ in ε, we get

$$0 = \varepsilon\left(\{z_i, a_{j,1}(z)\} + \{a_{i,1}(z), z_j\}\right) + \sum_{n\geq 2} \varepsilon^n \sum_{p+q=n} \{a_{i,p}(z), a_{j,q}(z)\}, \tag{10.9}$$

and, identifying term by term the coefficients of the series,

$$\{z_i, a_{j,1}(z)\} + \{a_{i,1}(z), z_j\} = 0.$$

According to lemma 26, there exists g_1 such that for any $1 \leq i \leq 2n$ we have

$$a_{i,1} = \{z_i, g_1\} = -L_{g_1} z_i. \tag{10.10}$$

Hence we deduce that

$$\begin{aligned} e^{\varepsilon L_{g_1}} T z_i &= e^{\varepsilon L_{g_1}} (z_i + \sum_{n\geq 1} \varepsilon^n a_{i,n}(z)) \\ &= z_i + \sum_{n\geq 1} \varepsilon^n \sum_{p+q=n} \frac{L_{g_1}^p}{p!} a_{i,q}(z) \\ &= z_i + \varepsilon L_{g_1} z_i + a_{i,1}(z) + \sum_{n\geq 2} \varepsilon^n \sum_{p+q=n} \frac{L_{g_1}^p}{p!} a_{i,q}(z) \\ &= z_i + \sum_{n\geq 2} \varepsilon^n a_{i,n}^{(1)}(z). \end{aligned} \tag{10.11}$$

• Let us suppose the property (10.8), satisfied at rank $k-1$. Since $e^{\varepsilon^p L_{g_p}}$ is a canonical transformation, for any $1 \leq p \leq k-1$, we deduce that $e^{\varepsilon^{k-1} L_{g_{k-1}}} \cdots e^{\varepsilon L_{g_1}} T$ is a canonical transformation, and consequently, for any $1 \leq i, j \leq 2n$, we have

$$\{z_i + \sum_{m\geq k} \varepsilon^m a_{i,m}^{(k-1)}(z), z_j + \sum_{m\geq k} \varepsilon^m a_{j,m}^{(k-1)}(z)\} = \{z_i, z_j\}.$$

In the same way, we deduce that

$$0 = \left(\{z_i, a_{j,k}^{(k-1)}(z)\} + \{a_{i,k}^{(k-1)}(z), z_j\}\right) + \sum_{m\geq k+1} \varepsilon^m \sum_{kp+q=m} \{a_{i,p}^{(k-1)}(z), a_{j,q}^{(k-1)}(z)\},$$

which, identifying the terms of degree k, gives

$$\{z_i, a_{j,k}^{(k-1)}(z)\} + \{a_{i,k}^{(k-1)}(z), z_j\} = 0$$

and proves the existence of g_k, such that, for any $1 \leq i \leq 2n$, we have

$$a_{i,k}^{(k-1)} = \{z_i, g_k\} = -L_{g_k} z_i.$$

Similarly, we deduce that

$$\begin{aligned} e^{\varepsilon^k L_{g_k}} \cdots e^{\varepsilon L_{g_1}} T_{z_i} &= e^{\varepsilon^k L_{g_k}} \cdots e^{\varepsilon L_{g_1}} (z_i + \sum_{m \geq k} \varepsilon^m a_{i,m}^{(k-1)}(z)) \\ &= z_i + \sum_{m\geq k} \varepsilon^m \sum_{kp+q=m} \frac{L_{g_k}^p}{p!} a_{i,q}^{(k-1)}(z) \\ &= z_i + \varepsilon^k \left(L_{g_k} z_i + a_{i,k}^{(k-1)}(z)\right) + \sum_{m\geq k+1} \varepsilon^m \sum_{kp+q=m} \frac{L_{g_k}^p}{p!} a_{i,q}^{(k-1)}(z) \\ &= z_i + \sum_{m\geq k+1} \varepsilon^m a_{i,m}^{(k)}(z). \end{aligned} \tag{10.12}$$

***

In order to demonstrate the second part of proposition 25, we use following lemma:

LEMMA 28 (Dragt-Finn [14]). — *If T is a canonical transformation, there exists a function $h(Tz, \varepsilon)$ such that*

$$\frac{\partial Tz}{\partial \varepsilon} = -\{h(Tz, \varepsilon), Tz\}.$$

Proof. — From $\{Tz_i, Tz_j\} = \{z_i, z_j\}$, we deduce that, for any $1 \leq i, j \leq 2n$,

$$\sum_{p,q\geq 0} \varepsilon^p \varepsilon^q \{a_{i,p}(z), a_{j,q}(z)\} = \{z_i, z_j\}. \tag{10.13}$$

When differentiating the previous equation with respect to ε, we have

$$\sum_{p,q\geq 0} \varepsilon^p \varepsilon^{q-1} \{a_{i,p}(z), q a_{j,q}(z)\} + \varepsilon^{p-1} \varepsilon^q \{p a_{i,p}(z), a_{j,q}(z)\} = 0 \tag{10.14}$$

therefore

$$\{\sum_{p\geq 0} \varepsilon^p a_{i,p}(z), \sum_{q\geq 0} q\varepsilon^{q-1} a_{j,q}(z)\} + \{\sum_{p\geq 0} p\varepsilon^{p-1} a_{i,p}(z), \sum_{q\geq 0} \varepsilon^q a_{j,q}(z)\} = 0,$$

that is to say,

$$\{Tz_i, \frac{\partial Tz_j}{\partial \varepsilon}\} = \{Tz_j, \frac{\partial Tz_i}{\partial \varepsilon}\},$$

with the Poisson bracket taken in z, and T canonical, we have, according to (8.24),

$$\{Tz_i, \frac{\partial Tz_j}{\partial \varepsilon}\}_z = \{Tz_i, \frac{\partial Tz_j}{\partial \varepsilon}\}_{Tz},$$

and since the Tz_i are canonical coordinates, there exists, according to lemma 28, a function $h(Tz, \varepsilon)$, which satisfies, for any $1 \leq i \leq 2n$,

$$\frac{\partial Tz_i}{\partial \varepsilon} = -\{h(Tz, \varepsilon), Tz_i\} = -T\{h(z, \varepsilon), z_i\} = -TL_h z_i. \tag{10.15}$$

Then we have

$$\frac{\partial T}{\partial \varepsilon} = -TL_h. \tag{10.16}$$

Let us pose $h = \frac{\partial w}{\partial \varepsilon}$, to be consistent with previous material, and to later obtain more concise formulas. Specifically if $w = \varepsilon^n w_n$, we again come up with the exponential of a Lie operator.

The existence of w is subordinate to the equality

$$\{Tz_i, \frac{\partial Tz_j}{\partial \varepsilon}\} = \{Tz_j, \frac{\partial Tz_i}{\partial \varepsilon}\}, \tag{10.17}$$

which, for example, is satisfied if T is of class C^2, or satisfies for any $1 \leq i, j < 2n$

$$\frac{\partial^2 Z_i}{\partial z_j \partial \varepsilon} = \frac{\partial^2 Z_i}{\partial \varepsilon \partial z_j}. \tag{10.18}$$

10.2. Deprit Transformation

The type T_f Lie transformation is also called the Deprit transformation. It was introduced by Deprit [11]. Given a generating series

$$w = \sum_{n \geq 1} \varepsilon^n w_n, \tag{10.19}$$

we focus on the mapping, at time ε, of the non-autonomous hamiltonian flow given by the hamiltonian function $\frac{\partial w}{\partial \varepsilon}$. Let us denote T_w, a one parameter family, we have,

$$T_w(0) = I, \quad \frac{\partial T_w}{\partial \varepsilon} = -T_w L_{\frac{\partial w}{\partial \varepsilon}}. \tag{10.20}$$

From (10.20), we deduce that T_w^{-1} verifies the differential equation

$$T_w^{-1}(0) = I, \quad \frac{\partial T_w^{-1}}{\partial \varepsilon} = L_{\frac{\partial w}{\partial \varepsilon}} T_w^{-1}. \tag{10.21}$$

The solution is expanded in the small parameter ε, and we get inductively

$$T_w = \sum_{n \geq 0} \varepsilon^n (T_n)_n, \quad T_w^{-1} = \sum_{n \geq 0} \varepsilon^n (T_w^{-1})_n, \tag{10.22}$$

with

$$(T_w)_0 = I, \quad (T_w)_n = -\sum_{p=1}^{n} \tfrac{p}{n} (T_w)_{n-p} L_{w_p}, \tag{10.23}$$

$$(T_w^{-1})_0 = I, \quad (T_w^{-1}) = \sum_{p=1}^{n} \tfrac{p}{n} L_{w_p} (T_w^{-1})_{n-p}. \tag{10.24}$$

Whenever there will be no ambiguity, we will denote T_n for $(T_w)_n$ and T_n^{-1} for $(T_w^{-1})_n$.

Remark. — This transformation, which I shall take the liberty of calling the Deprit transformation, is not, strictly speaking, the Deprit transformation. In [11], it in fact denotes the mapping of the hamiltonian flow of w (instead of $\frac{\partial w}{\partial \varepsilon}$). I used the other notation for the sake of conciseness in the formulas, and also because this formulation is used, specifically, to provide estimates (cf. [22, 25, 52, 24]).

The Deprit transformation, just as its inverse, is a canonical transformation, since it is the mapping, at time ε, of a hamiltonian flow, and direct computation shows that it is linear, that it maintains the product and Poisson bracket (cf. [11, 22] or (6.30) page 16).

The general term T_n is expanded into

$$T_n = \sum_{k_1+\cdots+k_j=n} (-1)^j \tfrac{k_1}{k_1} \tfrac{k_2}{k_1+k_2} \cdots \tfrac{k_j}{k_1+\cdots+k_j} L_{w_{k_1}} \cdots L_{w_{k_j}}, \tag{10.25}$$

and T_n^{-1} into

$$T_n^{-1} = \sum_{k_1+\cdots+k_j=n} \tfrac{k_1}{k_1} \tfrac{k_2}{k_1+k_2} \cdots \tfrac{k_j}{k_1+\cdots+k_j} L_{w_{k_1}} \cdots L_{w_{k_j}}. \tag{10.26}$$

The previous sums are the sums for all possible j-uplets of strictly positive integers, whose sum is n. The number t_n of terms in (10.25) or (10.26) is the number of increasing sequences $\{1, \ldots, n\}$, that is to say that

$$t_n = 2^{n-1}. \tag{10.27}$$

For $f = \sum_{p\geq 0} \varepsilon^p f_p$ we have, for the first terms of $F = \sum_{p\geq 0} \varepsilon^p F_p = T_w^{-1} f$,

$$F_0 = f_0$$

$$F_1 = f_1 - \{f_0,\, w_1\}$$

$$F_2 = f_2 - \{f_0,\, w_2\} - \{f_1,\, w_1\} + \tfrac{1}{2}\, \{\{f_0,\, w_1\},\, w_1\}$$

$$\begin{aligned} F_3 = {} & f_3 - \{f_0,\, w_3\} - \{f_1,\, w_2\} - \{f_2,\, w_1\} + \tfrac{2}{3}\, \{f_0,\, \{w_1,\, w_2\}\} + \{\{f_0,\, w_2\},\, w_1\} \\ & + \tfrac{1}{2}\, \{\{f_1,\, w_1\},\, w_1\} - \tfrac{1}{6}\, \{\{\{f_0,\, w_1\},\, w_1\},\, w_1\} \end{aligned}$$

$$\begin{aligned} F_4 = {} & f_4 - \{f_0,\, w_4\} - \{f_1,\, w_3\} - \{f_2,\, w_2\} - \{f_3,\, w_1\} + \tfrac{3}{4}\, \{f_0,\, \{w_1,\, w_3\}\} \\ & + \tfrac{2}{3}\, \{f_1,\, \{w_1,\, w_2\}\} + \tfrac{1}{2}\, \{\{f_0,\, w_2\},\, w_2\} + \{\{f_0,\, w_3\},\, w_1\} + \{\{f_1,\, w_2\},\, w_1\} \\ & + \tfrac{1}{2}\, \{\{f_2,\, w_1\},\, w_1\} - \tfrac{1}{4}\, \{f_0,\, \{w_1,\, \{w_1,\, w_2\}\}\} - \tfrac{2}{3}\, \{\{f_0,\, \{w_1,\, w_2\}\},\, w_1\} \\ & - \tfrac{1}{2}\, \{\{\{f_0,\, w_2\},\, w_1\},\, w_1\} - \tfrac{1}{6}\, \{\{\{f_1,\, w_1\},\, w_1\},\, w_1\} \\ & + \tfrac{1}{24}\, \{\{\{\{f_0,\, w_1\},\, w_1\},\, w_1\},\, w_1\}. \end{aligned}$$

10.3. Exponential

Given a series

$$k = \sum_{n \geq 1} \varepsilon^n k_n$$

we consider $T_{k(\varepsilon)}$, the mapping at time ε of the flow associated with the autonomous hamiltonian system

$$\varepsilon \frac{dz}{d\tau} = \{z, k\}. \tag{10.28}$$

Therefore, we have $\varepsilon \dfrac{\partial T_{k(\varepsilon)}}{\partial \tau} = -T_{k(\varepsilon)} L_k$ and consequently, $T_{k(\varepsilon)} = e^{-L_k}$, $\quad T_{k(\varepsilon)}^{-1} = e^{L_k}$. The general term T_n^{-1} of $T_{k(\varepsilon)}^{-1}$ is expanded into

$$T_n^{-1} = \sum_{k_1 + \cdots + k_j = n} \frac{L_{k_1} \cdots L_{k_j}}{j!}. \tag{10.29}$$

The number of elements in (10.29) is the number of increasing sequences of $\{1, \ldots, n-1\}$, that is to say, 2^{n-1}. For $f = \sum_{p \geq 0} \varepsilon^p f_p$, the first terms of $F = \sum_{p \geq 0} \varepsilon^p F_p = e^{L_k} f$ are

$$
\begin{aligned}
F_0 &= f_0 \\
F_1 &= f_1 - \{f_0,\, k_1\} \\
F_2 &= f_2 - \{f_0,\, k_2\} - \{f_1,\, k_1\} + \tfrac{1}{2}\,\{\{f_0,\, k_1\},\, k_1\} \\
F_3 &= f_3 - \{f_0,\, k_3\} - \{f_1,\, k_2\} - \{f_2,\, k_1\} + \tfrac{1}{2}\,\{f_0,\, \{k_1,\, k_2\}\} + \{\{f_0,\, k_2\},\, k_1\} \\
&\quad + \tfrac{1}{2}\,\{\{f_1,\, k_1\},\, k_1\} - \tfrac{1}{6}\,\{\{\{f_0,\, k_1\},\, k_1\},\, k_1\} \\
F_4 &= f_4 - \{f_0,\, k_4\} - \{f_1,\, k_3\} - \{f_2,\, k_2\} - \{f_3,\, k_1\} + \tfrac{1}{2}\,\{f_0,\, \{k_1,\, k_3\}\} + \tfrac{1}{2}\,\{f_1,\, \{k_1,\, k_2\}\} \\
&\quad + \tfrac{1}{2}\,\{\{f_0,\, k_2\},\, k_2\} + \{\{f_0,\, k_3\},\, k_1\} + \{\{f_1,\, k_2\},\, k_1\} + \tfrac{1}{2}\,\{\{f_2,\, k_1\},\, k_1\} \\
&\quad - \tfrac{1}{6}\,\{f_0,\, \{k_1,\, \{k_1,\, k_2\}\}\} - \tfrac{1}{2}\,\{\{f_0,\, \{k_1,\, k_2\}\},\, k_1\} - \tfrac{1}{2}\,\{\{\{f_0,\, k_2\},\, k_1\},\, k_1\} \\
&\quad - \tfrac{1}{6}\,\{\{\{f_1,\, k_1\},\, k_1\},\, k_1\} + \tfrac{1}{24}\,\{\{\{\{f_0,\, k_1\},\, k_1\},\, k_1\},\, k_1\}.
\end{aligned}
$$

10.4. Dragt-Finn Transformation

Let us consider the infinite product, at time ε^n, of autonomous hamiltonian flows g_n. The Dragt-Finn transformation, as well as its inverse, are infinite products

$$M_g = e^{-\varepsilon L_{g_1}} \cdots e^{-\varepsilon^n L_{g_n}} \cdots, \quad M_g^{-1} = \cdots e^{\varepsilon^n L_{g_n}} \cdots e^{\varepsilon L_{g_1}}. \tag{10.30}$$

The Dragt-Finn transform is an infinite product of Deprit transformations, which makes it a canonical transformation.

Remark. — It seems that the idea of this transformation might have been Kamel's, in the early 70's. However, the name Dragt-Finn remains attached to this transformation.

We can expand M_g and M_g^{-1} as a function of the parameter ε:

$$M_g = \sum_{n \geq 0} \varepsilon^n M_n, \quad M_g^{-1} = \sum_{n \geq 0} \varepsilon^n M_n^{-1}, \tag{10.31}$$

with the general form

$$M_0^{-1} = I, \quad M_n^{-1} = \sum_{\wp(1,n;n)} \frac{L_{g_n}^{m_n} \cdots L_{g_1}^{m_1}}{m_n! \cdots m_1!}. \tag{10.32}$$

Here, $\wp(1,n;n)$ is the set of partitions of n, defined in (6.39), that is to say

$$\wp(1,n;n) = \{(m_1,\ldots,m_n), m_1 + 2m_2 + \cdots + nm_n = n\}.$$

We know (cf. Andrews [1]) that the number p_n of partitions of n is equivalent, whenever n tends toward the infinite, at

$$\frac{e^{\pi\sqrt{\frac{2n}{3}}}}{4n\sqrt{3}}, \tag{10.33}$$

that is to say that there are radically fewer terms in (10.32) than in the other expansions corresponding to the other transformations.

For $f = \sum_{p\geq 0} \varepsilon^p f_p$, the first terms of $F = \sum_{p\geq 0} \varepsilon^p F_p = M_g^{-1} f$ are

$$\begin{aligned}
F_0 &= f_0\\
F_1 &= f_1 - \{f_0, g_1\}\\
F_2 &= f_2 - \{f_0, g_2\} - \{f_1, g_1\} + \tfrac{1}{2}\,\{\{f_0, g_1\}, g_1\}\\
F_3 &= f_3 - \{f_0, g_3\} - \{f_1, g_2\} - \{f_2, g_1\} + \{f_0, \{g_1, g_2\}\} + \{\{f_0, g_2\}, g_1\}\\
&\quad + \tfrac{1}{2}\,\{\{f_1, g_1\}, g_1\} - \tfrac{1}{6}\,\{\{\{f_0, g_1\}, g_1\}, g_1\}\\
F_4 &= f_4 - \{f_0, g_4\} - \{f_1, g_3\} - \{f_2, g_2\} - \{f_3, g_1\} + \{f_0, \{g_1, g_3\}\} + \{f_1, \{g_1, g_2\}\}\\
&\quad + \tfrac{1}{2}\,\{\{f_0, g_2\}, g_2\} + \{\{f_0, g_3\}, g_1\} + \{\{f_1, g_2\}, g_1\} + \tfrac{1}{2}\,\{\{f_2, g_1\}, g_1\}\\
&\quad - \tfrac{1}{2}\,\{f_0, \{g_1, \{g_1, g_2\}\}\} - \{\{f_0, \{g_1, g_2\}\}, g_1\} - \tfrac{1}{2}\,\{\{\{f_0, g_2\}, g_1\}, g_1\}\\
&\quad - \tfrac{1}{6}\,\{\{\{f_1, g_1\}, g_1\}, g_1\} + \tfrac{1}{24}\,\{\{\{\{f_0, g_1\}, g_1\}, g_1\}, g_1\}.
\end{aligned}$$

11. Computation of Inverse Transforms

The computation of inverse transforms can be done by evaluating the expansion formulas of the transformations. But, since the number of the monomials is considerable, we compute the transforms directly.

In order to compute the inverse transform by the Deprit transformation, we use the algorithm proposed by Henrard [29], which takes into consideration the inductive relations between the T_p^{-1}, and incidentally, is more convenient, from the computation point of view, than the algorithm proposed by Deprit. This algorithm is often used to carry out estimations of the radius of convergence, as, for example, in Giorgilli's, Galgani's or Simò's work. [25, 22, 8].

For the Dragt-Finn transformation, since we do not know *a priori* of any closed formula between the M_p^{-1}, we use the fact that the transformation is an infinite product of exponentials of homogeneous terms.

Given a formal series $f = \sum_{n\geq 0} \varepsilon^n f_n$, we denote F the inverse transform of f:

$$F = T^{-1} f = \sum_{n\geq 0} \varepsilon^n F_n. \tag{11.1}$$

11.1. Deprit Transformation

From $F_n = \sum_{k=0}^{p} T_p^{-1} f_{n-p}$ and the relations $T_p^{-1} f_q = \sum_{k=1}^{p} \frac{k}{p} L_{w_k} T_{p-k}^{-1} f_q$, by-products of (10.24), we successively construct $F_{p,q} = T_p^{-1} f_q$ for $p + q \leq r$. We must then fill out a chart

$$\begin{array}{cccccccc}
f_0 & f_1 & f_2 & \cdots & f_i & \cdots & \cdots & f_n \\
 & F_{1,0} & F_{1,1} & F_{1,2} & & \ddots & \vdots & F_{1,n-1} \\
 & & \ddots & \ddots & & \ddots & & \\
 & & & \ddots & \ddots & & \ddots & \vdots \\
 & & & & F_{i,0} & F_{i,1} & \cdots & F_{i,n-i} \\
 & & & & & \ddots & & \vdots \\
\downarrow & \downarrow & \downarrow & \cdots & \downarrow & \cdots & \cdots & F_{n,0} \\
F_0 & F_1 & F_2 & \cdots & F_i & \cdots & & F_n
\end{array} \tag{11.2}$$

column by column. Each term $F(j, i-j)$ is deduced from the previous ones $F(k, i-j)$, which are in the same diagonal, the diagonals being independent from each other. In particular if $f_i = F(i,0) = 0$, then we have $F(k,i) = 0$ for any k. Each term F_i is found as the sum of the elements in column $F(j, i-j)$.

With initial values $F_0 = f_0, \ldots, F_r = f_r$, we use the loop:

for $i := 1$ **to** r **do**
  for $j := 1$ **to** i **do**
    for $k := 1$ **to** j **do** $F_{j,i-j} := F_{j,i-j} + \frac{k}{j}\{w_k, F_{j-k,i-j}\}$;
    $F_i := F_i + F_{j,i-j}$.

In the previous algorithm, we evaluate a Poisson bracket of a term of order ε^p, and a term of order ε^q, whenever $k = p$, and $j - k + i - j = q$, or, similarly, whenever $k = p$, and $i = p + q$. The number of such Poisson brackets is the number of elements of $\{j; p \leq j \leq p+q\}$, that is to say, $q + 1$.

The number of computed Poisson brackets gives a good indication of the cost involved, but one must also take into consideration the number of terms stored in memory during the computation. In the algorithm proposed for the Deprit transformation, one must store all the terms $F_{p,q}$, for $p + q \leq r$ in memory, in order to compute $F_0, \ldots, F_r$. Furthermore, it is a good idea to store them, in the event one should have to compute F_{r+1}.

The inverse Deprit transformation thus requires evaluating $q + 1$ Poisson brackets involving terms of order p and q, and storing in memory $k + 1$ terms of order k.

Remark. — Although it does not affect the number of necessary evaluations, it might prove more interesting to compute the elements of the chart, diagonal per diagonal. Each diagonal represents a $T^{-1} f_i$. This technique gives us the option of storing in memory one single diagonal, – sum of the previous diagonals–. However, if we wish to compute F_{r+1}, we must compute the entire chart again.

11.2. Dragt-Finn Transformation

Since there is, *a priori*, no closed formula between the various M_n^{-1}, the computation cost of $\sum_{m=0}^{n} M_{n-m}^{-1} f_m$ ought to be equivalent to the cost of the evaluation of $\sum_{i \le n} p(i)$ Lie monomials, which is, all things considered, substantially less than 2^n. But the fact that M^{-1} is a product of exponentials of homogeneous terms compels us to use another algorithm.

Given a function $f = \sum_{n \ge 0} \varepsilon^n f_n$, we will construct an auxiliary sequence of series,

$$F_0 = f, F^k = e^{\varepsilon^k L_{g_k}} F^{k-1}.$$

Then we have

$$F_n^k = \sum_{jk+p=n} \tfrac{1}{j!} L_{g_k}^j F_p^{k-1}, \tag{11.3}$$

and consequently $F_n = F_n^n$. Given $F_{|r}^{k-1} = \sum_{p=0}^{r} \varepsilon^p F_p^{k-1}$, we construct successively $F_{|r}^{k,0} = F_{|r}^{k-1}$, $F_{|r}^{k,i} = \frac{1}{i} L_{g_k} F_{|r}^{k,i-1}$ so that $i \le [\frac{r}{k}] = \max\{n \in N; n \le \frac{r}{k}\}$. Then we get $F_{|r}^{k}$ as $\sum_{ki \le r} F_{|r}^{k,i}$.

Next, we construct the chart

$$\begin{array}{cccccccc}
f_0 & f_1 & f_2 & \cdots & f_i & \cdots & \cdots & f_r \\
 & F_1^1 & F_2^1 & F_3^1 & & \ddots & \vdots & F_r^1 \\
 & & \ddots & \ddots & & \ddots & & \\
 & & & \ddots & \ddots & & \ddots & \vdots \\
 & & & & F_i^i & F_{i+1}^i & \cdots & F_r^i \\
 & & & & & \ddots & & \vdots \\
\downarrow & \downarrow & \downarrow & \cdots & \downarrow & \cdots & \cdots & F_r^r \\
F_0 & F_1 & F_2 & \cdots & F_i & \cdots & & F_r
\end{array} \tag{11.4}$$

with the algorithm:

for $i := 1$ **to** r **do**
  for $k := i$ **to** r **do**
    for $j := 1$ **to** $[\frac{k}{i}]$ **do** $F_{i+j-1,k} := \frac{1}{j}\{g_i, F_{i+j-2,k-i}\}$;
  for $k := i$ **to** r **do**
    for $j := 2$ **to** $[\frac{k}{i}]$ **do** $F_{i,k} := F_{i,k} + F_{i+j-1,k}$;
    $F_{i,k} = F_{i,k} + F_{i-1,k}$.

The chart is filled out line by line, which is the main difference with the algorithm for the Deprit transformation. We temporarily store the $F_k^{i,j}$ in the same chart, on the $(i+j-1)^{\text{th}}$ line.

We get a Poisson bracket between a term of order ε^p, and a term of order ε^q when $i = p$ and $k - i = q$, thus the number of such Poisson brackets is equal to $[\frac{p+q}{p}]$.

On the other hand, the algorithm uses a chart which is partially erased at every loop. This chart is filled by the $F_p^{i,j}$ for $ij \le p \le r$. During the computation of the F^i, it is filled, on the one hand, by $F^{i'}$ for $0 \leq i' \leq i$, which makes $\inf(i, k+1)$ terms of order k yielded, on the other hand, by the $F^{i,j}$, which also yields $[\frac{k}{i}]$ terms of order k. Therefore, we have simultaneously $\inf(i, k+1) + [\frac{k}{i}]$ terms of order k in the chart at the end of the i^{th} loop. If $i \geq k$ then $\inf(i, k+1) + [\frac{k}{i}] = k+1$, else $\inf(i, k+1) + [\frac{k}{i}] = i + [\frac{k}{i}] \leq i + \frac{k}{i} \leq k+1$, so that the chart contains at most $k+1$ terms of order k.

11.3. Cost of Algorithms

From $[\frac{p+q}{p}] \leq \frac{p+q}{p} \leq q+1$, we deduce that the computation of the inverse Dragt-Finn transform costs less in evaluations of Poisson brackets of terms of order p and q, than an inverse Deprit transform. The total number of Poisson brackets needed to compute $F_0, \ldots, F_r$ by the inverse Deprit transformation is

$$t'_r = \sum_{1 \leq p+q \leq r} q+1 = \sum_{i=1}^{r} \sum_{q=0}^{i-1} q+1 = \sum_{i=1}^{r} \tfrac{i(i+1)}{2} = \frac{r(r+1)(r+2)}{6} = \frac{r^3}{6} + O(r^2). \quad (11.5)$$

For the Dragt-Finn transformation, we find

$$m'_r = \sum_{1 \leq p+q \leq r} [\tfrac{p+q}{p}] = \sum_{i=1}^{r} \sum_{p=1}^{i} (\tfrac{i}{p} + O(1)) = \sum_{i=1}^{r} (i \log i + O(i)) = \tfrac{1}{2} r^2 \log r + O(r^2). \quad (11.6)$$

Let us recapitulate in a chart for given perturbation orders, the number of products of Lie operators in $\sum_{k \leq r} T_k^{-1}$ (resp. $\sum_{k \leq r} M_k^{-1}$) and the number of Poisson brackets evaluated when computing $F_{|r} = T^{-1} f$ (resp. M^{-1}) with the previous algorithms.

Cost of Transformations

Order	Lie Operators		Evaluated Poisson Brackets	
	$\sum_{k<r} M_k^{-1}$	$\sum_{k<r} T_k^{-1}$	$F_{\|r} = M^{-1} f$	$F_{\|r} = T^{-1} f$
1	1	1	1	1
2	3	3	4	4
3	6	7	9	10
4	11	15	17	20
5	18	31	27	35
6	29	63	41	56
7	44	127	57	84
8	66	255	77	120
9	96	511	100	165
10	138	1023	127	220
15	683	32767	314	680
20	2713	1048575	600	1540
30	28628	1073741823	1492	4960

Table 1

°°°

The Dragt-Finn transform uses fewer Poisson brackets than the Deprit transform. Furthermore, the Deprit transformation and the Dragt-Finn transformation imply storing $k+1$ terms of order k (they may be polynomials of degree k or $k+2$). The difference becomes apparent at order 3, and is obvious at order 10 and beyond. This order may seem high, but while computing normal forms for the moon (cf. Henrard [29]), or for the stability study of the point $\mathcal{L}_4$, we carry the computation to at least order 30 (cf. Simò [52], Giorgilli *et al.* [25], Galgani & Giorgilli [8]).

It would undoubtedly be most interesting to search, for each of these two transformations, for the minimum number of evaluations of Lie brackets necessary to compute the various transforms. This could possibly be carried out by using rewriting techniques, based on a distributed representation of Lie polynomials.

Chapter III
Perturbations

In this section, we study certain aspects of the perturbation theory for the hamiltonian systems. We recall the results of this theory by contributing some specific information, as well as some answers to the questions which arise during implementation attempts.

A perturbed system implies a hamiltonian system approximating an integrable hamiltonian system. Here, the study of perturbed hamiltonian systems consists, on the one hand, of searching after formal invariants, and on the other hand, whenever possible, it involves a qualitative study of the system's stability in the vicinity of points of equilibrium.

In a specified framework, Lie transformations may provide methods which construct step by step formal invariants in the form of an expansion into Lie series which do not generally converge. When the first terms of these series are known, we can deduce finer asymptotic estimations for regions of stability (cf. Giorgilli *et al.* [25], or Giorgilli's and Galgani's work, or Simò's work [52]). This theory is comparable to Nekhoroshev's, in so far as we only consider the case of unperturbed Hamiltonians, which are linear in the actions.

In this section, we recall the two methods suggested by the two types of transformations previously mentioned: the Dragt-Finn transformation, and the Deprit transformation. We prove that in some cases, these two methods are equivalent.

We present various ways of solving the perturbation equation. This discussion essentially rests on the possibility of reducing a linear hamiltonian operator in a symplectic basis.

12. Invariants

In the following discussion, we place ourselves in the vicinity of a point of equilibrium of the hamiltonian system, that is to say we can write

$$h = \sum_{k\geq 0} \varepsilon^k h_k, \tag{12.1}$$

where h_k is a polynomial of degree $k+2$, and h_0, a quadratic form which we will presume non-degenerate. h_0 is, for example, the Hamiltonian of n uncoupled harmonic oscillators

$$h_0 = \tfrac{1}{2} \sum_{l=1}^{n} \omega_l (p_l^2 + q_l^2). \tag{12.2}$$

For a Hamiltonian such as this, the invariants are the actions $I_l = \frac{1}{2}(p_l^2 + q_l^2)$, and we are obviously concerned with the search for invariants of the perturbed system.

We can search for invariants directly, in the form of series, which are considered like perturbations of the actions. We then search for

$$\xi_l = I_l + \sum_{n\geq 1} \varepsilon^n \xi_{l,n}, \{\xi_l, h\} = 0, \quad 1 \leq l \leq n. \tag{12.3}$$

Therefore, we must seek to solve, for any $n \geq 0$,

$$0 = \{h, \xi_l\} = \sum_{n\geq 0} \varepsilon^n \sum_{p+q=n} \{h_p, \xi_{l,q}\} \tag{12.4}$$

$$= \sum_{n\geq 0} \varepsilon^n [\{h_0, \xi_{l,n}\} + \sum_{p=1}^{n} \{h_p, \xi_{l,n-p}\}] \tag{12.5}$$

$$= \sum_{n\geq 0} \varepsilon^n [\{h_0, \xi_{l,n}\} - \psi_{l,n}] \tag{12.6}$$

in which $\psi_{l,n}$ is known provided that we know $\xi_{l,p}$, for $0 \leq p \leq n-1$.

At this point, we give the following method in an attempt to formally construct the invariants: solve successively, for $n \geq 1$,

$$L_{h_0} \xi_{l,n} = \psi_{l,n}, \tag{12.7}$$

which does not necessarily have a solution.

Let us therefore suppose that $h_0 = \frac{1}{2} \sum_{l=1}^{n} \omega_l (p_l^2 + q_l^2)$. In this case, the canonical change of variables

$$x_l = \frac{1}{\sqrt{2}} (p_l + \mathrm{i} q_l), y_l = \frac{\mathrm{i}}{\sqrt{2}} (p_l - \mathrm{i} q_l), \quad 1 \leq l \leq n, \tag{12.8}$$

transforms h_0 into $\mathrm{i} \sum_{l=1}^{n} \omega_l x_l y_l$. L_{h_0} is then diagonalizable in the basis of monomials $x^i y^j$. We have

$$L_{h_0} x^j y^k = \mathrm{i}(k-j).\omega x^j y^k, \tag{12.9}$$

and we can easily find the solution of (12.7), as long as we have established that $\psi_{l,n}$ is in the range of L_{h_0}. *A priori*, there is no reason for this to happen, since at each step, $\xi_{l,n}$ is defined modulo an element of the kernel of h_0. In some cases, we show directly (cf. [23, 22]) that at every step, we can construct $\xi_{l,n}$, so that the equation (12.7) may be consistent. One of the conditions, for example, is the linear independence over $\mathbb{Q}$ of the ω_l (non-resonance condition) and, on the other hand, (cf. [8]), the reversibility of h, that is to say that $h(p,q) = h(p,-q)$.

⁂

Another approach consists of implementing a change of variables which transforms I_l into ξ_l, for any l. This change of variables is a canonical transformation T, or rather, a family of canonical transformations, which transforms h into $K = T^{-1}h$.

$K = T^{-1}h$ is the Hamiltonian of the system in the variables Tz_i. K must be constructed in such a way as to allow the reading of invariants. After setting an order r, we require, for any $1 \leq l \leq n$

$$\{K_p, I_l\} = 0 \quad \text{for} \quad 0 \leq p \leq r, \tag{12.10}$$

which means that the I_l are invariants at the order r. Let us denote then

$$\xi_l^{(r)} = TI_l = \sum_{p\geq 0} \xi_{l,p}^{(r)} = I_l + \sum_{p\geq 1} \xi_{l,p}^{(r)}. \tag{12.11}$$

From (12.10), we deduce that

$$\{I_l, K\} = \sum_{p\geq r+1} \varepsilon^p \{I_l, K_p\}, \tag{12.12}$$

next that

$$\begin{aligned} T\{I_l, K\} &= \sum_{n\geq 0} \varepsilon^n \sum_{p+q=n} T_{n-p}\{I_l, K_q\} \\ &= \sum_{n\geq r+1} \varepsilon^n \sum_{p=r+1}^{n} T_{n-p}\{I_l, K_p\}. \end{aligned} \tag{12.13}$$

But

$$\left\{\xi_l^{(r)}, h\right\} = \sum_{n\geq 0} \varepsilon^n \sum_{p+q=n}^{r} \left\{\xi_{l,p}^{(r)}, h_q\right\} \tag{12.14}$$

and, since T is canonical, we have

$$T\{I_l, K\} = \{TI_l, TK\} = \left\{\xi_l^{(r)}, h\right\} \tag{12.15}$$

and

$$\left\{\xi_l^{(r)}, h\right\} = \sum_{n\geq r+1} \varepsilon^n \sum_{p+q=n} \left\{\xi_{l,p}^{(r)}, h_q\right\}, \tag{12.16}$$

and, for any $0 \leq n \leq r$,

$$\sum_{p=0}^{n} \left\{\xi_{l,p}^{(r)}, h_{n-p}\right\} = \left\{\xi_{l,n}^{(r)}, h_0\right\} + \sum_{p=1}^{n} \left\{\xi_{l,p}^{(r)}, h_{n-p}\right\} = 0. \tag{12.17}$$

In particular, $\xi_l^{(r)}$ is an invariant of h, up to order $r+1$.

Generally speaking, a Hamiltonian K, satisfying an equation of type (12.10) is called a normal form. The normal form is determined by the type of invariants sought after. We shall now show how normal forms are constructed formally.

13. Normal Form

The normal form of a Hamiltonian h, refers to a function K which is the image of h by a (symplectic) canonical change of variables on which the invariants can be plainly read. The notion of a normal form is somewhat vague, and strongly depends on the use that is to be made of it.

For example, if h is time-independent, if $h_0 = \frac{1}{2}\sum_{l=1}^{n} \omega_l(p_l^2 + q_l^2)$ and if the ω_l are rationally independent, we shall say that K is a normal form of h, if K only depends on the actions, or if, –which amounts to the same thing in this case– h_0 is an invariant of K. We search after K in the form

$$K = K_0 + \sum_{n\geq 1} \varepsilon^n K_n(I_1, \dots, I_n). \tag{13.1}$$

We shall say that K is a normal form at the order r if

$$K = \mathcal{Z}^{(r)} + \mathcal{R}^{(r)} \quad \text{where} \quad \begin{cases} \mathcal{Z}^{(r)} &= \sum_{i=0}^{r} \mathcal{Z}_i^{(r)}(I_1, \dots, I_l) \\ \mathcal{R}^{(r)} &= \sum_{i\geq r+1} \mathcal{R}_i^{(r)}. \end{cases} \tag{13.2}$$

If h verifies the same hypotheses as previously, but the ω_l are no longer rationally independent, the module of resonance $\mathcal{M}_\omega \subset \mathbb{Z}^n$ may be introduced,

$$\mathcal{M}_\omega = \{(k_1, \dots, k_n); k.\omega = \sum_{l=1}^{n} k_l \omega_l = 0\}, \tag{13.3}$$

or any submodule $\mathcal{M}$ of $\mathbb{Z}^n$. If we now take our discussion from the point of view of the complex variables (x_l, y_l), defined in (12.8), we shall say that K is a normal form, with respect to $\mathcal{M}$, at the order r if

$$K = \mathcal{Z}^{(r)} + \mathcal{R}^{(r)} \quad \text{where} \quad \begin{cases} \mathcal{Z}^{(r)} &= \sum_{i=0}^{r} \sum_{\substack{|k|+|l|=r \\ k-l\in\mathcal{M}}} z_{k,l} x^k y^l \\ \mathcal{R}^{(r)} &= \sum_{i\geq r+1} R_i^{(r)}. \end{cases} \tag{13.4}$$

We notice that, for $\mathcal{M} = \{0\}$, we have

$$\mathcal{Z}^{(r)} = \sum_{i=0}^{r} \sum_{\substack{|k|+|l|=r \\ k-l\in\mathcal{M}}} z_{k,l} x^k y^l = \sum_{i=0}^{r} \sum_{2|k|=r} z_k x^k y^k = \sum_{i=0}^{r} \sum_{2|k|=r} z_k I^k, \tag{13.5}$$

and we come up again with the previous normal form.

If, for example, h is quasi-periodic, that is to say if h admits an expansion in reciprocal circular functions, we will be able to have as a normal form a Hamiltonian which is no longer time-dependent.

The main idea behind the construction of the normal form, is the construction of formal invariants.

13.1. Construction of the Normal Form

Here, we study the construction of normal forms, in the autonomous or non-autonomous case. However, in all that follows, we suppose that $h = \sum_{n\geq 0} \varepsilon^n h_n$ where h_0 is integrable. In all cases, we attempt to search for a canonical transformation near the identity,

$$T = \sum_{n\geq 0} \varepsilon^n T_n, \tag{13.6}$$

which, –contrary to the other methods, which give the change of variables in the form of implicit functions– offers the advantage of giving explicit change of variables. We suppose that the new Hamiltonian is written as

$$K = \sum_{n\geq 0} \varepsilon^n K_n. \tag{13.7}$$

In the general case, we look for a canonical transformation $T(\varepsilon, t)$, of class C^2. Let us call $Z = Tz$, we have, for $1 \le i, j \le 2n$,

$$\{Z_i, Z_j\} = \{z_i, z_j\} = \sum_{k,l=1}^{2n} \frac{\partial Z_i}{\partial z_k} J_{k,l} \frac{\partial Z_j}{\partial z_l}. \tag{13.8}$$

But T is of class C^2, therefore, the partial derivatives commute, and we have

$$\begin{aligned}
\frac{\partial}{\partial t}\{Z_i, Z_j\} &= \sum_{k,l=1}^{n} \frac{\partial^2 Z_i}{\partial t \partial z_k} J_{k,l} \frac{\partial Z_j}{\partial z_l} + \frac{\partial Z_i}{\partial z_k} J_{k,l} \frac{\partial^2 Z_j}{\partial t \partial z_l} \\
&= \sum_{k,l=1}^{n} \frac{\partial}{\partial z_k} \left[\frac{\partial Z_i}{\partial t}\right] J_{k,l} \frac{\partial Z_j}{\partial z_l} + \frac{\partial Z_i}{\partial z_k} J_{k,l} \frac{\partial}{\partial z_l} \left[\frac{\partial Z_j}{\partial t}\right] \\
&= \left\{\frac{\partial Z_i}{\partial t}, Z_j\right\} + \left\{Z_i, \frac{\partial Z_j}{\partial t}\right\} = 0,
\end{aligned} \tag{13.9}$$

and in the same fashion,

$$\left\{\frac{\partial Z_i}{\partial \varepsilon}, Z_j\right\} + \left\{Z_i, \frac{\partial Z_j}{\partial \varepsilon}\right\} = 0. \tag{13.10}$$

From the use of lemma 26, we deduce the existence of two functions u and v, such that, for any $1 \le i \le 2n$, we have

$$\frac{\partial Z_i}{\partial \varepsilon} = \{Z_i, u\} \quad \text{and} \quad \frac{\partial Z_i}{\partial t} = \{Z_i, v\}, \tag{13.11}$$

that is to say,

$$\frac{\partial T}{\partial \varepsilon} = -TL_u \quad \text{and} \quad \frac{\partial T}{\partial t} = -TL_v, \tag{13.12}$$

and for the inverses,

$$\frac{\partial T^{-1}}{\partial \varepsilon} = L_u T^{-1} \quad \text{and} \quad \frac{\partial T^{-1}}{\partial t} = L_v T^{-1}. \tag{13.13}$$

Besides, for the new variables Z_i, we have (cf. 9.32)

$$\begin{aligned}
\dot{Z}_i &= \frac{\partial Z_i}{\partial t} + \{Z_i, h(z, t, \varepsilon)\} && (13.14)\\
&= \frac{\partial Z_i}{\partial t} + \left\{Z_i, h(T^{-1}Z, t, \varepsilon)\right\} && (13.15)\\
&= \{Z_i, v\} + \{Z_i, H(Z, t, \varepsilon)\} && (13.16)\\
&= \{Z_i, K(Z, t, \varepsilon)\}. && (13.17)
\end{aligned}$$

Here,

$$K = H + v = T^{-1}h + v \tag{13.18}$$

is the Hamiltonian of the system expressed in the new variables $Z_i = Tz_i$. Taking the derivative of (13.18) with respect to ε, we find

$$\frac{\partial T}{\partial \varepsilon} K + T \frac{\partial K}{\partial \varepsilon} = \frac{\partial h}{\partial \varepsilon} + \frac{\partial Tv}{\partial \varepsilon}, \tag{13.19}$$

therefore, according to (13.12),

$$-L_u K + \frac{\partial K}{\partial \varepsilon} = T^{-1}\frac{\partial h}{\partial \varepsilon} + T^{-1}\frac{\partial T v}{\partial \varepsilon}. \tag{13.20}$$

But

$$\begin{aligned}
L_{\frac{\partial T v}{\partial \varepsilon}} &= \frac{\partial}{\partial \varepsilon}\left[L_{Tv}\right] && (13.21)\\
&= \frac{\partial}{\partial \varepsilon}\left[T L_v T^{-1}\right] && (13.22)\\
&= \frac{\partial}{\partial \varepsilon}\left[T \frac{\partial}{\partial t} T^{-1}\right] && (13.23)\\
&= \frac{\partial T}{\partial \varepsilon}\frac{\partial}{\partial t} T^{-1} + T\frac{\partial^2 T^{-1}}{\partial \varepsilon \partial t} && (13.24)\\
&= -T L_u \frac{\partial}{\partial t} T^{-1} + T\frac{\partial^2 T^{-1}}{\partial t \partial \varepsilon} && (13.25)\\
&= -T L_u \frac{\partial}{\partial t} T^{-1} + T\frac{\partial}{\partial t}\left[L_u T^{-1}\right] && (13.26)\\
&= T L_{\frac{\partial u}{\partial t}} T^{-1} && (13.27)\\
&= L_{T\frac{\partial u}{\partial t}}. && (13.28)
\end{aligned}$$

Therefore, within an independent constant of the phase variables, we have

$$\frac{\partial T v}{\partial \varepsilon} = T\frac{\partial u}{\partial t}. \tag{13.29}$$

Equation (13.20) becomes

$$\frac{\partial u}{\partial t} = \frac{\partial K}{\partial \varepsilon} - T^{-1}\frac{\partial h}{\partial \varepsilon} - L_u K. \tag{13.30}$$

Let us now call $u = \frac{\partial w}{\partial \varepsilon}$, so that the notations may be coherent with the previous sections, and let us search for w in the form $w = \sum_{n \geq 1} \varepsilon^n w_n$. T and T^{-1} are defined by their series, and we have

$$T_0 = T_0^{-1} = I, \quad T_n = -\sum_{p=1}^{n} \tfrac{p}{n} T_{n-p} L_{w_p} \quad T_n^{-1} = \sum_{p=1}^{n} \tfrac{p}{n} L_{w_p} T_{n-p}^{-1}. \tag{13.31}$$

Identifying term by term the elements of the series (13.20), we have

$$n\frac{\partial w_n}{\partial t} = nK_n - \sum_{p=1}^{n} p T_{n-p}^{-1} h_p - \sum_{p=1}^{n} L_{u_p} K_{n-p}, \tag{13.32}$$

therefore

$$n\frac{\partial w_n}{\partial t} = nK_n - nh_n - \sum_{p=1}^{n-1} p T_{n-p}^{-1} h_p - \sum_{p=1}^{n-1} L_{u_p} K_{n-p} - \{w_n, h_0\}. \tag{13.33}$$

We get the equation of perturbation, which we have to solve, for any n,

$$\frac{\partial w_n}{\partial t} + \{w_n, h_0\} = K_n - h_n - \sum_{p=1}^{n-1} \tfrac{p}{n}\left[T_{n-p}^{-1} h_p + L_{w_p} K_{n-p}\right]. \tag{13.34}$$

This equation is solved with respect to w_n and K_n, using (9.40). K_n must be chosen so that the equation has a solution in w_n, and K_n has the proper form. We shall specify these conditions later.

13.2. Deprit Method

The Deprit method is the method previously mentioned, since it involves the Deprit transformation. This method is used primarily for autonomous Hamiltonians. However, we saw that it adapts perfectly to its non-autonomous counterpart, provided that the perturbation equation can be solved (13.34).

In the autonomous case, we have $K = T^{-1}h$, that is to say, for each term

$$\begin{aligned} K_n &= \sum_{p=0}^{n} T_p^{-1} h_{n-p} && (13.35)\\ &= T_n^{-1} h_0 + h_n + \sum_{p=1}^{n-1} T_p^{-1} h_{n-p} && (13.36)\\ &= \{w_n, h_0\} + h_n + \sum_{p=1}^{n-1} \tfrac{p}{n}[L_{w_p} T_{n-p}^{-1} h_0] + T_p^{-1} h_{n-p}. && (13.37)\end{aligned}$$

We get the perturbation equation

$$\{h_0, w_n\} + K_n = h_n + \sum_{p=1}^{n-1} \tfrac{p}{n} L_{w_p} T_{n-p}^{-1} h_0 + T_p^{-1} h_{n-p}. \tag{13.38}$$

We can also directly use (13.34), which yields

$$\{w_n, h_0\} = K_n - h_n - \sum_{p=1}^{n-1} \tfrac{p}{n}\left[T_{n-p}^{-1} h_p + L_{w_p} K_{n-p}\right], \tag{13.39}$$

that is to say,

$$\{h_0, w_n\} + K_n = h_n + \sum_{p=1}^{n-1} \tfrac{p}{n}\left[T_{n-p}^{-1} h_p + L_{w_p} K_{n-p}\right]. \tag{13.40}$$

13.3. Dragt-Finn Method

With the Dragt-Finn method, we search for the transformation, in the form of the Dragt-Finn transformation. In the autonomous case, from $K = M^{-1}h$, we deduce that

$$\begin{aligned} K_n &= \sum_{p=0}^{n} M_p^{-1} h_{n-p} && (13.41)\\ &= M_n^{-1} h_0 + h_n + \sum_{p=1}^{n-1} T_p^{-1} h_{n-p} && (13.42)\\ &= \{g_n, h\}_0 + h_n + \sum_{\wp(1,n-1;n)} \frac{L_{g_{n-1}}^{m_{n-1}} \cdots L_{g_1}^{m_1}}{m_1! \cdots m_{n-1}!} h_0 + \sum_{p=1}^{n-1} M_p^{-1} h_{n-p}. && (13.43)\end{aligned}$$

Hence we deduce, as in the case of the Deprit method, the perturbation equation

$$\{h_0, g_n\} + K_n = h_n + \sum_{\wp(1,n-1;n)} \frac{L_{g_{n-1}}^{m_{n-1}} \cdots L_{g_1}^{m_1}}{m_1! \cdots m_{n-1}!} h_0 + \sum_{p=1}^{n-1} M_p^{-1} h_{n-p}. \tag{13.44}$$

We can also use the auxiliary sequence

$$K^{(0)} = h, K^{(p)} = e^{\varepsilon^p L_{g_p}} K^{(p-1)}. \tag{13.45}$$

The successive elements are defined by

$$K_k^{(n)} = \sum_{np+q=k} \frac{L_{g_n}^p}{p!} K_q^{(n-1)}, \tag{13.46}$$

and, in particular,

$$K_n = K_n^{(n)} = \{g_n, h_0\} + K_n^{(n-1)}. \tag{13.47}$$

We then get the perturbation equation the form

$$\{h_0, g_n\} + K_n = K_n^{(n-1)}. \tag{13.48}$$

***

In the non-autonomous case, we add a degree of freedom to phase space (cf [7]), which is the time t, so that we must have as the new Poisson bracket $\{,\}'$:

$$L'_f g = \{f, g\}' = \{f, g\} + \frac{\partial f}{\partial t}\frac{\partial g}{\partial e} - \frac{\partial f}{\partial e}\frac{\partial g}{\partial t}. \tag{13.49}$$

From $\dot{t} = 1 = \frac{\partial h'}{\partial e}$ and $\dot{e} = -\frac{\partial h'}{\partial t}$, we simply have to take $h' = h + e$, that is to say,

$$h' = h_0 + e + \sum_{n\geq 1} \varepsilon^n h_n. \tag{13.50}$$

We then look for a series,

$$g(\varepsilon, t) = \sum_{n\geq 1} \varepsilon^n g_n(t), \tag{13.51}$$

and a new Hamiltonian

$$K = M'^{-1} h = \cdots e^{\varepsilon^n L'_{g_n}} \cdots e^{\varepsilon L'_{g_1}} h. \tag{13.52}$$

Here, since $\frac{\partial h_n}{\partial e} = 0$, if $n \geq 1$, we have

$$L'_{g_k} h_n = L_{g_k} h_n, \tag{13.53}$$

and for the first term h_0

$$L'_{g_k} h_0 = L_{g_k} h_0 + \tfrac{\partial g_k}{\partial t} \quad \text{and} \quad L'_{g_p} L'_{g_q} h_0 = L_{g_p} L_{g_q} h_0. \tag{13.54}$$

In particular, for the Dragt-Finn transformation, we have

$$K_n = \sum_{p=0}^{n} M'^{-1}_p h'_{n-p} \tag{13.55}$$

$$= M'^{-1}_n h_0 + \sum_{p=0}^{n-1} M_p^{-1} h_{n-p} \tag{13.56}$$

$$= L'_{g_n} h'_0 + \sum_{\wp(1,n-1;n)} \frac{L_{g_{n-1}}^{m_{n-1}} \cdots L_{g_1}^{m_1}}{m_1! \cdots m_{n-1}!} h'_0 + \sum_{p=0}^{n-1} M_p^{-1} h_{n-p} \tag{13.57}$$

$$= \frac{\partial g_n}{\partial t} + L_{g_n} h_0 + \sum_{\wp(1,n-1;n)} \frac{L_{g_{n-1}}^{m_{n-1}} \cdots L_{g_1}^{m_1}}{m_1! \cdots m_{n-1}!} h_0 + \sum_{p=0}^{n-1} M_p^{-1} h_{n-p} \tag{13.58}$$

$$= \frac{\partial g_n}{\partial t} + \sum_{p=0}^{n} M_p^{-1} h_{n-p}, \tag{13.59}$$

which yields the perturbation equation

$$\frac{\partial g_n}{\partial t} + \{g_n, h_0\} = K_n - h_n - \sum_{\wp(1,n-1;n)} \frac{L_{g_{n-1}}^{m_{n-1}} \cdots L_{g_1}^{m_1}}{m_1! \cdots m_{n-1}!} h_0 - \sum_{p=1}^{n-1} M_p^{-1} h_{n-p}. \tag{13.60}$$

Using the auxiliary sequence

$$K^{(0)} = h, K^{(p)} = e^{\varepsilon^p L'_{g_p}} K^{(p-1)}, \tag{13.61}$$

we get, as for the autonomous case,

$$\frac{\partial g_n}{\partial t} + \{g_n, h_0\} = K_n - K_n^{(n-1)}. \tag{13.62}$$

13.4. Computation Algorithms of the Normal Form and the Generating Series

Here, we suppose that we have an algorithm to solve the perturbation equation involved in the various methods. We denote by **solve** such an algorithm. For the Dragt-Finn method as for the Deprit method, we can use almost the same algorithms as for the computation of the inverse transformations (cf. pages 49-51). Let us take $h = h_0 + \sum_{n\geq 1} \varepsilon^n h_n$.

For the Deprit method, we construct the same chart $K^w(p,q)$, but, at every loop, we must solve the perturbation equation (13.38)

$$L_{h_0} w_n + K_n^w = R_n^w = \sum_{p=1}^{n-1} \tfrac{p}{n} \{w_p, K^w(n-p, 0)\} + \sum_{p=1}^{n-1} K^w(n-p, p). \tag{13.63}$$

We use the following algorithm:

for $i := 1$ **to** r **do**
  for $j := 1$ **to** $i - 1$ **do**
    for $k := 1$ **to** j **do** $K_{j,i-j} := K_{j,i-j} + \frac{k}{j}\{w_k, K_{j-k,i-j}\}$;
    $R_i := R_i + K_{j,i-j}$;
  for $k := 1$ **to** $i - 1$ **do**
    $R_i := R_i + \frac{k}{i}\{w_k, K_{i-k,0}\}$
    $K_{i,0} := K_{i,0} + \frac{k}{i}\{w_k, K_{i-k,0}\}$;
  solve $L_{h_0} w_i + K_i = R_i$
  $K_{i,0} := K_{i,0} + \{w_i, K_{0,i}\}$.

With the Dragt-Finn method, the computation of $K_n^{n,1}$ is replaced by the solution of the equation (13.40)

$$L_{h_0} g_n + K_n^{n,1} = K_n^{n-1}. \tag{13.64}$$

The algorithm used is the following:

for $i := 1$ **to** r **do**
  solve $L_{h_0} g_i + K_{i,i} = K_{i-1,i}$;
  for $k := i+1$ **to** r **do**
    for $j := 1$ **to** $[\frac{k}{i}]$ **do** $K_{i+j-1,k} := \frac{1}{j}\{g_i, K_{i+j-2,k-i}\}$;
  for $k := i$ **to** r **do**
    for $j := 2$ **to** $[\frac{k}{i}]$ **do** $K_{i,k} := K_{i,k} + K_{i+j-1,k}$
    $K_{i,k} := K_{i,k} + K_{i-1,k}$.

The cost of the solution to the perturbation equation by the algorithm **solve** being the same for both methods, the difference is established by the number of respective evaluations, which tips the scale in favor of the Dragt-Finn method. These results are in part confirmed by the study carried out by Fried & Ezra ([21]), on the Hénon & Heiles Hamiltonian. However, in the absence of a description of their algorithm to compute the Dragt-Finn transformation, one is led to think that it could be improved, which would confirm that the Dragt-Finn method is faster.

14. Solution of the Perturbation Equation

Whether with the Dragt-Finn method or the Deprit method, the computation algorithm of the normal form of a Hamiltonian h requires that the perturbation equations (13.34), (13.38), (13.40), (13.44), (13.48), (13.60) or (13.62) must be solved. In the autonomous case, the perturbation equation may be summarized as

$$\{h_0, w_n\} + K_n = R_n, \tag{14.1}$$

and for the non-autonomous case, as

$$\frac{\partial w_n}{\partial t} + \{w_n, h_0\} = K_n - R_n. \tag{14.2}$$

At each step, (14.1) or (14.2) must be solved by choosing K_n in order to simplify it, or to express it in terms of the invariants. The equation (14.2) may be solved, after choosing K_n, by acting with the flow of h_0 and by using (9.40).

14.1. The Case of Uncoupled Harmonic Oscillators

Let us now suppose that

$$h_0 = \tfrac{1}{2}\sum_{l=1}^{n} \omega_l (p_l^2 + q_l^2),$$

where the ω_l are rationally independent. By implementing the canonical change of variables (12.8),

$$x_l = \frac{1}{\sqrt{2}}\left(p_l + \mathrm{i} q_l\right), y_l = \frac{\mathrm{i}}{\sqrt{2}}\left(p_l - \mathrm{i} q_l\right), \tag{14.3}$$

h_0 becomes $\mathrm{i}\sum_{l=1}^{n} \omega_l x_l y_l$ and L_{h_0} is diagonalized in the basis of the monomials $x^i y^j$. In particular, the range and the kernel of L_{h_0} are in direct sum. Furthermore, since the ω_l are rationally independent, $x^j y^k \in \mathrm{Ker} L_{h_0}$ if

$$L_{h_0} x^j y^k = \mathrm{i}(k-j).\omega x^j y^k = 0, \tag{14.4}$$

that is to say, if $k = j$. Noting that

$$x^k y^k = x_1^{k_1} y_1^{k_1} \cdots x_n^{k_n} y_n^{k_n} = \left(\tfrac{\mathrm{i}}{2}(p_1^2 + q_1^2)\right)^{k_1} \cdots \left(\tfrac{\mathrm{i}}{2}(p_n^2 + q_n^2)\right)^{k_n}, \tag{14.5}$$

the kernel of L_{h_0} is then made up of linear combinations of products of actions.

So the solution of the perturbation equation (14.1) is the decomposition of R_n onto the range and the kernel of L_{h_0}.

In the basis of the eigenvectors $x^j y^k$, we proceed as follows: if

$$R_n = \sum_{|j|+|k|=n} r_{j,k} x^j y^k, \tag{14.6}$$

we take

$$K_n = \sum_{\substack{|j|+|k|=n \\ j=k}} r_{j,k} x^j y^k \quad \text{and} \quad w_n = \sum_{\substack{|j|+|k|=n \\ j\neq k}} \frac{r_{j,k}}{i(j-k).\omega} x^j y^k. \tag{14.7}$$

Another way of proceeding is to consider K_n as the projection of R_n on the kernel of L_{h_0}, then to solve

$$L_{h_0} L_{h_0} w_n = L_{h_0} R_n, \tag{14.8}$$

which has a unique solution, since L_{h_0} is an isomorphism on its range. This method may prove to be better if the change of variables (14.1) is costly.

In all cases, we construct K, step by step, so that h_0 is an invariant. From

$$h_0 = \tfrac{1}{2} \sum_{l=1}^{n} \omega_l (p_l^2 + q_l^2) = \sum_{l=1}^{n} r_l \tag{14.9}$$

where the r_l are in involution, we deduce that the r_l are n invariants of K.

14.1.1. Study of the Pendulum at Order 4 Using the Dragt-Finn Method

We use the Cartesian coordinates, and

$$h = \tfrac{1}{2}p^2 + (1 - \cos q) = \tfrac{1}{2}\left(p^2 + q^2\right) - \tfrac{1}{24}q^4 + \tfrac{1}{720}q^6 + \cdots. \tag{14.10}$$

The first equation based on (13.44) yields

$$\{h_0, g_1\} + K_1 = h_1 = 0, \tag{14.11}$$

which may be solved by $K_1 = h_1 = g_1 = 0$. Then, for $n = 2$, we have

$$L_{h_0} L_{h_0} g_2 = L_{h_0}\left(-\tfrac{1}{24}q^4\right) \quad \text{which gives} \quad g_2 = -\tfrac{5}{192}q^3 p - \tfrac{1}{64}qp^3, \tag{14.12}$$

and we get

$$K_2 = \{g_2, h_0\} + h_2 = -\tfrac{1}{64}\left(p^2+q^2\right)^2. \tag{14.13}$$

For $n = 3$, we get as previously

$$g_3 = 0, K_3 = 0. \tag{14.14}$$

Finally

$$L^2_{h_0} g_4 = L_{h_0}\left(-\tfrac{49}{30720}q^6 - \tfrac{23}{6144}q^4p^2 + \tfrac{3}{2048}q^2p^4 + \tfrac{1}{2048}p^6\right), \tag{14.15}$$

which yields

$$g_4 = -\tfrac{17}{15360}q^5p - \tfrac{1}{384}q^3p^3 + \tfrac{1}{1024}qp^5. \tag{14.16}$$

Therefore we have

$$K_4 = \{g_4, h_0\} - \tfrac{49}{30720}q^6 - \tfrac{23}{6144}q^4p^2 + \tfrac{3}{2048}q^2p^4 + \tfrac{1}{2048}p^6 = -\tfrac{1}{2048}(p^2+q^2)^3. \tag{14.17}$$

We thus find, at order 4

$$K = \tfrac{1}{2}(p^2+q^2) - \tfrac{1}{64}(p^2+q^2)^2 - \tfrac{1}{2048}(p^2+q^2)^3 + \cdots. \tag{14.18}$$

14.2. Reduction of the Linear Part

Since all the linear Hamiltonians are not Hamiltonians of uncoupled oscillators, we may ask ourselves whether we can by a change of variables bring them to a diagonal form.

In the case where we can find a symplectic change of variables which diagonalizes L_{h_0}, we will then be able to solve the perturbation equation by going to the complex variables (12.8).

Let us call $H = L_{h_0}$, the operator of the linear portion of h. From $JH^T + HJ = 0$, we deduce that the eigenvalues of H are in inverse pairs, and since H is a real operator, that they are complex conjugates. On the other hand, if $Hx = \lambda x$ and $Hy = \mu y$ then

$$\omega(x, Hy) + \omega(Hx, y) = (\lambda + \mu)\omega(x, y) = 0, \tag{14.19}$$

therefore if $\lambda + \mu \neq 0$, we have $\omega(x, y) = 0$.

***

Let us first demonstrate a lemma which may bring us back to systems of one degree of freedom.

LEMMA 29. — *If h_0 is positive definite, or if the eigenvalues of $H = L_{h_0}$ are distinct, then the phase space E decomposes into*

$$E = \bigoplus_{i=1}^{n} P_i, \tag{14.20}$$

where the P_i are stable planes for L_{h_0}, orthogonals for ω.

• If $h_0 = \frac{1}{2}\omega(x, H(x))$ is positive definite, h_0 defines a scalar product on the phase space. Therefore, there exists a matrix U of the linear group such that $U^T JHU = I$. But then, $J_1 = U^T JU$ is the antisymmetric matrix of an endomorphism $u = -u^*$. From $u^{2*} = (-u)(-u) = u^2$, we deduce that u^2 is self-adjoint for the scalar product, defined by I, therefore diagonalizable. Given x an eigenvector of u^2, then $P_x =< x, u(x) >$ is a stable plane by u (the only eigenvectors of u are the elements of the kernel), just as its orthogonal is stable by u^*, therefore by u. So we can decompose the space into a sum of stable orthogonal planes by u: $E = \bigoplus_{i=1}^{n} P_i$.

• In the general case, the characteristic polynomial of H is written $\prod_{l=1}^{n}(x^2 - \lambda_l^2)$, where the λ_l are eigenvalues of H. Hence we deduce that, if the λ_l are distinct, –though it is not always necessary– then

$$E = \bigoplus_{l=1}^{n} \ker\left(H^2 - \lambda_l^2 I\right) = \bigoplus_{l=1}^{n} P_l, \tag{14.21}$$

where the P_l are the eigenspaces associated with $H^2 - \lambda_l^2 I$, and are two by two orthogonals (according to 14.19) for the symplectic form ω.

⁂

Later, we examine the case where the space E decomposes into a direct sum of orthogonal planes for ω. After a change of the symplectic basis, we can then write (cf. [53]),

$$h_0 = \sum_{l=1}^{n} r_l, \quad \text{where} \quad r_l = \tfrac{1}{2}\alpha_l q_l^2 + \beta_l q_l p_l + \tfrac{1}{2}\gamma_l p_l^2, \tag{14.22}$$

and, in particular, for any $1 \leq i, j \leq n$,

$$\left[L_{r_l}, L_{r_j}\right] = L_{\{r_l, r_j\}} = 0. \tag{14.23}$$

⁂

By simple computation, or using (8.12), we deduce the form of the matrix of L_{r_l} in the basis p_l, q_l:

$$\begin{pmatrix} -\beta_l & \gamma_l \\ \alpha_l & \beta_l \end{pmatrix}. \tag{14.24}$$

The eigenvalues, $\pm\lambda_l$, of L_{r_l}, are given by

$$\lambda_l = \sqrt{\beta_l^2 - \alpha_l \gamma_l}. \tag{14.25}$$

Depending on the values of the $\alpha_l, \beta_l, \gamma_l$, we find a set of eigenvectors which form a symplectic basis (not necessarily real), except if one of the λ_l is zero, in which case, L_{r_l} is nilpotent.

• If r_l is defined, then $\beta_l^2 - \alpha_l\gamma_l < 0$ and

$$\lambda_l = \pm\mathrm{i}\sqrt{\alpha_l\gamma_l - \beta_l^2} = \pm\mathrm{i}\omega_l. \tag{14.26}$$

Given $e_l + if_l$, such that $H(e_l + if_l) = i\omega_l(e_l + if_l)$. We then have, for example, in the positive definite case, $He_l = -\omega_l f_l$, $Hf_l = \omega_l e_l$ and

$$h(f_l) = \tfrac{1}{2}\omega(f_l, Hf_l) = \tfrac{1}{2}\omega_l\omega(f_l, e_l) > 0. \tag{14.27}$$

Let us pose $P_l = \sqrt{\frac{2}{\omega_l}}e_l, Q_l = \sqrt{\frac{2}{\omega_l}}f_l$. We have $\omega(P_l, Q_l) = 1$ and, consequently, (P_l, Q_l) forms a symplectic basis, in which

$$r_l = \pm\tfrac{1}{2}w_l(P_l^2 + Q_l^2). \tag{14.28}$$

To conclude, if the restriction of h in every eigensubspace of H is defined, h_0 may be made orthogonal in a real symplectic basis, L_{h_0} is diagonalizable and, in particular, its range and its kernel are in direct sum, which makes the perturbation equation self-consistent.

• If $\beta_l^2 - \alpha_l\gamma_l > 0$, let us pose e_l and f_l, the eigenvectors of L_{r_l}, $He_l = \lambda_l e_l$ and $Hf_l = -\lambda_l f_l$. e_l and f_l are independent and $(P_l, Q_l = \frac{1}{\{e_l, f_l\}}f_l)$ forms a symplectic basis in which

$$r_l = \tfrac{1}{2}\lambda_l(P_l^2 - Q_l^2). \tag{14.29}$$

***

Since L_{h_0} is a derivation, we deduce that, if a and b are eigenvectors of L_{h_0}, associated with the eigenvalues λ_a and λ_b, then

$$\{h_0, ab\} = a\{h_0, b\} + \{h_0, a\}b = (\lambda_a + \lambda_b)ab, \tag{14.30}$$

which proves that ab is an eigenvector of L_{h_0}.

Let us suppose that we have diagonalized L_{h_0} in a symplectic basis (x, y) of E_1. The isomorphism $(p, q) \mapsto (x, y)$ of E_1 extends into a ring isomorphism over the ring of the polynomials E, so that for each degree n, the set of the products of the elements of the symplectic basis, is a basis of E_n, a basis of eigenvectors of the restriction of L_{h_0} to E_n.

14.3. General Autonomous Case

In this paragraph, let us go back to the idea developed by Steinberg ([53]), of using a Euclidian structure, and looking for adjoints of L_{h_0}. Nevertheless, we show that the interesting cases are reduced to the case where L_{h_0} is diagonalizable.

The idea behind the choice of K_n is the search for invariants in involution. The first being the construction of an invariant, we search for a function h^*, such that, for any n, we have $\{h^*, K_n\} = 0$. So we attempt to find h^*, such that the equation

$$L_{h_0}w_n + K_n = R_n \quad \text{and} \quad L_{h^*}K_n = 0 \tag{14.31}$$

will admit a solution. We search for h^*, such that L_{h^*} is an isomorphism on the range of $L^*_{h_0}$.

In the case where L_{h_0} is an isomorphism on its range (for example, diagonalizable in each homogeneous space), we can take $h^* = h_0$. But generally, L_{h_0} is not an isomorphism on its range.

The idea is to find h^*, such that L_{h^*} is the adjoint of L_{h_0}, for a given Euclidian structure (cf. Steinberg, [53]). For $x_{i,j} = p^i q^j \in E_k$, with $|i| + |j| = k$, let us consider the scalar product which makes the basis $x_{i,j}$ orthogonal, and such that

$$(x_{i,j} \mid x_{i,j}) = i_1! \cdots i_n! j_1! \cdots j_n!. \tag{14.32}$$

Noting that

$$(\frac{\partial x_{i,j}}{\partial q_l} \mid x_{r,s}) = (x_{i,j} \mid q_l x_{r,s}) \quad \text{and} \quad (\frac{\partial x_{i,j}}{\partial p_l} \mid x_{r,s}) = (x_{i,j} \mid p_l x_{r,s}), \tag{14.33}$$

we deduce by linearity, for $x_k, y_k \in E_k$, that

$$(\frac{\partial x}{\partial q_l} \mid y) = (x \mid q_l y) \quad \text{and} \quad (\frac{\partial x}{\partial p_l} \mid y) = (x \mid p_l y). \tag{14.34}$$

But then

$$(\{p_i p_j, x\} \mid y) = (p_i \frac{\partial x}{\partial q_j} + p_j \frac{\partial x}{\partial q_i} \mid y) = (x \mid q_j \frac{\partial y}{\partial p_i} + q_i \frac{\partial y}{\partial p_j}) = (x \mid -\{q_i q_j, y\}) \tag{14.35}$$

$$(\{p_i q_j, x\} \mid y) = (-p_i \frac{\partial x}{\partial p_j} + q_j \frac{\partial x}{\partial p_i} \mid y) = (x \mid -q_i \frac{\partial y}{\partial q_j} + p_j \frac{\partial y}{\partial p_i}) = (x \mid \{q_i p_j, y\}) \tag{14.36}$$

and we deduce that

$$(\{h_0(p,q), x\} \mid y) = (x \mid -\{h_0(-q,p), y\}), \tag{14.37}$$

which establishes that

$$h^*(p,q) = -h_0(-q,p) \tag{14.38}$$

satisfies

$$L_{h_0^*} = L_{h_0}^*. \tag{14.39}$$

In the case where $h^* = h_0^*$, equation (14.31) admits a solution for any $n \geq 1$. If, moreover, $\{h^*, h_0\} = 0$, then h^* is an invariant of the normal form. In the case where $\{h^*, h_0\} = 0$, then $[L_{h_0}, L_{h_0}^*] = 0$, so that L_{h_0} commutes with its adjoint and is therefore diagonalizable (eventually in a complex, non- symplectic basis). In particular, the kernel and the range of L_{h_0} are in direct sum. In fact, we can take h_0 as the invariant of the new Hamiltonian.

If $h_0 = \sum_{l=1}^n r_l$ where the r_l's are in involution, then

$$\{h^*, h_0\} = 0 \Leftrightarrow \{r_l, r_l^*\} = 0, \quad 1 \leq l \leq n. \tag{14.40}$$

If $r_l = \frac{1}{2}\alpha_l q_l^2 + \beta_l q_l p_l + \frac{1}{2}\gamma_l p_l^2$, then $r_l^* = -\frac{1}{2}\gamma_l q_l^2 + \beta_l q_l p_l - \frac{1}{2}\gamma_l p_l^2$ and

$$\{r_l, r_l^*\} = (\alpha_l + \gamma_l)(\beta_l q_l^2 + (\gamma_l - \alpha_l) q_l p_l - \beta_l p_l^2). \tag{14.41}$$

Hence we deduce that $\{r_l, r_l^*\} = 0$ if $r_l = \frac{1}{2}\omega_l(p_l^2 + q_l^2)$, or if $r_l = \frac{1}{2}\omega_l(p_l^2 - q_l^2 + \beta_l p_l q_l)$, which brings us back to the case $r_l = \frac{1}{2}\omega_l(P_l^2 - Q_l^2)$.

***

In conclusion, we are most often led to search for h_0 as the invariant of the new Hamiltonian. Whenever one of the r_l is nilpotent, this may pose a problem.

Depending on the cases, it may be more or less costly to reduce L_{h_0} in a symplectic basis in order to solve the perturbation equation, or to solve $L_{h^*} L_{h_0} x = L_{h^*} y$ directly by using linear algebra techniques.

14.4. Non-autonomous Equation

With the same hypotheses on h_0 (12.9), we can also solve the perturbation equation (14.2), since we know the flow of h_0. Going to action-angle variables

$$J_l = \tfrac{1}{2}(p_l^2 + q_l^2), \Phi_l = \operatorname{arctg}^{-1}\left(\frac{p_l}{q_l}\right), \quad 1 \le l \le n, \tag{14.42}$$

we have, under the action of h_0,

$$J_l(t) = J_l(0), \Phi_l(t) = \Phi_l(0) + \omega_l t, \quad 1 \le l \le n. \tag{14.43}$$

Whether the h_i do not depend on time, or depend on it through circular functions, we will solve the perturbation equation (14.2) by choosing K_n as the constant term of R_n. We then have

$$w_n(t) = S_{h_0}(0,t)\int_{t'} K_n(-t') - R_n(-t')dt'. \tag{14.44}$$

In the case where h is independent from time, we write

$$S_{h_0}(0,t)R_n(p(t),q(t)) = \sum_{|l|+|m|=n} r_{l,m} I^{\frac{l+m}{2}} e^{i(l-m).(\Phi+t\omega)} \tag{14.45}$$

and consequently, we take

$$K_n = \sum_{\substack{|l|+|m|=n\\ l=m}} r_{l,m} I^{\frac{l+m}{2}} = \sum_{l_1+\cdots+l_n=n} r_{l,l} I_1^{l_1}\cdots I_n^{l_n} \tag{14.46}$$

and

$$w_n = S(0,t)\sum_{\substack{|l|+|m|=n\\ l\neq m}} r_{l,m} J^{\frac{l+m}{2}} \int_{t'} e^{i(l-m).(\Phi-t'\omega)}dt' \tag{14.47}$$

$$= \sum_{\substack{|l|+|m|=n\\ l\neq m}} \frac{r_{l,m}}{i(l-m).\omega} J^{\frac{l+m}{2}} e^{i(l-m).\Phi}. \tag{14.48}$$

We come up with the same result as for the solution of (14.1).

14.4.1. Study of the Pendulum at Order 4 through the Deprit Method

Let us consider the Hamiltonian of the pendulum

$$h = \tfrac{1}{2}p^2 + (1-\cos q) = \tfrac{1}{2}(p^2+q^2) + \sum_{p\ge 1}\frac{(-1)^p}{(2p)!}q^{2p+2}. \tag{14.49}$$

The flow of h_0 is given by

$$q(t) = p(0)\sin(t) + q(0)\cos(t), p(t) = p(0)\cos(t) - q(0)\sin(t). \tag{14.50}$$

We use the action-angle coordinates, that is to say $j = \frac{1}{2}(p^2+q^2), \theta = \operatorname{arctg}^{-1}\left(\frac{q}{p}\right)$. The mapping at time t of the flow of h_0 becomes

$$j = j_0, \theta = \theta_0 + t. \tag{14.51}$$

We have

$$h_0 = j, h_1 = 0, h_2 = -\tfrac{j^2}{48}\left(3 + 4\cos(2\theta) - \cos(4\theta)\right), h_3 = 0,$$
$$h_4 = -\tfrac{j^3}{2080}\left(-10 + 15\cos(2\theta) - 6\cos(4\theta) + \cos(6\theta)\right).$$

The first equation based on (13.34) yields

$$\frac{\partial w_1}{\partial t} + \{w_1, h_0\} + K_1 = h_1 = 0 \tag{14.52}$$

which may be solved by taking $K_1 = h_1 = 0$ and $w_1 = 0$. For $n = 2$, we get

$$\frac{\partial w_2}{\partial t} + \{w_2, h_0\} = 2\left(K_2 + \tfrac{j^2}{48}\left(3 + 4\cos(2\theta) + \cos(4\theta)\right)\right), \tag{14.53}$$

which we solve by taking $K_2 = -\frac{1}{16}j^2$ and

$$w_2 = \int_t \tfrac{j^2}{24}\left(4\cos(2(\theta - t)) - \cos(4(\theta - t))\right) dt = \tfrac{j^2}{96}\left(8\sin(2\theta) - \sin(4\theta)\right). \tag{14.54}$$

Next $K_3 = h_3 = 0, g_3 = 0$, and finally

$$\frac{\partial w_4}{\partial t} + \{w_4, h_0\} = 4\left(K_4 + j^3\left(\tfrac{1}{256} - \tfrac{7}{384}\cos(2\theta) + \tfrac{1}{960}\cos(4\theta) + \tfrac{1}{640}\cos(6\theta)\right)\right), \tag{14.55}$$

which we solve by taking $K_4 = -\frac{1}{128}j^3$. So we find, at order 4

$$K = j - \tfrac{1}{48}j^2 - \tfrac{1}{256}j^3. \tag{14.56}$$

15. Uniqueness of the Normal Form

Here, we demonstrate the uniqueness of the normal form, under certain hypotheses. As a first step, we show the uniqueness of the normal form, whenever the canonical transformation is a Deprit transformation, then we show the uniqueness of the normal form, whether we choose a Deprit transformation or a Dragt-Finn transformation. The proof of this property also occurs in another form in the book by Siegel & Moser ([51]). First, let us give a proof of this classical proposition (the proof is sketched in [41]).

Proposition 30. — *Under the following hypotheses:*

(*i*) *$h_0 \in E_2$,*

(*ii*) *the kernel and the range of L_{h_0} are in direct sum.*

(*iii*) *for any $a, b \in \ker L_{h_0}$, we have $\{a, b\} = 0$,*

there exists a unique normal form K, and a generating series w such that $K = T_w^{-1}h$.

For $h_0 = \frac{1}{2}\sum_{l=1}^{n} \omega_l(p_l^2 + q_l^2)$ and the rationally independent ω_l's, the two first hypotheses are verified. Since the kernel of L_{h_0} is the vector subspace of E generated by products of actions, we deduce (*iii*) from

$$\{I_1^{k_1} \cdots I_n^{k_n}, I_1^{l_1} \cdots I_n^{l_n}\} = I_1^{k_1+l_1} \cdots I_n^{k_n+l_n} \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{k_i l_j}{I_i I_j}\{I_i, I_j\} = 0. \tag{15.1}$$

Given u and v two generating series, such that $K^u = T_u^{-1}h$ and $K^v = T_v^{-1}h$ are two normal forms of h, at a certain order r. Calling $T = T_u^{-1}T_v$ we deduce that $K^v = T^{-1}K^u$. By taking the derivative, we get

$$\frac{\partial T^{-1}}{\partial \varepsilon} = \frac{\partial T_v^{-1}}{\partial \varepsilon}T_u + T_v^{-1}\frac{\partial T_u}{\partial \varepsilon} = L(\tfrac{\partial v}{\partial \varepsilon})T_v^{-1}T_u - T_v^{-1}T_u L(\tfrac{\partial u}{\partial \varepsilon}) \tag{15.2}$$

therefore

$$\frac{\partial T^{-1}}{\partial \varepsilon} = \left[L(\tfrac{\partial v}{\partial \varepsilon}) - T^{-1}L(\tfrac{\partial u}{\partial \varepsilon})T\right]T^{-1} = L\left[\tfrac{\partial v}{\partial \varepsilon} - T^{-1}\tfrac{\partial u}{\partial \varepsilon}\right]T^{-1}. \tag{15.3}$$

T appears as the Lie transformation associated with w, where

$$\frac{\partial w}{\partial \varepsilon} = \frac{\partial v}{\partial \varepsilon} - T^{-1}\frac{\partial u}{\partial \varepsilon}, \tag{15.4}$$

which transforms the normal form K^u into the normal form K^v, at the same order r. Using (13.38) and $h_0 = K_0^u = K_0^v$, we deduce, for any n, the associated perturbation equation

$$\{h_0, w_n\} + K_n^v = R_n^u = K_n^u + \sum_{p=1}^{n-1} T_p^{-1}K_{n-p}^u + \sum_{p=1}^{n-1} \tfrac{p}{n}L(w_p)T_{n-p}^{-1}h_0. \tag{15.5}$$

Let us now prove recursively that, for any $0 \le n \le r$ and $1 \le i, i+j \le n$,

$$R_n^u = K_n^u = K_n^v, \quad \{h_0, w_n\} = 0 \quad \text{and} \quad T_i^{-1}K_j^u = 0. \tag{15.6}$$

Given an order r, (15.5) becomes, for $n = 1$,

$$\{h_0, w_1\} + K_1^v = K_1^u. \tag{15.7}$$

According to hypothesis (ii), since K_1^v and $K_1^u \in \ker L_{h_0}$, we have

$$\{h_0, w_1\} = 0 \quad \text{and} \quad K_1^u = K_1^v. \tag{15.8}$$

Let us suppose that we have (15.6), for $1 \le i < n \le r$, then (15.5) becomes

$$\begin{aligned}
\{h_0, w_n\} + K_n^v &= K_n^u + \sum_{p=1}^{n-1} T_p^{-1}K_{n-p}^u + \sum_{p=1}^{n-1} \tfrac{p}{n}L_{w_p}\underbrace{T_{n-p}^{-1}h_0}_{=0(15.6)} && (15.9)\\
&= K_n^u + \sum_{p=1}^{n-1}\sum_{q=1}^{p} \tfrac{q}{p}L_{w_q}T_{p-q}^{-1}K_{n-p}^u && (15.10)\\
&= K_n^u + \sum_{p=1}^{n-1}\sum_{q=1}^{p-1} \tfrac{q}{p}L_{w_q}\underbrace{T_{p-q}^{-1}K_{n-p}^u}_{=0(15.6)} + \sum_{p=1}^{n-1}\underbrace{L_{w_p}K_{n-p}^u}_{=0(iii)} && (15.11)\\
&= K_n^u && (15.12)
\end{aligned}$$

and consequently,

$$K_n^v = K_n^u \quad \text{and} \quad \{h_0, w_n\} = 0 \tag{15.13}$$

and, for $1 \leq i, i+j=n$, we have, according to (15.6),

$$T_i^{-1} K_j^u = \sum_{p=1}^{i-1} \tfrac{p}{i} \underbrace{T_{i-p}^{-1} K_j^u}_{=0(15.6)} + \underbrace{L_{w_p} K_j^u}_{=0(iii)} = 0. \tag{15.14}$$

Hence we deduce that, for any $n \leq r$,

$$\{h_0, w_n\} = 0 \quad \text{and} \quad K_n^u = K_n^v. \tag{15.15}$$

We thus have $K^u = K^v$, which proves the uniqueness of the normal form. However, the transformation T is not necessarily the identity, because it is sufficient that $w \in \ker L_{h_0}$.

15.1. Equivalence of the Methods

We will now deduce the equivalence of the Deprit and Dragt-Finn methods, as well as the uniqueness of the normal form.

PROPOSITION 31. — *Under the same conditions as for proposition 30, the two normal forms K^w and K^g obtained by the Deprit method and by the Dragt-Finn method, are equal.*

Let us pose M_g, the Dragt-Finn transformation which corresponds to the normal form K^g, that is to say we have $K^g = M_g^{-1} h$. According to proposition (15), there exists a series w' such that

$$T_{w'}^{-1} = M_g^{-1} \quad \text{therefore} \quad K^{w'} = T_{w'}^{-1} h = K^g. \tag{15.16}$$

By uniqueness of the normal form obtained with the Deprit method, we deduce that $K^w = K^g$.

Any normal form obtained by the Dragt-Finn method is the unique normal form obtained by the Deprit method. Consequently, it is unique. Therefore, both methods produce the same normal form, but the transformations used are not necessarily the same.

∘°∘

In the case of autonomous perturbed Hamiltonians, it will be to our advantage to use the Dragt-Finn method, rather than the Deprit method, since they provide the same objects at a reduced cost.

The solution of the perturbation equation may be carried out, either by diagonalizing L_{h_0}, which forces us to implement changes of variables, –which may prove very costly for high orders– or by solving linear systems, –which may also be very costly– since the dimensions of the n increase very rapidly. In any case, the use of the adjoint endomorphism does not yield any more answers in the case where the kernel and the range of L_{h_0} are not in direct sum.

Chapter IV
Estimations

In this section, we propose some estimations on the convergence of the series used with the perturbation methods. It is unusual for normal forms to converge, or to be able to find convergent invariants.

However, in light of Giorgilli's ([8, 25]) and Simò's ([52]) contributions, some results –concerning regions of stability in the vicinity of elliptical points– may be deduced from certain asymptotic estimations about the general term of the normal form.

There are several types of results. First of all, as we had done for the Deprit transformation, we give some estimations about the radius of convergence of the inverse transform of a series by the Dragt-Finn transformation. The results fall roughly in the same range as those found with the Lie transformation.

Next we use the results of [25, 52], in order to bound the gap between the new and old coordinates, then, in the same fashion, we give an estimation for the radius of convergence of the remainder of the normal form for any order r. This result is however inferior to the result obtained for the Deprit method, although I cannot say with certainty whether this phenomenon is due to greater precision in Giorgilli's results, *et al.* or to a property of the Deprit transformation.

Finally, since these results are inferior to those obtained for the Deprit method, we recall the way in which we deduced regions of stability. However, we shall see that the best estimations are not so much due to the Deprit transformation as to the existence of approximate invariants (12.16).

Remark. — I very recently learned of Fassò & Benettin's article ([18]), which also proposes some estimates for the same transformation. My goal, in this chapter, was to compare my estimations to those, given by Giorgilli for the Deprit method, by using the same norm. In this article, the authors remark, and rightly so, that they cannot carry out any comparisons, since the norms are vastly different. Despite this limitation, they eventually proceed to give stability results, which I do not propose to do.

16. Upper Bounds for the Transformations

In this section, we give an upper bound of the inverse transform of a function, by the Deprit or Dragt-Finn transformations.

Let us first recall a few useful inequalities. For a homogeneous polynomial of degree n

in the state variables $z_1, \dots z_{2n}$,

$$f = \sum_{|i|+|j|=n} f_{i,j} q^i p^j,$$

we denote

$$\|f\| = \sum_{|i|+|j|=n} |f_{i,j}|, \tag{16.1}$$

the norm L_1. We know (cf. [22]) that if f_1 and f_2 are polynomials of degree d_1 and d_2, we have

$$\|\{f_1, f_2\}\| \le d_1 d_2 \|f_1\| \, \|f_2\|, \tag{16.2}$$

where

$$L(f_1)f_2 = \{f_1, f_2\} = \sum_{i=1}^{n} \frac{\partial f_1}{\partial q_i}\frac{\partial f_2}{\partial p_i} - \frac{\partial f_1}{\partial p_i}\frac{\partial f_2}{\partial q_i}$$

is a polynomial of degree $d_1 + d_2 - 2$.

16.1. Upper bound for the Deprit Transformation

We know the following result (cf. [25]):

THEOREM 32 ([25]). — *Given $w = \sum_{k\ge 1} w_k$ and $f = \sum_{k\ge 1} f_k$ two functions of the phase space. We suppose that there exist constants a, b, c, d, such that, for any $n \ge 1$,*

— *w_n is of degree $n+2$ and $\|w_n\| \le a^{n-1}b$,*

— *f_n is of degree n and $\|f_n\| \le c^{n-1}d$,*

so, the function $F = \sum_{n\ge 0} F_n = T_w^{-1} f$ satisfies, for any $n \ge 0$,

$$\|F_n\| \le (\alpha a + \beta b + \gamma c)^{n-1}\, d, \tag{16.3}$$

where $\alpha = \frac{8}{3}, \beta = 3, \gamma = 1$.

16.2. Upper Bound for the Dragt-Finn Transformation

We call $M_g^{-1} = \cdots e^{L_{g_n}} \cdots e^{L_{g_1}}$, the inverse Dragt-Finn transformation associated with g. We propose to demonstrate the following result:

THEOREM 33. — *Given $g = \sum_{k\ge 1} g_k$ and $f = \sum_{k\ge 1} f_k$ two functions of the phase space. We suppose that there exist constants a, b, c, d, such that, for any $n \ge 0$,*

— *g_n is of degree $n+2$ and $\|g_n\| \le a^{n-1}b$,*

— *f_n is of degree n and $\|f_n\| \le c^{n-1}d$.*

So, the function $F = \sum_{n\ge 0} F_n = M_g^{-1} f$ satisfies, for any $n \ge 0$,

$$\|F_n\| \le (\alpha a + \beta b + \gamma c)^{n-1}\, d, \tag{16.4}$$

where α, β, γ are constants which will be determined later.

First of all, we demonstrate the following lemma:

LEMMA 34. — *Based on the hypotheses of theorem 33, we have, for* $F = \sum_{n\geq 0} F_n = e^{L_{g_k}} f$

$$\|F_n\| \leq C_{(k)}^{n-1} d \quad \textit{where} \quad C_{(k)} = \left(k(k+2)\|g_k\| + c^k\right)^{\frac{1}{k}}. \tag{16.5}$$

Proof. — According to (16.2), we have, based on the hypotheses of theorem 33

$$\|L(g_k)^p f_q\| \leq q(q+k)\cdots(q+(p-1)k)(k+2)^p \|g_k\|^p c^{q-1} d, \tag{16.6}$$

therefore

$$\begin{aligned}\sum_{kp+q=n} \frac{\|L(g_k)^p f_q\|}{p!} &\leq d \sum_{kp+q=n} \frac{q(q+k)\cdots(q+(p-1)k)}{p!}((k+2)\|g_k\|)^p c^{n-kp-1} \\ &= d \sum_{kp+q=n} \frac{(\frac{n}{k}-1)\cdots(\frac{n}{k}-p)}{p!}(k(k+2)\|g_k\|)^p c^{n-kp-1}. \end{aligned} \tag{16.7}$$

Let us denote, for $m \in \mathbb{Q}$ and $n \in \mathbb{N}$,

$$(m)_n = \frac{m(m-1)\cdots(m-n+1)}{n!}. \tag{16.8}$$

We then have

$$\begin{aligned}\|F_n\| &\leq d \sum_{kp+q=n} (\tfrac{n}{k}-1)_p (k(k+2)\|g_k\|)^p c^{n-kp-1} \\ &= d \sum_{p=0}^{[\frac{n}{k}]} (\tfrac{n}{k}-1)_p (k(k+2)\|g_k\|)^p c^{n-kp-1}. \end{aligned} \tag{16.9}$$

• If $n = kr$ is a multiple of k, then (16.9) becomes

$$\begin{aligned}\|F_n\| &\leq d \sum_{kp+q=kr} \frac{(r-1)\cdots(r-p)}{p!}(k(k+2)\|g_k\|)^p c^{k(r-1-p)+(k-1)} \\ &= dc^{k-1}\left(k(k+2)\|g_k\| + c^k\right)^{r-1} \\ &= d\left(k(k+2)\|g_k\| + c^k\right)^{\frac{n-1}{k}} \left[c^{k-1}\left(k(k+2)\|g_k\| + c^k\right)^{r-1-\frac{n-1}{k}}\right] \\ &= d\left(k(k+2)\|g_k\| + c^k\right)^{\frac{n-1}{k}} \left[c^{k-1}\left(k(k+2)\|g_k\| + c^k\right)^{\frac{1}{k}-1}\right] \\ &\leq d\left(k(k+2)\|g_k\| + c^k\right)^{\frac{n-1}{k}} c^{k-1}\left(c^k\right)^{(\frac{1}{k}-1)} \\ &= d\left(k(k+2)\|g_k\| + c^k\right)^{\frac{n-1}{k}} \end{aligned} \tag{16.10}$$

therefore

$$\|F_n\| \leq \sum_{kp+q=n} \frac{\|L(g_k)^p f_q\|}{p!} \leq d\left(k(k+2)\|g_k\| + c^k\right)^{\frac{n-1}{k}} = dC_{(k)}^{n-1}. \tag{16.11}$$

• If $n \neq kr$, we have $[\frac{n}{k}] < \frac{n}{k}$ therefore $[\frac{n}{k}] \leq \frac{n-1}{k}$ else we would have $n-1 < k[\frac{n}{k}] < n$, which is impossible. On the other hand, $\frac{n}{k} - 1 \leq \frac{n-1}{k}$ therefore $(\frac{n}{k}-1)_p \leq (\frac{n-1}{k})_p$ for $p \leq \frac{n-1}{k} - 1$, hence

$$\begin{aligned}
\sum_{kp+q=n} \frac{\|L(g_k)^p f_q\|}{p!} &\leq d\sum_{p=0}^{[\frac{n}{k}]} (\tfrac{n}{k}-1)_p (k(k+2)\,\|g_k\|)^p (c^k)^{\frac{n-1}{k}-p} \\
&\leq d\sum_{p=0}^{[\frac{n}{k}]} (\tfrac{n-1}{k})_p (k(k+2)\,\|g_k\|)^p (c^k)^{\frac{n-1}{k}-p} \\
&= dc^{n-1}\sum_{p=0}^{[\frac{n}{k}]} (\tfrac{n-1}{k})_p (k(k+2)\,\|g_k\|\,c^{-k})^p. \qquad (16.12)
\end{aligned}$$

According to Taylor's formula at the order $[\frac{n}{k}]$, where the remainder is expressed as an integral, we have, for $\frac{n-1}{k} \geq 1$,

$$(1+x)^{\frac{n-1}{k}} = \sum_{p=0}^{[\frac{n}{k}]} (\tfrac{n-1}{k})_p x^p + \int_0^1 \frac{(1-t)^{[\frac{n}{k}]}}{[\frac{n}{k}]!} (\tfrac{n-1}{k})_{[\frac{n}{k}]+1} (1+t)^{\frac{n-1}{k}-[\frac{n}{k}]-1} dt. \qquad (16.13)$$

But

$$\int_0^1 \frac{(1-t)^{[\frac{n}{k}]}}{[\frac{n}{k}]!} \underbrace{(\tfrac{n-1}{k})_{[\frac{n}{k}]+1}}_{\geq 0} (1+t)^{\frac{n-1}{k}-[\frac{n}{k}]-1} dt \geq 0, \qquad (16.14)$$

whence we deduce, by using (16.12), that

$$\begin{aligned}
\sum_{kp+q=n} \frac{\|L(g_k)^p f_q\|}{p!} &\leq dc^{n-1}(1 + k(k+2)\,\|g_k\|\,c^{-k})^{\frac{n-1}{k}} \qquad (16.15) \\
&= d(k(k+2)\,\|g_k\| + c^k)^{\frac{n-1}{k}}. \qquad (16.16)
\end{aligned}$$

Hence we deduce lemma 34, and if $\|g_k\| \leq a^{k-1}b$, that

$$\begin{aligned}
C^k_{(k)} &\leq k(k+2)\,\|g_k\| + c^k \\
&\leq (k+2)\left((k-1)a^k + b^k\right) + c^k \\
&= (k+2)(k-1)a^k + (k+2)b^k + c^k. \qquad (16.17)
\end{aligned}$$

***

Let us call $F^0 = f, \ldots, F^k = e^{L_{g_k}} F^{k-1}$, and demonstrate the following lemma:

LEMMA 35. — *For any k, we can write*

$$\left\|F^k_n\right\| \leq C_k^{n-1} d \quad \textit{where} \quad C^k_k \leq a^k_k a^k + b^k_k b^k + c^k_k c^k. \qquad (16.18)$$

Proof. — For $k = 1$, we have $C_1 = 3b + c$. So we can write (16.18) with

$$a_1 = 0, b_1 = 3, c_1 = 1. \tag{16.19}$$

According to (16.17), and supposing (16.18) for $k-1$, we have

$$C_k^k \leq (k+2)(k-1)a^k + (k+2)b^k + \left(a_{k-1}^{k-1}a^{k-1} + b_{k-1}^{k-1}b^{k-1} + c_{k-1}^{k-1}c^{k-1}\right)^{\frac{k}{k-1}}. \tag{16.20}$$

But the function $k \mapsto \left(\frac{x^k+y^k+z^k}{3}\right)^{\frac{1}{k}}$ is increasing (cf. [47]), which implies that we have

$$\begin{aligned} C_k^k &\leq (k+2)(k-1)a^k + (k+2)b^k + 3^{\frac{1}{k-1}}\left(a_{k-1}^k a^k + b_{k-1}^k b^k + c_{k-1}^k c^k\right) & (16.21)\\ &\leq a_k^k a^k + b_k^k b^k + c_k^k c^k & (16.22)\end{aligned}$$

with

$$\left\{\begin{array}{lcl} a_k^k &=& (k-1)(k+2) + 3^{\frac{1}{k-1}}a_{k-1}^k \\ b_k^k &=& k+2+3^{\frac{1}{k-1}}b_{k-1}^k \\ c_k^k &=& 3^{\frac{1}{k-1}}c_{k-1}^k. \end{array}\right. \tag{16.23}$$

For k=2, we have

$$C_2^2 \leq 4a^2 + 4b^2 + (3b+c)^2 \leq 4a^2 + 4b^2 + 4(3b^2+c^2) = 4a^2 + 16b^2 + 4c^2, \tag{16.24}$$

so we can therefore take

$$a_2 = 2, b_2 = 4, c_2 = 2, \tag{16.25}$$

which is a better bound than (16.23).

⁂

We conclude this proof by the lemma

LEMMA 36. — *a_k, b_k, c_k are bounded by constants α, β, γ.*

Proof. — For the last equation (16.23), we get

$$c_k = c_1 3^{\frac{1}{k-1}-\frac{1}{k}+\dots+1-\frac{1}{2}} = 3^{1-\frac{1}{k}}. \tag{16.26}$$

For the other equations, let us pose, for any p,

$$(p-1)(p+2) = \alpha_p 3^{\frac{1}{p-1}}a_{p-1}^p \quad \text{and} \quad (p+2) = \beta_p 3^{\frac{1}{p-1}}b_{p-1}^p, \tag{16.27}$$

therefore, with (16.23),

$$a_k^k = (1+\alpha_k)3^{\frac{1}{k-1}}a_{k-1}^k \quad \text{and} \quad b_k^k = (1+\beta_k)3^{\frac{1}{k-1}}b_{k-1}^k. \tag{16.28}$$

We deduce that

$$\left\{\begin{array}{lcl} a_k &=& a_{k-1}(1+\alpha_k)^{\frac{1}{k}}3^{\frac{1}{k(k-1)}} \\ b_k &=& b_{k-1}(1+\beta_k)^{\frac{1}{k}}3^{\frac{1}{k(k-1)}}, \end{array}\right. \tag{16.29}$$

whence

$$\left\{\begin{array}{lcll} a_k &=& a_2 3^{\frac{1}{2}-\frac{1}{k}} \prod_{p=3}^{k}(1+\alpha_p)^{\frac{1}{p}} & \geq\ a_2\ 3^{\frac{1}{2}-\frac{1}{k}} \\ b_k &=& b_2 3^{\frac{1}{2}-\frac{1}{k}} \prod_{p=3}^{k}(1+\beta_p)^{\frac{1}{p}} & \geq\ b_2\ 3^{\frac{1}{2}-\frac{1}{k}}, \end{array}\right. \tag{16.30}$$

therefore

$$\left\{\begin{array}{lcllll} \alpha_k &=& \dfrac{(k-1)(k+2)}{3^{\frac{1}{k-1}} a_{k-1}^k} & \leq & \dfrac{(k-1)(k+2)}{3^{\frac{1}{k-1}} a_2^k 3^{\frac{k}{2}-\frac{k}{k-1}}} & = \ 3\dfrac{(k-1)(k+2)}{\left(a_2\sqrt{3}\right)^k} \\ \beta_k &=& \dfrac{(k+2)}{3^{\frac{1}{k-1}} b_{k-1}^k} & \leq & \dfrac{(k+2)}{3^{\frac{1}{k-1}} b_2^k 3^{\frac{k}{2}-\frac{k}{k-1}}} & = \ 3\dfrac{(k+2)}{\left(b_2\sqrt{3}\right)^k}. \end{array}\right. \tag{16.31}$$

But

$$a_2\sqrt{3} = 2\sqrt{3} > 1 \quad \text{and} \quad b_2\sqrt{3} = 4\sqrt{3} > 1 \tag{16.32}$$

and, for $k \geq 3$,

$$\alpha_k \leq 3\frac{(k-1)(k+2)}{\left(a_2\sqrt{3}\right)^k} \quad \text{and} \quad \beta_k \leq 3\frac{(k+2)}{\left(b_2\sqrt{3}\right)^k} \tag{16.33}$$

which are the main terms of a convergent series, which proves that

$$\prod_{p\geq 2}(1+\alpha_p)^{\frac{1}{p}} \leq \prod_{p\geq 2}(1+\alpha_p) \quad \text{and} \quad \prod_{p\geq 2}(1+\beta_p)^{\frac{1}{p}} \leq \prod_{p\geq 2}(1+\beta_p) \tag{16.34}$$

are convergent and bounded. Using (16.30), we deduce that a_k and b_k are convergent and bounded by α and β. Using the fact that the L_p are decreasing norms, we find

$$C_k \leq \left(\alpha^k a^k + \beta^k b^k + \gamma^k c^k\right)^{\frac{1}{k}} \leq \alpha a + \beta b + \gamma c. \tag{16.35}$$

We can find explicit upper bounds for α and β, by using the usual upper bounds on entire series, but it is clear that the upper bounds can be improved if we know the respective values of a, b, c.

16.3. Upper Bounds of the Remainders

For $r \in \mathbb{N}$, we denote the remainder at the order r of $F = M_g^{-1} f$ (resp. $F = T_w^{-1} f$) by

$$\mathcal{R}^{(r)} = \sum_{p\geq r+1} F_p. \tag{16.36}$$

We denote, for the Dragt-Finn transformation, as well as for the Deprit transformation,

$$R^* = (\alpha a + \beta b + \gamma c)^{-1}, \tag{16.37}$$

the radius of convergence of F. We demonstrate the following result:

THEOREM 37 ([25]). — *Based on the hypotheses of theorems (32) and (33), F is convergent in ball $D_R = \left\{(x,y) \in \mathbb{C}^{2n}, \|(x,y)\|_2 \le R\right\}$ where $R < R^*$ and*

$$|F|_{D_R} \le \left(1 - \frac{R}{R^*}\right)^{-1} Rd, \left|R^{(r)}\right|_{D_R} \le \left(\frac{R}{R^*}\right)^r \left(1 - \frac{R}{R^*}\right)^{-1} Rd, \tag{16.38}$$

Proof. — From $\|F_n\| \le R^{*1-n}d$, we deduce that $|F_n|_{D_R} \le dR\left(\frac{R}{R^*}\right)^{n-1}$. Hence we deduce that

$$|F|_{D_R} \le dR \sum_{n\ge 0} \left(\tfrac{R}{R^*}\right)^n = dR\frac{1}{1-\frac{R}{R^*}} = \left(1 - \tfrac{R}{R^*}\right)^{-1} Rd \tag{16.39}$$

and, in the same fashion,

$$\left|\mathcal{R}^{(r)}\right|_{D_R} \le \left(\tfrac{R}{R^*}\right)^r \left(1 - \tfrac{R}{R^*}\right)^{-1} Rd. \tag{16.40}$$

THEOREM 38 ([25]). — *Based on the same hypotheses as in theorems (32) and (33), and R^* defined in (16.37), we have, for any $R^* > \rho R > R$, for any $(X,Y) \in D_R$ and $(x,y) = M_g^{-1}(X,Y)$ (resp. $(x,y) = T_w^{-1}(X,Y)$)*

$$|(x,y) - (X,Y)|_{D_R} \le |(X,Y)|_{D_R} \sqrt{2}(\rho-1)^{-1}. \tag{16.41}$$

Proof. — Posing Z, one of the coordinates of (X,Y), we have

$$Z - z = \sum_{p\le 2} Z_p(X,Y) \tag{16.42}$$

where Z_p is a homogeneous polynomial of degree p. Using norm L_1, we have

$$\|Z_p\| \le (\alpha a + \beta b)^{p-1} \tag{16.43}$$

since we can take $c = 0$ and $d = 1$ for $f(X,Y) = (X,Y)$.

In this case, we have

$$\begin{aligned}
|z - Z|_{D_R} &\le \sum_{p\ge 2} \|Z_p\| \, |(X,Y)|^p_{D_R} && (16.44)\\
&\le |(X,Y)|_{D_R} \sum_{p\ge 1} (\alpha a + \beta b)^p \, |(X,Y)|^p_{D_R} && (16.45)\\
&= |(X,Y)|_{D_R} \frac{|(X,Y)|_{D_R} (\alpha a + \beta b)}{1 - (\alpha a + \beta b)\, |(X,Y)|_{D_R}} && (16.46)\\
&\le |(X,Y)|_{D_R} \rho^{-1} \frac{1}{1-\rho^{-1}} && (16.47)\\
&= |(X,Y)|_{D_R} (\rho - 1)^{-1}. && (16.48)
\end{aligned}$$

Since $|(X,Y)| \le \sqrt{2} \sup(|X|, |Y|)$, hence we deduce the stated property.

17. Upper Bounds for the Normal Form

Here, we give an estimation for the radius of convergence of the normal form at a given order. We pose, for any $k \geq 1$,

$$\alpha_k = \min_{0<|k|\leq r} (|k.\omega|) = \min_{\sum_{l=1}^{n}|k_l|\leq r} (|\textstyle\sum_{l=1}^{n} k_l\omega_l|). \tag{17.1}$$

Here, we suppose that $h = \sum_{k\geq 0} h_k$ where h_k is of degree $k+2$ and satisfies, for any k, $\|h_k\| \leq a^k b$.

17.1. Deprit Method

We construct the normal form of h using the Deprit method. We call w, the generating function of T_w, the Deprit transformation which corresponds to the method. We get the following result [25] :

Lemma 39 ([25]). — *For any $k \leq r$,*

$$\|w_k\| \leq \frac{3ab}{\alpha_k}\left[\frac{12k}{5}a\left(1+\frac{b}{\alpha_r}\right)\right]^{k-1}. \tag{17.2}$$

Hence we deduce, by applying theorem 32, that

Theorem 40 ([25]). — *For $w = \sum_{k=1}^{r} w_k$ and $K = T_w^{-1}h$, the normal form at the order r, we have*

$$\|K_n\| \leq A^n d, \quad 1 \leq n \quad \textit{where} \quad A = a\left[\left(9+\tfrac{32}{5}r\right)\frac{b}{\alpha_r} + \left(1+\tfrac{32}{5}r\right)\right]. \tag{17.3}$$

Posing $R_r^* = A^{-1}$, we get, for the normal form $T_w^{-1}h = \mathcal{Z}^{(r)} + \mathcal{R}^{(r)}$ at the order r

- *$T_w^{-1}h$ is convergent in any ball D_R with $R < R^*$,*
- *the general term of the remainder is bounded by $\left\|\mathcal{R}_n^{(r)}\right\| \leq \left(\frac{R}{R_r^*}\right)^n d$ and*

$$\left|\mathcal{R}^{(r)}\right|_{D_R} \leq bR\left(\frac{R}{R^*}\right)^r\left(1-\frac{R}{R^*}\right)^{-1}. \tag{17.4}$$

17.2. Dragt-Finn Method

We construct $K^{(0)} = h$, $K^{(k)} = e^{L_{g_k}}K^{(k-1)}$ for $k \geq 0$. The generator g_k verifies $L_{h_0}g_k + K_k^{(k)} = K_k^{(k-1)}$.

Lemma 41. — *For any $k, n \in N$, we have*

$$\left\|K_n^{(k)}\right\| \leq a_k^n b \quad \textit{and} \quad \|g_k\| \leq \frac{a_{k-1}^k b}{\alpha_k}, \tag{17.5}$$

where

$$a_k = 2a\prod_{p=1}^{k}\left(1+\tfrac{p(p+2)b}{\alpha_p}\right)^{\frac{1}{p}}. \tag{17.6}$$

Proof. — Let us define a_k, b_k so that we have, for any k and any n,

$$\left\|K_n^{(k)}\right\| \leq a_k^n b_k. \tag{17.7}$$

From $L_{h_0}g_k + K_k^{(k)} = K_k^{(k-1)}$, we first get, by projection (cf. [25]), and since α_k (cf. 17.1) is the smallest of the eigenvalues of L_{h_0} on E_k,

$$\left\|K_k^{(k)}\right\| \leq \left\|K_k^{(k-1)}\right\| \quad \text{and} \quad \|g_k\| \leq \tfrac{1}{\alpha_k}\left\|K_k^{(k-1)}\right\|. \tag{17.8}$$

For $n \in \mathbb{N}$, we have

$$\begin{aligned}
\left\|K_n^{(k)}\right\| &\leq \sum_{kp+q=n} \frac{1}{p!}\left\|L_{g_k}^p K_q^{(k-1)}\right\| \\
&\leq \sum_{kp+q=n} \frac{(q+2)(q+k+2)\cdots(q+(p-1)k+2)}{p!}(k+2)^p \|g_k\|^p \left\|K_q^{(k-1)}\right\| \\
&\leq \sum_{kp+q=n} \frac{(\frac{n+2}{k}-1)\cdots(\frac{n+2}{k}-p)}{p!}\left(k(k+2)\|g_k\|\right)^p \left\|K_q^{(k-1)}\right\| \\
&\leq a_{k-1}^n b_{k-1} \sum_{p=0}^{[\frac{n}{k}]} (\tfrac{n+2}{k}-1)_p \left(\tfrac{k(k+2)b_{k-1}}{\alpha_k}\right)^p.
\end{aligned} \tag{17.9}$$

We must now distinguish between several cases.

- If $k = 1$, we have

$$\begin{aligned}
\left\|K_n^{(1)}\right\| &\leq b\sum_{p=0}^{n} \tbinom{n+1}{p}\left(\tfrac{3ab}{\alpha_1}\right)^p a^{n-p} \\
&= a^n b\left[\left(1+\tfrac{3b}{\alpha_1}\right)^{n+1} - \left(\tfrac{3b}{\alpha_1}\right)^{n+1}\right] \\
&\leq b_1\left(a(n+1)^{\frac{1}{n}}(1+\tfrac{3b}{\alpha_1})\right)^n,
\end{aligned} \tag{17.10}$$

and we can take

$$b_1 = b, a_1 = a\max_n\left((n+1)^{\frac{1}{n}}\right)\left(1+\tfrac{3b}{\alpha_1}\right) = 2a\left(1+\tfrac{3b}{\alpha_1}\right). \tag{17.11}$$

- If $k \geq 2$, we have $\frac{n+2}{k} - 1 \leq \frac{n}{k}$ and

$$\begin{aligned}
\left\|K_n^{(k)}\right\| &\leq a_{k-1}^n b_{k-1}\sum_{p=0}^{[\frac{n}{k}]} (\tfrac{n}{k})_p \left(\tfrac{k(k+2)b_{k-1}}{\alpha_k}\right)^p \\
&\leq a_{k-1}^n b_{k-1}\left(1+\tfrac{k(k+2)a_{k-1}^k b_{k-1}}{\alpha_k}\right)^{\frac{n}{k}}.
\end{aligned} \tag{17.12}$$

This inequality is obtained by writing Taylor's formula with integral remainder

$$\begin{aligned}
(1+x)^{\frac{n}{k}} &= \sum_{p=0}^{[\frac{n}{k}]} (\tfrac{n}{k})_p x^p + \int_0^1 \frac{(1-t)^{[\frac{n}{k}]}}{[\frac{n}{k}]!}\overbrace{(\tfrac{n}{k})_{[\frac{n}{k}]+1}}^{\geq 0}(1+t)^{\frac{n}{k}-[\frac{n}{k}]-1}dt && (17.13)\\
&\geq \sum_{p=0}^{[\frac{n}{k}]} (\tfrac{n}{k})_p x^p. && (17.14)
\end{aligned}$$

We can now take $b_k = b_{k-1}$ and $a_k = a_{k-1}\left(1 + \frac{k(k+2)b_{k-1}}{\alpha_k}\right)^{\frac{1}{k}}$.

⁂

Hence we deduce a result on the convergence of the normal form.

THEOREM 42. — *If $h = \sum_{k\geq 0} h_k$ satisfies, for any $k \geq 0$, $\|h_k\| \leq a^k b$, we get, with the notations of lemma 41, for the normal form at the order r:*

$$K^{(r)} = \mathcal{Z}^{(r)} + \mathcal{R}^{(r)} = e^{L_{g_r}} \cdots e^{L_{g_1}} h = {M^{(r)}}^{-1} h, \tag{17.15}$$

— *$K^{(r)}$ is convergent in any ball D_R with*

$$R < R^*_r = 2a\left[\prod_{p=1}^{r}\left(1 + \frac{p(p+2)b}{\alpha_p}\right)^{\frac{1}{p}}\right]^{-1}, \tag{17.16}$$

— *the general term of the remainder is bounded by $\left\|\mathcal{R}^{(r)}_n\right\| \leq R^{*n}_r d$, and we have*

$$\left|\mathcal{R}^{(r)}_n\right|_{D_R} \leq bR\left(\frac{R}{R^*}\right)^r\left(1 - \frac{R}{R^*}\right)^{-1}. \tag{17.17}$$

The proof follows from lemma 41, as well as classical upper bounds on numerical series.

⁂

THEOREM 43. — *Based on the hypotheses of the previous theorem, given $f = \sum_{n\geq 1} f_n$ where f_n is a polynomial of degree n and satisfies $\|f_n\| \leq c^{n-1}d$. Posing $\rho = \max(\frac{c}{a}, 1)$, we have*

— *${M^{(r)}}^{-1} f$ is convergent in the disc $D_R = \left\{(x,y) \in \mathbb{C}^{2n}, \|(x,y)\| \leq R\right\}$ where*

$$R < R^*_r = 2a\left[\prod_{p=1}^{r}\left(\rho^p + \frac{p(p+2)b}{\alpha_p}\right)^{\frac{1}{p}}\right]^{-1}, \tag{17.18}$$

— *$F^{(r)} = {M^{(r)}}^{-1} f$ is convergent in D_R and, for any $n \in \mathbb{N}$, $\left\|F^{(r)}_n\right\| \leq c^{n-1}_r d$, where*

$$c_r = 2a\prod_{p=1}^{r}\left(\rho^p + \frac{p(p+2)b}{\alpha_p},\right)^{\frac{1}{p}}, \tag{17.19}$$

—

$$\left|F^{(r)}\right|_{D_R} \leq \left(1 - \frac{R}{R^*}\right)^{-1} Rd, \quad \left|\mathcal{R}^{(r)}\right|_{D_R} \leq \left(\frac{R}{R^*}\right)^r\left(1 - \frac{R}{R^*}\right)^{-1} Rd. \tag{17.20}$$

Proof. — As for the previous theorem, let us define

$$F^{(0)} = f, F^{(k)} = e^{L_{g_k}} F^{(k-1)}. \tag{17.21}$$

In virtue of lemma 34, we can pose, for any $n \in \mathbb{N}$, $\left\|F_n^{(1)}\right\| \leq c_1^{n-1} d$ where

$$c_1 \leq 3\|g_1\| + \rho a = a\left(\rho + \tfrac{3b}{\alpha_1}\right). \tag{17.22}$$

Let us now suppose that $\left\|F_n^{(k-1)}\right\| \leq c_{k-1}^n d$ with

$$c_{k-1} \leq 2a \prod_{p=1}^{k-1} \left(\rho^k + \tfrac{p(p+2)b}{\alpha_p}\right)^{\frac{1}{p}}. \tag{17.23}$$

Hence we deduce that

$$
\begin{aligned}
c_k^k &\leq \left(k(k+2)\|g_k\| + c_{k-1}^k\right) && (17.24)\\
&\leq 2^k a^k \left(\tfrac{k(k+2)b}{\alpha_k}\left[\prod_{p=1}^{k-1}\left(1+\tfrac{p(p+2)b}{\alpha_p}\right)^{\frac{1}{p}}\right]^k + \rho^k \left[\prod_{p=1}^{k-1}\left(\rho^p+\tfrac{p(p+2)b}{\alpha_p}\right)^{\frac{1}{p}}\right]^k\right) && (17.25)\\
&\leq 2^k a^k \left[\prod_{p=1}^{k-1}\left(\rho^p+\tfrac{p(p+2)b}{\alpha_p}\right)^{\frac{1}{p}}\right]^k \left(\rho^k + \tfrac{k(k+2)b}{\alpha_k}\right) && (17.26)\\
&= 2^k a^k \left[\prod_{p=1}^{k}\left(\rho^p+\tfrac{p(p+2)b}{\alpha_p}\right)^{\frac{1}{p}}\right]^k, && (17.27)
\end{aligned}
$$

and recursively that, for any $k, n \geq 1$, we have

$$\left\|F_n^{(k)}\right\| \leq c_k^{n-1} d \quad \text{where} \quad c_k = 2a \prod_{p=1}^{k}\left(\rho^p + \tfrac{p(p+2)b}{\alpha_p}\right)^{\frac{1}{p}}. \tag{17.28}$$

The other upper bounds follow from elementary manipulations on the numerical series.

17.3. Comparisons

Since the α_r's tend toward 0, it is most unlikely that the normal forms will converge. Let us take for example the classical diophantian approximation for α_r,

$$\alpha_r = \frac{k}{r^{m-1}} \tag{17.29}$$

where k and $m-1$ are positive constant. From (17.2), we deduce, for the Deprit transformation, that

$$R_r^{*-1} = 9 + \tfrac{32}{5} r \tfrac{b}{k} r^{m-1} + 1 + \tfrac{32}{5} r \simeq \lambda_1 r^m,$$

whereas, for the Dragt-Finn transformation, upper bound (17.18) yields

$$
\begin{aligned}
R_r^{*-1} &= 2a \textstyle\prod_{p=1}^{r}\left(1 + kp(p+2)p^{m-1}\right)^{1/p}\\
&\simeq \lambda_2 e^{\sum_{p=1}^{r} \frac{m+1}{p}\log p}\\
&\simeq \lambda_2 e^{\frac{m+1}{2}\log^2 r}\\
&\simeq \lambda_2 r^{\frac{m+1}{2}\log r}.
\end{aligned} \tag{17.30}
$$

Consequently, the upper bound for the Deprit transformation proposed by Giorgilli is better inasmuch as

$$\log r \geq 2\frac{m}{m+1}. \tag{17.31}$$

18. Estimation of the Diffusion

Using an application of the mean value theorem, we search for a ball centered at the point of equilibrium, such that, for the longest possible time, we may predict that the trajectories originating in this ball, do not get too far away from it. The bounds given here are not the best possible, but they give some indication of what could be expected. In fact, it appears necessary to adapt an upper bound *ad hoc* to every type of problem encountered.

The technique (cf. [25]) primarily consists of bounding the derivative of the actions, which are the distances from the origin. At this point, we search for an optimal perturbation order r_{opt} for the truncation.

Given $h = \sum_{n\geq 0} h_n$ with $h_0 = \frac{1}{2}\sum_{l=1}^{n} \frac{\omega_l}{2}(p_l^2 + q_l^2)$. We also suppose that the ω_l are rationally independent. Let us denote $T^{(r)}$, the Dragt-Finn or Deprit transformation which enabled us to construct the normal form at the order r, $K^{(r)} = \mathcal{Z}^{(r)} + \mathcal{R}^{(r)}$, as well as $(Q^{(r)}, P^{(r)}) = T^{(r)}(q,p)$. Let us now pose $I_l^{(r)} = \dfrac{1}{2}({P^{(r)}}^2 + {Q^{(r)}}^2)$, we get the following theorem where R_r^* denotes the radius of convergence of $\mathcal{R}^{(r)}$:

Theorem 44 ([25]). — *For $R < R_r^*$, we have the upper bound*

$$\left|\dot{I}^{(r)}\right|_{D_R} \leq \tfrac{d}{c} R\left(\frac{R}{R_r^*}\right)\left(R + 1 - r\frac{R}{R_r^*}\right)\left(1 - \frac{R}{R_r^*}\right)^{-2}. \tag{18.1}$$

The upper bound is obtained by using the previous results, and noticing that

$$\dot{I}_l^{(r)} = \left\{I_l^{(r)}, \mathcal{R}^{(r)}\right\}. \tag{18.2}$$

***

The following step consists of finding an optimal order r_{opt} so that the upper bound of $\dot{I}^{(r)}$ is minimal. We then deduce that, for a time $t < T$, determined by the previously introduced constants, we have, for a given $\rho > 1$:

Any particle located at $t = 0$ in the ball D_R, is located in the ball $D_{\rho R}$ as long as $t < T$, –since T now depends on the bounds chosen for h, R, the α_i's, ρ and, especially with this method, the transformation used, since T depends on the upper bounds on the remainder.

Since the estimation for the Dragt-Finn method is *a priori* not as good as that obtained by [25] or [52] for the Deprit method, we do not explain the upper bounds obtained in detail at this point.

These upper bounds only depend on the upper bounds obtained theoretically, and the transformation is only used to compute intermediate bounds on the remainder of the normal form.

We may now ask ourselves what purpose effectively computing the normal forms or the transformations might serve. It so happens that, in an article subsequent to [25], the results obtained by using information based on the computation of the first terms of the Lie transformation, –whether a Deprit or Dragt-Finn transformation– are radically better.

The results rely essentially on the existence of approximate invariants (12.16) of the hamiltonian system, whose construction is insured by the Deprit or Dragt-Finn method.

Therefore, let us presume given a Hamiltonian h. The existence, at any order r, of a function $\xi_l^{(r)} = \sum_{p\geq 0} \xi_{l,p}^{(r)}$ which satisfies, for any $1 \leq l \leq n$,

$$\left\{\xi_l^{(r)}, h\right\} = \; = \sum_{n\geq r+1} \varepsilon^n \sum_{p+q=n}{}' \left\{\xi_{l,p}^{(r)}, h_q\right\}, \tag{18.3}$$

is insured by (12.16). Let us consider then

$$\Phi_l^{(r)} = \sum_{p=0}^{r} \xi_l^{(r)} = \sum_{p=0}^{r} \Phi_l^{(r)}. \tag{18.4}$$

The first terms verify, for $0 \leq n \leq r$,

$$\Phi_{l,0}^{(r)} = I_l, \sum_{p=0}^{n} \left\{\Phi_{l,p}^{(r)}, h_{n-p}\right\} = \left\{\Phi_{l,n}^{(r)}, h_0\right\} + \sum_{p=1}^{n} \left\{\xi_{l,p}^{(r)}, h_{n-p}\right\} = 0. \tag{18.5}$$

So we have

$$\dot{\Phi}_l^{(r)} = \left\{\Phi_l^{(r)}, h\right\} \; = \sum_{n\geq 0} \sum_{p+q=n} \left\{\Phi_{l,p}^{(r)}, h_q\right\} = \sum_{p=0}^{r} \sum_{n\geq r} \varepsilon^n \left\{\Phi_{l,p}^{(r)}, h_{n-p}\right\}. \tag{18.6}$$

The idea, in [8], is to use (18.6), as well as the first terms of $\Phi_l^{(r)}$, to bound I_l in time. In [8], we write

$$|I_l(t) - I_l(0)| \leq \left|I_l(t) - \Phi_l^{(r)}(t)\right| + \left|I_l(0) - \Phi_l^{(r)}(0)\right| + \left|\Phi_l^{(r)}(t) - \Phi_l^{(r)}(0)\right|. \tag{18.7}$$

The two first terms are of the form $\left|\sum_{p=1}^{r} \Phi_{l,p}^{(r)}\right|$, for which we have some estimates, either because they have been computed, or we have computed in part $\Phi_{l,0}^{(r)} \cdots \Phi_{l,\tilde{r}}^{(r)}$, and we have secured estimates of $\|h_k\|$. In this case, we deduce estimates for the $\Phi_{l,p}^{(r)}$, by using (18.5). We increase the last term, using the mean value theorem, as well as an upper bound deduced from (18.6).

The proof is rather technical, but it appears to yield far better bounds than the method which consists of giving an estimation of the remainder. In [8], the estimation of the stability region is compared to the results achieved by Simò ([52]), and it appears that the latter technique yields better results. This shows that the knowledge of the first terms of the normal form and of the related transformation can contribute major improvements.

18.1. Example

Here, we recall an application of the previous results to the study of the Hamiltonian of the restricted three-body problem, in the vicinity of the Lagrange (cf. [52, 25]) point $\mathcal{L}_4$.

We consider the following system: 2 bodies of great relative masses μ and $1-\mu$, in uniform rotation around their center of mass. The distance between these bodies is one unit, the rotation period is 2π. We study a third body subjected to the attraction of these two primary bodies. Its mass m, is considered of negligible influence on the motion of the two primary bodies. We place ourselves in the rotating frame of reference, that is to say where the positions of the body of mass μ and $1-\mu$ are respectively $(1-\mu, 0, 0)$ and $(\mu, 0, 0)$. The third body has the coordinates:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(t) & \sin(t) & 0 \\ -\sin(t) & \cos(t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{18.8}$$

The energy of the system T is given by

$$\begin{aligned} T &= \tfrac{1}{2}(\dot{X}^2+\dot{Y}^2+\dot{Z}^2) = \tfrac{1}{2}\left((\tfrac{d}{dt}(x\cos(t)-y\sin(t)))^2+(\tfrac{d}{dt}(x\sin(t)+y\cos(t)))^2+\dot{z}^2\right) \\ &= \tfrac{1}{2}\left(((\dot{x}-y)\cos(t)-(\dot{y}+x)\sin(t))^2+((\dot{x}-y)\sin(t)+(\dot{y}+x)\cos(t))^2+\dot{z}^2\right) \\ &= \tfrac{1}{2}(((\dot{x}-y)^2+(\dot{y}-x)^2+\dot{z}^2), \end{aligned} \tag{18.9}$$

and the potential energy by

$$U = -\tfrac{\mu}{d_1} - \tfrac{1-\mu}{d_2}, \tag{18.10}$$

where d_1 and d_2 are the distances from the primary bodies:

$$d_1^2 = (x-1+\mu)^2+y^2+z^2, \quad d_2^2 = (x-\mu)^2+y^2+z^2. \tag{18.11}$$

The hamiltonian system is given by the canonical variables

$$p_x = (\dot{x}-y), p_y = (\dot{y}+x), p_z = \dot{z}, \tag{18.12}$$

and the Hamiltonian

$$\begin{aligned} h &= p_x\dot{x}+p_y\dot{y}+p_z\dot{z}-T+U && (18.13)\\ &= p_x(p_x+y)+p_y(p_y-x)-\tfrac{1}{2}(p_x^2+p_y^2+p_z^2)+U && (18.14)\\ &= \tfrac{1}{2}(p_x^2+p_y^2+p_z^2)+p_xy-p_yx+U. && (18.15) \end{aligned}$$

The points of equilibrium of the system verify

$$\dot{x} = p_x+y = \frac{\partial h}{\partial x} = 0, \ \dot{y} = p_y - x = \frac{\partial h}{\partial y}, \ \dot{z} = p_z = \frac{\partial h}{\partial z} = 0, \tag{18.16}$$

which yields, as its general solution

$$z = p_z = 0, \ p_x + y = -p_y + \frac{\partial U}{\partial x} = 0, \ p_y - x = p_x + \frac{\partial U}{\partial y} = 0, \tag{18.17}$$

given,

$$z = p_z = 0, \ x + \frac{\partial U}{\partial x} = y + \frac{\partial U}{\partial y} = 0, \ p_x + p_y = -y + \frac{\partial U}{\partial x}, \ p_x - p_y = -x - \frac{\partial U}{\partial y}. \tag{18.18}$$

Let us call

$$V = \tfrac{1}{2}(x^2+y^2+z^2)+U = \tfrac{1}{2}(\mu d_1^2+(1-\mu)d_2^2-\mu(1-\mu))+U. \tag{18.19}$$

If $y=0$, we find three solutions that are the three Lagrange points $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$, aligned on the straight line which joins the primary bodies. If $y \neq 0$ then $(x,y,z) \mapsto (d_1, d_2, z)$ is a diffeomorphism, and $V = V(d_1, d_2, z)$ is differentiable. $DV = 0$ is written as

$$\mu d_1 - \frac{\mu}{d_1^2} = 0,\ (1-\mu)d_2 - \frac{1-\mu}{d_2^2} = 0,\ z=0, \tag{18.20}$$

that is to say $d_1 = d_2 = 1$. We find the two Lagrange points $\mathcal{L}_4, \mathcal{L}_5$ which form an equilateral triangle with the primary bodies.

After a change of origin at the point $\mathcal{L}_4$, we find the new Hamiltonian

$$h = \tfrac{1}{2}(p_x^2+p_y^2+p_z^2)+yp_x-xp_y-(\mu-\tfrac{1}{2})-y\tfrac{\sqrt{3}}{2}-\tfrac{\mu}{r_1}-\tfrac{1-\mu}{r_2}, \tag{18.21}$$

where

$$r_1^2 = 1-x+\sqrt{3}y+x^2+y^2+z^2, r_2^2 = 1+x+\sqrt{3}y+x^2+y^2+z^2. \tag{18.22}$$

Posing $a = -3(1-2\mu)\frac{\sqrt{3}}{4}$, we can write $h = \sum_{n\geq 0} h_n$ with

$$\begin{aligned} h_0 &= \tfrac{1}{2}(p_x^2+p_y^2+p_z^2)+yp_x-xp_y+\tfrac{1}{8}x^2-\tfrac{5}{8}y^2-axy+z^2 && (18.23)\\ h_n &= \mu r^n P_n(\tfrac{-x-\sqrt{3}y}{2r})+(1-\mu)r^n P_n(\tfrac{x-\sqrt{3}y}{2r}),\quad n\geq 1, && (18.24)\end{aligned}$$

where the P_k are the Legendre polynomials of degree k.

The hamiltonian operator L_{h_0} admits, as its characteristic polynomial,

$$(\lambda^4+\lambda^2+\tfrac{27}{16}-a^2)(\lambda^2+1)=0. \tag{18.25}$$

A pair of eigenvalues corresponds to $\pm\mathrm{i}$ for the plane $(0, z, p_z)$, and the others are multiples of i, and distinct whenever $\mu(1-\mu) < \frac{1}{27}$, that is to say whenever $0 < \mu < \frac{1}{2}(1-\frac{\sqrt{69}}{9})$. In this case, $\mathcal{L}_4$ is an elliptical point of equilibrium of the system.

This Hamiltonian may then be handled using perturbation methods possibly diagonalizing L_{h_0} (the ω_l's are of opposite signs) (cf. [25, 52, 41]).

If we keep masses μ and $1-\mu$ as parameters, the computation of the normal form proves too large to be carried out at high orders using the algorithms which I implemented.

Nevertheless, the normal forms were computed for low orders (cf. [41]). At the order 2, in the action-angle coordinates:

$$\begin{aligned} h_0 &= \omega_1 I_1 - \omega_2 I_2 && (18.26)\\ h_2 &= \tfrac{1}{2}(AI_1^2+2BI_1I_2+CI_2^2) && (18.27)\end{aligned}$$

where

$$\begin{aligned} A &= \tfrac{1}{72}\omega_2^2\frac{(81-696\omega_1^2+124\omega_1^4)}{(1-2\omega_1^2)(1-5\omega_1^2)},\ C(\omega_1,\omega_2)=A(\omega_2,\omega_1), && (18.28)\\ B &= -\tfrac{1}{6}\frac{\omega_1\omega_2(43+64\omega_1^2\omega_2^2)}{(1-2\omega_1^2)(1-5\omega_1^2)(1-2\omega_2^2)(1-5\omega_2^2)}, && (18.29)\end{aligned}$$

However, the study was completed for numerical values corresponding to the Sun and Jupiter as the primary bodies in [8, 25, 52]. The normal forms were computed to the degree 30, which made it possible to define a non-negligible stability zone for a period approximating the age of the universe (10^{10} rotation periods of Jupiter around the sun). In [25], using theoretical bounds based on the estimations of the Hamiltonian h, we find a stability zone of a few kilometers. The particles wander in a region 1.005 times the size of the original stability zone during a time comparable to the age of the universe. In [52] or [8], we find a stability zone of the order of a few thousand kilometers, for the same period, by computing effective bounds on the normal form computed at a high order.

$\circ{}^{\circ}\circ$

The estimations given in this section for the Dragt-Finn method, do not, in and of themselves, improve the stability results that could be deduced, by mere estimation of the truncated invariants, by using the estimations carried out on the Lie transformation by Giorgilli *et al.* . Undoubtedly, the use of combinatorial properties on partitions would enable us to improve the estimations for the Dragt-Finn transformation, but who is to say whether they would be better than those taken directly from invariants ?

Knowing the first terms of the invariants, as well as using recursive formulas relating to the successive terms, enables us to significantly improve the asymptotic estimations which provide information concerning the system's stability. Effective computation of normal forms and truncated invariants, is one of the essential tools which provide stability information, and the Dragt-Finn method is definitely an efficient means of achieving this goal.

Chapter V
Symplectic Integrators

We frequently have to compute the time evolution of hamiltonian systems. In celestial mechanics, we use multi-step integration methods at fairly high orders. The disadvantage of such methods, is that the error on the positions increases quadratically in time.

Symplectic integrators are numerical methods to compute, on a very long term basis, the time evolution of a hamiltonian system. At every step, the transformation is symplectic, which holds the advantage of preserving an energy approximating the system's energy[3]. On the other hand, a disadvantage is that they are, for the most part, considerably more costly than multi-step integrators of the same order.

Symplectic integrators can be considered like mappings of perturbed hamiltonian flows with respect to the original Hamiltonian. So these integrators can be represented either as Deprit transformations, or as exponentials, or as Dragt-Finn transformations.

The construction which we propose here relies mainly on the possibility of generating identities of exponential type, by using the algorithms of the first section.

This brings us back to the general construction ideas, inspired by Forest & Ruth's [20] and Yoshida's [60, 61] contributions. We represent symplectic integrators either by Lie transformations, which holds the advantage of not requiring the computation of exponential identities, or by exponentials, which may have symmetry properties. First, we show the strict equivalence of these representations, that is to say the sets of solutions obtained coincide from an algebraic point of view. Next, we give an exhaustive list of the symplectic integrators for low orders. The low orders are obtained by searching for the solutions to a system of polynomial equations, using algebraic tools such as standard bases. Some of the solutions were thus far unknown. We show elsewhere that the method proposed by Yoshida [61] provides all possible solutions.

In the case where the Hamiltonian is written as the sum of a kinetic energy and of a potential energy, we decrease the number of terms necessary to the manufacture of such integrators. In the case where the Hamiltonian is the sum of three terms, we give an exhaustive list of order 2 and 4 integrators.

The purpose of this section is not to compare the respective merits of the various integrators used to implement very long term integrations, but to supply the means to construct symplectic integrators at the highest possible orders.

[3]This statement is rigorously exact only for linear problems. Therefore, in cases for which the Hamiltonian is preserved, one must gauge how much artificial chaos is induced by the symplectic integrator. N. T.

19. General Methods of Construction

Let us consider a hamiltonian system, given by $h = A+B$, in which we seek to compute the time evolution $S_h(t)$. $S_h(t)$ is the solution to the differential equation

$$\frac{\partial S_h}{\partial t} = -S_h(t)L_h. \tag{19.1}$$

If A and B are autonomous, we have

$$S_h(t) = e^{-tL_h} = e^{-tL_A - tL_B}, \tag{19.2}$$

else, $S_h(t)$ is a Lie transformation

$$S_h(t) = T_{\bar{h}(t)}, \quad \frac{d\bar{h}(t)}{dt} = h, \tag{19.3}$$

which can be written as an exponential (cf. [42]).

Whenever S_h cannot be computed easily, it might be advisable to approximate it by a product of $S_A(t)$ and $S_B(t)$, when these are simpler.

If we consider Euler's classical method of integration, we get, for a step τ,

$$p(t+\tau) = p(t) - \tau\frac{\partial H}{\partial q}, \quad q(t+\tau) = q(t) + \tau\frac{\partial H}{\partial p}. \tag{19.4}$$

This transformation is not usually symplectic, as is, for example, the case for $h = \frac{\omega}{2}(p^2+q^2)$ where (19.4) becomes

$$p(t+\tau) = p(t) - \omega\tau q(t), \quad q(t+\tau) = q(t) + \omega\tau p(t), \tag{19.5}$$

in which the determinant is equal to $1+\omega^2\tau^2$. At every step, the value of the total energy of the Hamiltonian h, is multiplied by $1+\omega^2\tau^2$.

On the other hand, the product of the two symplectic transformations

$$\left\{\begin{array}{lll} p & \mapsto & p - \tau\dfrac{\partial H}{\partial q} \\ q & \mapsto & q \end{array}\right. \quad \text{and} \quad \left\{\begin{array}{lll} p & \mapsto & p \\ q & \mapsto & q + \tau\dfrac{\partial H}{\partial p} \end{array}\right., \tag{19.6}$$

is a symplectic transformation. For $h = \frac{\omega}{2}(p^2+q^2)$ we get

$$\begin{pmatrix} q(t+\tau) \\ p(t+\tau) \end{pmatrix} = \begin{pmatrix} 1-\omega^2\tau^2 & \omega\tau \\ -\omega\tau & 1 \end{pmatrix} \begin{pmatrix} q(t) \\ p(t) \end{pmatrix}. \tag{19.7}$$

In this case, the transformation obtained is the product

$$T(\tau) = e^{-\tau L_K} = e^{-\frac{\omega}{2}\tau L_{q^2}} e^{-\frac{\omega}{2}\tau L_{p^2}} \tag{19.8}$$

where, according to the Baker-Campbell-Hausdorff formula,

$$K = \tfrac{\omega}{2}(p^2+q^2) - \tau\tfrac{\omega}{2}pq + \tau^2\tfrac{\omega}{12}(p^2+q^2) - \tau^3\tfrac{\omega}{12}pq + \cdots. \tag{19.9}$$

We note that K is invariant by (19.8), but is not the Hamiltonian which corresponds to the transformation (19.8). The corresponding Hamiltonian satisfies

$$\frac{d}{dt}T(t) = -L_{W(t)}T(t), \quad \text{i.e.} \quad L_{W(t)} = -T^{-1}(t)\frac{d}{dt}T(t). \tag{19.10}$$

Using (6.46), we get

$$\begin{aligned} W(t) &= \tfrac{\omega}{2}(e^{-\frac{\omega}{2}tL_{q^2}}p^2+q^2) \\ &= \tfrac{\omega}{2}(q^2+p^2-\tfrac{\omega t}{2}\{q^2,p^2\}+\tfrac{\omega^2t^2}{8}\{q^2,\left\{q^2,p^2\right\}\}) \\ &= \tfrac{\omega}{2}(p^2+q^2)+2t\omega pq+t^2\tfrac{\omega^2}{2}q^2. \end{aligned} \tag{19.11}$$

$\star^{\star}\star$

The typical case where the symplectic integrators demonstrate their efficiency is the case in which

$$h = T(p) + V(q), \tag{19.12}$$

that is to say when, for example, h is the sum of uncoupled kinetic and potential energies. In this case, since

$$\{T(p),p\}=0,\ \{T(p),q\}=-\frac{\partial T}{\partial p} \quad\text{and}\quad \{V(q),q\}=0,\ \{V(q),p\}=\frac{\partial V}{\partial p} \tag{19.13}$$

we have

$$\{T,\{T,q\}\}=0 \quad\text{and}\quad \{V,\{V,p\}\}=0. \tag{19.14}$$

When we compute the time evolution of the state variables, we get the extremely simple form:

$$S_T(t)(p,q)=e^{-tT(p)}(p,q)=\sum_{n\geq 0}\frac{(-tL_T)^n}{n!}(p,q)=(p,q+t\frac{\partial V}{\partial p}), \tag{19.15}$$

$$S_V(t)(p,q)=e^{-tV(q)}(p,q)=\sum_{n\geq 0}\frac{(-tL_V)^n}{n!}(p,q)=(p-t\frac{\partial T}{\partial q},q). \tag{19.16}$$

The main idea is to construct

$$S^{(n)}(t)=S_T(c_1t)S_V(d_1t)\cdots S_T(c_nt)S_V(d_nt), \tag{19.17}$$

in such a way that, for a given integer k,

$$S^{(n)}(t)=S_h(t)+o(t^k). \tag{19.18}$$

In this case, we get $S_h(\tau)$ as a sequence of transformations

$$q^0=q,\ p^0=p,\ \ldots,\ p^{i+1}=p^i+\tau t d_i\frac{\partial T}{\partial p}(q^{i+1}),\ q^{i+1}=q^i+\tau c_i\frac{\partial T}{\partial p}(p^{i+1}), \tag{19.19}$$

and $p_\tau=p^n,\ q_\tau=q^n$ appear after n evaluations of $\dfrac{\partial T}{\partial p}$ and $\dfrac{\partial V}{\partial q}$.

19.1. Direct Determination of Symplectic Integrators

Given $h = A+B$ and $S_A(t)=e^{-tL_A}, S_B(t)=e^{-tL_B}$. For a preset n and k, we search for a minimal set $c_1,\ldots,c_n,d_1,\ldots d_n$, such that we have

$$S^{(n)}(t)=S_A(c_1t)S_B(d_1t)\cdots S_A(c_nt)S_B(d_nt)=S_h(t)+o(t^k). \tag{19.20}$$

Let us denote $R = \mathbb{Q}[c_1, \ldots, c_n, d_1, \ldots, d_n]$, $\tilde{L}_p(\{A, B\})$, the submodule of $\tilde{L}(\{A, B\}, R)$ generated by the elements of weight p, where $|A|_* = |B|_* = 1$.

We also denote by $\tilde{L}_p\{z, A, B\}$, the submodule of $\tilde{L}(\{z, A, B\}, R)$, generated by the elements of weight p, which contain only once the letter z, where $|z|_* = 0, |A|_* = |B|_* = 1$.

Remark. — $\tilde{L}_p\{z, A, B\}$ is of dimension 2^p. Indeed, if $w \in \{z, A, B\}^*_{p+1}$, with $z < A < B$, then if z appears once, and only once, in w, w is a Lyndon word if and only if $w = zw_1$ where $w_1 \in \{A, B\}^*_p$, that is to say there are exactly 2^p Lyndon words of length $p+1$ containing once, and only once, the letter z.

***

In order to solve (19.20), we must first express exponential identities, by placing ourselves in a free Lie algebra, and by using the methods from the first section. First, we show the algebraic equality of the solutions obtained, by representing the integrators in various ways. We have the following property:

LEMMA 45. — *Let us consider the three following problems*

1. *Having represented $S^{(n)}(t)$ as $\exp(-\sum_{p\geq 1} t^p L_{K_p})$, solve in $L(\{A, B\})$: $K_1 = A + B, K_2 = \cdots = K_k = 0$,*
2. *Having represented $S^{(n)}(t)$ as $\exp(-tL_{G_1}) \cdots \exp(t^p L_{G_p}) \cdots$, solve in $L(\{A, B\})$: $G_1 = A + B, G_2 = \cdots = G_k = 0$,*
3. *Having represented $S^{(n)}(t)$ as $T_{\sum_{p\geq 1} t^p W_p}$, solve in $L(\{A, B\})$: $W_1 = A + B, W_2 = \cdots = W_k = 0$.*

The varieties, solutions to the 3 problems, are defined by the same ideal.

Proof. — Denoting $x_{p,1}, \ldots, x_{p,l_p}$ the Lyndon basis of $\mathcal{L}_p(\{A, B\})$, we can write

$$K_p = \sum_{i=1}^{l_p} K_{p,i} x_{p,i}, G_p = \sum_{i=1}^{l_p} G_{p,i} x_{p,i}, W_p = \sum_{i=1}^{l_p} W_{p,i} x_{p,i}. \tag{19.21}$$

For every one of these methods, the solutions are the zeros of the ideals:

$$\begin{aligned} \mathfrak{I}_K^{(k)} &= (K_{1,1} - 1, K_{1,2} - 1, K_{2,1}, K_{3,1}, \ldots, K_{k,1}, \ldots), \\ \mathfrak{I}_G^{(k)} &= (G_{1,1} - 1, G_{1,2} - 1, G_{2,1}, G_{3,1}, \ldots, G_{k,1}, \ldots), \\ \mathfrak{I}_W^{(k)} &= (W_{1,1} - 1, W_{1,2} - 1, W_{2,1}, W_{3,1}, \ldots, W_{k,1}, \ldots). \end{aligned} \tag{19.22}$$

Note that S_W is the Deprit transformation, associated, not with $W(t)$, but with the integral of W, which cancels out at 0. Then, we have the following relations, extracted from the first section (cf. proposition 14):

$$\begin{aligned} K_p &= \tfrac{1}{p} W_p + R_p^W, \quad R_p^W \in L_p(W_1, \ldots, W_{p-1}), \\ \tfrac{1}{p} W_p &= G_p + R_p^G, \quad R_p^G \in L_p(G_1, \ldots, G_{p-1}), \\ G_p &= K_p + R_p^K, \quad R_p^K \in L_p(K_1, \ldots, K_{p-1}). \end{aligned} \tag{19.23}$$

For $p > 1$, every R_p^W, is expressed in the Lyndon basis $\mathcal{L}_p(\{A, B\})$, and the coefficients are polynomials in the $\{W_{1,1}, W_{1,2}, W_{2,1}, W_{3,1}, \ldots, W_{p-1,1}, \ldots, W_{p-1,l_{p-1}}\}$. The monomials of these polynomials all contain a $W_{i,j}$ with $i > 1$, and so they are in $\mathfrak{I}_W^{(k)}$. Hence we deduce that $\mathfrak{I}_K^{(k)} \subset \mathfrak{I}_W^{(k)}$.

Using the same argument, we deduce that $\mathfrak{I}_W^{(k)} \subset \mathfrak{I}_G^{(k)}$ and $\mathfrak{I}_G^{(k)} \subset \mathfrak{I}_K^{(k)}$, that is to say that

$$\mathfrak{I}_K^{(k)} = \mathfrak{I}_W^{(k)} = \mathfrak{I}_G^{(k)}. \tag{19.24}$$

⁂

Let us now consider the following lemma:

LEMMA 46. — *Given the following problems:*

2. *Having represented* $S^{(n)}(t)$ *as* $\exp(-tL_{G_1})\cdots\exp(t^pL_{G_p})\cdots$,
solve in $L(\{A,B\})$: $G_1 = A+B, G_2 = \cdots = G_k = 0$

4. *Having posed* $Z = S^{(n)}(t)z$, *solve in* $\tilde{L}_p(\{z,A,B\})$:
$Z_0 = (\exp(-tL_{A+B})z)_0, \ldots, Z_k = (\exp(-tL_{A+B})z)_k$.

The varieties solutions to the two above problems are defined by the same ideal.

Proof. — For any $p \geq 1$, Z_p decomposes into $\tilde{L}_p(\{z,A,B\})$. Let us denote $z_{p,1},\ldots,z_{p,2^p}$ the Lyndon basis of this subspace which corresponds to the images of zw, where $w \in \{A,B\}_p^*$, by the parenthesizing mapping Λ. So we have, for any p, $Z_p = \sum_{q=1}^{2^p} Z_{p,q}z_{p,q}$, and the solutions to problem **4.** are the zeros of the ideal

$$\mathfrak{I}_Z^{(k)} = (Z_{i,j} - a_{i,j}, 1 \leq i \leq k, 1 \leq j \leq 2^i) \tag{19.25}$$

where the $a_{i,j}$ are the coefficients obtained by decomposing $\exp(-tL_{A+B})z$ in $\tilde{L}\{z,A,B\}$.

From $S^{(n)}(t)z - \exp(-tL_{A+B})z = \left(S^{(n)}(t) - \exp(-tL_{A+B})\right)z = o(t^k)$, we deduce term by term, for $p \leq k$, that

$$Z_p - \tfrac{(-1)^p}{p!}L_{A+B}^p z = \left[\sum_{\wp(1,p;p)} (-1)^{m_1+\cdots+m_p}\frac{L_{G_1}^{m_1}\cdots L_{G_p}^{m_p}}{m_p!\cdots m_1!} - \tfrac{(-1)^p}{p!}L_{A+B}^p\right]z = 0. \tag{19.26}$$

Hence we deduce, for $p = 1$, that

$$Z_1 - \{z, A+B\} = -(L_{G_1} - L_{A+B})z = \{z, G_1 - (A+B)\} = 0, \tag{19.27}$$

that is to say, on the basis $([z,A],[z,B])$,

$$Z_{1,1} - 1 = G_{1,1} - 1 = 0, \quad Z_{1,2} - 1 = G_{1,2} - 1 = 0. \tag{19.28}$$

At the order $1 \leq p \leq k$, (19.26) is written

$$\begin{aligned} Z_p - \tfrac{(-1)^p}{p!}L_{A+B}^p z &= \left[\sum_{\wp(1,p;p)} (-1)^{m_1+\cdots+m_p}\frac{L_{G_1}^{m_1}\cdots L_{G_p}^{m_p}}{m_p!\cdots m_1!} - \tfrac{(-1)^p}{p!}L_{A+B}^p\right]z \\ &= \tfrac{(-1)^p}{p!}\left(L_{G_1}^p - L_{A+B}^p\right)z - L_{G_p}z + \qquad (19.29) \\ &\qquad \sum_{\wp(1,p-1;p), m_1<p} (-1)^{m_1+\cdots+m_{p-1}}\frac{L_{G_1}^{m_1}\cdots L_{G_p}^{m_p}}{m_p!\cdots m_1!}z. \qquad (19.30) \end{aligned}$$

$L^p_{G_1} - L^p_{A+B}$ is written

$$\sum_{k_1+\cdots+k_p+l_1+\cdots+l_p=p} \left(G_{1,1}^{k_1+\cdots+k_p} G_{1,2}^{l_1+\cdots+l_p} - 1\right) L_A^{k_1} L_B^{l_1} \cdots L_A^{k_p} L_B^{l_p}. \tag{19.31}$$

A coefficient of (19.31) is then written

$$G_{1,1}^k G_{1,2}^l - 1 = (G_{1,1}^k - 1) G_{1,2}^l + (G_{1,2}^l - 1) \in (G_{1,1} - 1, G_{1,2} - 1) \subset \mathfrak{I}_G^{(p)}, \tag{19.32}$$

On the other hand, every $L_A^{k_1} L_B^{l_1} \cdots L_A^{k_p} L_B^{l_p} z$ decomposes linearly onto the Lyndon basis, and consequently, the coefficients of term $(L^p_{G_1} - L^p_{A+B})z$ belong to $(G_{1,1} - 1, G_{1,2} - 1) \subset \mathfrak{I}_G^{(p)}$. The other terms of (19.29) and (19.30) also decompose, and the coefficients are in

$$(G_{2,1}, \ldots, G_{p,1}, \ldots G_{p,l_p}) \subset \mathfrak{I}_G^{(p)}. \tag{19.33}$$

Let us assume proven, for any $m < p$, that $\mathfrak{I}_Z^{(m)} = \mathfrak{I}_G^{(m)}$. So, at the order p,

- we deduce from (19.26), by identifying every coefficient, that $\mathfrak{I}_Z^{(p)} \subset \mathfrak{I}_G^{(p)}$,
- from (19.26), we also deduce that

$$\begin{aligned} -L_{G_p} z &= \overbrace{Z_p - \frac{(-1)^p}{p!} L^p_{A+B} z}^{(I)} - \overbrace{\frac{(-1)^p}{p!} \left(L^p_{G_1} - L^p_{A+B}\right) z}^{(II)} - \\ & \underbrace{\sum_{\wp(1,p-1;p), m_1 < n} (-1)^{m_1+\cdots+m_{p-1}} \frac{L_{G_1}^{m_1} \cdots L_{G_p}^{m_p}}{m_p! \cdots m_1!} z}_{(III)}. \end{aligned} \tag{19.34}$$

All the terms, on the right hand side of the previous equation, decompose onto the basis $z_{p,q}$. The coefficients of (I) belong to $\mathfrak{I}_Z^p$, and, by assumption, those of (II) or (III) are in $(G_{1,1} - 1, G_{1,2} - 1) \subset \mathfrak{I}_G^{(p-1)} \subset \mathfrak{I}_Z^{(p-1)}$. Therefore $L_{G_p} z$ can be decomposed on the $z_{p,q}$ and the coefficients are in $\mathfrak{I}_Z^{(p)}$. Thus G_p can decompose on the $x_{p,q}$, by solving a linear system with coefficients in $\mathfrak{I}_Z^{(p)}$.

So we have, $\mathfrak{I}_G^{(p)} \subset \mathfrak{I}_Z^{(p)}$.

***

In order to get symplectic integrators directly, we will be able to use the three following methods, which are equivalent in virtue of the lemma 46.

19.1.1. Direct Method

The problem (19.20) may be solved directly by searching after, for any z

$$S^{(n)}(t)z = S_A(c_1 t) S_B(d_1 t) \cdots S_A(c_n t) S_B(d_n t) z = S_h(t) z + o(t^k). \tag{19.35}$$

19.1.2. Invariant Function

Problem (19.35) may be reformulated by expressing $S^{(n)}$ as an exponential, and by searching after $c_1, \ldots, c_n, d_1, \ldots d_n$ such that

$$S_A(c_1 t) S_B(d_1 t) \cdots S_A(c_n t) S_B(d_n t) = e^{-tL_K} \tag{19.36}$$

with $K = h + o(t^{k-1})$. K is not the Hamiltonian of the hamiltonian system associated with $S^{(n)}$, but, simply, an invariant function of the system. This technique is used by Yoshida.

19.1.3. Perturbed Hamiltonian

$S^{(n)}(t)$ can also be written as a Lie transformation

$$S_A(c_1 t) S_B(d_1 t) \cdots S_A(c_n t) S_B(d_n t) = S_W \tag{19.37}$$

where $W = h + o(t^{k-1})$. The condition is achieved by writing that

$$\frac{d}{dt} S_W = -S_W L_W = -S_h L_h + o(t^{k-1}). \tag{19.38}$$

The Hamiltonian of the associated hamiltonian system is then given by W.

***

The three methods yield equal solutions, which seemed clear. Lemmas 45 and 46 prove that these methods steer us toward the study of the same algebraic objects.

19.2. Computation of the First Integrators

For every preset order, by one method or another, we search for n such that $S^{(n)}(t)$ is an integrator of order k. We attempt to find the solutions to a system of polynomial equations, that is to say we search for the points of the variety defined by the ideal of the polynomial equations. Here, we use algebraic methods of elimination, or computational methods in standard bases. The software used is, either MACAULAY, which computes standard bases in a ring $\mathbb{Z}/p\mathbb{Z}$, or whenever possible, AXIOM.

For low orders, these methods provide symplectic integrators. Besides, we use the representation by the Lie transformations (Deprit), which has the advantage of not requiring the computation of exponential identities.

Using the Lie transformations, we get, at the order 4

$$\begin{aligned} W = {} & W_{1,1} A + W_{1,2} B + \tfrac{1}{2} W_{2,1}[A,\ B] + \tfrac{1}{3} W_{3,1}[A,\ [A,\ B]] + \tfrac{1}{3} W_{3,2}[[A,\ B],\ B] \\ & + \tfrac{1}{4} W_{4,1}[A,\ [A,\ [A,\ B]]] + \tfrac{1}{4} W_{4,2}[A,\ [[A,\ B],\ B]] + \tfrac{1}{4} W_{4,3}[A,\ [[A,\ B],\ B]], \end{aligned}$$

where the $W_{i,j}$ are polynomials. For $n = 4$, and 7 variables $c_1, c_2, c_3, c_4, d_1, d_2, d_3$, we have

$$W_{1,1} = c_3 + c_2 + c_1 + c_0 - 1$$

$$W_{1,2} = d_3 + d_2 + d_1 - 1$$

$$W_{2,1} = (c_3 - c_2 - c_1 - c_0)\ d_3 + (c_3 + c_2 - c_1 - c_0)\ d_2 + (c_3 + c_2 + c_1 - c_0)\ d_1$$

$$W_{3,1} = (\tfrac{1}{2}\ c_3^2 + (-c_2 - c_1 - c_0)\ c_3)\ d_3 + (\tfrac{1}{2}\ c_3^2 + (c_2 - c_1 - c_0)\ c_3 + \tfrac{1}{2}\ c_2^2 + (-c_1 - c_0)\ c_2)\ d_2$$

$$+ (\tfrac{1}{2}\, c_3^2 + (c_2 + c_1 - c_0)\ c_3 + \tfrac{1}{2}\, c_2^2 + (c_1 - c_0)\ c_2 + \tfrac{1}{2}\, c_1^2 - c_0\ c_1)\ d_1$$

$$W_{3,2} = (\tfrac{1}{2}\, c_2 + \tfrac{1}{2}\, c_1 + \tfrac{1}{2}\, c_0)\ d_3^2 + ((-c_2 + c_1 + c_0)\ d_2 + (-c_2 - c_1 + c_0)\ d_1)\ d_3 + (\tfrac{1}{2}\, c_1 + \tfrac{1}{2}\, c_0)\ d_2^2 + (-c_1 + c_0)\ d_1\ d_2 + \tfrac{1}{2}\, c_0\ d_1^2$$

$$W_{4,1} = (\tfrac{1}{6}\, c_3^3 + (-\tfrac{1}{2}\, c_2 - \tfrac{1}{2}\, c_1 - \tfrac{1}{2}\, c_0)\ c_3^2)\ d_3 + (\tfrac{1}{6}\, c_3^3 + (\tfrac{1}{2}\, c_2 - \tfrac{1}{2}\, c_1 - \tfrac{1}{2}\, c_0)\ c_3^2 + (\tfrac{1}{2}\, c_2^2 + (-c_1 - c_0)\ c_2)\ c_3 + \tfrac{1}{6}\, c_2^3 + (-\tfrac{1}{2}\, c_1 - \tfrac{1}{2}\, c_0)\ c_2^2)\ d_2 + (\tfrac{1}{6}\, c_3^3 + (\tfrac{1}{2}\, c_2 + \tfrac{1}{2}\, c_1 - \tfrac{1}{2}\, c_0)\ c_3^2 + (\tfrac{1}{2}\, c_2^2 + (c_1 - c_0)\ c_2 + \tfrac{1}{2}\, c_1^2 - c_0\ c_1)\ c_3 + \tfrac{1}{6}\, c_2^3 + (\tfrac{1}{2}\, c_1 - \tfrac{1}{2}\, c_0)\ c_2^2 + (\tfrac{1}{2}\, c_1^2 - c_0\ c_1)\ c_2 + \tfrac{1}{6}\, c_1^3 - \tfrac{1}{2}\, c_0\ c_1^2)\ d_1$$

$$W_{4,2} = (\tfrac{1}{2}\, c_2 + \tfrac{1}{2}\, c_1 + \tfrac{1}{2}\, c_0)\ c_3\ d_3^2 + (((-c_2 + c_1 + c_0)\ c_3 - \tfrac{1}{2}\, c_2^2 + (c_1 + c_0)\ c_2)\ d_2 + ((-c_2 - c_1 + c_0)\ c_3 - \tfrac{1}{2}\, c_2^2 + (-c_1 + c_0)\ c_2 - \tfrac{1}{2}\, c_1^2 + c_0\ c_1)\ d_1)\ d_3 + ((\tfrac{1}{2}\, c_1 + \tfrac{1}{2}\, c_0)\ c_3 + (\tfrac{1}{2}\, c_1 + \tfrac{1}{2}\, c_0)\ c_2)\ d_2^2 + ((-c_1 + c_0)\ c_3 + (-c_1 + c_0)\ c_2 - \tfrac{1}{2}\, c_1^2 + c_0\ c_1)\ d_1\ d_2 + (\tfrac{1}{2}\, c_0\ c_3 + \tfrac{1}{2}\, c_0\ c_2 + \tfrac{1}{2}\, c_0\ c_1)\ d_1^2$$

$$W_{4,3} = (-\tfrac{1}{6}\, c_2 - \tfrac{1}{6}\, c_1 - \tfrac{1}{6}\, c_0)\ d_3^3 + ((\tfrac{1}{2}\, c_2 - \tfrac{1}{2}\, c_1 - \tfrac{1}{2}\, c_0)\ d_2 + (\tfrac{1}{2}\, c_2 + \tfrac{1}{2}\, c_1 - \tfrac{1}{2}\, c_0)\ d_1)\ d_3^2 + ((-\tfrac{1}{2}\, c_1 - \tfrac{1}{2}\, c_0)\ d_2^2 + (c_1 - c_0)\ d_1\ d_2 - \tfrac{1}{2}\, c_0\ d_1^2)\ d_3 + (-\tfrac{1}{6}\, c_1 - \tfrac{1}{6}\, c_0)\ d_2^3 + (\tfrac{1}{2}\, c_1 - \tfrac{1}{2}\, c_0)\ d_1\ d_2^2 - \tfrac{1}{2}\, c_0\ d_1^2\ d_2 - \tfrac{1}{6}\, c_0\ d_1^3$$

- A solution for $k = 1$ is given by $c_1 = d_1 = 1$ and

$$S_1 = S^{(1)}(t) = S_A(\tfrac{t}{2})S_B(\tfrac{t}{2}). \tag{19.39}$$

- A solution for $k = 2$ is given by

$$c_1 + c_2 = 1, d_1 = 1, (c_2 - c_1)d_1 = 0, \tag{19.40}$$

that is to say, $c_1 = c_2 = \frac{1}{2}, d_1 = 1$ and

$$S_2(t) = S^{(2)}(t) = S_A(\tfrac{t}{2})S_B(t)S_A(\tfrac{t}{2}). \tag{19.41}$$

Mapping n time the operator $S_2(\tau)$ amount to mapping once $S_A(\frac{t}{2})$, then n times the operator S_1 of order 1 (19.39).

- The solutions for $k = 3$ are the solutions to the system of polynomial equations:

$$\left\{\begin{array}{rcl} c_3 + c_2 + c_1 & = & 1 \\ d_2 + d_1 & = & 1 \\ (c_3 - c_2 - c_1)d_2 + (c_3 + c_2 - c_1)d_1 & = & 0 \\ (\tfrac{1}{2}\, c_3^2 + (-c_2 - c_1)\ c_3)\ d_2 + (\tfrac{1}{2}\, c_3^2 + (c_2 - c_1)\ c_3 + \tfrac{1}{2}\, c_2^2 - c_1\ c_2)\ d_1 & = & 0 \\ (\tfrac{1}{2}\, c_2 + \tfrac{1}{2}\, c_1)\ d_2^2 + (-c_2 + c_1)\ d_1\ d_2 + \tfrac{1}{2}\, c_1\ d_1^2 & = & 0 \end{array}\right. \tag{19.42}$$

which, by successive eliminations, is reduced to

$$\left\{\begin{array}{rcl} d_2 + 2\ c_1 & = & 1 \\ d_1 - 2\ c_1 & = & 0 \\ c_3 + c_1 - \tfrac{1}{2} & = & 0 \\ c_2 - \tfrac{1}{2} & = & 0 \\ c_1^2 - \tfrac{1}{2}\ c_1 + \tfrac{1}{12} & = & 0 \end{array}\right. . \tag{19.43}$$

We notice that c_3 and c_1 are conjugate roots just as $d_1 = 2c_1$ and $d_2 = 2c_3$. On the other hand, the solutions are not real. We must include an added term, and we get, for example, as a real (and even rational) solution:

$$c_1 = \tfrac{2}{3}, d_3 = \tfrac{1}{24}, d_2 = -\tfrac{3}{4}, d_1 = -\tfrac{7}{24}, c_3 = 1, c_2 = -\tfrac{2}{3}. \tag{19.44}$$

The previous 6 term solution was also found by Forest & Ruth ([20]).

- After successive eliminations, the system of polynomial equations is equivalent for $k = 4$ to

$$\left\{\begin{array}{rcl} d_3 + 288\, c_1^4 - 312\, c_1^3 + 96\, c_1^2 - 14\, c_1 + \frac{1}{2} & = & 0 \\ d_2 - 288\, c_1^4 + 312\, c_1^3 - 96\, c_1^2 + 16\, c_1 - \frac{3}{2} & = & 0 \\ d_1 - 2\, c_1 & = & 0 \\ c_4 + 144\, c_1^4 - 156\, c_1^3 + 48\, c_1^2 - 7\, c_1 + \frac{1}{4} & = & 0 \\ c_3 + c_1 - \frac{1}{2} & = & 0 \\ c_2 - 144\, c_1^4 + 156\, c_1^3 - 48\, c_1^2 + 7\, c_1 - \frac{3}{4} & = & 0 \\ c_1^5 - \frac{5}{4}\, c_1^4 + \frac{13}{24}\, c_1^3 - \frac{1}{8}\, c_1^2 + \frac{1}{64}\, c_1 - \frac{1}{1152} & = & 0 \end{array}\right. . \tag{19.45}$$

The last polynomial is factored into

$$(c_1^2 - \tfrac{1}{4}\, c_1 + \tfrac{1}{24})\,(c_1^3 - c_1^2 + \tfrac{1}{4}\, c_1 - \tfrac{1}{48}), \tag{19.46}$$

in which the first factor has no real solution.

If c_1 is the root of $(c_1^3 - c_1^2 + \frac{1}{4}\, c_1 - \frac{1}{48})$, we get

$$\left\{\begin{array}{rcl} d_3 - 2\, c_1 & = & 0 \\ d_2 + 4\, c_1 - 1 & = & 0 \\ d_1 - 2\, c_1 & = & 0 \\ c_4 - c_1 & = & 0 \\ c_3 + c_1 - \frac{1}{2} & = & 0 \\ c_2 + c_1 - \frac{1}{2} & = & 0 \\ c_1^3 - c_1^2 + \frac{1}{4}\, c_1 - \frac{1}{48} & = & 0 \end{array}\right. , \tag{19.47}$$

or else, for example for $c_1 = \frac{1}{2(\sqrt[3]{2}-1)}$,

$$c_4 = c_1 = \tfrac{1}{2(\sqrt[3]{2}-1)}, d_3 = d_1 = \tfrac{1}{\sqrt[3]{2}-1}, d_2 = \tfrac{\sqrt[3]{2}-3}{\sqrt[3]{2}-1}, c_2 = c_3 = \tfrac{1}{2}\tfrac{\sqrt[3]{2}-2}{\sqrt[3]{2}-1}. \tag{19.48}$$

We then come up again with the solutions obtained by Forest & Ruth ([20]) or Yoshida ([61]).

If c_1 is the root of $(c_1^2 - \frac{1}{4}\, c_1 + \frac{1}{24})$, we get two additional solutions

$$d_3 + 2\, c_1 = \tfrac{1}{2}, d_2 = \tfrac{1}{2}, d_1 = 2\, c_1, c_4 + c_1 = \tfrac{1}{4}, c_3 + c_1 = \tfrac{1}{2}, c_2 - c_1 = \tfrac{1}{4}, \tag{19.49}$$

or similarly

$$c_4 = \bar{c_1}, c_3 = \bar{c_2} = \tfrac{1}{2} - c_1, d_3 = \bar{d_1} = 2\bar{c_1}, d_2 = \tfrac{1}{2}. \tag{19.50}$$

- For $k = 5$, we do not find solutions directly. So far, using MACAULAY, we can say that there are 46 solutions for integrators obtained as product of 11 operators (the computation is carried out in $\mathbb{Z}/p\mathbb{Z}$, for $p = 31991$). As for the AXIOM system, it fails to complete the computation.

- For $k = 6$, there is no solution whose product would have fewer than 13 operators. Above 13, I was not able to find any reasonable means to know whether or not the algebraic variety of the solutions was empty.

We notice that, for $k = 2$ and $k = 4$, the integrators of real values are reversible, that is to say, if $S(t)$ is an integrator of order k, we have

$$S(-t) = S^{-1}(t). \tag{19.51}$$

This observation inspires us to search for reversible symplectic integrators. The reversibility is an additional stability criterion.

19.3. Search for Reversible Integrators

If we suppose that the integrator is reversible, we have the following result ([61]):

LEMMA 47. — *If $S(t) = e^{-tL_{K(t)}}$ is a reversible integrator, then $K(t) = K(-t)$.*

The existence of K is warranted by the Campbell-Hausdorff theorem. Since

$$\left[e^{-tL_{K(t)}}\right]^{-1} = e^{tL_{K(t)}} \tag{19.52}$$

we deduce that $L_{K(t)} = L_{K(-t)}$, therefore that $K(t) = K(-t)$.

⁂

So now, we can directly search for reversible integrators of $2k$ order. We search for $c_0, \dots, c_n, d_1, \dots, d_n$ and $S_r^{(n)} = S_h(t) + o(t^{2k})$ in the form

$$S_A(c_nt)S_B(d_nt)\cdots S_A(c_1t)S_B(d_1t)S_A(c_0t)S_B(d_1t)S_A(c_1t)\cdots S_B(d_nt)S_A(c_nt). \tag{19.53}$$

We can express $S_r^{(n)}$, either in the form of an exponential e^{-tL_K}, or in the form of a Lie transformation S_W. The solutions found are strictly the same, owing to lemma 46. If we express $S_r^{(n)}$ in the form of an exponential, we get an even Hamiltonian K, and the search for the integrator of order $2k$ is achieved by simultaneously canceling polynomial equations of odd degree, in number $\dim L_{2p+1}(\{A, B\})$. Else, we get a Hamiltonian W, that is not even, and we must, *a priori*, consider the terms of odd degree. However, the following proposition enables us to consider the terms of even order only:

PROPOSITION 48. — *If $S_{W(t)}$ is reversible, and if, for a given $k \in \mathbb{N}$, we have $W(t) = h + o(t^{2k-1})$, then $W(t) = h + o(t^{2k})$.*

Let us define $K(t) = \sum_{i \geq 0} t^i K_{i+1} \in \tilde{L}(\{K_1, \dots, K_n\})$, such that

$$S_W = \exp(-tL_K) = S_h(t) + o(t^{2k}). \tag{19.54}$$

Here, the basic ring is $\mathbb{Q}[c_0, \dots, c_n, d_1, \dots, d_n]$. K and W are connected by expressions of the form (cf. proposition 14):

$$K_p = \tfrac{1}{p}W_p + W_p', \quad p \geq 1, \quad \text{where} \quad W_p' \in \tilde{L}(K_1, \dots, K_{p-1}). \tag{19.55}$$

Therefore, let us suppose that $W = h + o(t^{2k-1})$, that is to say $W_1 = h, W_2 = \cdots = W_{2k-1} = 0$. Since S_W is reversible, we deduce that $K_{2k} = 0$, that is to say $W_{2k} + W_{2k}' = 0$. But $W_{2k}' \in \tilde{L}(K_1, \dots, K_{2k-1})$, vanishes, because its weight is $2k$. Consequently, $W_{2k} = 0$ and $W = h + o(t^{2k})$.

⁂

In the case of the search for reversible integrators in the form of Lie transformations, all that is required is to cancel the terms of even order. The Lie transformation is all the more useful that the generator W is deduced without using exponential identities. For low orders, the direct search is implemented as follows:

— we compute the polynomials to be canceled, that is to say the coefficients of the Lie series,

— we compute a standard basis, –for the order of the elimination– of the ideal defining the variety . Besides, we use MACAULAY software to compute the standard bases modulo p. Only in the case where the variety of the solutions is not void in $\mathbb{Z}/p\mathbb{Z}$, do we attempt to compute it on $\mathbb{Q}$, by using AXIOM.

Let us give specifications for proposition 48.

PROPOSITION 49. — *Given* $\mathfrak{I}_W^{(2k)} = (W_{1,1}-1, W_{1,2}-1, W_{2,1}, \dots, W_{2k,1}, \dots, W_{2k,l_{2k}})$, *the ideal defining the variety of the solutions to* $S_r^{(n)}(t) = S_{W(t)} = S_h + o(t^{2k})$, *so, for any* $p \le k$, *we have*

$$\{W_{2p,1}, \dots, W_{2p,l_{2p}}\} \subset \mathfrak{I}_W^{(2p-1)} \quad \textit{that is to say} \quad \mathfrak{I}_W^{(2p)} = \mathfrak{I}_W^{(2p-1)}. \tag{19.56}$$

For any $p \ge 1$, we get, by searching for the solutions to $S_r^{(n)}(t)$ in exponential form $\exp(-\sum_{i\ge1} t^i K_i)$, the ideal

$$\mathfrak{I}_K^{(p)} = (K_{1,1}-1, K_{1,2}-1, K_{2,1}, \dots, K_{p,1}, K_{p,l_p}) \tag{19.57}$$

where the $K_{i,j}$ are homogeneous polynomials of degree i. According to the lemma 45, we have $\mathfrak{I}_K^{(p)} = \mathfrak{I}_W^{(p)}$, for any $p \le 2k$. However, since S_W is reversible, we have $K_{2p} = 0$, for any p, that is to say $\mathfrak{I}_K^{2p} = \mathfrak{I}_K^{2p-1}$ and $\mathfrak{I}_W^{2p} = \mathfrak{I}_W^{2p-1}$.

⁂

The previous lemma enables us to consider the polynomials of odd degree only.

• For $k = 4$, we find, at the order 3,

$$\begin{aligned} W &= (2\, d_1 + 2\, d_2)\, A + (c_0 + 2\, c_1)\, B \\ &+ \tfrac{t^2}{3}((-\tfrac12\, d_1^2 - d_2\, d_1 - \tfrac12\, d_2^2)\, c_0 + (2\, d_1^2 - 2\, d_2\, d_1 - d_2^2)\, c_1)[A,\, [A,\, B]] \\ &+ \tfrac{t^2}{3}((\tfrac12\, d_1 + \tfrac12\, d_2)\, c_0^2 + (-d_1 + 2\, d_2)\, c_1\, c_0 + (-d_1 + 2\, d_2)\, c_1^2)\ [[A,\, B],\, B]. \end{aligned}$$

The general solution is given by

$$c_0 + 4\, d_2 - 1 = 0, c_1 - 2\, d_2 = 0,\ d_1 + d_2 - \tfrac12 = 0, d_2^3 - d_2^2 + \tfrac14\, d_2 - \tfrac{1}{48} = 0. \tag{19.58}$$

We come up again with three of the solutions obtained with the previous method, and specifically with the real solution:

$$d_2 = \tfrac{1}{2(\sqrt[3]{2}-1)}, d_1 = \tfrac12 \tfrac{\sqrt[3]{2}-2}{\sqrt[3]{2}-1}, c_1 = \tfrac{1}{\sqrt[3]{2}-1}; c_0 = \tfrac{\sqrt[3]{2}-3}{\sqrt[3]{2}-1}. \tag{19.59}$$

• For $k = 6$, we find, with $n = 8$, a variety of dimension 0, by using MACAULAY. The degree being 39, we deduce that there are at most 39 solutions. These solutions would provide integrators of order 6 as the reversible product of 15 operators. Unfortunately, we cannot directly find the solutions on $\mathbb{Q}$. However, we still find these solutions by using the method proposed by Yoshida ([61]), which we will mention in the next section.

Number of Equations				
Order	Non Reversible		Reversible	
1	2		2	
2	1	**3**		**2**
3	2		2	
4	3	**8**		**4**
5	6		6	
6	9	**23**		**10**
7	18		18	
8	30	**71**		**28**
9	56		56	
10	99	**226**		**84**

Table 2

***

For higher orders, we find the system of polynomial equations to be solved, but we do not find any more solutions than with the direct method. Here, we give (table 2) the number of polynomial equations to solve, in the case where we only compute the even terms, or all the terms. The number of equations is, in fact, the dimension of $L_n(\{A, B\})$.

20. Search for Integrators of Higher Order

The advantage of considering only reversible integrators also lies in the fact that we can find an integrator of order $2k+2$, as a function of an integrator of order $2k$. Indeed, we have lemma (cf. [54]):

Lemma 50. — *If $S_{2k}(t)$ is a symplectic integrator of order $2k$, then*

$$S(t) = S_{2k}(\tfrac{1}{2-\omega_{2k+1}}t)S_{2k}(-\tfrac{\omega_{2k+1}}{2-\omega_{2k+1}}t)S_{2k}(\tfrac{1}{2-\omega_{2k+1}}t) \tag{20.1}$$

is a symplectic integrator of order $2k+2$, insofar as $\omega_{2k+1}^{2k+1} = 2$.

We simply need to write

$$S_{2k}(t) = e^{-t(K_1 + t^{2k}K_{2k+1}^{(2k)} + o(t^{2k+1}))} \tag{20.2}$$

and

$$\log\left[S_{2k}(at)S_{2k}(bt)S_{2k}(at)\right] = t(2a+b)K_1 + t^{2k+1}(2a^{2k+1} + b^{k+1})K_{2k+1}^{(2k)} + o(t^{2k+2}). \tag{20.3}$$

We must have $2a+b=1, 2a^{2k+1}+b^{2k+1}=0$ and $b=-\omega_{2k+1}a, a(2-\omega_{2k+1})=1$, that is to say,

$$a = \tfrac{1}{2-\omega_{2k+1}},\ b = -\tfrac{\omega_{2k+1}}{2-\omega 2k+1}. \tag{20.4}$$

With the integrator $S_2(t) = e^{-\frac{1}{2}tL_A}e^{-tL_B}e^{-\frac{1}{2}tL_A}$, we come up again with the integrator of order 4

$$S_4(t) = S_2(\tfrac{1}{2-\sqrt[3]{2}}t)S_2(\tfrac{\sqrt[3]{2}}{2-\sqrt[3]{2}}t)S_2(\tfrac{1}{2-\sqrt[3]{2}}t). \tag{20.5}$$

This method enables us to construct integrators of even order, but an integrator of order $2k$ is a product of $2.3^{k-1}+1$, which are operators of order 1, S_A or S_B. This way, we would find an integrator of order 6, a product of 19 elementary operators, as well as an integrator of order 8, as a product of 55 operators.

20.1. Construction of Integrators as Products of Integrators of Order 2

Since we do not directly find solutions by searching for reversible integrators for high orders, Yoshida ([61]) proposes the search for reversible integrators, as reversible products of integrators of order 2.

Given $S_2(t)$, the integrator of order 2 defined in (19.41). We search for reversible integrators that are the products of S_2, in the form

$$S^{(n)}(t) \;=\; S_2(c_nt)\cdots S_2(c_1t)S_2(c_0t)S_2(c_1t)\cdots S_2(c_nt) = e^{-tL_{K^{(n)}}}. \tag{20.6}$$

Similarly, we can write, based on

$$\log S_2(t) = \sum_{i\geq 0} t^{2i+1}K^{(n)}_{2i+1}, \tag{20.7}$$

$$K^{(0)} \;=\; \sum_{i\geq 0} t^{2i}c_0^{2i}K_{2i+1}, \tag{20.8}$$

$$-tK^{(n+1)} \;=\; \log\left[e^{-\sum_{i\geq 0}t^{2i+1}c_n^{2i}L_{K_{2i+1}}}e^{-tL_{K^{(n)}(t)}}e^{-\sum_{i\geq 0}t^{2i+1}c_n^{2i}L_{K_{2i+1}}}\right]. \tag{20.9}$$

Considering the alphabet $X = \{K_{2i+1}, i\geq 0, |K_{2i+1}|_* = 2i+1\}$, $K^{(n)}$ is expressed as a formal Lie series of $\tilde{L}(X,R)$, where R is the ring of the polynomials with coefficients in $\mathbb{Q}$.

Let us denote the Lyndon basis of $\tilde{L}_n(X)$ by $\{K_{n,1},\ldots,K_{n,l_n}\}$, where $l_n = \dim\tilde{L}_n(X)$. We can

$$K^{(n)} = \sum_{i\geq 0} t^{2i}\sum_{j=1}^{l_{2i+1}} k^{(n)}_{2i+1,j}K_{2i+1,j}. \tag{20.10}$$

Then, we find the equations which define the coefficients $c_0,\ldots,c_n$, by expressing recursively the $k^{(n+1)}_{2i+1,j}$ in function of the $k^{(n)}_{2i+1,j}$. In order to do this,

— we compute the various Lyndon bases for the odd orders,

— we formally compute, at a given order, $\log\left[S_2(c_nt)e^{-tL_K}S_2(c_nt)\right]$, where

$$K = \sum_{i\geq 0}\sum_{j=1}^{l_{2i+1}} k_{2i+1,j}K_{2i+1,j} \tag{20.11}$$

is a generic element of the Lie algebra.

The Lyndon basis, at order 9, of $\tilde{L}(X)$ is

$$K_{1,1}=k_1, K_{3,1}=k_3, K_{5,1}=k_5, K_{5,2}=[k_1,[k_1,\ k_3]],$$
$$K_{7,1}=k_7, K_{7,2}=[[k_1,\ k_3],\ k_3], K_{7,3}=[k_1,\ [k_1,\ k_5]], K_{7,4}=[k_1,\ [k_1,\ [k_1,\ [k_1,\ k_3]]]],$$
$$K_{9,1}=k_9, K_{9,2}=[[k_1,\ k_5],\ k_3], K_{9,3}=[k_1,\ [k_3,\ k_5]], K_{9,4}=[k_1,\ [k_1,\ k_7]],$$
$$K_{9,5}=[[k_1,\ [k_1,\ k_3]],\ [k_1,\ k_3]], K_{9,6}=[k_1,\ [k_1,\ [[k_1,\ k_3],\ k_3]]],$$
$$K_{9,7}=[k_1,\ [k_1,\ [k_1,\ [k_1,\ k_5]]]], K_{9,8}=[k_1,\ [k_1,\ [k_1,\ [k_1,\ [k_1,\ [k_1,\ k_3]]]]]].$$

Then we find, at order 9, the following relations:

$$k_{1,1}^{(n+1)}=k_{1,1}^{(n)}+2\ c_n,$$

$$k_{3,1}^{(n+1)}=k_{3,1}^{(n)}+2\ c_n^3,$$

$$k_{5,1}^{(n+1)}=k_{5,1}^{(n)}+2\ c_n^5,$$

$$k_{5,2}^{(n+1)}=k_{5,2}^{(n)}+(-\tfrac{1}{6}\ c_n\ k_{1,1}^{(n)}-\tfrac{1}{6}\ c_n^2)\ k_{3,1}^{(n)}+\tfrac{1}{6}\ c_n^3\ {k_{1,1}^{(n)}}^2+\tfrac{1}{6}\ c_n^4\ k_{1,1}^{(n)},$$

$$k_{7,1}^{(n+1)}=k_{7,1}^{(n)}+2\ c_n^7,$$

$$k_{7,2}^{(n+1)}=k_{7,2}^{(n)}+(-\tfrac{1}{6}\ c_n\ k_{1,1}^{(n)}-\tfrac{1}{6}\ c_n^2)\ a_{5,1}+\tfrac{1}{6}\ c_n^5\ {k_{1,1}^{(n)}}^2+\tfrac{1}{6}\ c_n^6\ k_{1,1}^{(n)},$$

$$k_{7,3}^{(n+1)}=k_{7,3}^{(n)}+\tfrac{1}{6}\ c_n\ {k_{3,1}^{(n)}}^2+(-\tfrac{1}{6}\ c_n^3\ k_{1,1}^{(n)}+\tfrac{1}{6}\ c_n^4)\ a_{3,1}-\tfrac{1}{6}\ c_n^6\ k_{1,1}^{(n)},$$

$$k_{7,4}^{(n+1)}=k_{7,4}^{(n)}+(-\tfrac{1}{6}\ c_n\ k_{1,1}^{(n)}-\tfrac{1}{6}\ c_n^2)\ k_{5,2}^{(n)}+(\tfrac{1}{360}\ c_n\ {k_{1,1}^{(n)}}^3+\tfrac{1}{45}\ c_n^2\ {k_{1,1}}^2+\tfrac{7}{180}\ c_n^3\ k_{1,1}^{(n)}$$
$$+\tfrac{7}{360}\ c_n^4)\ k_{3,1}^{(n)}-\tfrac{1}{360}\ c_n^3\ {k_{1,1}^{(n)}}^4-\tfrac{1}{45}\ c_n^4\ {k_{1,1}^{(n)}}^3-\tfrac{7}{180}\ c_n^5\ {k_{1,1}^{(n)}}^2-\tfrac{7}{360}\ c_n^6\ k_{1,1}^{(n)},$$

$$k_{9,1}^{(n+1)}=k_{9,1}^{(n)}+2\ c_n^9,$$

$$k_{9,2}^{(n)}=k_{9,2}^{(n)}+(-\tfrac{1}{6}\ c_n\ k_{1,1}^{(n)}-\tfrac{1}{6}\ c_n^2)\ k_{7,1}^{(n)}+\tfrac{1}{6}\ c_n^7\ {k_{1,1}^{(n)}}^2+\tfrac{1}{6}\ c_n^8\ k_{1,1}^{(n)},$$

$$k_{9,3}^{(n+1)}=k_{9,3}^{(n)}+(\tfrac{1}{6}\ c_n\ k_{3,1}^{(n)}-\tfrac{1}{3}\ c_n^3\ k_{1,1}^{(n)}-\tfrac{1}{6}\ c_n^4)\ k_{5,1}^{(n)}+(\tfrac{1}{6}\ c_n^5\ k_{1,1}^{(n)}+\tfrac{1}{3}\ c_n^6)\ k_{3,1}^{(n)}-\tfrac{1}{6}\ c_n^8\ k_{1,1}^{(n)},$$

$$k_{9,4}^{(n+1)}=k_{9,4}^{(n)}+(\tfrac{1}{3}\ c_n\ k_{3,1}^{(n)}-\tfrac{1}{6}\ c_n^3\ k_{1,1}^{(n)}+\tfrac{1}{6}\ c_n^4)\ k_{5,1}^{(n)}+(-\tfrac{1}{6}\ c_n^5\ k_{1,1}^{(n)}+\tfrac{1}{6}\ c_n^6)\ k_{3,1}^{(n)}-\tfrac{1}{3}\ c_n^8\ k_{1,1}^{(n)},$$

$$k_{9,5}^{(n+1)}=k_{9,5}^{(n)}+(-\tfrac{1}{6}\ c_n\ k_{1,1}^{(n)}-\tfrac{1}{6}\ c_n^2)\ k_{7,2}^{(n)}+(\tfrac{1}{360}\ c_n\ {k_{1,1}^{(n)}}^3+\tfrac{1}{45}\ c_n^2\ {k_{1,1}^{(n)}}^2+\tfrac{7}{180}\ c_n^3\ k_{1,1}^{(n)}$$
$$+\tfrac{7}{360}\ c_n^4)\ k_{5,1}^{(n)}-\tfrac{1}{360}\ c_n^5\ {k_{1,1}^{(n)}}^4-\tfrac{1}{45}\ c_n^6\ {k_{1,1}^{(n)}}^3-\tfrac{7}{180}\ c_n^7\ {k_{1,1}^{(n)}}^2-\tfrac{7}{360}\ c_n^8\ k_{1,1}^{(n)},$$

$$k_{9,6}^{(n+1)}=k_{9,6}^{(n)}+(-\tfrac{1}{6}\ c_n\ k_{1,1}^{(n)}-\tfrac{1}{6}\ c_n^2)\ k_{7,3}^{(n)}+(\tfrac{1}{6}\ c_n\ k_{3,1}^{(n)}+\tfrac{1}{6}\ c_n^3\ k_{1,1}^{(n)}+\tfrac{1}{3}\ c_n^4)\ k_{5,2}^{(n)}+(-\tfrac{1}{120}\ c_n\ {k_{1,1}^{(n)}}^2$$
$$-\tfrac{2}{45}\ c_n^2\ k_{1,1}^{(n)}-\tfrac{7}{180}\ c_n^3)\ {k_{3,1}^{(n)}}^2+(\tfrac{1}{120}\ c_n^3\ {k_{1,1}^{(n)}}^3+\tfrac{1}{45}\ c_n^4\ {k_{1,1}^{(n)}}^2-\tfrac{7}{180}\ c_n^5\ k_{1,1}^{(n)}-\tfrac{7}{120}\ c_n^6)\ k_{3,1}^{(n)}$$
$$+\tfrac{1}{45}\ c_n^6\ {k_{1,1}^{(n)}}^3+\tfrac{7}{90}\ c_n^7\ {k_{1,1}^{(n)}}^2+\tfrac{7}{120}\ c_n^8\ a_{1,1},$$

$$k_{9,7}^{(n+1)}=k_{9,7}^{(n)}+(-\tfrac{1}{3}\ c_n\ k_{3,1}+\tfrac{1}{6}\ c_n^3\ k_{1,1}^{(n)}-\tfrac{1}{6}\ c_n^4)\ k_{5,2}^{(n)}+(\tfrac{1}{360}\ c_n\ {k_{1,1}^{(n)}}^2+\tfrac{1}{30}\ c_n^2\ k_{1,1}^{(n)}+\tfrac{1}{45}\ c_n^3)\ {k_{3,1}^{(n)}}^2$$
$$+(-\tfrac{1}{360}\ c_n^3\ {k_{1,1}^{(n)}}^3-\tfrac{2}{45}\ c_n^4\ {k_{1,1}^{(n)}}^2-\tfrac{1}{180}\ c_n^5\ k_{1,1}^{(n)}+\tfrac{7}{360}\ c_n^6)\ k_{3,1}^{(n)}+\tfrac{1}{90}\ c_n^6\ {k_{1,1}}^3-\tfrac{1}{60}\ c_n^7\ {k_{1,1}^{(n)}}^2$$
$$-\tfrac{7}{360}\ c_n^8\ k_{1,1}^{(n)},$$

$$k_{9,8}^{(n+1)}=k_{9,8}^{(n)}+(-\tfrac{1}{6}\ c_n\ k_{1,1}^{(n)}-\tfrac{1}{6}\ c_n^2)\ k_{7,4}^{(n)}+(\tfrac{1}{360}\ c_n\ {k_{1,1}^{(n)}}^3+\tfrac{1}{45}\ c_n^2\ {k_{1,1}^{(n)}}^2+\tfrac{7}{180}\ c_n^3\ k_{1,1}^{(n)}$$

$$+ \tfrac{7}{360}\, c_n^4)\, k_{5,2}^{(n)} + (-\tfrac{1}{15120}\, c_n\, {k_{1,1}^{(n)}}^5 - \tfrac{1}{1260}\, c_n^2\, {k_{1,1}^{(n)}}^4 - \tfrac{53}{15120}\, c_n^3\, {k_{1,1}^{(n)}}^3 - \tfrac{13}{1890}\, c_n^4\, {k_{1,1}^{(n)}}^2$$
$$- \tfrac{31}{5040}\, c_n^5\, k_{1,1}^{(n)} - \tfrac{31}{15120}\, c_n^6)\, k_{3,1}^{(n)} + \tfrac{1}{15120}\, c_n^3\, {k_{1,1}^{(n)}}^6 + \tfrac{1}{1260}\, c_n^4\, {k_{1,1}^{(n)}}^5 + \tfrac{53}{15120}\, c_n^5\, {k_{1,1}^{(n)}}^4$$
$$+ \tfrac{13}{1890}\, c_n^6\, {k_{1,1}^{(n)}}^3 + \tfrac{31}{5040}\, c_n^7\, {k_{1,1}^{(n)}}^2 + \tfrac{31}{15120}\, c_n^8\, k_{1,1}^{(n)}.$$

⁂

With $k_{1,1}^{(1)} = c_0,\ k_{3,1}^{(1)} = c_0^3,\ k_{2i+1}^{(1)} = c_0^{2i+1}$, we find for the various orders:

- For $k = 4$ and $n = 1$,

$$k_{1,1}^{(2)} = c_0 + 2c_1,\ k_{3,1}^{(2)} = c_0^3 + 2c_1^3 \tag{20.12}$$

and the solution, to get an integrator of order 4 is $k_{1,1}^{(2)} = 1,\ k_{3,1}^{(2)} = 0$, that is to say,

$$c_1 = \tfrac{1}{2-\sqrt[3]{2}},\ c_0 = \tfrac{\sqrt[3]{2}}{2-\sqrt[3]{2}}. \tag{20.13}$$

- For $k = 6$ and $n = 4$, we find, for the coefficients,

$$k_{1,1}^{(2)} = c_0 + 2c_1 + 2c_2 + 2c_3,$$
$$k_{3,1}^{(2)} = c_0^3 + 2c_1^3 + 2c_2^3 + 2c_3^3,$$
$$k_{5,1}^{(2)} = c_0^5 + 2c_1^5 2\ c_2^5 + 2\ c_3^5,$$
$$\begin{aligned} k_{5,2}^{(2)} = {} & (\tfrac{1}{3}\, c_2 + \tfrac{1}{3}\, c_1 + \tfrac{1}{6}\, c_0)\, c_3^4 + (\tfrac{2}{3}\, c_2^2 + (\tfrac{4}{3}\, c_1 + \tfrac{2}{3}\, c_0)\, c_2 + \tfrac{2}{3}\, c_1^2 + \tfrac{2}{3}\, c_0\, c_1 + \tfrac{1}{6}\, c_0^2)\, c_3^3 \\ & + (-\tfrac{1}{3}\, c_2^3 - \tfrac{1}{3}\, c_1^3 - \tfrac{1}{6}\, c_0^3)\, c_3^2 + (-\tfrac{2}{3}\, c_2^4 + (-\tfrac{2}{3}\, c_1 - \tfrac{1}{3}\, c_0)\, c_2^3 + (-\tfrac{2}{3}\, c_1^3 - \tfrac{1}{3}\, c_0^3)\, c - \tfrac{2}{3}\, c_1^4 \\ & - \tfrac{1}{3}\, c_0\, c_1^3 - \tfrac{1}{3}\, c_0^3\, c_1 - \tfrac{1}{6}\, c_0^4)\, c_3 + (\tfrac{1}{3}\, c_1 + \tfrac{1}{6}\, c_0)\, c_2^4 + (\tfrac{2}{3}\, c_1^2 + \tfrac{2}{3}\, c_0\, c_1 + \tfrac{1}{6}\, c_0^2)\, c_2^3 \\ & + (-\tfrac{1}{3}\, c_1^3 - \tfrac{1}{6}\, c_0^3)\, c_2^2 + (-\tfrac{2}{3}\, c_1^4 - \tfrac{1}{3}\, c_0\, c_1^3 - \tfrac{1}{3}\, c_0^3\, c_1 - \tfrac{1}{6}\, c_0^4)\, c_2 + \tfrac{1}{6}\, c_0\, c_1^4 + \tfrac{1}{6}\, c_0^2\, c_1^3 \\ & - \tfrac{1}{6}\, c_0^3\, c_1 c_1^2 - \tfrac{1}{6}\, c_0^4\, c_1. \end{aligned}$$

After reduction, we get c_0 as the root of an irreducible polynomial of degree 39, which admits 3 real roots,

$$\begin{aligned} & c_0^{39} + 4\, c_0^{38} - 18\, c_0^{37} - \tfrac{232}{3}\, c_0^{36} + \tfrac{6469}{45}\, c_0^{35} + \tfrac{8108}{15}\, c_0^{34} - \tfrac{82144}{135}\, c_0^{33} - \tfrac{239008}{135}\, c_0^{32} + \tfrac{870652}{675}\, c_0^{31} + \\ & \tfrac{5898416}{2025}\, c_0^{30} - \tfrac{618824}{675}\, c_0^{29} - \tfrac{5158016}{2025}\, c_0^{28} + \tfrac{2525372}{30375}\, c_0^{27} + \tfrac{32135888}{30375}\, c_0^{26} - \tfrac{1377776}{10125}\, c_0^{25} - \\ & \tfrac{33361568}{91125}\, c_0^{24} + \tfrac{536566}{10125}\, c_0^{23} + \tfrac{35651416}{455625}\, c_0^{22} - \tfrac{19660868}{1366875}\, c_0^{21} - \tfrac{8051504}{455625}\, c_0^{20} + \tfrac{5636474}{1366875}\, c_0^{19} + \\ & \tfrac{11313208}{4100625}\, c_0^{18} - \tfrac{17674448}{20503125}\, c_0^{17} - \tfrac{8733536}{20503125}\, c_0^{16} + \tfrac{1302268}{6834375}\, c_0^{15} + \tfrac{87632}{2460375}\, c_0^{14} - \tfrac{624184}{20503125}\, c_0^{13} + \\ & \tfrac{288448}{922640625}\, c_0^{12} + \tfrac{3333844}{922640625}\, c_0^{11} - \tfrac{716752}{922640625}\, c_0^{10} - \tfrac{127664}{553584375}\, c_0^{9} + \tfrac{143264}{922640625}\, c_0^{8} - \\ & \tfrac{136499}{4613203125}\, c_0^{7} - \tfrac{19996}{8303765625}\, c_0^{6} + \tfrac{117142}{41518828125}\, c_0^{5} - \tfrac{33848}{41518828125}\, c_0^{4} + \tfrac{17431}{124556484375}\, c_0^{3} - \\ & \tfrac{9668}{622782421875}\, c_0^{2} + \tfrac{656}{622782421875}\, c_0 - \tfrac{64}{1868347265625}, \end{aligned}$$

and $c_1 = P_1(c_0), c_2 = P_2(c_0), c_3 = P_3(c_0)$ where P_1, P_2, P_3 are polynomials of degree 38 (this result is obtained in 20' of C.P.U. time, with AXIOM, on an IBM Risc6000 computer). We have thus found all possible solutions for the reversible integrators of order 6, since we know that there are at most 39 solutions. The obtained integrators are products of 15 elementary operators.

• Likewise, by using numerical methods, Yoshida found an integrator of order 8 with $n = 7$, that is to say as a product of 15 integrators S_2 or 31 operators of order 1: S_A and S_B. We have been able to verify that these integrators are not the reversible product of integrators of order 4, which would also have totaled 31 terms.

***

The same methods may be applied when searching for reversible integrators as the reversible product of integrators of order 4. The technique is identical, and consists of searching for a K of the form

$$K^{(n)} = k_{1,1}^{(n)} K_{1,1} + k_{5,1}^{(n)} K_{5,1} + k_{7,1}^{(n)} K_{7,1} + k_{7,2}^{(n)} K_{7,2} + \cdots. \tag{20.14}$$

We first compute the Lyndon basis:

$K_{1,1} = k_1,$

$K_{5,1} = k_5,$

$K_{7,1} = k_7,\ K_{7,2} = [k_1,\ [k_1,\ k_5]],$

$K_{9,1} = k_9,\ K_{9,2} = [k_1,\ [k_1,\ k_7]],\ K_{9,3} = [k_1,\ [k_1,\ [k_1,\ [k_1,\ k_5]]]]$

We find the relations between the $k_{i,j}^{(n)}$ at order 9:

$k_{1,1}^{(n+1)} = k_{1,1}^{(n)} + 2\ c_n,$

$k_{5,1}^{(n+1)} = k_{5,1}^{(n)} + 2\ c_n^5,$

$k_{7,1}^{(n+1)} = k_{7,1}^{(n)} + 2\ c_n^7,$

$k_{7,2}^{(n+1)} = k_{7,2}^{(n)} + (-\frac{1}{6}\ c_n\ k_{1,1}^{(n)} - \frac{1}{6}\ c_n^2)\ k_{5,1}^{(n)} + \frac{1}{6}\ c_n^5\ {k_{1,1}^{(n)}}^2 + \frac{1}{6}\ c_n^6\ k_{1,1}^{(n)},$

$k_{9,1}^{(n+1)} = k_{9,1}^{(n)} + 2\ c_n^9,$

$k_{9,2}^{(n)} = k_{9,2}^{(n)} + (-\frac{1}{6}\ c_n\ k_{1,1}^{(n)} - \frac{1}{6}\ c_n^2)\ k_{7,1}^{(n)} + \frac{1}{6}\ c_n^7\ {k_{1,1}^{(n)}}^2 + \frac{1}{6}\ c_n^8\ k_{1,1}^{(n)},$

$$\begin{aligned} k_{9,3}^{(n+1)} &= k_{9,3}^{(n)} + (-\tfrac{1}{6}\ c_n\ k_{1,1}^{(n)} - \tfrac{1}{6}\ c_n^2)\ k_{7,2}^{(n)} + (\tfrac{1}{360}\ c_n\ {k_{1,1}^{(n)}}^3 + \tfrac{1}{45}\ c_n^2\ {k_{1,1}^{(n)}}^2 + \tfrac{7}{180}\ c_n^3\ k_{1,1}^{(n)} \\ &\quad + \tfrac{7}{360}\ c_n^4)\ k_{5,1}^{(n)} - \tfrac{1}{360}\ c_n^5\ {k_{1,1}^{(n)}}^4 - \tfrac{1}{45}\ c_n^6\ {k_{1,1}^{(n)}}^3 - \tfrac{7}{180}\ c_n^7\ {k_{1,1}^{(n)}}^2 - \tfrac{7}{360}\ c_n^8\ k_{1,1}^{(n)}. \end{aligned}$$

• For $k = 6$, we find a solution with $n = 2$: $c_0 + 2c_1 = 1, c_0^5 + 2c_1^5 = 0$, that is to say

$$c_0 = \frac{1}{2-\sqrt[5]{2}}, c_1 = \frac{\sqrt[5]{2}}{2-\sqrt[5]{2}}. \tag{20.15}$$

• For $k = 8$ and $n = 3$, the variety of the solutions is empty, which proves that the integrator of order 8 found by Yoshida, ([61]) is not a reversible product of integrators of order 4, which would also have had 31 factors.

***

In chart 3, for each order and method, we give the number of polynomial equations, and the number of solutions obtained, with the number of operators in the product. In the

	Polynomials				Operators			Solutions		
			Reversibles			Reversibles			Reversibles	
Order	D	L	RL	S_2	L	RL	S_2	L	RL	S_2
1	2	2	2		1			1		
2	4	1		1	3	3	3	1		
3	8	2	4		5			3		
					6			∞		
4	16	3		1	7	7	7	5-1	3-1	3-1
5	32	6	6		11?			46?		
6	64	9		2	≥ 13	15	15		39	39-3
7	128	18	18							
8	256	30		4			31			?-5
9	512	56	56							
10	1024	99		8						

Table 3 — Symplectic Integrators —

For example, in the line corresponding to order 4, the number of polynomials defining the zeros is respectively 30 polynomials (the sum in the column) with the direct method, 8 by using Lie transformations, 6 by searching for reversible integrators, and 2 by searching for reversible products of S_2. In the same line, we find solutions which provide integrators of order 4 as a product of 7 operators. We find 5 solutions with the Lie transformations (1 real solution), and 3 with the other methods. In the line that corresponds to order 3, there are 3 integrators as the product of 5 operators, but none has a real value, and there is an infinite number of them as the product of 6 operators.

first column, D means *using the direct method (19.35)*, L means *representing the operator by a Lie transformation, or an exponential (19.37), or (19.36)*, RL means *searching for reversible integrators represented by the Lie transformations, using proposition (49))*, S_2 means *searching for products of reversible integrators* S_2. In the column "solutions", whenever there are two integers, the first one is the number of solutions and the second one, the number of realizations .

20.2. Higher Number of Operators

We can generalize the construction of symplectic integrators in the case where h is a finite sum of terms (see for example Suzuki [54]).

In this section, we search for integrators, in the case where $h = A + B + C$, using the previous algorithms. Such integrators may prove useful when we study the perturbed Kepler motion, whose Hamiltonian may be written in the previous form, by using Poincaré's canonical heliocentrical variables (see 21.1).

We give an exhaustive list for orders 2 and 4.

• The integrator of order 1 is obtained as

$$S_1(t) = S_A(t)S_B(t)S_C(t). \tag{20.16}$$

• For the integrator of order 2, from $S_2(t) = S_B(\frac{t}{2})S_C(t)S_B(\frac{t}{2})$, we deduce the integrator

$$S_2(t) = S_A(\tfrac{t}{2})S_B(\tfrac{t}{2})S_C(t)S_B(\tfrac{t}{2})S_A(\tfrac{t}{2}). \tag{20.17}$$

If we search directly for solutions of the form

$$S_2(t) = S_A(c_1t)S_B(d_1t)S_C(e_1t)\cdots S_A(c_nt)S_B(d_nt)S_C(e_nt), \tag{20.18}$$

we find a system of polynomial equations for $n = 3$:

$$\left\{\begin{array}{r} c_3 + c_2 + c_1 - 1 = 0 \\ d_2 + d_1 - 1 = 0 \\ e_2 + e_1 - 1 = 0 \\ (c_3 - c_2 - c_1)\ d_2 + (c_3 + c_2 - c_1)\ d_1 = 0 \\ (c_3 + c_2 - c_1)\ e_2 + (c_3 - c_2 - c_1)\ e_1 = 0 \\ (d_2 - d_1)\ e_2 + (-d_2 - d_1)\ e_1 = 0 \end{array}\right. . \tag{20.19}$$

The previous system is reduced to a standard basis

$$\left\{\begin{array}{rcl} e_2 + e_1 - 1 &=& 0 \\ (d_1 - 1)\ e_1 - d_1 + \frac{1}{2} &=& 0 \\ c_2\ e_1 + c_1 - \frac{1}{2} &=& 0 \\ (c_1 - \frac{1}{2})\ e_1 - \frac{1}{2}\ c_2 - c_1 + \frac{1}{2} &=& 0 \\ d_2 + d_1 - 1 &=& 0 \\ c_2\ d_1 - c_2 - c_1 + \frac{1}{2} &=& 0 \\ (c_1 - \frac{1}{2})\ d_1 + \frac{1}{2}\ c_2 &=& 0 \\ c_3 + c_2 + c_1 - 1 &=& 0 \\ c_2^2 + (2\,c_1 - 1)\ c_2 + 2\ c_1^2 - 2\,c_1 + \frac{1}{2} &=& 0 \end{array}\right. . \tag{20.20}$$

The last polynomial equation is written as a sum of 2 squares

$$(c_2 + c_1 - \tfrac{1}{2})^2 + (c_1 - \tfrac{1}{2})^2. \tag{20.21}$$

So the real solutions are given by

$$\left\{\begin{array}{rcl} c_1 = c_3 &=& \frac{1}{2} \\ c_2 &=& 0 \\ d_1 + d_2 &=& 1 \\ e_1 + e_2 &=& 1 \\ (d_1 - 1)e_2 + \frac{1}{2} &=& 0 \end{array}\right. . \tag{20.22}$$

In the case where we search for the least possible number of operators, we again come up with the previous integrator.

• For the integrator of order 4, we could, by searching for reversible integrators, apply the same technique, namely, if

$$S_4(t) = S_A(d_2t)S_B(c_1t)S_A(d_1t)S_B(c_0t)S_A(d_1t)S_B(c_1t)S_A(d_2t) \tag{20.23}$$

where

$$d_2 = \tfrac{1}{2(\sqrt[3]{2}-1)},\ d_1 = \tfrac{1}{2}\tfrac{\sqrt[3]{2}-2}{\sqrt[3]{2}-1},\ c_1 = \tfrac{1}{\sqrt[3]{2}-1},\ c_0 = \tfrac{\sqrt[3]{2}-3}{\sqrt[3]{2}-1}, \tag{20.24}$$

is the integrator of order 4 (20.5), then

$$S_C(d_2t)S_4(c_1t)S_C(d_1t)S_4(c_0t)S_C(d_1t)S_4(c_1t)S_C(d_2t) \tag{20.25}$$

is an integrator of order 4 for $h = A + B + C$, which shows up as the product of 25 operators of order 1: S_A, S_B or S_C.

However, if we use lemma 50, we deduce that an operator of order 4 can be written as the product of 3 reversible integrators of order 2, therefore, as the product of 13 operators. Specifically, we find

$$S_A(\tfrac{a}{2}t)S_B(\tfrac{a}{2}t)S_C(at)S_B(\tfrac{a}{2}t)S_A(\tfrac{a+b}{2}t)S_B(\tfrac{b}{2}t)S_C(bt)$$
$$S_B(\tfrac{b}{2}t)S_A(\tfrac{a+b}{2}t)S_B(\tfrac{a}{2}t)S_C(at)S_B(\tfrac{a}{2}t)S_A(\tfrac{a}{2}t).$$

where

$$a = \tfrac{1}{2-\omega_3}, b = -\tfrac{\omega_3}{2-\omega_3}. \tag{20.26}$$

If we search directly for all the reversible integrators which are written as a 13 term product, we must search for a product of the form

$$S_A(c_nt)S_C(e_nt)S_B(d_nt)\cdots$$
$$S_C(e_1t)S_B(d_1t)S_A(c_0t)S_B(d_1t)S_C(e_1t)\cdots S_B(d_nt)S_C(e_nt)S_A(c_nt).$$

Since the A, B, C, play a symmetrical part, the previously found integrator requires that 9 variables be used (17 terms, *a priori*) $c_0, c_1 = 0, c_2, d_1, d_2 = 0, d_3, e_1, e_2, e_3$. If we search for 9 variables integrators, that is to say $n = 3$, we get a variety of null dimension of degree 12 in which a standard basis is

$$e_1 + \tfrac{4887}{100}\,c_0^{11} - \tfrac{3051}{100}\,c_0^{10} - \tfrac{11277}{100}\,c_0^{9} + \tfrac{15417}{100}\,c_0^{8} - \tfrac{3123}{100}\,c_0^{7} - \tfrac{1707}{20}\,c_0^{6} + \tfrac{1494}{25}\,c_0^{5} + \tfrac{1767}{100}\,c_0^{4}$$
$$- \tfrac{4461}{100}\,c_0^{3} + \tfrac{2823}{100}\,c_0^{2} - \tfrac{903}{100}\,c_0 + \tfrac{151}{150} = 0,$$
$$e_2 - \tfrac{8451}{50}\,c_0^{11} + \tfrac{2349}{25}\,c_0^{10} + \tfrac{19071}{50}\,c_0^{9} - \tfrac{12708}{25}\,c_0^{8} + \tfrac{2727}{25}\,c_0^{7} + \tfrac{2811}{10}\,c_0^{6} - \tfrac{9873}{50}\,c_0^{5} - \tfrac{2691}{50}\,c_0^{4}$$
$$+ \tfrac{3714}{25}\,c_0^{3} - \tfrac{2402}{25}\,c_0^{2} + \tfrac{1519}{50}\,c_0 - \tfrac{323}{75} = 0,$$
$$e_3 + \tfrac{2403}{20}\,c_0^{11} - \tfrac{1269}{20}\,c_0^{10} - \tfrac{5373}{20}\,c_0^{9} + \tfrac{7083}{20}\,c_0^{8} - \tfrac{1557}{20}\,c_0^{7} - \tfrac{783}{4}\,c_0^{6} + \tfrac{1377}{10}\,c_0^{5} + \tfrac{723}{20}\,c_0^{4} - \tfrac{2079}{20}\,c_0^{3}$$
$$+ \tfrac{1357}{20}\,c_0^{2} - \tfrac{427}{20}\,c_0 + \tfrac{14}{5} = 0,$$
$$d_3 + \tfrac{2403}{20}\,c_0^{11} - \tfrac{1269}{20}\,c_0^{10} - \tfrac{5373}{20}\,c_0^{9} + \tfrac{7083}{20}\,c_0^{8} - \tfrac{1557}{20}\,c_0^{7} - \tfrac{783}{4}\,c_0^{6} + \tfrac{1377}{10}\,c_0^{5} + \tfrac{723}{20}\,c_0^{4} - \tfrac{2079}{20}\,c_0^{3}$$
$$+ \tfrac{1357}{20}\,c_0^{2} - \tfrac{427}{20}\,c_0 + \tfrac{14}{5} = 0,$$
$$d_2 - \tfrac{2403}{20}\,c_0^{11} + \tfrac{1269}{20}\,c_0^{10} + \tfrac{5373}{20}\,c_0^{9} - \tfrac{7083}{20}\,c_0^{8} + \tfrac{1557}{20}\,c_0^{7} + \tfrac{783}{4}\,c_0^{6} - \tfrac{1377}{10}\,c_0^{5} - \tfrac{723}{20}\,c_0^{4} + \tfrac{2079}{20}\,c_0^{3}$$
$$- \tfrac{1357}{20}\,c_0^{2} + \tfrac{437}{20}\,c_0 - \tfrac{33}{10} = 0,$$
$$d_1 - \tfrac{1}{2}c_0 = 0,$$
$$c_2 + \tfrac{8451}{50}\,c_0^{11} - \tfrac{2349}{25}\,c_0^{10} - \tfrac{19071}{50}\,c_0^{9} + \tfrac{12708}{25}\,c_0^{8} - \tfrac{2727}{25}\,c_0^{7} - \tfrac{2811}{10}\,c_0^{6} + \tfrac{9873}{50}\,c_0^{5} + \tfrac{2691}{50}\,c_0^{4}$$
$$- \tfrac{3714}{25}\,c_0^{3} + \tfrac{2402}{25}\,c_0^{2} - \tfrac{747}{25}\,c_0 + \tfrac{571}{150} = 0,$$
$$c_1 - \tfrac{8451}{50}\,c_0^{11} + \tfrac{2349}{25}\,c_0^{10} + \tfrac{19071}{50}\,c_0^{9} - \tfrac{12708}{25}\,c_0^{8} + \tfrac{2727}{25}\,c_0^{7} + \tfrac{2811}{10}\,c_0^{6} - \tfrac{9873}{50}\,c_0^{5} - \tfrac{2691}{50}\,c_0^{4}$$
$$+ \tfrac{3714}{25}\,c_0^{3} - \tfrac{2402}{25}\,c_0^{2} + \tfrac{1519}{50}\,c_0 - \tfrac{323}{75} = 0,$$
$$c_0^{12} - c_0^{11} - 2\,c_0^{10} + 4\,c_0^{9} - 2\,c_0^{8} - \tfrac{4}{3}\,c_0^{7} + \tfrac{17}{9}\,c_0^{6} - \tfrac{2}{9}\,c_0^{5} - c_0^{4} + \tfrac{26}{27}\,c_0^{3} - \tfrac{4}{9}\,c_0^{2} + \tfrac{1}{9}\,c_0 - \tfrac{1}{81} = 0.$$

We notice the relations

$$d_3 = e_3, d_1 = \tfrac{1}{2}c_0, c_1 = e_2. \tag{20.27}$$

The last polynomial is factorized into

$$(c_0^9 - 2\,c_0^8 + c_0^7 + \tfrac{2}{3}\,c_0^6 - c_0^5 + \tfrac{2}{3}\,c_0^3 - \tfrac{5}{9}\,c_0^2 + \tfrac{2}{9}\,c_0 - \tfrac{1}{27})(c_0^3 + c_0^2 - c_0 + \tfrac{1}{3}) \tag{20.28}$$

and we get two families of solutions

$$\begin{gathered} c_0 = 2e_1 = 2d_1, \\ c_1 = e_2 = d_2 = +27\,c_0^8 - \tfrac{81}{2}\,c_0^7 + \tfrac{9}{2}\,c_0^6 + \tfrac{45}{2}\,c_0^5 - 15\,c_0^4 - 9\,c_0^3 + \tfrac{27}{2}\,c_0^2 - \tfrac{15}{2}\,c_0 + 2, \\ c_2 = d_3 = e_3 = -c_1 - \tfrac{1}{2}c_0 - \tfrac{1}{2}, \\ c_0^9 - 2\,c_0^8 + c_0^7 + \tfrac{2}{3}\,c_0^6 - c_0^5 + \tfrac{2}{3}\,c_0^3 - \tfrac{5}{9}\,c_0^2 + \tfrac{2}{9}\,c_0 = \tfrac{1}{27}. \end{gathered}$$

and the previously known solutions

$$c_1 = e_2 = 0,\ d_2 = d_3 = e_3 = \tfrac{1}{2}c_2 = -\tfrac{1}{4}\,c_0 + \tfrac{1}{4}, d_1 = \tfrac{1}{2}c_0,\ e_1 = \tfrac{1}{4}c_0 + \tfrac{1}{4},\ c_0^3 + c_0^2 - c_0 = -\tfrac{1}{3}.$$

The first family provides integrators as a 17 term product, while the last family provides integrators of order 4 as a 13 term products. These integrators are the product of 4 operators S_A, of 6 operators S_B, and 3 operators S_C. The Hamiltonian with the least costly computation of the mapping of the flow should be used for B.

21. Special Kinetic Energy

In the preceding discussion, we considered h as a Hamiltonian, sum of A and B, without taking into account the special form of A and B. In numerous cases, if $h = T(p) + V(q)$, the kinetic energy is a polynomial of degree 2 in p. From this we deduce that $\{T, V\}$ is a polynomial of degree 1 in p, and that $\{\{T, V\}, V\} = V_1(q)$ only depends on q, and finally, that

$$\{\{\{T, V\}, V\}, V\} = 0. \tag{21.1}$$

If we take (21.1) into consideration, we do not find any integrators (at least for the first orders), as the product of fewer operators S_T or S_V.

Here, instead of $S_V(t)$, let us consider the operator

$$S_{\alpha,\beta}(t) = e^{-t(\alpha V + t^2\beta\{\{T,V\},V\})}. \tag{21.2}$$

We have, as before

$$S_{\alpha,\beta}(t)p = p - t\alpha\frac{\partial V}{\partial q} - t^3\beta\frac{\partial}{\partial q}\{\{T, V\}, V\} \quad \text{and} \quad S_{\alpha,\beta}(t)q = q. \tag{21.3}$$

Let us search for integrators of the form

$$S^{(n)}(t) = S_{c_n,z_n}(t)S_T(d_nt)\cdots S_{c_1,z_1}(t)S_T(d_0t)S_{c_1,z_1}(t)\cdots S_T(d_nt)S_{c_n,z_n}(t) \tag{21.4}$$

or, –which amounts to the same thing– of the form

$$S^{(n)}(t) = S_{c_n,z_n}(t)\cdots S_T(d_1t)S_{c_0,z_0}(t)S_T(d_1t)\cdots S_{c_n,z_n}(t). \tag{21.5}$$

• For $n = 1$ and $k = 4$, we find the relations

$$d_0 + 2d_1 = 1,\ 2c_1 = 1,\ (\tfrac{1}{2}d_0^2 - d_1 d_0 - d_1^2)c_1 = 0,\ 2z_1 + (-\tfrac{1}{2}d_0 + 2d_1)c_1^2 = 0, \tag{21.6}$$

which admit as a solution

$$z_1 + \tfrac{3}{8}d_1 = \tfrac{1}{16},\ c_1 = \tfrac{1}{2},\ d_0 + 2d_1 = 1,\ d_1^2 - d_1 = -\tfrac{1}{6}, \tag{21.7}$$

or

$$2d_1 = 1,\ c_0 + 2c_1 = 1,\ (-\tfrac{1}{2}c_0 + 2\ c_1)d_1^2,\ 2z_1 + z_0 + (\tfrac{1}{2}c_0^2 - c_1 c_0 - c_1^2)d_1, \tag{21.8}$$

which admit as a general solution

$$z_1 + \tfrac{1}{2}z_0 + \tfrac{1}{48} = 0, d_1 - \tfrac{1}{2} = 0, c_0 - \tfrac{2}{3} = 0, c_1 - \tfrac{1}{6} = 0. \tag{21.9}$$

We can now take the second solution

$$z_1 = 0,\ z_0 = -\tfrac{1}{24},\ d_1 = \tfrac{1}{2},\ c_0 = \tfrac{2}{3},\ c_1 = \tfrac{1}{6}. \tag{21.10}$$

S_4 is the product of an operator $S_{\alpha,\beta}$, of two operators S_T, of two operators S_V, totaling 5 operators. At each step, 6 evaluations must be implemented, which represents some progress with respect to the 7 evaluations found in the general case.

• Likewise, for $k = 6$, we find, with $n = 2$:

$$z_2 = 0, z_1 - \tfrac{15}{16}c_2^2 + \tfrac{1}{4}\ c_2 - \tfrac{1}{96} = 0, z_0 - \tfrac{3}{8}c_2^2 - \tfrac{1}{4}\ c_2 + \tfrac{1}{48} = 0, d_2 + 3c_2 - \tfrac{1}{2} = 0,$$
$$d_1 - 3c_2 = 0, c_0 - 30c_2^2 + 12c_2 - 1 = 0, c_1 + 15c_2^2 - 5c_2 = 0, c_2^3 - \tfrac{1}{2}c_2^2 + \tfrac{1}{18}c_2 - \tfrac{1}{540} = 0.$$

The integrator of order 6 is a product of 9 operators (instead of 15), and each step is computed after 12 evaluations.

***

If $T(p) = \sum_{i,j=1}^n t_{i,j}p_i p_j$, we have $\{T, V\} = \sum_{i,j=1}^n t_{i,j}p_i \frac{\partial V}{\partial q_j} + p_j \frac{\partial V}{\partial q_i}$ and

$$\{\{T,V\},V\} = 2\sum_{i,j=1}^{n} t_{i,j}\frac{\partial V}{\partial q_j}\frac{\partial V}{\partial q_i} = 2T(\nabla V). \tag{21.11}$$

21.1. Integrator for the Problem of the Perturbed Kepler Motion

Symplectic integrators can demonstrate their efficiency in the perturbed Kepler problem (cf. for example, Wisdom [59], Channel & Scovel [9, 10]).

Please refer to [37] for more details on what follows. We focus on the problem of $n+1$-body. Denoting G the barycenter of the $n+1$ bodies, u_i the position vectors with respect to G, and $\Delta_{i,j} = \|u_i - u_j\|$, the Hamiltonian of the system is classically written as

$$H = T + U = \tfrac{1}{2}\sum_{i=0}^{n} m_i \|\dot{u}_i\|^2 - G\sum_{0\le i<j\le n}\tfrac{m_i m_j}{\Delta_{i,j}}. \tag{21.12}$$

Calling $\tilde{u}_i = m_i \dot{u}_i$, the conjugate momentum of u_i, we get, in the canonical coordinates

$$T = \tfrac{1}{2}\sum_{i=0}^{n}\frac{\|\tilde{u}_i\|}{m_i}. \tag{21.13}$$

The idea is to use the canonical heliocentrical variables: $r_0 = u_0, r_i = u_i - u_0$. In order for the transformation to be canonical, we pose

$$\tilde{r}_0 = \sum_{i=0}^{n} u_i = 0, \tilde{r}_i = \tilde{u}_i, \quad 1 \leq i \leq n \tag{21.14}$$

which is

$$\tilde{u}_0 = -\sum_{i=1}^{n} \tilde{r}_i, \tilde{u}_i = \tilde{r}_i, \quad 1 \leq i \leq n. \tag{21.15}$$

So, in the heliocentrical coordinates, we have

$$T = \tfrac{1}{2}\sum_{i=1}^{n} \frac{\|\tilde{r}_i\|^2}{m_i} + \tfrac{1}{2}\frac{\|\sum_{i=1}^{n} \tilde{r}_i\|^2}{m_0} \tag{21.16}$$

$$= \tfrac{1}{2}\sum_{i=1}^{n} \|\tilde{r}_i\|^2 \left[\tfrac{1}{m_i} + \tfrac{1}{m_0}\right] + \sum_{0<i<j} \frac{\tilde{r}_i.\tilde{r}_j}{m_0} \tag{21.17}$$

and

$$U = -G\sum_{i=1}^{n} \frac{m_0 m_i}{r_i} - G \sum_{0<i<j\leq n} \frac{m_i m_j}{\Delta_{i,j}}. \tag{21.18}$$

So we can write $H = H_0 + H_1$ with $H_0 = T_0 + U_0$, $H_1 = T_1 + U_1$ where H_0 appears as a sum of two-body problems, and H_1, as an interaction perturbation. We will have

$$T_0 = \tfrac{1}{2}\sum_{i=1}^{n} \|\tilde{r}_i\|^2 \left[\tfrac{1}{m_i} + \tfrac{1}{m_0}\right], U_0 = -G\sum_{i=1}^{n} \frac{m_0 m_i}{r_i} \tag{21.19}$$

$$T_1 = \sum_{0<i<j} \frac{\tilde{r}_i.\tilde{r}_j}{m_0}, U_1 = -G \sum_{0<i<j\leq n} \frac{m_i m_j}{\Delta_{i,j}}. \tag{21.20}$$

H_1 has a particularly simple form, since the kinetic energy and the potential energy are decoupled. H_0 can be interpreted as n Hamiltonians of Keplerian motion of bodies of mass $\frac{m_0 m_i}{m_0 + m_i}$ around a stationary center of mass $m_0 + m_i$.

By writing $H = H_0 + T_1 + U_1$, where T_1 has a form which is particularly easy to integrate, we can attempt to integrate H numerically, by using the integrator of order 4, where S_{T_1} is used 6 times, S_{U_1}, 4 times, and S_{H_0}, the Hamiltonian of n two-body Keplerian problems, 3 times.

The interpretation is the following: the n bodies gravitate around the central body. S_{H_0} represents the gravitation of the n bodies, independently around the central body. S_{T_1} amounts to modifying the velocities, and S_{U_1}, surreptitiously, the positions, as if it were a matter of putting them back on their correct trajectories.

°°°

In this section, we finally gave an exhaustive list of the symplectic integrators all the way to order 4. Some were previously known, except for 2 integrators of order 4 with a complex value, and two integrators of order 3. At order 5, there seem to be non reversible solutions (46) obtained as a product of 11 operators. There are no integrators of order 6 with a product of fewer than 13 terms. All the reversible integrators were found. There are exactly 39, which are written as the product of 15 operators. Three are real, and had been known by Yoshida ([61]).

We showed that the various methods of searching for integrators were algebraically equivalent.

In the case where the Hamiltonian is the sum of a kinetic energy of degree 2, and of a potential energy, we improved on the previous results.

We gave an exhaustive list of the reversible integrators when the Hamiltonian is written as the sum of 3 terms, which is the case for the Hamiltonian in the problem of the perturbed Kepler motion.

Chapter VI
Implementations

In this section, we describe the various implementations of the algorithms quoted in the previous sections. The formal computation program used is AXIOM (NAG), or Scratchpad (an AXIOM prototype). Some algorithms were carried out in Scratchpad in 1989, and have not been transferred into AXIOM as yet.

First of all, we describe the implementations of the algorithms that are specific to free Lie algebras, that is to say, primarily the decompositions onto the bases. In fact, we propose two representations for the magma elements, the tree representation, or a representation by list, which enables us to consider the order of the Hall basis as a lexicographical order.

Next we mention the implementations carried out to write the relations between the generators of the various Lie transformations. It is interesting to note that these issues have long interested optical physicists, namely Dragt's team. In fact, a software program (MARYLIE) –which computes exponential identities within the framework of the composition of the Dragt-Finn transformations– is already in place for optical applications.

Next we shall describe a set of functions implemented to compute symplectic integrators.

Last, we mention various packages and domains created to compute normal forms in specific cases, in particular when the perturbed Hamiltonian is written as a series of polynomials, or in the action-angles coordinates. Some non-autonomous cases can also be handled. In order to do so, a domain of the Poisson series with one or several variables was needed.

22. AXIOM IN SHORT

Just one word to briefly introduce the AXIOM formal computation system. We suggest that the reader read the official documentation ([30]), the first section of M. Petitot's thesis [45], or D. Pinchon's article [46]. AXIOM is the commercial version –distributed by NAG– of the formal computation prototype system Scratchpad II. Scratchpad II was developed as early as the 70's, at the Yorktown Heights IBM Research Center.

AXIOM is a strongly typed formal computation system. The objects handled by the system all have a name, a value, and a type. Objects of a same type all belong to the same domain. The domains are constructed from different categories.

A category is defined by a collection of operations given by their signature –a series of properties (or attributes)–, and it is eventually parametrized by other categories. At this point, it is possible to do a semantic control of the types used.

For example, category *Ring* of the rings, consists of a set and two operations, –addition and multiplication. The multiplication is not necessarily commutative, but category *CommutativeRing* includes this attribute. The category of square matrices of constant size, is parametrized by the ring category –which corresponds to the basic ring–, and by a *NonNegativeInteger* integer. This category has the basic properties of a unitary ring, and a mapping *determinant* when the basic ring is commutative. With the new version of AXIOM, it is possible to implement certain functions by default. In this case, the implementation does not predict which representation will be used in the domains.

A domain defined as a category *Ring* domain (all domains belong to a category) therefore has an addition and a multiplication, and all the properties of the category. If the category had been *CommutativeRing*, the domain would have been commutative. It is however possible to add specific operations to every domain. The inheritance (the part which is dependent on the category), as well as the operations and specific attributes form what we commonly call the public part of the domain. The private part of the domain is essentially the implementation of the algorithms, once the internal representation of the objects has been chosen. The domains themselves can be parametrized by other domains.

So AXIOM offers a library of domains and categories directly accessible by the user. In addition, a number of packages contain functions which have not been implemented from the point of view of domain definition, because they are either too specific, or they involve several domains. For example, the package *COMBINAT* contains a number of functions used in integer combinatorial. Although not officially included in the currently available version, the compiler may also be used, – which is what we have done– in order to create one's own categories, domains, and packages.

23. LIE MONOMIALS

In view of the implementation, in AXIOM, of a domain of the Lie algebras, one needs to consider several aspects. The notion of inheritance in AXIOM makes it possible to implement only the algorithms that are typical of the domain considered, or that strongly depend on the structure of the data used.

In order to implement a domain of Lie algebras, it is necessary to first consider the fact that a Lie algebra on an alphabet X, with coefficients in a ring R, is first and foremost an algebra on R, and a free module on R, with support in the magma $M(X)$. These notions do not depend on the basis chosen to represent the Lie polynomials. We shall first implement the category of free Lie algebras.

One problem that immediately arises, is to know whether every free Lie algebra of the defined category is a module with support in every one of the possible bases, which are subsets of the magma. It seems clear that, first and foremost, it is necessary to implement the magma domain which will be parametrized by the alphabet X.

Should one define the Lyndon or Hall bases as subsets of the magma? We could define a category of the submagmas whose domains could be the magma $M(X)$, $\mathcal{L}$, $\mathcal{H}$, etc..., with every one of these domains equipped with the needed elementary operations.

23.1. Magma

Independently of any representation of the data, the elementary operations to consider on the magma are:

— comparison operations, and the definition of an order relation,

— computation of the length, and unparenthesizing operation,

— finding left and right factors,

— an evaluation or substitution morphism; for example, the substitution in $[[a,b],[b,c]]$ of $\{[a,b],c\}$ by $\{[[a,b],c],[a,b]\}$ will be

$$[[[a,b],c],[b,[a,b]]].$$

The three latter operations are specific to the magma. On the other hand, the order relations are specific to the bases considered. The lexicographical order is not a total order relation on the magma ($[a,[b,c]]$, and $[[a,b],c]$ are not comparable), contrary to the order relation used for the Hall basis.

Furthermore, unlike the Hall and Lyndon bases, $M(X)$ has a monoid structure, since the parenthesized product of two elements of the basis does not belong to the basis.

Therefore, it appeared simpler to define the magma $M(X)$, as a domain that is an ordered set, and also contains X, with the following basic operations:

```
MG SE: ORDSET  is a domain constructor
Abbreviation for Magma is MG
This constructor is exposed in this frame.
Issue )edit magma.spad to see algebra source code for MG

------------------------------- Operations--------------------------
?*? : ($,$) -> $                        ?<? : ($,$) -> BOOLEAN
?=? : ($,$) -> BOOLEAN                  ad : $ -> ($ -> $)
coerce : $ -> LIST SE                   coerce : SE -> $
coerce : $ -> OUTFORM                   eval : ($,LIST SE,LIST $)-> $
eval : ($,LIST $,LIST $) -> $           hall : ($,$) -> BOOLEAN
left : $ -> $                           length : $ -> INT
lexico : ($,$) -> BOOLEAN               lexicoInv : ($,$) -> BOOLEAN
lvar : $ -> LIST SE                     max : ($,$) -> $
min : ($,$) -> $                        retract : $ -> SE
right : $ -> $                          signature : $ -> LIST SE
signatureOrder : ($,$) -> BOOLEAN
retractIfCan : $ -> Union(SE,"failed")
```

Part of these operations are inherited from the category `OrderedSet`

```
OrderedSet is a category constructor.
Abbreviation for OrderedSet is ORDSET

------------------------------- Operations------------------------

?<? : ($,$) -> BOOLEAN                  ?=? : ($,$) -> BOOLEAN
coerce : $ -> OUTFORM                   max : ($,$) -> $
min : ($,$) -> $
```

and from the category `RetractableTo`

```
 RETRACT S: OBJECT  is a category constructor
 Abbreviation for RetractableTo is RETRACT

------------------------------- Operations------------------------
 coerce : S -> $                        retract : $ -> S
 retractIfCan : $ -> Union(S,"failed")
```

It might prove interesting to enumerate the elements of $\mathcal{H}$ or $\mathcal{L}$ of bounded length, but we are not interested in this information from the point of view of Lie polynomial manipulation.

23.2. Monomial Representation by binary trees

The classic representation of a Lie monomial, –that is to say a parenthesized word– is the binary tree structure. This structure allows us to carry out the usual operations in a straightforward manner, such as the product, or the search for the left or right factor. Where the computation of the length, the Hall order, and the lexicographical order on the unparenthesized word are concerned, a recursive definition is easy to implement –which is what was done in the first version.

Having defined the Magma $M(X)$ as an ordered set containing the alphabet `SE`, by the command

```
)abbrev domain LMS LieMonomSorted
```

++ Here we define the abbreviation `LMS` for the domain

```
LieMonomSorted(SE:OrderedSet): Export == Implement where
      LSE ==> List SE
      I ==> Integer
      O ==> OutputForm
      B ==> Boolean
      NNI ==> NonNegativeInteger
```

++ We declare `LieMonomSorted` as dependent on an ordered set `SE`. We introduce a number of abbreviations. The first line expresses the fact that the category on which the domain depends is defined by `Export`, and that the function declarations pertaining to the domain (local) will be declared and implemented in `Implement`.

```
      Export == Join(OrderedSet,RetractableTo SE) with
```

++ We have defined `Export`. The domain is all at once an ordered set, containing `SE`, and possessing a coercion into `SE`.

```
        lvar    : $ -> LSE
        "*"     : ($,$) -> $
        lex     : ($,$) -> B
        lexinv  : ($,$) -> B
        hall    : ($,$) -> B
        length  : $ -> I
        coerce  : SE -> $
        coerce  : $ -> LSE
        left    : $ -> $
        right   : $ -> $
        signature : $ -> LSE
        eval    : ($,List $,List $) -> $
      Implement == add
```

++ We declare the functions and their signatures (the dollar sign designates the domain itself). Then, the choice of the data representation is made by the command

```
RC :=Record(left:$,right:$)
Rep:=Union(SE,RC)
```

++ which indicates that an element of the domain is either an element of the alphabet, or a two-field recording (`left` and `right`), which are elements of the domain. `Rep` is a key-word, meant to designate the internal representation.

At this time, implementation of the algorithms can begin.

— `length` returns the length of a monomial, and is defined straightforwardly as follows.

```
length(x) ==
   x case RE => 1$I
   x case RC => length(x.left) + length(x.right)
```

— `left` and `right` return, whenever possible, the right and left factor, respectively, or a error message. For example, `right` was implemented as follows.

```
left(x) ==
  x case RC => x.left
  x case SE => error "is of length  1"
```

— `x*y` returns the product of x and y, and its straightforward implementation is as follows.

```
x*y == [x,y]$RC
```

— `coerce(x):LSE` sends back the corresponding unparenthesized word, in the form of a leaf list. The recursive definition is immediate.

```
coerce(x):LSE ==
   x case SE => construct(x)$LSE
   x case RC => append(coerce(left x),coerce(right x))
```

— From the unparenthesizing function, we are able to redefine the lexicographical order (AXIOM's –on lists– does not work out). We can also exhibit the list of letters involved in the monomial, or the `signature` (multi-degree commutative evaluation), useful in refining the lexicographical order on the Lyndon words, since Lie monomials are related if and only if they have the same signature. This function can be implemented by

```
signature(x) ==
  x case SE => construct(x)$LSE
  x case RC => sort(append(signature left x,signature right x))
```

— `eval(x,ly,lz)` is the evaluation morphism. `eval(x,ly,lz)` sends back x, in which all monomials of ly that are factors of x, are evaluated in the corresponding monomials of lz. This function is implemented recursively, and must assume that there is no ambiguity.

```
eval(x,ly,lz) ==
   i:=position(x,ly)
   i>0 => lz.i
   x case RC => eval(left x,ly,lz) * eval(right x,ly,lz)
   x case SE => x
```

— `x=y` tests the equality of x and y

```
x=y ==
  (x case SE) and (y case SE) => x:SE =$SE y:SE
  (x case RC) and (y case RC) =>
              left(x)=left(y) and right(x)=right(y)
  false
```

— `coerce(x):O` is the printing function, and it is straightforwardly defined recursively

```
coerce(x:$): O ==
  x case SE => x::SE::O
  xx:= x::RC
  bracket([xx.left::O,xx.right::O])$O
```

— `coerce(m:SE)` enables us to inject the letters of X into $M(X)$

```
coerce(m:SE):$ == m::SE::Rep
```

— `x<y` tests whether $x < y$. We have chosen a total order on the monomials. This order must be organized in such a way that the monomials of lesser length be printed first in the writing of a Lie polynomial. For this reason, I chose the inverse Hall order.

```
x<y == hall(y,x)
```

— `hall(x,y)` tests whether $x < y$ for the Hall order. A recursive implementation is easy with

```
hall(x,y) ==
  length x = length y =>
        (x case SE) and (y case SE) => x:SE <$SE y:SE
        (x case RC) and (y case RC) =>
               x.left = y.left => hall(x.right,y.right)
               hall(x.left,y.left)
  length x < length y => true
  false
```

This representation offers the advantage of being more straightforward, and is easier to implement. But, for example, for the function `hall`, a number of redundancies appear, and derecursifying this algorithm did not seem simple to me.

Indeed, let us suppose that x and y have the same length. In order to compute it, we had to compute the respective lengths of the left and right subtrees. In order to determine whether $x < y$, one must then compute again the lengths of the left subtrees of x and y. This had already been done, since we know the total lengths. This process is repeated, as long as the lengths on the left are equal. Since the derecursification of this algorithm did not appear straightforward, it would be advisable to use a different data structure.

With respect to the lexicographical order, we encounter similar drawbacks, since we carry out a comparison on totally unparenthesized words, where all that might be required is to know the first letters.

23.3. List Representation

The other representation, which proves more efficient, is the following. We represent a parenthesized word u by a list $\mathtt{lp}(u)$, which corresponds to the word length enumeration under every knot or leaf.

For example

— a is represented by: `(a)`

— $[a, b]$ is represented by: `(2,a,b)`

— $[a, [b, c]]$ is represented by: `(3,a,2,b,c)`

— $[[a, b], c]$ is represented by: `(3,2,a,b,c)`

— $[[a, b], [[c, d], e]]$ is represented by: `(5,2,a,b,3,2,c,d,e)`

The representation is automatically constructed as follows:

— If $|u| = 1$ then $\mathtt{lp}(u) = \mathtt{(u)}$

— Else $u = [u_1, u_2]$ and $\mathtt{lp}(u) = |u|\,\mathtt{lp}(u_1)\mathtt{lp}(u_2)$

The length of the position list `lp` is the number of leaves and knots on the tree, namely $2n - 1$, if n is the length of the word.

So the position list may be interpreted as the list –for all the knots and leaves on the tree– of the number of leaves under the trees, –knots and leaves are sorted from left to right.

PROPOSITION 51. — *The Hall order is the lexicographical order on the position lists, where $x < n$, for any $x \in X$ and any $1 < n \in \mathbb{N}$.*

Given to compare u and v, two Lie monomials.

- if $|u| = |v| = 1$ then $\mathtt{lp}(u) < \mathtt{lp}(v) \iff u < v$
- if $|u| \neq |v|$ then $\mathtt{lp}(u) < \mathtt{lp}(v) \iff |u| < |v| \iff u < v$
- else, we can pose $u = [u_1, u_2]$ and $v = [v_1, v_2]$. In this case,

$$\mathtt{lp}(u) < \mathtt{lp}(v) \iff \mathtt{lp}(u_1)\mathtt{lp}(u_2) < \mathtt{lp}(v_1)\mathtt{lp}(v_2)$$

  - if $|u_1| \neq |v_1|$ then since $|u_1|$ and $|v_1|$ are the first elements in the lists to compare, we have
$$\mathtt{lp}(u) < \mathtt{lp}(v) \iff |u_1| < |v_1| \iff u < v$$
  - if $|u_1| = |v_1|$ then $\mathtt{lp}(u_1)$ and $\mathtt{lp}(v_1)$ are of same length. With the properties of the lexicographical order, we have

$$\mathtt{lp}(u) < \mathtt{lp}(v) \iff \begin{cases} \mathtt{lp}(u_1) < \mathtt{lp}(v_1) \\ \quad \text{or} \\ \mathtt{lp}(u_1) = \mathtt{lp}(v_1) \quad \text{and} \quad \mathtt{lp}(u_2) < \mathtt{lp}(v_2) \end{cases}$$

In all cases, we do have the articulated proposition, since it is true for monomials of length 1.

Rather than using this single list `lp`, which has the disadvantage of containing elements of different types, we use two lists, `lv` and `lp`, which are, for `lv`, the list of leaves, that is to say, the unparenthesized word, and for the second one, the same list as `lp`, where 1's will have been substituted to letters.

With this representation, the Lyndon order is the lexicographical order on `lv`. The Hall order is only very slightly different, and is not quite the lexicographical order on `lp`, as is apparent in the following example.

Let us call $u = [[a, b], [[c, d], e]]$ and $v = [[a, e], [b, [c, d]]]$. We have $u < v$ for the Hall order, since $|u| = |v|$, and $[a, b] < [a, e]$. If we use the proposed representation, we have:

— `lv`(u) =`(a,b,c,d,e)` and `lp`(u) =`(5,2,1,1,3,2,1,1,1)`

— `lv`(v) =`(a,e,b,c,d)` and `lp`(v) =`(5,2,1,1,3,1,2,1,1)`

therefore `lp`(v) < `lp`(u).

So one test must be added in order to differentiate the `1` in the `lp` lists. This is done by simultaneously running lists `lp` and `lv`.

The Hall order is implemented as follows:

```
hall(x,y) ==
     lpx:=x.lp
     lpy:=y.lp
     lvx:=x.lv
     lvy:=y.lv
     while ^null lpx and ^null lpy and lpx.1=lpy.1 repeat

         if lpx.1=1 then
             if lvx.1=lvy.1 then
                    lvx:=rest lvx
                    lvy:=rest lvy
             else return lvx.1<lvy.1
         lpx:=rest lpx
         lpy:=rest lpy
     null lpy => false
     null lpx => true
     lpx.1<lpy.1 => true
     false
```

23.3.1. Implementation

The representation declaration is done by

```
Rep:=Record(lv:List SE,lp:LP)
```

where `LP` denotes the type `list of integers`.

— `length` is defined by

```
length x == x.lp.1
```

— `left(x)` and `right(x)` are also readily defined, for we know that the first element of `lp` is the length of x, the second one is the length of the left factor, and the length of `lp` is $2|x| - 1$.

```
        left x ==
           [first(x.lv,x.lp.2::NNI),rest
first(x.lp,2*(x.lp.2::NNI))]

        right x ==
           [rest(x.lv,x.lp.2::NNI),rest(x.lp,2*(x.lp.2::NNI))]
```

— `x*y` is computed by concatenation of lists and by appending on top of `lp`, of the length of $[x, y]$, that is to say, the sum of the first elements of `lp(x)` and `lp(y)`.

```
        x*y ==
[append(x.lv,y.lv),concat(x.lp.1+y.lp.1,append(x.lp,y.lp))]
```

— `x<y` is the lexicographical order on `lp`, then the lexicographical order on `lv`. It is a total order

```
        x<y ==
             lpx:=x.lp
             lpy:=y.lp
             lvx:=x.lv
             lvy:=y.lv
             while ^null lpx and ^null lpy and lpx.1=lpy.1 repeat

                lpx:=rest lpx
                lpy:=rest lpy
             null lpx and null lpy => lexico(y,x)
             null lpy => true
             null lpx => false
             lpy.1<lpx.1 => true
             false
```

— `x=y` tests the equality of x and y. We compare the list of leaves (words), and the list of positions (structure)

```
        x=y == x.lv=y.lv and x.lp=y.lp
```

— `coerce` transforms a letter into a word

```
        coerce(v:SE):$ == [[v],[1]]
```

— `coerce` returns the list of leaves, that is to say the unparenthesized word

```
        coerce(x):LSE == x.lv
```

— `lvar x` yields the list of the variables which make up x, which consists of eliminating the doubles in the list of leaves `lv`, and sorting it out.

```
        lvar x == sort(setUnion(x.lv,[]))
```

— `signature x` yields the signature, that is to say the commutative multi-degree. It is a sorting on `lv(x)`

```
        signature x == sort(x.lv)
```

— `retractIfCan` tests whether $x \in X$ and, eventually, returns x as an element of X

```
retractIfCan x ==
   x.lp.1 = 1 => x.lv.1
   "failed"
```

— `retract x` transforms x into an element of X, or fails during execution

```
retract x ==
  if (u:=retractIfCan x) case SE then u
  else error "Not of Length 1"
```

— `lexico(x,y)` tests the lexicographical order on x and y. This order was rewritten, since the order which had been implemented by default on the lists was not suitable

```
lexico(x,y) ==
     lvx:=x.lv
     lvy:=y.lv
     while ^null lvx and ^null lvy and lvx.1=lvy.1 repeat

         lvx:=rest lvx
         lvy:=rest lvy
     null lvy => false
     null lvx => true
     lvx.1<lvy.1 => true
     false
```

— `eval(x,ly,lz)` evaluates the monomial x, by substituting to any factor in ly, the corresponding element of lz. This function was derecursified by using the following procedure. The list `lp(x)` is run from left to right, and for every knot or leaf, we verify whether the tree under this is an element of `ly`. The result is progressively constructed with `lvr` and `lpr`.

```
eval(x:$,ly:List $,lz:List $) ==
   lvr:=[]
   lpr:=[]
   lvx:=x.lv
   lpx:=x.lp
   while ^null lpx repeat
        n:NNI:=lpx.1 pretend NNI
        lvrx:LSE:=first(lvx,n)
        lprx:LP:=first(lpx,(2*n -1) pretend NNI)
        if (i:=position([lvrx,lprx],ly))>0 then
            lvr:=append(lvr,lz.i.lv)
            lvx:=rest(lvx,n)
            lpx:=rest(lpx,(2*n-1) pretend NNI)
            if ^((k:=lz.i.lp.1 - lprx.1)=0) then
               nb1:=0
               lpp:=reverse lpr
               lpr:=[]
               while ^null lpp repeat
                 if lpp.1=1 then
                      nb1:=nb1+1
                      lpr:=concat(lpp.1,lpr)
```

```
                        else
                          if nb1<lpp.1 then lpr:=concat(lpp.1 + k,lpr)
                          else lpr:=concat(lpp.1,lpr)
                        lpp:=rest lpp
                   lpr:=append(lpr,lz.i.lp)
               else
                   if lpx.1=1 then
                         lvr:=concat(lvr,lvx.1)
                         lvx:=rest lvx
                   lpr:=concat(lpr,lpx.1)
                   lpx:=rest lpx
       [lvr,lpr]
```

— `coerce x` prints x. The programming is recursive.

```
coerce(x:$):O ==
   x.lp.1=1 => x.lv.1 :: O
   bracket([coerce left x,coerce right x])
```

This algorithm could have been derecursified, but the benefit is not significant, particularly because `coerce` is only called once for printing purposes.

However, if we consider the list of positions, run from left to right, an opening parenthesis must be added each time we encounter a true knot, and the corresponding letter each time we encounter a leaf. In order to place the closing parentheses, we must compute the number of leaves which have been placed since we encountered the last knot, and add one, as many times as the number of leaves reaches the values of the previously visited knots.

Here, let us describe the algorithm for which we will have opted for a list of positions where the leaves are letters rather than 1's.

`ln:=[]` – list of visited knots
`res:=""` – result
for all elements of `lp` **do**
  if the first element of `lp` is a number p
    add p at the beginning of `ln`
    add `"("` at the end of `res`
  else the first element of `lp` is a letter a
    add a at the end of `res`
    substract 1 from all the elements of `ln`
    add as many `")"` at the end of `res` as there are 0's beginning of `ln`
    take away all the 0's from `ln`
return `res`

23.4. Comparisons

Let us compare the complexity of these algorithms, according to the representation –either tree or list– used.

— **length**

— *Tree Representation*
In order to compute the length of a tree of length n, one has to
Test whether or not the root is a leaf.
Compute the lengths of the left and right subtrees, and add them up.
The number of tests $t(n)$ is therefore defined by $t(n) = 1 + t(p) + t(q)$ for $p + q = n$ and $t(1) = 1$. The number of additions is defined by $a(1) = 0$ and $a(n) = 1 + a(p) + a(q)$. So we have, for any n, $t(n) = 2n - 1$ and $a(n) = n - 1$.

— *List Representation*
All that is needed is to read the first element of the list of positions, which costs one test.

Regardless of the size reached by the stack while computing the length of a tree, the difference speaks for itself.

— **Lexicographical Order**

— *Tree Representation*
As the order has been implemented, we initiate the process by searching for all the leaves on the trees, then we carry out a lexicographical comparison on the lists we have thus obtained. As in the previous procedure, this computation costs $2n-1$ tests of leaves, and $2n$ concatenations of lists. In this case, the cost is higher by these lengths.

It is conceivable to search for the leaves progressively from left to right until we can be determine which of the two trees is the smallest. But in the worst case scenario, the search for the first leaf may cost $\frac{n}{2}$ tests, if all the right subtrees of the successive left subtrees are of length 1.

— *List Representation*
It amounts to the cost of the lexicographical order on both lists.

— **Hall Order**

— *Tree Representation*
If the two trees are not of the same length, the cost $h(m, n)$ is that of the computation of the length of both trees, or $2(n + m - 1)$ tests of leaves and $n + m - 2$ additions, if n and m are the respective lengths.
If the lengths are equal, the left subtrees –and possibly the right subtrees– must be compared. The cost becomes, for the leaf tests, $h(n, m) = t(n) + t(m) + h(p_n, p_m) + h(q_n, q_m)$, for the computation of the lengths, and the Hall order on the subtrees. In the worst case, (the equality case, for example), recursively, $h(n, n) = 2t(n) + \sup_{p+q=n}(h(p, p) + h(q, q)) \geq 2t(n) + h(n-1, n-1) + h(1)$. But $h(1) = 2$ (the lengths must be computed on two leaves), therefore, $h(n, n) \geq h(n-1, n-1) + 4n = 2n^2 + 2n - 2$. As for the number of additions resulting from the computation of the lengths, we also find a second-degree polynomial, and we have not computed the number of leaf comparisons which, in all cases, is also involved in the list representation. At each step, the lengths are compared, therefore, $2n-1$ integer comparisons, which corresponds to the number of knots.

— *List Representation*
With this representation, the Hall order is the lexicographical order on the list

of positions and leaves. In the case of the equality, there are $2n-1$ integer comparisons, and $2(2n-1)$ equality tests to 1 (which correspond to the leaf tests), and the comparison tests on the leaves (n). There is not a single addition.

Aside from the additions, there is a factor n on the number of leaf tests in favor of list representation.

— **Equality**

— *Tree Representation*
For each tree, we test whether the root is a leaf, then we compare the left subtrees and the right subtrees, which yields a linear cost in function of the length.

— List Representation
We test on the equality of 2 lists, which costs, –depending on list size– $3n-1$ tests, at most, if the words are equal, and of size n.

List representation enables us to immediately detect any difference in the lengths, whereas for tree representation, the whole tree must be run.

24. Lie Polynomials

Having available a magma depending on an alphabet X, which is an ordered set providing a few basic operations, we can define the category of the free Lie algebras as an algebra on R, a free module on R, with support in M. The elementary operations –aside from those that result from the inheritance of the algebraic and free module structures are:

— Access to the coefficients, to the various Lie monomials,

— access to the letters of the alphabet involved

— an evaluation morphism,

— whenever possible, factorization by a Lie monomial

— an injection (coercion) of $M(X)$ into $\mathfrak{L}$.

Most of these basic operations can be implemented by default, at the category level, since they do not depend on either the representation of the chosen data, or the chosen bases. However, since the free module category is not defined with Axiom, I have opted to define L as a non-associative algebra on R, which enables me to retrieve the following basic operations:

```
NAALG R: COMRING  is a category constructor
Abbreviation for NonAssociativeAlgebra is NAALG
This constructor is exposed in this frame.
Issue )edit naalgc.spad to see algebra source code for NAALG

------------------------------ Operations--------------------------------
?*? : (R,$) -> $                        ?*? : ($,R) -> $
?*? : ($,$) -> $                        ?*? : (INT,$) -> $
```

```
?*? : (NNI,$) -> $                     ?*? : (PI,$) -> $
?**? : ($,PI) -> $                     ?+? : ($,$) -> $
?-? : ($,$) -> $                       ?-? : ($,$) -> Union($,"failed")
-? : $ -> $                            ?=? : ($,$) -> BOOLEAN
0 : () -> $                            antiCommutator : ($,$) ->$
associator : ($,$,$) -> $              coerce : $ -> OUTFORM
commutator : ($,$) -> $                leftPower : ($,PI) -> $
plenaryPower : ($,PI) -> $             rightPower : ($,PI) -> $
zero? : $ -> BOOLEAN
```

L is also an element of the Indexed Direct Products category, which enables me to inherit the operations specific to the free module structure:

```
IDPC(A: SETCAT,S: ORDSET)  is a category constructor
Abbreviation for IndexedDirectProductCategory is IDPC
This constructor is exposed in this frame.
Issue )edit indexedp.spad to see algebra source code for IDPC

------------------------------ Operations--------------------------
?=? : ($,$) -> BOOLEAN                 coerce : $ -> OUTFORM
leadingCoefficient : $ -> A            leadingSupport : $ -> S
map : ((A -> A),$) -> $                monomial : (A,S) -> $
reductum : $ -> $
```

24.1. Domains of Lie Algebras

Once a basis is chosen, the implementation only amounts to an implementation of the order relation specific to the basis, and the algorithm of multiplication of two monomials. Order relations have been defined at the magma level, that is to say we have opted to represent the elements of the Hall basis and the Lyndon basis in the same fashion.

The underlying natural structure is the free module structure available in AXIOM, but the difficulty arises from the fact that this structure is a domain structure. Therefore, for the internal representation, we have chosen the free module structure on R with support in $M(X)$. The only algorithms to implement –regardless of the data structures chosen for the magma– are the multiplication of two basic monomials.

One of the great strengths of multiplication algorithms, –for the Lyndon basis as well as for the Hall basis– is that there is absolutely no need to know the bases. It might however be interesting to know the bases constructed on a finite subset of letters of the alphabet.

24.2. Category of Free Lie Algebras

First of all, we define the category of free Lie Algebras, and we implement by default the operations that do not depend on either the basis, nor the internal representation chosen.

```
FLA(R: COMRING,SE: ORDSET)  is a category constructor
Abbreviation for FreeLieAlgebra is FLA
This constructor is exposed in this frame.
Issue )edit flac.spad to see algebra source code for FLA

------------------------------ Operations----------------------------
?*? : (MG SE,MG SE) -> $               ?*? : (R,$) -> $
```

```
?*? : ($,R) -> $                      ?*? : ($,$) -> $
?*? : (INT,$) -> $                    ?*? : (NNI,$) -> $
?*? : (PI,$) -> $                     ?**? : ($,PI) -> $
?+? : ($,$) -> $                      ?-? : ($,$) -> $
?-? : ($,$) -> Union($,"failed")      -? : $ -> $
?=? : ($,$) -> BOOLEAN                0 : () -> $
antiCommutator : ($,$) -> $           associator : ($,$,$) -> $
coerce : SE -> $                      coerce : MG SE -> $
coerce : $ -> OUTFORM                 commutator : ($,$) -> $
degree : $ -> NNI                     eval : ($,LIST MG SE,LIST$) -> $
inBasis? : MG SE -> BOOLEAN           leadingCoefficient : $ -> R
leadingMonomial : $ -> $              leadingSupport : $ -> MG SE
leftAd : $ -> ($ -> $)                leftDivide : ($,MG SE) -> LIST $
leftPower : ($,PI) -> $               listOfCoefficient : $ -> LIST R
listOfMonomials : $ -> LIST MG SE     lvar : $ -> LIST SE
map : ((R -> R),$) -> $               monomial : (R,MG SE) -> $
plenaryPower : ($,PI) -> $            reductum : $ -> $
rightAd : $ -> ($ -> $)               rightDivide : ($,MG SE) -> LIST $
rightPower : ($,PI) -> $              size : $ -> INT
zero? : $ -> BOOLEAN
basis : (LIST SE,NNI) -> LIST LIST MG SE
boundedBasis : (LIST SE,MG SE,NNI) -> LIST LIST MG SE
boundedWeightedBasis : (LIST LIST SE,MG SE,NNI) -> LIST LIST MG SE
eval : ($,LIST MG SE,LIST MG SE) -> $
mkGenerator? : (MG SE,MG SE) -> BOOLEAN
weightedBasis : (LIST LIST SE,NNI) -> LIST LIST MG SE
```

Note that we have defined a multiplication on the magma, –of value in the Lie algebra– which will only be implemented within each domain. After defining the category of free Lie algebras by

```
)abbrev category FLA FreeLieAlgebra

FreeLieAlgebra(R:CommutativeRing,SE:OrderedSet): Category ==
Join(NonAssociativeAlgebra(R),IndexedDirectProductCategory(R,Magma SE)) with
     .
     .
     .(list of functions and attributes)
     .
   add
```

Algorithm implementation can begin

— `coerce m` injects an element m of $M(X)$ in the free Lie algebra, and it is implemented recursively, since m is not necessarily an element of the basis.

```
coerce(m:Magma SE):$ ==
  retractIfCan m case SE => monomial(1$R,m)
  coerce(left m) * coerce(right m)
```

— `listOfCoefficient`, `listOfMonomials`, `size`, `lvar`, `leadingMonomial` are functions that may be defined on the basis of `leadingSupport`, `leadingCoefficient` and `reductum`.

— x*y is the multiplication. It is defined if there exists a multiplication of two basic monomial in free Lie algebra.

```
x:$ * y:$ ==
  x=0 or y=0 => 0
  lx:=listOfCoefficient x
  mx:=listOfMonomials x
  ly:=listOfCoefficient y
  my:=listOfMonomials y
  res:$:=0
  for u in lx for a in mx repeat
      for v in ly for b in my repeat
          res:=res + (u*v)*(a*b)
  res
```

Here, u*v is the multiplication of two elements of the basic ring, and a*b, the multiplication of two basic elements in algebra. This multiplication is implemented in each one of the domains.

— leftDivide and rightDivide are a left or right pseudo-division by an element of the magma considered as a Lie polynomial. These functions return a dividend and a remainder. They are implemented by default, by testing –for each monomial– whether the left or right factor is the element of the magma that we are looking for. They can eventually by re-implemented in each domain of the category.

```
leftDivide(x:$,m:Magma SE) ==
 x=0 => [0,0]
 mx:=listOfMonomials x
 lx:=listOfCoefficient x
 res1:$:=0
 res2:$:=0
 for u in lx for v in mx repeat
  if v=m then res1:=res1+monomial(lx.1,mx.1)
  else if length v = 1 then res2:=res2+monomial(lx.1,mx.1)
       else if left v = m then res1:=res1+monomial(lx.1,right mx.1)
            else if right v = m then
                    res1:=res1 -monomial(lx.1,left mx.1)
                 else res2:=res2+monomial(lx.1,mx.1)
  mx:=rest mx
  lx:=rest lx
[res1,res2]
```

— leftAd and rightAd denotes the left and right adjoint operators, respectively. We generally denote leftAd by *ad*.

— Finally, we implement a number of evaluation functions. The first one uses the function eval of the magma, and the coercion in Lie algebra. The second one uses a local function ev, and it is defined recursively, since we cannot access the internal data representation.

```
eval(x:$,ly:List Magma SE,lz:List Magma SE):$ ==
    x=0 => x
    +/ [u*(eval(v,ly,lz)::$) for u in listOfCoefficient x
                                      for v in listOfMonomials x]
```

```
ev(m:Magma SE,ly:List Magma SE,lz:List $):$ ==
    m=1 => m::$
    (i:=position(m,ly))>0 => lz.i
    retractIfCan m case SE => m::$
    ev(left m,ly,lz) * ev(right m,ly,lz)

eval(x:$,ly:List Magma SE,lz:List $):$ ==
    x=0 => x
    +/ [u*ev(v,ly,lz) for u in listOfCoefficient x for
                                  v in listOfMonomials x]
```

The rest of the operations is either inherited from the categories which define the free Lie algebras, or implemented within the domains that are the free Lie algebras equipped with various bases. These operations consist, in fact, of nothing more than multiplying two basic monomials, and finding elements of the basis –of given length. The addition of two polynomials, which is inherited from the algebraic structure, has not been implemented, because it depends on the data structure.

24.3. Free Lie Algebra using the Hall Basis

This domain is defined by

```
)abbrev domain HFLA HallFreeLieAlgebra

HallFreeLieAlgebra(R,SE): Export == Implement where
      R:CommutativeRing
      SE:OrderedSet
      LSE ==> List SE
      I ==> Integer
      O ==> OutputForm
      B ==> Boolean
      NNI ==> NonNegativeInteger
      MG ==> Magma(SE)
      FM ==> FreeModule(R,MG)

      Export == FreeLieAlgebra(R,SE)

      Implement == FM add
```

++ This domain is of the `FLA` category and as such, it inherits its operations. For the internal representation, we have used the `FreeModule` domain for any operations related to the module structure. Their implementation is the default implementation in the `FreeModule` domain.

We define the multiplication of two elements of the Hall basis. They are, in fact, *a priori*, elements of the magma, but while the operation is in process, they are necessarily elements of the Hall basis. We use an auxiliary function `mult`, thereby avoiding comparisons when we know that $a < b$.

```
a*b ==
  a=b => 0
  hall(b,a) => -mult(b,a)
  mult(a,b)

mult(a,b):$ ==
```

```
length(b)=1 or ^(hall(a,left b)) => monomial(1$R,a*b)
x:=mult(a,right b)
bb:=left b
+/ [u*mult(bb,v) for u in listOfCoefficient x
                 for v in listOfMonomials x]
    - monomial(1$R,right b)*mult(a,bb)
```

We have redefined the union (`union`) of two sorted lists, since it is incorrectly implemented in AXIOM. This enables us to construct the elements of the basis.

— `basis(lx,n)` constructs a basis with length smaller than n on the alphabet lx.

```
basis(lx,n) ==
  l:List List MG:=[[] for i in 1..n]
  l.1 := sort(hall,[u::MG for u in lx])
  for i in 2..n repeat
    for j in 1..(i quo 2) repeat
      k:=(i-j)::NNI
      for u in l.j repeat
        l.i:=union(l.i,[u*v for v in l.k | mkGenerator?(u,v)])
  l
```

— `weightedBasis` constructs a basis of weight smaller than n, on a weighted alphabet llx. This will be useful at a later stage. llx is a list of lists of letters of X, of increasing weights. `weightedBasis` returns a list of lists of elements, this time sorted by weight rather than length. It is the same algorithm, but elements of excessive weight have been eliminated. The order on the Lie brackets is the same.

```
weightedBasis(llx,n) ==
   m:=min(n,#llx)
   l:List List MG:=[[] for i in 1..n]
   for i in 1..m repeat l.i:=sort(hall,[u::MG for u in llx.i])
   for i in 2..n repeat
     for j in 1..(i-1) repeat
      k:=(i-j)::NNI
      for u in l.j repeat
       l.i:=union(l.i,[u*v for v in l.k | mkGenerator?(u,v)])
   l
```

— `boundedWeightedBasis(llx,n)` and `boundedBasis(llx,n)` compute bases of the free Lie algebra, – weighted or not – in which all the elements have a bounded (commutative multi-degree) signature. The implementations are the same as previously, but one test has been added.

```
boundedWeightedBasis(llx,n) ==
   m:=min(n,#llx)
   l:List List MG:=[[] for i in 1..n]
   for i in 1..m repeat l.i:=sort(hall,[u::MG for u in llx.i])
   for i in 2..n repeat
     for j in 1..(i-1) repeat
      k:=(i-j)::NNI
      for u in l.j repeat
       l.i:=union(l.i,[u*v for v in l.k |
              mkGenerator?(u,v) and signatureOrder(u*v,a)])])
```

```
          l

        boundedBasis(lx,n) ==
          l:List List MG:=[[] for i in 1..n]
          l.1 := sort(hall,[u::MG for u in lx])
          for i in 2..n repeat
            for j in 1..(i quo 2) repeat
              k:=(i-j)::NNI
              for u in l.j repeat
                l.i:=union(l.i,[u*v for v in l.k |
                        mkGenerator?(u,v) and signatureOrder(u*v,a)])])
```

— `mkGenerator?` tests whether the product of two elements is an element of the basis.

```
mkGenerator?(a,b) ==
    retractIfCan b case SE => hall(a,b)
    hall(a,b) and ^hall(a,left b))
```

24.4. Free Lie Algebra using the Lyndon Basis

Strictly the same principle is at work here. The functions `a*b` and `mult` are defined by

```
a*b ==
  a=b => 0
  lexico(b,a) => -mult(b,a)
  mult(a,b)

mult(a,b) ==
  length(a)=1 or ^lexico(right a,b) => monomial(1$R,a*b)
  aa:=left a
  x:=mult(right a,b)
  +/ [u*mult(aa,v) for u in listOfCoefficient x
                   for v in listOfMonomials x]
     + mult(aa,b)*monomial(1$R,right a)
```

The operations to compute the bases are implemented as follows.

```
weightedBasis(llx,n) ==
   m:=min(n,#llx)
   l:List List MG:=[[] for i in 1..n]
   for i in 1..m repeat l.i:=sort(lexico,[u::MG for u in llx.i])
   for i in 2..n repeat
     for j in 1..(i-1) repeat
      k:=(i-j)::NNI
      for u in l.j repeat
       l.i:=union(l.i,[u*v for v in l.k | mkGenerator?(u,v)])
   l

basis(lx,n) ==
  l:List List MG:=[[] for i in 1..n]
  l.1 := sort(lexico,[u::MG for u in lx])
  for i in 2..n repeat
    for j in 1..(i-1) repeat
```

```
        k:=(i-j)::NNI
        for u in l.j repeat
          l.i:=union(l.i,[u*v for v in l.k | mkGenerator?(u,v)])
    l

mkGenerator?(a,b) ==
    retractIfCan a case SE => lexico(a,b)
    lexico(a,b) and ^lexico(right a,b)
```

25. Lie Series

Here, rather than creating a domain of Lie series, we have written a package which manipulates lists of elements of a previously defined free Lie algebra.

In this package, we have basic operations such as scalar addition and multiplication; Derivation and integration are done with respect to a fictitious parameter. The composition of two lists consists of multiplication, taken as in the Lie series. We have also implemented the three transformations and their inverses, and, given a triangular system, –factorized or not– implemented its solution. Finally, considering the developed form of a canonical transformation, –made up of several transformations– we can give it as a Lie transformation, an exponential, or a Dragt-Finn transformation.

```
 LS(R: COMRING,FLA: FLA(R,SYMBOL))  is a package constructor
 Abbreviation for LieSeries is LS
 This constructor is exposed in this frame.
 Issue )edit ls.spad to see algebra source code for LS

------------------------------- Operations--------------------------------
 derivate : LIST FLA -> LIST FLA       integrate : LIST FLA -> LIST FLA
 mult : (FLA,LIST FLA) -> LIST FLA     mult : (R,LIST FLA) -> LIST FLA
 addi : (LIST FLA,LIST FLA) -> LIST FLA
 compose : (LIST FLA,LIST FLA) -> LIST FLA
 exponential : (LIST FLA,LIST FLA) -> LIST FLA
 exponentialGenerators : (NNI,LIST FLA,MG SYMBOL) -> LIST FLA
 exponentialGenerators : (NNI,LIST FLA,MG SYMBOL,LIST FLA) -> LIST FLA
 factoredProductGenerators : (NNI,LIST FLA,MG SYMBOL) -> LIST FLA
 factoredProductGenerators : (NNI,LIST FLA,MG SYMBOL,LIST FLA) -> LIST FLA
 factoredProductTransform : (LIST FLA,LIST FLA) -> LIST FLA
 inverseFactoredProductGenerators : (NNI,LIST FLA,MG SYMBOL) -> LIST FLA
 inverseFactoredProductGenerators : (NNI,LIST FLA,MG SYMBOL,LIST FLA) -> LIST FLA
 inverseFactoredProductTransform : (LIST FLA,LIST FLA) -> LIST FLA
 inverseFactoredProductTransform : (LIST FLA,LIST FLA) -> LIST FLA
 inverseLieGenerators : (NNI,LIST FLA,MG SYMBOL) -> LIST FLA
 inverseLieGenerators : (NNI,LIST FLA,MG SYMBOL,LIST FLA) -> LIST FLA
 inverseLieTransform : (LIST FLA,LIST FLA) -> LIST FLA
 inverseLieTransform : (LIST FLA,LIST FLA) -> LIST FLA
 lieGenerators : (NNI,LIST FLA,MG SYMBOL) -> LIST FLA
 lieGenerators : (NNI,LIST FLA,MG SYMBOL,LIST FLA) -> LIST FLA
 lieTransform : (LIST FLA,LIST FLA) -> LIST FLA
 mkLieMon : (SYMBOL,NNI,NNI) -> LIST MG SYMBOL
 mkLiePol : (SYMBOL,NNI,NNI) -> LIST FLA
 mkSymbol : (SYMBOL,NNI,NNI) -> LIST SYMBOL
 triangularFactorSystemSolve : (LIST MG SYMBOL,LIST FLA,LIST FLA,MG SYMBOL) ->\
LIST FLA
```

```
triangularSystemSolve : (LIST MG SYMBOL,LIST FLA,LIST FLA) -> LIST FLA
```

A Lie series is assumed to be of the form

$$f = \sum_{n \geq 0} t^n f_n$$

where t is a parameter equal to 1. f_n is considered of weight n. We give ourselves two elements, `f` and `g`, in the form `f = (f0,f1,...,fn)` and `g = (g0,g1,...,gm)`.

— `mkSymbol(s,m,n)` creates a list of symbols (taken as in AXIOM) $s_m, \cdots, s_n$, where `s` is a symbol.

— `mkLieMon` and `mkLiePol` create the same elements as before, but the result is a list of elements of the magma or the free Lie algebra.

— `mult` is a multiplication by a scalar or an element of the free Lie algebra. `addi` carries out a sum of two lists term by term. The first element of each list is of degree 0.

```
mult(r:R,x:LF) == [r*u for u in x]

mult(x:FLA,y:LF) == [x*u for u in y]

addi(x,y) ==
   n:=min(#x,#y)
   [x.i + y.i for i in 1..n]
```

— `derivate` and `integrate` differentiates and integrates lists with respect to a fictitious parameter.

```
derivate(x) ==
   #x=0 => []
   concat([i::I * x.((i+1)::NNI) for i in 1..((#x-1)::NNI)],0$FLA)

integrate(x) ==
  #x=0 => []
  concat(0$FLA,[recip(j::I::R)$R pretend R * x.j
                                   for j in 1..((#x-1)::NNI)])
```

— `compose(x,y)` computes $[x, y]$. The length of the result is the length of y.

```
compose(x,y) ==
   #x=0 or #y=0 => []
   n:=#x
   m:=#y
   [+/ [x.((j-i+1)::NNI) * y.i for i in 1..j | j<n+i] for j in 1..m]
```

— `exponential` is the exponential of L_x applied to y.

```
exponential(x,y) ==
   #x=0 or #y=0 => []
   lx:=y
   Exp:LF:=y
   for i in 1..(#y-1) repeat
       r:=recip(i::I::R)$R pretend R
       lx:=mult(r,compose(x,lx))
       Exp:=addi(Exp,lx)
   Exp
```

— `inverseLieTransform(w,f)` computes the inverse Lie transform of f by T_w. `w` and `f` are given in the form of lists of elements of free Lie algebra.

```
inverseLieTransform(w,f) ==
  n:=#f
  m:=#w
  Lie:List LF:=concat(f,[[0$FLA for i in 1..n] for j in 1..n])
  m=1 => f
  for i in 1..n repeat
     for j in 1..i repeat
       r:=recip(j::I::R)$R pretend R
       jj:=(j+1)::NNI
       ii:=(i-j+1)::NNI
       Lie.jj.ii := r*(+/[k * w.((k+1)::NNI) * Lie.((jj-k)::NNI).ii
                                                  for k in 1..j | k<m])
  [+/ [Lie.j.((i-j+1)::NNI) for j in 1..i] for i in 1..n]
```

— `inverseFactoredProductTransform(g,f)` computes the inverse Dragt-Finn transform

```
inverseFactoredProductTransform(g,f) ==
  n:=#f;m:=#g
  F:=[f.i for i in 1..n]
  w:=[0 for i in 1..m]
  for i in 2..m repeat
     w.i:=g.i
     F:=exponential(w,F)
     w.i:=0
  F
```

— `exponentialGenerators(n,l,h)` computes, at the order n, the generator of exponential $\exp(L_x)$, which verifies $\exp(L_x)h = (l_0, \ldots, l_n)$. Here, h is a Lie monomial and l, a truncated Lie series. The technique consists of formally computing $\exp(L_x)h$, and formally solving a triangular system that is factorizable by h. In the second version, we suppose that the first terms of the result (i.e. `kk1`) are already known.

```
exponentialGenerators(n,l,h) ==
  kk:LF:=concat(0,mkLiePol(k::SE,1,n))
  Exp:LF:=[0$FLA for i in 0..n]
  Exp.1:=h::MG::FLA
  lx:LF:=Exp
  r:R:=1$R
  for i in 1..n repeat
     j:=(i+1)::NNI
     lx:=mult(recip(i::I::R)$R pretend R,compose(kk,lx))
     kk.j:=leftDivide(Exp.j+lx.j-l.j,h).1 + kk.j
     ls:List MG:=mkLieMon(k::SE,i,i)
     lx:=append(first(lx,i),[eval(u,ls,[kk.j]) for u in rest(lx,i)])
     Exp:=append(first(Exp,i),[eval(u,ls,[kk.j]) for u in rest(Exp,i)])
     Exp:=addi(Exp,lx)
  kk

exponentialGenerators(n,l,h,kk1) ==
  n<#kk1 => kk1
```

```
      kk:LF:=append(kk1,mkLiePol(k::SE,#kk1,n))
      Exp:LF:=[0$FLA for i in 0..n]
      Exp.1:=h::MG::FLA
      lx:LF:=Exp
      r:R:=1$R
      for i in 1..(#kk1-1)::NNI repeat
         j:=(i+1)::NNI
         lx:=mult(recip(i::I::R)$R pretend R,compose(kk,lx))
         Exp:=addi(Exp,lx)
      for i in (#kk1)..n repeat
         j:=(i+1)::NNI
         lx:=mult(recip(i::I::R)$R pretend R,compose(kk,lx))
         kk.j:=leftDivide(Exp.j+lx.j-l.j,h).1 + kk.j
         ls:List MG:=mkLieMon(k::SE,i,i)
         lx:=append(first(lx,i),[eval(u,ls,[kk.j]) for u in rest(lx,i)])
         Exp:=append(first(Exp,i),[eval(u,ls,[kk.j]) for u in rest(Exp,i)])
         Exp:=addi(Exp,lx)
      kk
```

— `inverseLieGenerators`, `inverseFactoredProductGenerators` compute the same operations as previously shown, but this time, for the Lie transformation or the Dragt-Finn transformation.

```
    inverseLieGenerators(n,l,h) ==
      ww:=[0$FLA for i in 1..n]
      r:R:=1$R
      for i in 1..n repeat
         j:=(i+1)::NNI
         r:=recip(i::I::R)$R pretend R
         ww.i:=leftDivide((+/ [k * r * ww.k * l.((j-k)::NNI)
                                  for k in 1..(i-1)::NNI])-l.j,h).1
      concat(0$FLA,ww)

    inverseLieGenerators(n,l,h,ww1) ==
      n<#ww1 => ww1
      ww:=append(rest(ww1),[0$FLA for i in #ww1..n])
      r:R:=1$R
      for i in ((#ww1-1)::NNI)..n repeat
         j:=(i+1)::NNI
         r:=recip(i::I::R)$R pretend R
         ww.i:=leftDivide((+/ [k * r * ww.k * l.((j-k)::NNI)
                                for k in 1..(i-1)::NNI])-l.j,h).1
      concat(0$FLA,ww)

    inverseFactoredProductGenerators(n,l,h) ==
       gg:LF:=[0$FLA for i in 0..n]
       g:=0$FLA
       Fac:LF:=[0$FLA for i in 0..n]
       Fac.1:=h::MG::FLA
       for i in 1..n repeat
          j:NNI:=(i+1)::NNI
          g:=leftDivide(Fac.j - l.j,h).1
          gg.j:=g
          Fac:=exponential(concat([0$FLA for k in 1..i],g),Fac)
```

```
  gg

inverseFactoredProductGenerators(n,l,h,gg1) ==
  n<#gg1 => gg1
  gg:=append(gg1,[0$FLA for i in (#gg1)..n])
  Fac:LF:=[0$FLA for i in 0..n]
  Fac.1:=h::MG::FLA
  Fac:=inverseFactoredProductTransform(first(gg,#gg1),Fac)
  for i in #gg1..n repeat
     j:NNI:=(i+1)::NNI
     g:=leftDivide(Fac.j - l.j,h).1
     gg.j:=g
     Fac:=exponential(concat([0$FLA for k in 1..i],g),Fac)
  gg
```

— `triangularSystemSolve(l,lf,lg)` solves a system of equations `lf=lg` with respect to the list of monomials `l`. `lf` and `lg` are lists of elements of free algebra. The system must be triangular, otherwise an error occurs during execution. The basic procedure is the following: for each line `lf.i-lg.i`, we locate the monomial with support `l.i`: a`l.i`, and the solution is `l.i` $= \frac{1}{a_i}$`lf.i-lg.i`$-a$`l.i`. Next, we evaluate the following lines with the new value of `l.i`.

```
triangularSystemSolve(l,lf,lg) ==
   n:NNI:=min(#l,min(#lf,#lg))
   lgg:LF:=[lg.i - lf.i for i in 1..n]
   lm:List MG:=listOfMonomials(lgg.1)
   lc:List R:=listOfCoefficient(lgg.1)
   j:NNI:=position((l.1)::MG,lm)::NNI
   lgg.1:=(recip(lc.j)$R pretend R) * (lc.j * ((lm.j)::FLA) - lgg.1)
   for i in 2..n repeat
      k:=(i-1)::NNI
      lgg.i:=eval(lgg.i,first(l,k),first(lgg,k))
      lm:List MG:=listOfMonomials(lgg.i)
      lc:List R:=listOfCoefficient(lgg.i)
      j:NNI:=position((l.i)::MG,lm)::NNI
      lgg.i:=(recip(lc.j)$R pretend R) * (lc.j * ((lm.j)::FLA) - lgg.i)
   lgg
```

— `triangularFactorSystemSolve(l,lf,lg,s)` solves the same system after inner derivation by a. The process is the same as before, but `s` will have been extracted at every step.

```
triangularFactorSystemSolve(l,lf,lg,s) ==
   n:=min(#l,min(#lf,#lg))
   lgg:LF:=[lg.i - lf.i for i in 1..n]
   lgg.1:=leftDivide(lgg.1,s).1
   lm:List MG:=listOfMonomials(lgg.1)
   lc:List R:=listOfCoefficient(lgg.1)
   j:NNI:=position((l.1)::MG,lm)::NNI
   lgg.1:=(recip(lc.j)$R pretend R) * (lc.j * ((lm.j)::FLA) - lgg.1)
   for i in 2..n repeat
      k:NNI:=(i-1)::NNI
      lgg.i:=leftDivide(eval(lgg.i,first(l,k),first(lgg,k)),s).1
      lm:=listOfMonomials(lgg.i)
```

```
        lc:=listOfCoefficient(lgg.i)
        j:=position((l.i)::MG,lm)::NNI
        lgg.i:=(recip(lc.j)$R pretend R)*(lc.j * ((lm.j)::FLA)-lgg.i)
    lgg
```

Remark. — s is assumed smaller than all the letters of the alphabet. Consequently we must verify that this is true. In a future version, the first step will be to ascertain that s is indeed the smallest letter of the alphabet.

Samples of the obtained will be given in the appendix.

25.1. Composition of Transformations

Here, let us show how we have written –in a directory read by AXIOM– the few functions that enable us to compose Lie transformations, in order to construct symplectic integrators.

First of all, we give ourselves the basic ring which is the ring of polynomials with coefficients in $\mathbb{Q}$. We construct the free Lie algebra –using the Lyndon basis–, and we load the package with the corresponding Lie series.

```
R:=POLY FRAC INT
L:=LFLA(Symbol,R)
LL:=LieSeries(R,L)
```

We give ourselves a list of variables –which are in fact scalars in R.

```
lc:List R:=[char(i)::ST::SE::R for i in 65..90]
ld:List R:=[char(i)::ST::SE::R for i in 97..122]
le:=reverse ld
```

Then we implement the functions which will construct the compositions of Lie transformations. Starting from $S_A(t) = \exp(tL_A)$ and $S_B(t) = \exp(tL_B)$, we construct successively

$$S^{(n)}(t) = S_A(c_1t)S_B(d_1t)\cdots S_A(c_nt)S_B(d_nt), \tag{25.1}$$

using $S^{(2p+1)}(t) = S^{(2p)}(t)S_A(c_{p+1}t)$ and $S^{(2p+2)}(t) = S^{(2p+1)}(t)S_B(d_{p+1}t)$. Since the transformations $S^{(n)}$ are considered Lie transformations, they are represented by the derivative of their generating series. $S_A(t)$ is represented by A and $S_B(t)$ by B. If $S^{(n)}(t) = T^{-1}_{w^{(n)}(t)}$, we will have the following relations:

$$\frac{\partial}{\partial t}w^{(2p+1)} = S_A(c_pt)\frac{\partial}{\partial t}w^{(2p)} + c_{p+1}A, \quad \frac{\partial}{\partial t}w^{(2p+2)} = S_B(d_pt)\frac{\partial}{\partial t}w^{(2p+1)} + d_{p+1}B. \tag{25.2}$$

The c_i and the d_i are taken in lc and ld.

```
symplectic(n:NNI,r:NNI):List L ==
    l:List L  := concat(lc.1 * (A::L),[0$L for i in 2..r])
    ll:List L := [0$L for i in 1..r]
    n=1 => l
    i:NNI:=2
    while i<(n+1) repeat
        j:= i quo 2 + 1
        if (i rem 2)=0 then ll.1:=ld.((j-1)::NNI) * (B::L)
        else ll.1:=lc.j * (A::L)
        l:=addi(ll,exponential(integrate ll,l))
        i:=i+1
    l
```

The same thing can be done supposing that the products are reversible.

```
symplecticr(n:NNI,r:NNI):List L ==
    i:NNI:=n+1
    j:= i quo 2
    if (i rem 2)=0 then l:List L := concat(ld.j * B::L,[0$L for i in 2..r])
    else l:List L := concat(lc.j * A::L,[0$L for i in 2..r])
    ll:List L := [0$L for i in 1..r]
    n=1 => l
    i:=n
    while i>1 repeat
        j:= i quo 2
        if (i rem 2)=0 then ll.1 := ld.j * (B::L)
        else ll.1 := lc.j * (A::L)
        l:=addi(ll,exponential(integrate ll,l))
        i:=i-1
    i:=3
    while i<n+2 repeat
        j:= i quo 2
        if (i rem 2)=0 then ll.1 := ld.j * (B::L)
        else ll.1 := lc.j * (A::L)
        l:=addi(ll,exponential(integrate ll,l))
        i:=i+1
    l
```

We can study the same products, but this time, using 3 operators S_A, S_B and S_C.

```
symplex3(n:NNI,r:NNI):List L ==
    l:List L := concat(lc.1 * A::L,[0$L for i in 2..r])
    ll:List L := [0$L for i in 1..r]
    n=1 => l
    i:NNI:=2
    while i<(n+1) repeat
      j:=i quo 3 + 1
      if (i rem 3)=2 then ll.1:=ld.j * (B::L)
      else if (i rem 3)=0 then ll.1:=le.((j-1)::NNI) * (C::L)
           else ll.1:=lc.j * (A::L)
      l:=addi(ll,exponential(integrate ll,l))
      i:=i+1
    l

symplex3r(n:NNI,r:NNI):List L ==
    i:=n+2
    j:= i quo 3
    if (i rem 3)=0 then l:List L := concat(lc.j * (A::L),[0$L for i in 2..r])
    else if (i rem 3)=1 then l:List L:=concat(ld.j * (B::L),[0$L for i in 2..r])
         else l:List L := concat(le.j * (C::L),[0$L for i in 2..r])
    ll:List L := [0$L for i in 1..r]
    n=1 => l
    i:NNI:=n+1
    while i>2 repeat
      j:=i quo 3
      if (i rem 3)=0 then ll.1:=lc.j * (A::L)
      else if (i rem 3)=1 then ll.1:=ld.j * (B::L)
           else ll.1:=le.j * (C::L)
```

```
    l:=addi(ll,exponential(integrate ll,l))
    i:=i-1
  i:=4
  while i<n+3 repeat
    j:=i quo 3
    if (i rem 3)=0 then ll.1:=lc.j * (A::L)
    else if (i rem 3)=1 then ll.1:=ld.j * (B::L)
         else ll.1:=le.j * (C::L)
    l:=addi(ll,exponential(integrate ll,l))
    i:=i+1
  l
```

From these results, we extract the coefficients, –which will be analyzed later– by using other procedures.

For computational examples, as well as methods of computing symplectic integrators, see the appendix.

26. Computation of Normal Forms

Several domains and packages were implemented in order to compute the normal forms on a few concrete examples. Unfortunately, the complexity of interesting problems is such, that for the time being, we had to limit our scope to elementary problems.

At first glance, it might appear very difficult, if not unrealistic, to construct packages that are general enough to handle most cases. We have witnessed the development of highly specialized software on one class of problems. In my opinion, these difficulties are related to the fact that every problem has a specific representation, which cannot be handled by unspecialized software.

26.1. Manipulations of the Poisson Series

As we have seen, it would be interesting to be able to handle polynomials with trigonometric expressions on a ring. Therefore, we have created a number of domains of Poisson series.

We only handle the case where h_0 has a simple enough form so that its flow is known; for example, if

$$h_0 = \tfrac{1}{2} \sum_{i=1}^{n}(p_i^2 + \omega_i^2 q_i^2), \tag{26.1}$$

the curve solutions are, in the case of the classic coordinates,

$$q_i(t) = q_i(0)\ cos\ (\omega_i t) + p_i(0)\ sin\ (\omega_i t) \tag{26.2}$$

$$p_i(t) = p_i(0)\ cos\ (\omega_i t) - q_i(0)\ sin\ (\omega_i t), \tag{26.3}$$

in the case of the action-angle variables,

$$j_i(t) = j_i(0)\ \ ,\quad \phi_i = \phi_i(0) + \omega_i t. \tag{26.4}$$

We simultaneously developed two types of algebraic manipulations, polynomials in the cartesian variables on the one hand, and expressions in the action-angle coordinates on the other.

26.1.1. In Cartesian Coordinates:

We suppose that for any n, h_n is a polynomial in p_i, q_i, with coefficients in a ring containing trigonometric expressions in the form cos $(P\ t)$, sin $(P\ t)$, where P is a rational fraction with several variables, and closed under derivation and integration.

In order to manipulate such expressions, we were led to create new domains and packages (cf. [30]).

- Domain `MTGD(x,lv1,lv2)`
  Poisson series of a list of variables $lv1$, of time x, where the frequencies are polynomials with several variables $lv2$, with coefficients in $\mathbb{Q}$. The coefficients are in the field of its fractions. For example :
$$u = \frac{\omega}{\Omega^2 - \omega^2} \cos\ (\ \Omega t + q_1 - q_2)\ . \tag{26.5}$$

- Derivation and integration package, with respect to time, for `MTGD`. For example,
$$\frac{\partial u}{\partial q_1} = -\frac{\omega}{\Omega^2 - \omega^2} \sin\ (\ \Omega t + q_1 - q_2)\ , \tag{26.6}$$
$$\int_t u = \frac{\omega}{\Omega^3 - \Omega\omega^2} \sin\ (\ \Omega t + q_1 - q_2)\ . \tag{26.7}$$

- Manipulation package in `DMP(lv,MTGD(t,lv1))`, where lv is the list of variables of the phase space considered. (`DMP` is the constructor of polynomials with several distributed variables).

- Domain : `OTGD(t,lv1)`
  Poisson series of a variable t, where the frequencies belong to the ring of polynomial with several variables $lv1$ on $\mathbb{Q}$, with coefficients in its field of fraction.

- Derivation and integration package, with respect to time, for `OTGD(t,lv1)`

- Manipulation package in `DMP(lv,OTGD(t,lv1))`, where lv is the list of the phase space variables.

Using these various packages, we can apply our version of Deprit's algorithm. At the start, we have h as the list of polynomials with $2n$ variables p_i, q_i, with coefficients in the ring of the Poisson series.

26.1.2. action-angle Coordinates

This second way of approaching the solution is, in fact, better adapted to the framework in which we have placed ourselves. We make the same hypotheses as previously. We are led to work in a Scratchpad domain which contains square roots, and we use `RadicalExtension`, which is the extension by radicals, of a given field.

- Domain `RTGD(x,lv1)`
  Poisson series of a list of variables $lv1$, of time x, where the frequencies are rational fractions with several variables (not predefined) with coefficients in $\mathbb{Q}$, the coefficients are in the extension by the radicals of the rational fractions on $\mathbb{Q}$. For example,
$$u = \sqrt{\tfrac{2j}{\omega}} \cos\ (\ \Omega t + \phi_1 - \phi_2)\ . \tag{26.8}$$

- Derivation and integration package, with respect to time, in the case of RTGD. For example,

$$\frac{\partial u}{\partial \phi_1} = -\sqrt{\tfrac{2j}{\omega}} \ \sin\ (\ \Omega t + \phi_1 - \phi_2)\ , \tag{26.9}$$

$$\int_j u = -\frac{2j}{3}\sqrt{\tfrac{2j}{\omega}} \ \sin\ (\ \Omega t + \phi_1 - \phi_2)\ . \tag{26.10}$$

- It was created for each one of these domains, with two packages containing the functions necessary for the algorithms of the Dragt-Finn and Deprit methods.

Here is, for example, a presentation of RTGD :

```
RTGD(x: E,lvp: L E,lvq: L E)  is a domain constructor
Abbreviation for RadicalTimeTrigoDomain is RTGD
This constructor is currently exposed.
Issue )edit RATRIGD2 SPAD A1 to see algebra source code for RTGD


--------------------------- Operations---------------------------------

?+? : ($,$) -> $                          ?*? : (RF RN,$) -> $
?*? : (RN,$) -> $                         ?*? :(RE RF RN,$) -> $
?*? : ($,$) -> $                          ?*? : ($,$) -> $
?*? : (I,$) -> $                          ?**? : ($,NNI) -> $
?-? : ($,$) -> $                          -? : $ -> $
?=? : ($,$) -> B                          cderiv : ($,SE) -> $
-- characteristic : () -> NNI             coerce : RE RF RN -> $
coerce : $ -> $                           coerce : I -> $
coerce : $ -> E                           -- coerce : E -> Union($,"failed")
const : $ -> $                            cos : (P RN,DP(# lvq,I)) -> $
deriv : $ -> $                            integ : $ -> $
p : PI -> SE                              pderiv : ($,OV lvq) -> $
pinteg : ($,OV lvq) -> $                  poisson : ($,$) -> $
q : PI -> OV lvq                          recip : $ -> Union($,"failed")
retract : $ -> RE RF RN                   retractable? : $ -> B
sin : (P RN,DP(# lvq,I)) -> $             1 : () -> $
0 : () -> $
new : Record(c: RE RF RN,s: RE RF RN,int1: P RN,int2: DP(# lvq,I)) -> $
retractIfCan : $ -> Union(RE RF RN,"failed")
```

Here, RN represents the rationals, RF the rational fractions, RE RadicalExtension etc.

26.2. Polynomials

Whenever the systems under study are time independent, one will be able to manipulate the polynomials. As a representation, we take the polynomials with several distributed variables. In order to solve the perturbation equation, on the one hand we compute recursively –from the Hessian of h_0– the matrix L_{h_0} and on the other, we transform the polynomials into a vector, and we use the packages already existing under Scratchpad II to invert the linear system. The Hamiltonians, as well as the generating functions, are given in the form of lists where the ith term is the term of order $i+2$.

Conclusion and Perspectives

The starting point of this study is the comparison of two perturbation methods commonly used in mechanics.

The first result, which is nothing new in itself, is that both methods are equivalent, insofar as they produce the same normal forms and the same formal invariants. This result is mostly due to the uniqueness of Birkhoff's normal form in the non-resonant case. Nevertheless, we have come to the conclusion that the representation of a formal canonical transformation has a definite influence on the complexity of the required computation.

The use of the Dragt-Finn transformation (an increasing product of exponentials) is justified for the computation of normal forms, since it requires fewer evaluations. The use of the exponential presents definite advantages in the search for reversible symplectic integrators, whereas the Lie transformation is quite convenient in composing canonical transformations. Furthermore, the Lie transformation is the classic representation for the mapping of a non-autonomous flow.

The asymptotic estimations achieved when using the Deprit transformation are better than those I obtained myself while using the Dragt-Finn transformation. In all likelihood, finer results could be obtained by using combinatorial results on the partitions. However, this phenomenon does not as yet have any bearing on the stability estimations that can be deduced from it, as shown in the results obtained by Giorgilli & Celleti [8].

From these results, we can deduce that it is necessary to effectively compute normal forms and formal invariants in order to obtain finer results, even if the orders used in their computation are frequently very low. They are so low, because the complexity of the computation, or the size of the data, are too great. One way of remedying this would be to keep the smallest possible number of parameters, and to proceed numerically. A follow-up study on the relative importance of the various parameters would be indicated.

Using the Lyndon basis enabled us to compute relations between the generators of the various canonical transformations near the identity, which would not be possible with the Hall basis. We further recover a computation method from the Baker-Campbell-Hausdorff formula. In my opinion, the fact that we are able to compute these identities while remaining in the free Lie algebra does not, in any way, increase the overall complexity of the computation. The dimensions of the modules under consideration are the same as when the bijection between the Lyndon words and the Lyndon basis is used. One does not need to know the parenthesizing of every element of the Lyndon basis.

Representation of the magma elements by lists saved a substantial amount of time. In this representation, it appears that the decomposition onto the Lyndon basis is faster than the decomposition onto the Hall basis.

Finally, the search for symplectic integrators –whose formalism reminds us of the motion-planning problem– ([45]) is connected, at high orders– with the search for real points in a variety. For orders under 6, we gave an exhaustive list of the symplectic integrators (except for order 5, where some doubt remains). A thorough study of the geometry of the solutions would undoubtedly enable us to give algebraic solutions at higher orders.

BIBLIOGRAPHY

[1] G.E. ANDREWS, *The Theory of Partitions*, Encyclopedia of Mathematics, Number Theory, vol 2, Addison-Wesley, 1976.

[2] V.I. ARNOLD, *Dynamical Systems IV*, Encyclopaedia of Mathematical Sciences, vol 4, Springer Verlag, 1978.

[3] V.I. ARNOLD, *Mathematical Methods for Classical Mechanics*, Graduate texts in Mathematics, vol 60, Springer Verlag, 1978.

[4] G. BENETTIN, L. GALGANI, A. GIORGILLI, *A Proof of Nekhoroshev's Theorem for the Stability Times in Nearly Integrable Hamiltonian Systems*, Celestial Mechanics **37** (1985), 1–25.

[5] M. BERZ, E. FOREST, *Canonical Integration and Analysis of Periodic Maps using Non-standard Analysis and Lie Methods*, in *Lie Methods in Optics II*, Springer Verlag **352** (1988).

[6] N. BOURBAKI, *Groupes et Algèbres de Lie*, Eléments de Mathématiques, Hermann, Paris, 1972.

[7] J.R. CARY, *Lie Transform Perturbation Theory for Hamiltonian Systems*, in *Physics Reports*, North-Holland Publishing Compagny **79-2** (1981), 129–159.

[8] A. CELLETTI, A. GIORGILLI, *On the Stability of the Lagrangian Points in the Spatial Restricted Problem of Three Bodies*, CARR Reports in Mathematical Physics **19/90** (1990).

[9] P.J. CHANNEL, J.C. SCOVEL, *Symplectic Integration of Hamiltonien Systems*, Nonlinearity **3** (1990), 231–259.

[10] P.J. CHANNEL, J.C. SCOVEL, *Integrators for Lie-Poisson Dynamical Systems*, Physica D **50** (1991), 80–88.

[11] A. DEPRIT, *Canonical Transformations Depending on a Small Parameter*, Celestial Mechanics **1** (1969), 12–30.

[12] J. DIAMOND, A.J. DRAGT, F. NERI, J. VAN ZEIJTS, *Analytic Third-order Transfer Map for Combined-Function Quadrupole*, preprint (1990).

[13] A.J. DRAGT, *Numerical Third-order Transfer Map for Solenoid*, Nuclear Instruments and Methods in Physics Research **A298** (1990), 441–459.

[14] J. Dragt, J. M. Finn, *Lie Series and Invariant Functions for Analytic Symplectic Maps*, Journal of Mathematical Physic **17** (1976), 2215–2227.

[15] J. Dragt, J. M. Finn, *Normal form for Mirror Machin Hamiltonians*, Journal of Mathematical Physic **20 n°12** (1979).

[16] A.J. Dragt, L.M. Healy, *Concatenation of Lie Algebraic Maps*, in *Lie Methods in Optics II*, Springer Verlag **352** (1988).

[17] F. Fassò, *On a Relation among Lie Series*, Celestial Mechanics and Dynamical Astronomy **46** (1989), 113–118.

[18] F. Fassò, G. Benettin, *Composition of Lie transforms with rigourous estimates and applications to Hamiltonian perturbation theory*, Zeitschrift für Angewandte Mathematik und Physik **40** (May 1989).

[19] J.M. Finn, *Lie Series: a Perspective, Local and Global Methods of nonlinear Dynamics*, Lecture Notes in Physics **252** (1984), 63–86.

[20] E. Forest, D. Ruth, *Fourth-Orther Symplectic Integration*, Physica D **43** (1990), 105–117.

[21] E. Fried, G. Ezra, *PERTURB: A Special-Purpose Algebraic Manipulation Program for Classical Perturbation Theory*, Journal of Computational Chemistery (1987), 397–411.

[22] A. Giorgilli, *Rigourous Results on the Power Expansions for the Integrals of a Hamiltonian System near a Elliptic Equilibrium Point*, Annales de l'Institut Henri Poincaré **48** (1988), 423–439.

[23] A. Giorgilli, L. Galgani, *Formal Integrals for an Autonomous Hamiltonian System near an equilibrium point*, Celestial Mechanics **17 94** (1977), 267–280.

[24] A. Giorgilli, L. Galgani, *Rigourous Estimates for the Series Expansions of Hamiltonian Perturbation Theory*, Celestial Mechanics **37** (1985), 95–112.

[25] Giorgilli, Delshams, Fontich, Galgani, Simò, *Effective Stability for a Hamiltonian System near an Elliptic Equilibrium, with an Application to the restricted Three Body Problem*, Journal of Diff. Equations (1988).

[26] W. Gröbner, *Die Lie-Reihen und Ihre Anwendungen*, VEB Deutscher Verlag der Wissenschaften, 1967.

[27] Goutelas 89, *Les Méthodes Modernes de la Mécanique Céleste*, Editions Frontières, D. Benest, C. Froeschle eds, 1989.

[28] R. Grossman, *Using Trees To compute Approximate Solutions to Ordinary Differentials Equations Exactly*, Differential Equations and Computer Algebra (1991), 29–60.

[29] J. Henrard, *The Algorithm of the Inverse for Lie transform*, Recent Advances in Dynamical Astronomy, D. Reidel Publishing Compagny (1973), 250–259.

[30] R.D. JENKS, R.S. SUTOR, AXIOM, *the Scientific Computation System*, Springer Verlag, NaG, 1992.

[31] A.A. KAMEL, *Perturbation methods in the theory of nonlinear oscillations*, Celestial Mechanics **3** (1970), 90–106.

[32] P.-V. KOSELEFF, *Quelques Bases et Formules dans les Algèbres de Lie libres*, Etude F.136, I.B.M. France, Paris (1988).

[33] P.-V. KOSELEFF, *Contributions aux méthodes de Lie pour la résolution de systèmes hamiltoniens*, Etude F.144, I.B.M. France, Paris (1989).

[34] P.-V. KOSELEFF, *Jeux de Mots dans les Algèbres de Lie libres: Quelques Bases et Formules*, Theoretical Computing Science (1991).

[35] P.-V. KOSELEFF, *Relations among Formal Lie Series and Construction of Symplectic Integrators*, AAECC'10 proceedings, in print (1993).

[36] P.-V. KOSELEFF, *Comparison between Deprit and Dragt-Finn perturbation methods*, Celestial Mechanics, in print (1993).

[37] J. LASKAR, *Analytical Framework in Poincaré Variables for the Motion of the Solar System*, Predictability, Stability and Chaos in N-Body Dynamical Systems (1991).

[38] M.A. LICHTENBERG, A.J. LIEBERMAN, *Regular and Stochastic Motion*, Applied Mathematical Sciences 92, Springer Verlag, 1988.

[39] P. LOCHAK, C. MEUNIER, *Multiphase Averaging for Classical Systems*, Applied Mathematical Sciences 78, Springer Verlag, New-York, 1988.

[40] G. MÉLANÇON, *Combinatorics of Hall trees and Hall Words*, Journal of Combinatorial Theory **A 59** (1992), 285–308.

[41] K. R. MEYER, G. R. HALL, *Introduction to Hamiltonian Dynamical Systems and the N-Body Problem*, Applied Mathematical Sciences 92, Springer Verlag New-York, 1992.

[42] J. MICHEL, *Bases des Algèbres de Lie et Série de Hausdorff*, Séminaire Dubreil, Paris **27, 6** (1974).

[43] J. MICHEL, *Bases des Algèbres de Lie libres, Étude des coefficients de la Formule de Campbell-Hausdorff*, Thèse de troisième cycle, Orsay, 1974.

[44] D. PERRIN, *Factorisation of free monoids*, in *Lothaire M., Combinatorics On Words, Chapitre 5*, Addison-Wesley (1983).

[45] M. PETITOT, *Algèbre Non-commutative en Scratchpad: Application au Problème de la Réalisation Minimale Analytique*, Thèse de Doctorat, Université de Lille I, 1991.

[46] D. PINCHON, *Langages de Calcul Formel*, Encyclopédie des Techniques de l'ingénieur (1991).

[47] G. PÒLYA, G. SZEGÖ, *Problems ans Theorems in Analysis I*, Springer Study Edition, Springer Verlag, 1972.

[48] QIN MENG-ZHAO, ZHU WEN-JIE, *Construction of Higher Order Symplectic Schemes by Composition*, Computing **47** (1992), 309–321.

[49] A.A. SAGLE, R.E WALDE, *Introduction to Lie Groups and Lie Algebras*, Pure and Applied Mathematics 51, Academic Press, 1973.

[50] J.-P. SERRE, *Lie Algebras and Lie Groups*, Lecture Notes in Mathematics, 1500, Springer Verlag, 1991.

[51] C.L. SIEGEL, J.K. MOSER, *Lectures on celestial Mechanics*, Die Grundlehren der matematischen Wissenschaften, band 187, Springer Verlag, 1971.

[52] C. SIMÒ, *Estabilitat de Sistemes Hamiltonians*, Memorias de la Real Academia de Ciencias y Artes de Barcelona **48** (1989), 303–348.

[53] S. STEINBERG, *Lie Series, Lie Transformations, and their Applications*, in *Lie Methods in Optics*, Lecture Notes in Physics **250** (1985).

[54] M. SUZUKI, *Fractal Decomposition of Exponential Operators with Applications to Many-Body Theories and Monte Carlo Simulations*, Physic Letters A **146** (1990), 319–323.

[55] G. VIENNOT, *Une Théorie Algébrique des Bases et Familles Basiques des Algèbres de Lie*, Séminaire Dubreuil **27, 5** (1974).

[56] G. VIENNOT, *Quelques Bases et Familles Basiques des Algèbres de Lie Libres Commodes pour les Calculs sur Ordinateurs*, Bull. Soc. Math. France, Mémoire 49-50 (1975), 201–209.

[57] G. VIENNOT, *Algèbres de Lie libres et Monoïdes Libres*, Lecture Notes in Mathematics, vol 691, Springer Verlag, 1978.

[58] L. WAUTHIER, *Utilisation du Calcul Formel en Théorie Analytique du Mouvement des Satellites Artificiels*, Thèse de Doctorat, Université de Rennes I, 1991.

[59] J. WISDOM, M. HOLMAN, *Symplectic Maps for the N-Body Problem*, The Astronomic Journal **102(4)** (1991), 1528–1538.

[60] H. YOSHIDA, *Conserved Quantities of Symplectic Integrators for Hamiltonian Systems*, Physica D (1990).

[61] H. YOSHIDA, *Construction Of Higher Order Symplectic Integrators*, Physics Letters A **150,** (1990), 262–268.

Appendix

In this appendix, we propose examples of normal form computation obtained through the use of the Dragt-Finn method. We also present examples of sessions in AXIOM which enable us to manipulate Lie polynomials. We will show, using AXIOM, how one gets the equations to solve for symplectic integrators, and using MACAULAY and AXIOM, how one computes the solutions, or the dimension and degree of the variety of the solutions.

A. NORMAL FORM COMPUTATION

The examples of normal form computation presented here are relatively simple cases. These results were obtained by using Scratchpad II. With several degrees of freedom, we frequently come across data that are too large to be properly handled. It appears necessary to adapt a specific program for every type of Hamiltonian to be dealt with.

A.1. Examples with one degree of freedom

A.1.1. Perturbed Oscillator

Let us consider the Hamiltonian $h = \frac{1}{2}\left(p^2+q^2\right) + \frac{1}{3}q^3$. Applying the Dragt-Finn method, we find a normalized Hamiltonian at order 20:

$$\begin{aligned} K = {} & \tfrac{1}{2}\left(p^2+q^2\right) - \tfrac{5}{48}\left(p^2+q^2\right)^2 - \tfrac{235}{3456}\left(p^2+q^2\right)^3 - \tfrac{38585}{497664}\left(p^2+q^2\right)^4 - \tfrac{2663129}{23887872}\left(p^2+q^2\right)^5 \\ & - \tfrac{156934505}{859963392}\left(p^2+q^2\right)^6 - \tfrac{13400341405}{41278242816}\left(p^2+q^2\right)^7 - \tfrac{7275692993855}{11888133931008}\left(p^2+q^2\right)^8 \\ & - \tfrac{12369775168615645}{10271347716390912}\left(p^2+q^2\right)^9 - \tfrac{27168477250016438585}{11093055533370218496}\left(p^2+q^2\right)^{10} . \end{aligned}$$

Time c.p.u : 3 minutes. (IBM4381)

A.1.2. Pendulum

Old Hamiltonian (at order 30):

$$\begin{aligned} h = {} & \tfrac{1}{2}p^2 + \tfrac{1}{2}q^2 - \tfrac{1}{24}q^4 + \tfrac{1}{720}q^6 + -\tfrac{1}{40320}q^8 + \tfrac{1}{3628800}q^{10} - \tfrac{1}{479001600}q^{12} + \tfrac{1}{87178291200}q^{14} \\ & - \tfrac{1}{20922789888000}q^{16} + \tfrac{1}{6402373705728000}q^{18} - \tfrac{1}{2432902008176640000}q^{20} \\ & + \tfrac{1}{1124000727777607680000}q^{22} - \tfrac{1}{620448401733239439360000}q^{24} + \tfrac{1}{403291461126605635584000000}q^{26} \\ & - \tfrac{1}{304888344611713860501504000000}q^{28} + \tfrac{1}{265252859812191058636308480000000}q^{30}. \end{aligned}$$

New Hamiltonian (at order 30):

$$\frac{1}{2}\left(q^2+p^2\right)-\frac{1}{64}\left(q^2+p^2\right)^2-\frac{1}{2048}\left(q^2+p^2\right)^3-\frac{5}{131072}\left(q^2+p^2\right)^4-\frac{33}{8388608}\left(q^2+p^2\right)^5$$
$$-\frac{63}{134217728}\left(q^2+p^2\right)^6-\frac{527}{8589934592}\left(q^2+p^2\right)^7-\frac{9387}{1099511627776}\left(q^2+p^2\right)^8$$
$$-\frac{175045}{140737488355328}\left(q^2+p^2\right)^9-\frac{422565}{2251799813685248}\left(q^2+p^2\right)^{10}$$
$$-\frac{4194753}{144115188075855872}\left(q^2+p^2\right)^{11}-\frac{1330745}{288230376151711744}\left(q^2+p^2\right)^{12}$$
$$-\frac{4403374207}{590295810358705651712}\left(q^2+p^2\right)^{13}-\frac{578183175}{472236648286964521369 6}\left(q^2+p^2\right)^{14}$$
$$-\frac{12308013927}{604462909807314587353088}\left(q^2+p^2\right)^{15}.$$

Generating function (at order 9):

$$-\frac{1}{64}qp^3-\frac{5}{192}q^3p-\frac{1}{1024}qp^5-\frac{1}{384}q^3p^3-\frac{17}{15360}q^5p-\frac{11}{131072}qp^7-\frac{121}{393216}q^3p^5-\frac{1513}{5898240}q^5p^3$$
$$-\frac{2657}{24772608}q^7p.$$

Old variables with respect to the new variables (at order 9):

$$Q=q-\frac{3}{64}qp^2-\frac{5}{192}q^3-\frac{37}{8192}qp^4-\frac{37}{4096}q^3p^2-\frac{11}{122880}q^5-\frac{39}{524288}qp^6-\frac{731}{1572864}q^3p^4$$
$$-\frac{727}{1572864}q^5p^2+\frac{1415}{14155776}q^7+\frac{83}{134217728}qp^8-\frac{463}{100663296}q^3p^6-\frac{1687}{67108864}q^5p^4$$
$$-\frac{51281}{1509949440}q^7p^2-\frac{438161}{54358179840}q^9.$$

These formulas were computed to order 30 in 2 minutes, using Scratchpad on an IBM4381.

A.1.3. Perturbed Harmonic Oscillator

Old Hamiltonian:

$$h=\frac{1}{2}\left(p^2+q^2\right)+a_2q^4+a_3q^6+a_4q^8+a_5q^{10}+a_6q^{12}+a_7q^{14}+a_8q^{16}+a_9q^{18}+a_{10}q^{20}.$$

New Hamiltonian (at order 20):

$$\frac{1}{2}(p^2+q^2)$$
$$+\frac{1}{2}(3a_2)\left(\frac{1}{2}(p^2+q^2)\right)^2$$
$$+\frac{1}{16}(-17a_2^2+10a_3)\left(\frac{1}{2}(p^2+q^2)\right)^3$$
$$+\frac{1}{8}(375a_2^3+330a_2a_3+70a_4)\left(\frac{1}{2}(p^2+q^2)\right)^4$$
$$+\frac{1}{8}(-10689a_2^4+12516a_2^2a_3-1572a_3^2+3024a_2a_4+504a_5)\left(\frac{1}{2}(p^2+q^2)\right)^5$$
$$+\frac{1}{8}(87549a_2^5+127932a_2^3a_3+33684a_2a_3^2+32368a_2^2+7112a_3a_4+6720a_2a_5+924a_6)\left(\frac{1}{2}(p^2+q^2)\right)^6$$
$$+\frac{1}{256}(-3132399a_2^6+5485374a_2^4a_3+2218788a_2^2a_3^2+117960a_3^3+1420296a_2^3+679824a_2a_3a_4+31880a_4^2+320760a_2^2$$
$$+62640a_3a_5+58608a_2a_6+6864a_7)\left(\frac{1}{2}(p^2+q^2)\right)^7$$
$$+\frac{1}{2048}(238225977a_2^7+486162630a_2^5a_3+265977900a_2^3a_3^2+33288840a_2a_3^3+127609830a_2^4+95859000a_2^2a_3$$
$$+7006680a_3^2a_4+6728760a_2a_4^2+30119400a_2^3+13215600a_2a_3+1115280a_2a_5+6173640a_2^2+1085040a_3a_6$$
$$+1009008a_2a_7+102960a_8)\left(\frac{1}{2}(p^2+q^2)\right)^8$$

$$
\begin{aligned}
&+\tfrac{1}{16384}(-18945961925a_2^8+44145981880a_2^6a_3+30477140760a_2^4a_3^2+6275802720a_2^2a_3^3+183218640a_3^4+11691155520a_2^5\\
&\quad+12040842880a_2^3a_3+2109803520a_2a_3^2a_4+1012431200a_2^2+136607680a_3a_4^2+2834774800a_2^4+1987740480a_2^2a_3\\
&\quad+134122560a_3^2+257579520a_2a_4a_5+9700160a_5^2+618280960a_2^3+250440960a_2a_3+19212160a_4a_6+116276160a_2^2\\
&\quad+18578560a_3a_7+17205760a_2a_8+1555840a_9)\left(\tfrac{1}{2}(p^2+q^2)\right)^9\\
&+\tfrac{1}{16384}(2956096a_{10}+194904116847a_2^9+510499888800a_2^7a_3+425756036376a_2^5a_3^2+124085886144a_2^3a_3^3\\
&\quad+9047682864a_2a_3^4+136034613880a_2^6+178463437680a_2^4a_3+52065396768a_2^2a_3^2+1900075584a_3^3a_4\\
&\quad+16647437136a_2^3+5469343968a_2a_3a_4^2+109591328a_4^3+33584619552a_2^5+32674055040a_2^3a_3+5368091520a_2a_3^2\\
&\quad+5151717120a_2^2+645507456a_3a_4a_5+304231488a_2a_5^2+7614132504a_2^4+5006220384a_2^2a_3+313702752a_3^2\\
&\quad+602431104a_2a_4+41587392a_5a_6+1547562016a_2^3+582149568a_2a_3+40936896a_4a_7+269178624a_2^2\\
&\quad+39426816a_3a_8+36406656a_2a_9)\left(\tfrac{1}{2}(p^2+q^2)\right)^{10}
\end{aligned}
$$

A.1.4. Perturbed Harmonic Oscillator

Hamiltonian:

$$h = \tfrac{\omega}{2}\left(p^2+q^2\right) + a_2q^4 + a_3q^6 + a_4q^8 + a_5q^{10}.$$

New Hamiltonian:

$$
\begin{aligned}
K &= \tfrac{\omega}{2}\left(p^2+q^2\right) + \tfrac{3a_2}{8}\left(p^2+q^2\right)^2 + \tfrac{1}{32}(-17a_2^2+10a_3\omega)\omega\left(p^2+q^2\right)^3 + \tfrac{1}{256}(375a_2^3-330a_2a_3\omega\\
&\quad+70a_4\omega^2)\omega^2\left(p^2+q^2\right)^4 + \tfrac{1}{2048}(-10689a_2^4+12516a_2^2a_3\omega-1572a_3^2-3024a_2a_4\omega^2\\
&\quad+504a_5\omega^3)\omega^3\left(p^2+q^2\right)^5
\end{aligned}
$$

This Hamiltonian frequently shows up, for example, when we study a system with several degrees of freedoms, which we reduce with respect to one of the degrees of freedom.

A.2. Examples with two degrees of freedom

A.2.1. Particle in a Magnetic Field

We use the model given in [15] of the Hamiltonian of a particle subjected to a magnetic field. We have, as the Hamiltonian

$$h = \tfrac{1}{2}\left(p_2^2+q_2^2\right) + \tfrac{1}{2}p_1^2 + \tfrac{1}{2}q_1^2q_2^2 + q_1^4q_2^2\ .$$

Applying the method of the adjoint operator, we find

$$h = \tfrac{1}{2}\left(p_2^2+q_2^2\right) + \tfrac{1}{2}p_1^2 + \tfrac{1}{2}q_1^2\left(\tfrac{1}{8}\left(\tfrac{1}{2}(p_2^2+q_2^2)\right)^2 + \left(\tfrac{1}{2}(p_2^2+q_2^2)\right)\right) + \tfrac{7}{8}q_1^4\left(\tfrac{1}{2}(p_2^2+q_2^2)\right)\ .$$

Posing $\Omega^2 = \frac{1}{8}\left(\frac{1}{2}(p_2^2+q_2^2)\right)^2 + \left(\frac{1}{2}(p_2^2+q_2^2)\right)$ and $e_2 = \frac{1}{2}(p_2^2+q_2^2)$, and carrying out the change of variables

$$P_1 = \sqrt{\Omega p_1} \quad \text{and} \quad Q_1 = \frac{q_1}{\sqrt{\Omega}}, \tag{A.1}$$

we end up with an Hamiltonian with one degree of freedom

$$H = e_2 + \tfrac{\Omega}{2}(P_1^2+Q_1^2) + \tfrac{7e_2}{\Omega^2}Q_1^4\ . \tag{A.2}$$

This Hamiltonian is in the previous form. Its normal form is

$$K = e_2 + \tfrac{\Omega}{2}(P_1^2 + Q_1^2) + \tfrac{3}{8}\left(\tfrac{7e_2}{\Omega^2}\right)(P_1^2 + Q_1^2)^2 , \tag{A.3}$$

$$K = \tfrac{1}{2}(p_2^2 + q_2^2) + \tfrac{\Omega}{2}(P_1^2 + Q_1^2) + \tfrac{21}{8}\left(\frac{(P_1^2+Q_1^2)^2}{1+\frac{1}{8}\left(\frac{1}{2}(p_2^2 + q_2^2)\right)}\right) \tag{A.4}$$

So, we have found, at order 4, 2 invariants in involution:

$$P_1^2 + Q_1^2 \quad \text{and} \quad p_2^2 + q_2^2 . \tag{A.5}$$

The first normalization was implemented in 90 seconds. The second one is just a series of evaluations and rewriting of the expressions.

A.3. Time Dependent Examples

Let us consider the following Hamiltonian:

$$h = \tfrac{1}{2}\left(p^2 + a\omega^2\left(1 + \sin(\Omega t)\right) q^2\right) . \tag{A.6}$$

We use the action-angle coordinates, so that

$$h = \omega j + 2aj\omega \sin(\Omega t)\sin(\phi)\cos(\phi) . \tag{A.7}$$

Using the Deprit method, we find

$$K = j\omega - j\tfrac{1}{4}\frac{a^2\omega^3}{(\Omega-2\omega)(\Omega+2\omega)} + j\frac{a^4\omega^5}{64} + j\frac{2\Omega^4-35\omega^2\Omega^2+120\omega^4}{(\Omega-2\omega)^3(\Omega+2\omega)^3(\Omega-\omega)(\Omega+\omega)} + \cdots \tag{A.8}$$

We can also consider the Hamiltonian

$h = \frac{1}{2}p^2 + a\omega^2\left(1 + \sin(\Omega t)\right)\left(1 - \cos(q)\right)$.

In this case, we find

$$K = \omega j - \tfrac{1}{4}\frac{a^2 j\omega^3}{(\Omega-2\omega)(\Omega+2\omega)} + \tfrac{1}{8}\frac{a^2 j^2\omega^2}{(\Omega-2\omega)(\Omega+2\omega)} + \frac{a^2 j\omega}{576(\Omega-2\omega)^3(\Omega+2\omega)^3(\Omega-\omega)(\Omega+\omega)(\Omega+4\omega)(\Omega-4\omega)}\Big(-3712j^2\omega^8-8640a^2\omega^{10}+(5824j^2\omega^6+5580a^2\omega^8)\Omega^2-(2472j^2\omega^4+603a^2\omega^6)\Omega^4+(376j^2\omega^2+18a^2\omega^4)\Omega^6 -16j^2\Omega^8\Big)$$

This model corresponds to a swing with a child sitting on it. The eigenfrequency of the swing is ω, and the child swings its legs with a frequency Ω. As we well know, the system resonates when $\Omega = \pm 2\omega$, but also, to a lesser extent, when $\Omega = \pm\omega$, or $\Omega = \pm 4\omega$.

B. Examples of Sessions

B.1. Decomposition of Lie Polynomials

Here is a brief example, which makes it possible to decompose an element of the magma onto the Hall basis or the Lyndon basis. In order to verify the computation, we are able to go from one basis to the other, through the magma.

The first step is to load the various modules.

```
)lo FLA FLA- MG LFLA HFLA LS )cond
   loading /cmat/formel/koseleff/Spad/spad7/FLA.NRLIB for category
   FreeLieAlgebra
   loading /cmat/formel/koseleff/Spad/spad7/FLA-.NRLIB for domain
   FreeLieAlgebra&
   loading /cmat/formel/koseleff/Spad/spad7/MG.NRLIB for domain Magma
   loading /cmat/formel/koseleff/Spad/spad7/LFLA.NRLIB for domain
   LyndonFreeLieAlgebra
   loading /cmat/formel/koseleff/Spad/spad7/HFLA.NRLIB for domain
   HallFreeLieAlgebra
   loading /cmat/formel/koseleff/Spad/spad7/LS.NRLIB for package LieSeries
```

We give a list of abbreviations:

```
macro ST == String

   (29)  STRING
                                                              Type: DOMAIN
                                                               Time: 0 sec
macro SE == Symbol

   (30)  SYMBOL
                                                              Type: DOMAIN
                                                               Time: 0 sec
macro LSE == List SE

   (31)  LIST SYMBOL
                                                              Type: DOMAIN
                                                Time: 0.01 (IN) = 0.01 sec
macro O == OUTFORM

   (32)  OUTFORM
                                                              Type: DOMAIN
                                                Time: 0.01 (OT) = 0.01 sec
macro R == POLY FRAC INT

   (33)  POLY FRAC INT
                                                              Type: DOMAIN
                                                               Time: 0 sec
macro L == LFLA(R,SE)

   (34)  LFLA(POLY FRAC INT,SYMBOL)
                                                              Type: DOMAIN
                                    Time: 0.01 (IN) + 0.01 (OT) = 0.02 sec
macro H == HFLA(R,SE)

   (35)  HFLA(POLY FRAC INT,SYMBOL)
                                                              Type: DOMAIN
                                    Time: 0.01 (IN) + 0.01 (OT) = 0.02 sec
macro LL ==LS(R,L)

   (36)  LS(POLY FRAC INT,LFLA(POLY FRAC INT,SYMBOL))
                                                              Type: DOMAIN
                                    Time: 0.01 (IN) + 0.01 (OT) = 0.02 sec
macro M == MG SE
```

```
   (38)  MG SYMBOL
                                                          Type: DOMAIN
                                              Time: 0.01 (IN) = 0.01 sec
```

We give ourselves an element of the magma:

```
a:M:=a

   (40)  a
                                                      Type: MG SYMBOL
                                 Time: 0.01 (IN) + 0.01 (OT) = 0.02 sec
b:M:=b

   (41)  b
                                                      Type: MG SYMBOL
                                              Time: 0.02 (OT) = 0.02 sec
c:M:=c

   (42)  c
                                                      Type: MG SYMBOL
                                              Time: 0.01 (IN) = 0.01 sec

m:=a*b*c*c*b*a*a*b*c

   (44)  [[[[[[[[a,b],c],c],b],a],a],b],c]
                                                      Type: MG SYMBOL
                                 Time: 0.06 (IN) + 0.02 (OT) = 0.08 sec
```

We decompose this element onto both bases, –we can actually find out the number of monomials:

```
mh:=m::H;
                                      Type: HFLA(POLY FRAC INT,SYMBOL)
                  Time: 4.54 (IN) + 0.04 (OT) + 0.59 (GC) = 5.17 sec
ml:=m::L;
                                      Type: LFLA(POLY FRAC INT,SYMBOL)
                                              Time: 0.51 (IN) = 0.51 sec
size mh

   (47)  128
                                                               Type: PI
                                              Time: 0.06 (OT) = 0.06 sec
size ml

   (48)  48
                                                               Type: PI
                                              Time: 0.01 (OT) = 0.01 sec
```

Finally, we can verify that both decompositions coincide:

```
reduce(+,[u*(v::H) for u in listOfCoefficient ml for v in
listOfMonomials ml])-mh

   (49)  0
                                      Type: HFLA(POLY FRAC INT,SYMBOL)
```

```
                Time: 0.52 (IN) + 51.45 (EV) + 0.25 (OT) + 4.34 (GC) = 56.56 sec
reduce(+,[u*(v::L) for u in listOfCoefficient mh for v in listOfMonomials mh])-ml

   (50)  0
                                                 Type: LFLA(POLY FRAC INT,SYMBOL)
                Time: 0.29 (IN) + 19.33 (EV) + 0.01 (OT) + 2.45 (GC) = 22.08 sec
```

We start over with another monomial:

```
m:=c*m

   (52)  [c,[[[[[[[[a,b],c],c],b],a],a],b],c]]
                                                                 Type: MG SYMBOL
                                                       Time: 0.03 (OT) = 0.03 sec
mh:=m::H;

                                                 Type: HFLA(POLY FRAC INT,SYMBOL)
                                         Time: 17.20 (IN) + 0.59 (GC) = 17.79 sec
ml:=m::L;

                                                 Type: LFLA(POLY FRAC INT,SYMBOL)
                                           Time: 3.39 (IN) + 0.60 (GC) = 3.99 sec
size mh

   (55)  229

   Type: PI
                                           Time: 0.01 (IN) + 0.01 (OT) = 0.02 sec
size ml

   (56)  153
                                                                        Type: PI
                                                       Time: 0.01 (IN) = 0.01 sec

reduce(+,[u*(v::H) for u in listOfCoefficient ml for v in listOfMonomials ml])-mh

   (57)  0
                                                 Type: HFLA(POLY FRAC INT,SYMBOL)
            Time: 0.03 (IN) + 249.33 (EV) + 0.03 (OT) + 15.79 (GC) = 265.18 sec


reduce(+,[u*(v::L) for u in listOfCoefficient mh for v in listOfMonomials mh])-ml

   (58)  0
                                                 Type: LFLA(POLY FRAC INT,SYMBOL)
                Time: 0.03 (IN) + 75.10 (EV) + 0.02 (OT) + 1.59 (GC) = 76.74 sec
```

Based on this example, it appears that the decomposition onto the Lyndon basis is faster than the decomposition onto the Hall basis.

B.2. Computation of Symplectic Integrators

Here, we compute the polynomial relations to be solved, in order to get reversible symplectic integrators as the product of integrators of order 2.

If $K = \sum_{a_{i,j}} k_{i,j}$ and $S_2(t) = \exp(\sum_{i\geq 1} t^{2i+1} L_{h_{2i+1}}$, we compute

$$\log\left[S_2(xt)e^{tL_K}S_2(xt)\right]. \tag{B.1}$$

The first step is the computation of the Lyndon basis for odd orders

```
kk:=mkSymbol(h,1,9)$LL

   (59)  [h ,h ,h ,h ,h ,h ,h ,h ,h ]
           1  2  3  4  5  6  7  8  9
                                                       Type: LIST SYMBOL
                            Time: 0.03 (IN) + 0.03 (EV) + 0.07 (OT) = 0.13 sec
kkk:=[list u for u in kk]

   (60)  [[h ],[h ],[h ],[h ],[h ],[h ],[h ],[h ],[h ]]
            1    2    3    4    5    6    7    8    9
                                                  Type: LIST LIST SYMBOL
                            Time: 0.02 (IN) + 0.01 (EV) + 0.07 (OT) = 0.10 sec
for i in 1..4 repeat kkk.(2*i) := []
                                                              Type: VOID
                            Time: 0.38 (IN) + 0.01 (EV) + 0.06 (OT) = 0.45 sec

bk:=weightedBasis(kkk,9)$L

   (62)
   [[h ], [], [h ], [[h ,h ]], [[h ,[h ,h ]],h ], [[h ,[h ,[h ,h]]],[h ,h ]],
      1        3      1  3        1   1  3    5       1   1   1  3     1  5
    [[h ,[h ,[h ,[h ,h ]]]],[h ,[h ,h ]],[[h ,h ],h ],h ],
       1   1   1   1  3       1   1  5      1  3   3   7
      [h ,[h ,h ]], [[h ,h ],h ], h ] ]
        1   3  5       1  5   3    9
                                               Type: LIST LIST MG SYMBOL
                            Time: 0.01 (IN) + 0.08 (EV) + 0.34 (OT) = 0.43 sec
for i in 1..4 repeat bk.(2*i) := []
                                                              Type: VOID
                                                 Time: 0.13 (IN) = 0.13 sec
bk:=[reverse u for u in bk];
                                               Type: LIST LIST MG SYMBOL
                                     Time: 0.01 (IN) + 0.18 (OT) = 0.19 sec
```

Next, we compute a generic Lie polynomial

```
ck:List List SE:=[[subscript(a,[i,j]) for j in 1..#(bk.i)] for i in 1..9]

   (65)
   [[a   ], [], [a   ], [], [a   ,a   ], [], [a   ,a   ,a   ,a   ], [],
      1,1          3,1          5,1  5,2          7,1  7,2  7,3  7,4

    [a   ,a   ,a   ,a   ,a   ,a   ,a   ,a   ]]
      9,1  9,2  9,3  9,4  9,5  9,6  9,7  9,8
                                                  Type: LIST LIST SYMBOL
                Time: 0.28 (IN) + 0.35 (EV) + 0.85 (OT) + 0.75 (GC) = 2.23 sec
hk:=[[(u::R)*(v::L) for u in xx for v in yy] for xx in ck for yy in bk];

                                  Type: LIST LIST LFLA(POLY FRAC INT,SYMBOL)
```

```
                              Time: 0.03 (IN) + 0.11 (EV) + 0.49 (OT) = 0.63 sec
hh:=concat(0$L,[reduce(+,u,0$L) for u in hk])

   (67)
   [a    h , 0, a    h , 0, a    h  + a    [h ,[h ,h ]], 0,
     1,1 1       3,1 3       5,2 5     5,1  1   1  3
    a    h  + a    [h ,[h ,h ]] + a    [[h ,h ],h ] + a    [h ,[h ,[h ,[h ,h ]]]],
     7,4 7     7,2  1   1  5       7,3   1  3   3      7,1  1   1   1   1  3
    0,

      a    h  + a    [h ,[h ,h ]] + a    [h ,[h ,h ]] + a    [[h ,h ],h ]
       9,8 9     9,5  1   1  7       9,6  1   3  5       9,7   1  5   3
    +
      a    [h ,[h ,[h ,[h ,h ]]]] + a    [h ,[h ,[[h ,h ],h ]]]
       9,2  1   1   1   1  5         9,3  1   1    1  3   3
    +
      a    [[h ,[h ,h ]],[h ,h ]] + a    [h ,[h ,[h ,[h ,[h ,[h ,h ]]]]]] ]
       9,4   1   1  3     1  3       9,1  1   1   1   1   1   1  3

                                         Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                              Time: 0.05 (IN) + 0.01 (EV) + 0.33 (OT) = 0.39 sec
```

We compute $S_2(xt)\exp(tL_k)S_2(xt)a$

```
ha:=[a::M::L]

   (69)  [a]
                                         Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                                        Time: 0.02 (IN) + 0.01 (OT) = 0.03 sec
ha:=concat(ha,[0$L for i in 1..9])

   (70)  [a,0,0,0,0,0,0,0,0,0]
                                         Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                              Time: 0.02 (IN) + 0.02 (EV) + 0.04 (OT) = 0.08 sec
hx:=[(x::R)**i * (v::L) for i in 1..9 for v in kk]

                2    3    4    5    6    7    8    9
   (71)  [x h ,x h ,x h ,x h ,x h ,x h ,x h ,x h ,x h ]
             1    2    3    4    5    6    7    8    9
                                         Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                              Time: 0.46 (IN) + 0.04 (EV) + 0.56 (OT) = 1.06 sec
for i in 1..4 repeat hx.(2*i) := 0$L
                                                                Type: VOID
                              Time: 0.03 (IN) + 0.01 (EV) + 0.01 (OT) = 0.05 sec
hx:=concat(0$L,hx)

                    3       5       7       9
   (73)  [0,x h ,0,x h ,0,x h ,0,x h ,0,x h ]
               1       3       5       7       9
                                         Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                                        Time: 0.01 (IN) + 0.03 (OT) = 0.04 sec
res:=exponential(hx,exponential(hh,exponential(hx,ha)));

                                         Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                  Time: 0.02 (IN) + 17.66 (EV) + 0.04 (OT) + 0.79 (GC) = 18.51 sec
```

which we compare to $\exp(t\sum_{L_{k_i}})a$

```
lkl:=concat(0$L,mkLiePol(k,1,9)$LL)

   (75)  [k ,k ,k ,k ,k ,k ,k ,k ,k ]
           1  2  3  4  5  6  7  8  9
                                          Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                              Time: 0.01 (IN) + 0.02 (EV) + 0.06 (OT) = 0.09 sec
exp:=exponential(lkl,ha);

                                          Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                                         Time: 0.02 (IN) + 8.54 (EV) = 8.56 sec
lk:=mkLieMon(k,1,9)$LL

   (78)  [k ,k ,k ,k ,k ,k ,k ,k ,k ]
           1  2  3  4  5  6  7  8  9
                                                         Type: LIST MG SYMBOL

triangularFactorSystemSolve(first(lk,9),rest exp,rest res,a)

                                          Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                              Time: 0.02 (IN) + 5.91 (EV) + 0.02 (OT) = 5.95 sec

triangularFactorSystemSolve(first(lk,5),rest exp,rest res,a)
   (79)
                                  3
   [(a     + 2x)h , 0, (a     + 2x )h , 0,
      1,1        1        3,1         3

                 5
      (a     + 2x )h
        5,2         5
    +
                  1          1  2        1  3   2  1  4
      (a     + (- - x a    - - x )a    + - x a    + - x a    )[h ,[h ,h ]] ]
        5,1       6    1,1  6     3,1   6    1,1   6    1,1   1   1  3

                                          Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                              Time: 0.01 (IN) + 0.21 (EV) + 0.11 (OT) = 0.33 sec
```

We have thus found the relations, which we now save in a file. The list of the polynomials is called `lp2`.

***

The following step is the construction of a function which computes the various relations, if we search for

$$S_r = S_2(c_n t)\cdots S_2(c_1 t)\cdots S_2(c_n t). \tag{B.2}$$

In fact, for technical reasons, we pose $c_1 = a, c_2 = b$ etc. `r` denotes the number of polynomials to compute. In order to have an integrator of order 4, one will need to take $r = 2$, $r = 4$, for an integrator of order 6, and $r = 8$, for an integrator of order 8.

```
eq2b(n:NNI,r:NNI):List R ==
  ch:R:=char(97)::ST::SE::R
```

```
  l:List R:=first([leadingMonomial u for u in lp2],r)
  lv:=reduce(append,[variables(u) for u in l])
  lv:=concat(lv,'b)
  l:=first([ch,ch**3,ch**5,0,ch**7,0,0,0,ch**9,0,0,0,0,0,0,0],r)
  for i in 2..n repeat
     ch:=char(96+i)::ST::SE::R
     l1:=concat(l,ch)
     l:=[eval(u,lv,l1) for u in first(lp4,r)]
  l.1 := l.1 -1
  l
```

For example, we get, for the various orders

```
(108) ->eq2b(2,2)
   loading /axiom/mnt/rios/algebra/VARIABLE.NRLIB for domain
Variable
   Compiling function eq2b with type (NNI,NNI) -> LIST POLY FRAC INT

                           3    3
   (108)  [2b + a - 1,2b  + a ]
                                                  Type: LIST POLY FRAC INT
                                 Time: 0.92 (IN) + 0.12 (EV) + 0.42 (OT) = 1.46 sec
(109) ->eq2b(4,4)

   (109)
                               3     3     3    3     5     5     5    5
   [2d + 2c + 2b + a - 1, 2d  + 2c  + 2b  + a , 2d  + 2c  + 2b  + a ,

       1     1     1    4    2  2    4     2       2  2   2      1  2  3
      (- c + - b + - a)d  + (- c  + (- b + - a)c + - b  + - a b + - a )d
       3     3     6         3       3     3       3      3       6
    +
         1  3   1  3   1  3  2
      (- - c  - - b  - - a )d
         3      3      6
    +
           2  4     2     1     3     2  3   1  3    2  4   1    3   1  3
         - - c  + (- - b - - a)c  + (- - b  - - a )c - - b  - - a b  - - a b
           3        3     3           3      3         3      3        3
       +
           1  4
         - - a
           6
     *
        d
    +
      1     1    4   2  2   2       1  2  3     1  3   1  3  2
     (- b + - a)c  + (- b  + - a b + - a )c  + (- - b  - - a )c
      3     6        3      3       6            3      6
    +
        2  4   1    3   1  3    1  4    1    4   1  2 3   1  3 2   1  4
     (- - b  - - a b  - - a b - - a )c + - a b  + - a b  - - a b  - - a b ]
        3      3        3       6        6        6        6        6

                                                  Type: LIST POLY FRAC INT
```

```
Time: 0.09 (EV) + 0.37 (OT) = 0.46 sec
```

The **groebner** function of AXIOM enables us to compute a standard basis for the elimination order

```
groebner %;

                                                 Type: LIST POLY FRAC INT
       Time: 0.03 (IN) + 202.51 (EV) + 0.24 (OT) + 41.89 (GC) = 244.67 sec
(24) ->%.last
```

(24)

$$
\begin{aligned}
&a^{39} + 4a^{38} - 18a^{37} - \frac{232}{3}a^{36} + \frac{6469}{45}a^{35} + \frac{8108}{15}a^{34} - \frac{82144}{135}a^{33} - \frac{239008}{135}a^{32} \\
&+ \frac{870652}{675}a^{31} + \frac{5898416}{2025}a^{30} - \frac{618824}{675}a^{29} - \frac{5158016}{2025}a^{28} + \frac{2525372}{30375}a^{27} \\
&+ \frac{32135888}{30375}a^{26} - \frac{1377776}{10125}a^{25} - \frac{33361568}{91125}a^{24} + \frac{536566}{10125}a^{23} + \frac{35651416}{455625}a^{22} \\
&+ - \frac{19660868}{1366875}a^{21} - \frac{8051504}{455625}a^{20} + \frac{5636474}{1366875}a^{19} + \frac{11313208}{4100625}a^{18} - \frac{17674448}{20503125}a^{17} \\
&+ - \frac{8733536}{20503125}a^{16} + \frac{1302268}{6834375}a^{15} + \frac{87632}{2460375}a^{14} - \frac{624184}{20503125}a^{13} + \frac{288448}{922640625}a^{12} \\
&+ \frac{3333844}{922640625}a^{11} - \frac{716752}{922640625}a^{10} - \frac{127664}{553584375}a^{9} + \frac{143264}{922640625}a^{8} - \frac{136499}{4613203125}a^{7} \\
&+ - \frac{19996}{8303765625}a^{6} + \frac{117142}{41518828125}a^{5} - \frac{33848}{41518828125}a^{4} + \frac{17431}{124556484375}a^{3} \\
&+ - \frac{9668}{622782421875}a^{2} + \frac{656}{622782421875}a - \frac{64}{1868347265625}
\end{aligned}
$$

```
                                                      Type: POLY FRAC INT
                        Time: 0.02 (IN) + 0.01 (EV) + 0.44 (OT) = 0.47 sec
```

Since the other polynomials are of the form $b = P_b(a), c = P_c(a), d = P_d(a)$, we deduce that the variety of the solutions is of null dimension, and contains 39 points.

Using the functions defined in the package of Lie series, we can compute the relations which must be satisfied by a reversible product of 15 (2.8-1) operators:

```
res:=symplecticr(8,5);
                                      Type: LIST LFLA(POLY FRAC INT,SYMBOL)
                        Time: 0.02 (IN) + 3.25 (EV) + 1.27 (GC) = 4.54 sec
res.4:=0$L;
```

```
lp:=reduce(append([[listOfCoefficient x for x in res]]));
res.1:=res.1 - 1;
res.2:=res.2 - 1;
```

The result, which is a list of polynomials, is converted in MACAULAY format, and we find a list of polynomials with integer coefficients

```
; 2D+2C+2B+2A-1 2d+2c+2b+a-1

; -2D2d-4DCd+4C2d-4DBd+8CBd+4B2d-4DAd+8CAd+8BAd+4A2d-2D2c-4DCc-2C2c-4DBc-4CBc \
;     +4B2c-4DAc-4CAc+8BAc+4A2c-2D2b-4DCb-2C2b-4DBb-4CBb-2B2b-4DAb-4CAb-4BAb \
;     +4A2b-D2a-2DCa-C2a-2DBa-2CBa-B2a-2DAa-2CAa-2BAa-A2a

; 4Dd2-2Cd2-2Bd2-2Ad2+8Ddc-4Cdc-4Bdc-4Adc+4Dc2+4Cc2-2Bc2-2Ac2+8Ddb-4Cdb-4Bdb \
;     -4Adb+8Dcb+8Ccb-4Bcb-4Acb+4Db2+4Cb2+4Bb2-2Ab2+4Dda-2Cda-2Bda-2Ada+4Dca \
;     +4Cca-2Bca-2Aca+4Dba+4Cba+4Bba-2Aba+Da2+Ca2+Ba2+Aa2

; -2D3d-6D2Cd+4C3d-6D2Bd+12C2Bd+12CB2d+4B3d-6D2Ad+12C2Ad+24CBAd+12B2Ad+12CA2d \
;     +12BA2d+4A3d-2D3c-6D2Cc-6DC2c-2C3c-6D2Bc-12DCBc-6C2Bc+4B3c-6D2Ac-12DCAc \
;     -6C2Ac+12B2Ac+12BA2c+4A3c-2D3b-6D2Cb-6DC2b-2C3b-6D2Bb-12DCBb-6C2Bb-6DB2b \
;     -6CB2b-2B3b-6D2Ab-12DCAb-6C2Ab-12DBAb-12CBAb-6B2Ab+4A3b-D3a-3D2Ca-3DC2a \
;     -C3a-3D2Ba-6DCBa-3C2Ba-3DB2a-3CB2a-B3a-3D2Aa-6DCAa-3C2Aa-6DBAa-6CBAa \
;     -3B2Aa-3DA2a-3CA2a-3BA2a-A3a

; 4D2d2+4DCd2-4C2d2+4DBd2-8CBd2-4B2d2+4DAd2-8CAd2-8BAd2-4A2d2+8D2dc+8DCdc \
;     -4C2dc+8DBdc-8CBdc-8B2dc+8DAdc-8CAdc-16BAdc-8A2dc+4D2c2+8DCc2+4C2c2 \
;     +4DBc2+4CBc2-4B2c2+4DAc2+4CAc2-8BAc2-4A2c2+8D2db+8DCdb-4C2db+8DBdb-8CBdb \
;     -4B2db+8DAdb-8CAdb-8BAdb-8A2db+8D2cb+16DCcb+8C2cb+8DBcb+8CBcb-4B2cb \
;     +8DAcb+8CAcb-8BAcb-8A2cb+4D2b2+8DCb2+4C2b2+8DBb2+8CBb2+4B2b2+4DAb2+4CAb2 \
;     +4BAb2-4A2b2+4D2da+4DCda-2C2da+4DBda-4CBda-2B2da+4DAda-4CAda-4BAda-2A2da \
;     +4D2ca+8DCca+4C2ca+4DBca+4CBca-2B2ca+4DAca+4CAca-4BAca-2A2ca+4D2ba+8DCba \
;     +4C2ba+8DBba+8CBba+4B2ba+4DAba+4CAba+4BAba-2A2ba+D2a2+2DCa2+C2a2+2DBa2 \
;     +2CBa2+B2a2+2DAa2+2CAa2+2BAa2+A2a2
```

B.3. A MACAULAY Session

Having initialized the MACAULAY environment, we can compute the Hilbert function of the ideal defined by the previous polynomials. We proceed in characteristic 31991.

```
<inhomog_std j w

hilb w

; codimension = 8
; degree      = 39
; genus       = 94
```

Hence we deduce that the variety thus defined is null in dimension of degree 39. All these solutions were found with reversible products of integrators of order 2.

With the same software, we can study the variety of the solutions in the search for an integrator of order 5, which is expressed as a product of 11 operators. We find in 140 minutes that the variety is of null dimension, and of degree 46.

```
1% hilb w

; codimension = 11
```

```
; degree        = 46
; genus         = 990

1% type w

; A46-90/77A45-4/105A44+6268A43+8040A42+74/95A41-12607A40+35/37A39+79/59A38 \
;     -837A37+66/115A36+4205A35+8812A34-34/27A33+207A32+4540A31+9576A30+53A29 \
;     -65/21A28+31/27A27-92A26-111/121A25-14668A24-143A23-87/80A22+31/108A21 \
;     -9910A20+55/63A19-14/115A18-4122A17-9707A16+52/51A15+2771A14+1/104A13 \
;     +83/109A12+95/102A11+45/17A10+111/103A9-101/63A8+3436A7-103/23A6+86/65A5 \
;     -98/97A4-2300A3-9386A2-13527A-107/119

; B+13653A45+52/5A44+66/103A43+4/15A42-12273A41-119/37A40+109/89A39+1178A38 \
;     -10603A37+7445A36+11306A35+124/119A34-8353A33-6566A32+114/43A31-37/84A30 \
;     -89/91A29+1787A28-68/65A27+85/53A26-21/2A25-21/61A24+85/74A23+676A22 \
;     -3/121A21-66/41A20+21/100A19+81/116A18+53/13A17-85/101A16-125/114A15 \
;     +11931A14-6290A13+103/95A12+89/97A11+57/56A10+1570A9+9122A8-9751A7 \
;     -18/7A6-14704A5+15899A4+124/55A3-111/52A2+13/82A+3983

; C-44/15BA22-16/47BA21+3905BA20-14321BA19-2/63BA18+8922BA17+35/59BA16 \
;     +37/8BA15+97/38BA14+65/124BA13+8774BA12+117/80BA11+11809BA10+6576BA9 \
;     +103/66BA8+9173BA7-837BA6-55/7BA5+1694BA4-83/41BA3-1158BA2-20/91BA \
;     +16/113B+8775A22+32/93A21+6949A20-10544A19-67/45A18-13002A17+103/38A16 \
;     -93/121A15+46/19A14-6007A13-63A12-49/92A11-11775A10+70/59A9-79/76A8 \
;     +8605A7+5509A6-75/49A5+113/116A4+9005A3-68/105A2+12163A-71/30

; D+108/47CA14-61/58CA13+119/47CA12+118/99CA11+81/76CA10-93/85CA9-7658CA8 \
;     +5127CA7+57/13CA6+67/122CA5+6967CA4+119/69CA3+8232CA2-99/25CA-31/115C \
;     -67/109BA15+9398BA14-53/52BA13+43/16BA12-5888BA11-10901BA10+12945BA9 \
;     +6692BA8+4395BA7-7687BA6-64/123BA5+79/70BA4-15805BA3+73/97BA2+89/124BA \
;     -25/88B+108/47A15+22/57A14-11084A13+55/117A12-19/30A11-109/55A10 \
;     -114/115A9+11/97A8+24/59A7-5/86A6+61/123A5+7861A4-37/64A3+11759A2-1868A \
;     -107/103

; E+C+A-1/2 F+D+B-1/2 a-2A b-2B+2A c-2C+2B-2A d-2D+2C-2B+2A e+2D+2B-1
```

C. Relation of Exponential Type

C.1. Campbell-Hausdorff Formula

$$
\begin{aligned}
\log(e^y e^x) = {} & x + y - \tfrac{1}{2}\,[x,\,y] + \tfrac{1}{12}\,[x,\,[x,\,y]] + \tfrac{1}{12}\,[[x,\,y],\,y] - \tfrac{1}{24}\,[x,\,[[x,\,y],\,y]] \\
& - \tfrac{1}{720}\,[x,\,[x,\,[x,\,[x,\,y]]]] + \tfrac{1}{180}\,[x,\,[x,\,[[x,\,y],\,y]]] + \tfrac{1}{180}\,[x,\,[[[x,\,y],\,y],\,y]] + \tfrac{1}{120}\,[[x,\,y],\,[[x,\,y],\,y]] \\
& + \tfrac{1}{360}\,[[x,\,[x,\,y]],\,[x,\,y]] - \tfrac{1}{720}\,[[[[x,\,y],\,y],\,y],\,y] + \tfrac{1}{1440}\,[x,\,[x,\,[x,\,[[x,\,y],\,y]]]] \\
& - \tfrac{1}{360}\,[x,\,[x,\,[[[x,\,y],\,y],\,y]]] - \tfrac{1}{240}\,[x,\,[[x,\,y],\,[[x,\,y],\,y]]] - \tfrac{1}{720}\,[x,\,[[x,\,[x,\,y]],\,[x,\,y]]] \\
& + \tfrac{1}{1440}\,[x,\,[[[[x,\,y],\,y],\,y],\,y]] + \tfrac{1}{30240}\,[x,\,[x,\,[x,\,[x,\,[x,\,[x,\,y]]]]]] - \tfrac{1}{5040}\,[x,\,[x,\,[x,\,[x,\,[[x,\,y],\,y]]]]] \\
& + \tfrac{1}{3780}\,[x,\,[x,\,[x,\,[[[x,\,y],\,y],\,y]]]] + \tfrac{1}{1680}\,[x,\,[x,\,[[x,\,y],\,[[x,\,y],\,y]]]] \\
& + \tfrac{1}{10080}\,[x,\,[x,\,[[x,\,[x,\,y]],\,[x,\,y]]]] + \tfrac{1}{3780}\,[x,\,[x,\,[[[[x,\,y],\,y],\,y],\,y]]] \\
& + \tfrac{13}{15120}\,[x,\,[[x,\,y],\,[[[x,\,y],\,y],\,y]]] + \tfrac{1}{1260}\,[x,\,[[x,\,[[x,\,y],\,y]],\,[x,\,y]]] \\
& - \tfrac{1}{5040}\,[x,\,[[[[[x,\,y],\,y],\,y],\,y],\,y]] + \tfrac{1}{1260}\,[[x,\,y],\,[[x,\,y],\,[[x,\,y],\,y]]] \\
& - \tfrac{1}{2016}\,[[x,\,y],\,[[[[x,\,y],\,y],\,y],\,y]] + \tfrac{1}{2016}\,[[x,\,[x,\,y]],\,[x,\,[[x,\,y],\,y]]] \\
& - \tfrac{1}{5040}\,[[[x,\,y],\,y],\,[[[x,\,y],\,y],\,y]] + \tfrac{1}{10080}\,[[x,\,[x,\,[x,\,y]]],\,[x,\,[x,\,y]]]
\end{aligned}
$$

$$+ \tfrac{1}{10080} [[x, [[x, y], y]], [[x, y], y]] - \tfrac{1}{1512} [[x, [[[x, y], y], y]], [x, y]]$$
$$- \tfrac{1}{5040} [[[x, [x, y]], [x, y]], [x, y]] + \tfrac{1}{30240} [[[[[x, y], y], y], y], y], y]$$
$$- \tfrac{1}{60480} [x, [x, [x, [x, [x, [[x, y], y]]]]]] + \tfrac{1}{10080} [x, [x, [x, [x, [[[x, y], y], y]]]]]$$
$$+ \tfrac{1}{20160} [x, [x, [x, [[x, y], [[x, y], y]]]]] + \tfrac{1}{15120} [x, [x, [x, [[x, [x, y]], [x, y]]]]]$$
$$- \tfrac{23}{120960} [x, [x, [x, [[[[x, y], y], y], y]]]] - \tfrac{13}{30240} [x, [x, [[x, y], [[[x, y], y], y]]]]$$
$$- \tfrac{1}{2520} [x, [x, [[x, [[x, y], y]], [x, y]]]] + \tfrac{1}{10080} [x, [x, [[[[[x, y], y], y], y], y]]]$$
$$- \tfrac{1}{2520} [x, [[x, y], [[x, y], [[x, y], y]]]] + \tfrac{1}{4032} [x, [[x, y], [[[[x, y], y], y], y]]]$$
$$- \tfrac{1}{4032} [x, [[x, [x, y]], [x, [[x, y], y]]]] + \tfrac{1}{10080} [x, [[[x, y], y], [[[x, y], y], y]]]$$
$$- \tfrac{1}{20160} [x, [[x, [x, [x, y]]], [x, [x, y]]]] - \tfrac{1}{20160} [x, [[x, [[x, y], y]], [[x, y], y]]]$$
$$+ \tfrac{1}{3024} [x, [[x, [[[x, y], y], y]], [x, y]]] + \tfrac{1}{10080} [x, [[[x, [x, y]], [x, y]], [x, y]]]$$
$$- \tfrac{1}{60480} [x, [[[[[[x, y], y], y], y], y], y]] - \tfrac{1}{1209600} [x, [x, [x, [x, [x, [x, [x, [x, y]]]]]]]]$$
$$+ \tfrac{1}{151200} [x, [x, [x, [x, [x, [x, [[x, y], y]]]]]]] - \tfrac{1}{56700} [x, [x, [x, [x, [x, [[[x, y], y], y]]]]]]$$
$$- \tfrac{1}{43200} [x, [x, [x, [x, [[x, y], [[x, y], y]]]]]] - \tfrac{1}{100800} [x, [x, [x, [x, [[x, [x, y]], [x, y]]]]]]$$
$$+ \tfrac{1}{75600} [x, [x, [x, [x, [[[[x, y], y], y], y]]]]] + \tfrac{11}{302400} [x, [x, [x, [[x, y], [[[x, y], y], y]]]]]$$
$$+ \tfrac{1}{75600} [x, [x, [x, [[x, [[x, y], y]], [x, y]]]]] + \tfrac{1}{75600} [x, [x, [x, [[[[[x, y], y], y], y], y]]]]$$
$$+ \tfrac{1}{100800} [x, [x, [[x, y], [[x, y], [[x, y], y]]]]] + \tfrac{1}{20160} [x, [x, [[x, y], [[[[x, y], y], y], y]]]]$$
$$+ \tfrac{1}{67200} [x, [x, [[x, [x, y]], [x, [[x, y], y]]]]] + \tfrac{1}{30240} [x, [x, [[[x, y], y], [[[x, y], y], y]]]]$$
$$+ \tfrac{11}{201600} [x, [x, [[x, [[x, y], y]], [[x, y], y]]]] + \tfrac{11}{151200} [x, [x, [[x, [[[x, y], y], y]], [x, y]]]]$$
$$+ \tfrac{1}{43200} [x, [x, [[[x, [x, y]], [x, y]], [x, y]]]] - \tfrac{1}{56700} [x, [x, [[[[[[x, y], y], y], y], y], y]]]$$
$$+ \tfrac{1}{6048} [x, [[x, y], [[x, y], [[[x, y], y], y]]]] - \tfrac{23}{302400} [x, [[x, y], [[[[[x, y], y], y], y], y]]]$$
$$+ \tfrac{23}{302400} [x, [[x, [x, y]], [x, [[[x, y], y], y]]]] + \tfrac{1}{37800} [x, [[x, [x, y]], [[x, [x, y]], [x, y]]]]$$
$$- \tfrac{11}{120960} [x, [[[x, y], y], [[[[x, y], y], y], y]]] + \tfrac{1}{25200} [x, [[x, [x, [[x, y], y]]], [x, [x, y]]]]$$
$$- \tfrac{1}{15120} [x, [[x, [[[x, y], y], y]], [[x, y], y]]] - \tfrac{1}{20160} [x, [[[x, y], [[x, y], y]], [[x, y], y]]]$$
$$+ \tfrac{1}{33600} [x, [[x, [[x, y], [[x, y], y]]], [x, y]]] - \tfrac{1}{7560} [x, [[x, [[[[x, y], y], y], y]], [x, y]]]$$
$$- \tfrac{17}{100800} [x, [[[x, [[x, y], y]], [x, y]], [x, y]]] + \tfrac{1}{151200} [x, [[[[[[[x, y], y], y], y], y], y], y]]$$
$$+ \tfrac{1}{10080} [[x, y], [[x, y], [[x, y], [[x, y], y]]]] - \tfrac{1}{7560} [[x, y], [[x, y], [[[[x, y], y], y], y]]]$$
$$- \tfrac{1}{15120} [[x, y], [[[x, y], y], [[[x, y], y], y]]] + \tfrac{1}{43200} [[x, y], [[[[[[x, y], y], y], y], y], y]]$$
$$+ \tfrac{1}{25200} [[x, [x, y]], [x, [[x, y], [[x, y], y]]]] - \tfrac{1}{17280} [[x, [x, y]], [x, [[[[x, y], y], y], y]]]$$
$$- \tfrac{1}{8400} [[x, [x, y]], [[x, [[x, y], y]], [x, y]]] + \tfrac{1}{33600} [[[x, y], y], [[[[x, y], y], y], y], y]]$$
$$+ \tfrac{1}{43200} [[x, [x, [x, y]]], [x, [x, [[x, y], y]]]] + \tfrac{1}{60480} [[x, [[x, y], y]], [x, [[[x, y], y], y]]]$$
$$+ \tfrac{1}{20160} [[x, [[x, y], y]], [[x, y], [[x, y], y]]] + \tfrac{1}{60480} [[[[x, y], y], y], [[[[x, y], y], y], y]]$$
$$+ \tfrac{1}{302400} [[x, [x, [x, [x, y]]]], [x, [x, [x, y]]]] + \tfrac{1}{67200} [[x, [x, [[x, y], y]]], [x, [[x, y], y]]]$$
$$+ \tfrac{1}{90720} [[x, [[[x, y], y], y]], [[[x, y], y], y]] - \tfrac{1}{25200} [[[x, [x, y]], [x, y]], [x, [[x, y], y]]]$$
$$- \tfrac{1}{21600} [[x, [x, [[[x, y], y], y]]], [x, [x, y]]] - \tfrac{1}{30240} [[x, [[x, [x, y]], [x, y]]], [x, [x, y]]]$$
$$+ \tfrac{1}{60480} [[x, [[[[x, y], y], y], y]], [[x, y], y]] + \tfrac{1}{15120} [[[x, y], [[[x, y], y], y]], [[x, y], y]]$$
$$- \tfrac{1}{10080} [[x, [[x, y], [[[x, y], y], y]]], [x, y]] + \tfrac{1}{21600} [[x, [[[[x, y], y], y], y], y]], [x, y]]$$
$$+ \tfrac{1}{20160} [[[x, [[x, y], y]], [[x, y], y]], [x, y]] + \tfrac{1}{10080} [[[x, [[[x, y], y], y]], [x, y]], [x, y]]$$
$$+ \tfrac{1}{50400} [[[[x, [x, y]], [x, y]], [x, y]], [x, y]] - \tfrac{1}{1209600} [[[[[[[x, y], y], y], y], y], y], y], y]$$
$$+ \tfrac{1}{2419200} [x, [x, [x, [x, [x, [x, [x, [[x, y], y]]]]]]]] - \tfrac{1}{302400} [x, [x, [x, [x, [x, [x, [[[x, y], y], y]]]]]]]$$
$$+ \tfrac{1}{604800} [x, [x, [x, [x, [x, [[x, y], [[x, y], y]]]]]]] - \tfrac{1}{403200} [x, [x, [x, [x, [x, [[x, [x, y]], [x, y]]]]]]]$$
$$+ \tfrac{37}{3628800} [x, [x, [x, [x, [x, [[[[x, y], y], y], y]]]]]] + \tfrac{1}{56700} [x, [x, [x, [x, [[x, y], [[[x, y], y], y]]]]]]$$
$$+ \tfrac{1}{37800} [x, [x, [x, [x, [[x, [[x, y], y]], [x, y]]]]]] - \tfrac{1}{67200} [x, [x, [x, [x, [[[[[x, y], y], y], y], y]]]]]$$
$$+ \tfrac{17}{604800} [x, [x, [x, [[x, y], [[x, y], [[x, y], y]]]]]] - \tfrac{11}{241920} [x, [x, [x, [[x, y], [[[[x, y], y], y], y]]]]]$$
$$+ \tfrac{1}{75600} [x, [x, [x, [[x, [x, y]], [x, [[x, y], y]]]]]] - \tfrac{1}{40320} [x, [x, [x, [[[x, y], y], [[[x, y], y], y]]]]]$$
$$+ \tfrac{1}{241920} [x, [x, [x, [[x, [x, [x, y]]], [x, [x, y]]]]]] - \tfrac{1}{43200} [x, [x, [x, [[x, [[x, y], y]], [[x, y], y]]]]]$$
$$- \tfrac{29}{453600} [x, [x, [x, [[x, [[[x, y], y], y]], [x, y]]]]] - \tfrac{1}{50400} [x, [x, [x, [[[x, [x, y]], [x, y]], [x, y]]]]]$$
$$+ \tfrac{37}{3628800} [x, [x, [x, [[[[[[x, y], y], y], y], y], y]]]] - \tfrac{1}{12096} [x, [x, [[x, y], [[x, y], [[[x, y], y], y]]]]]$$

$$
\begin{aligned}
&+ \tfrac{23}{604800}\, [x, [x, [[x, y], [[[[x, y], y], y], y], y]]]] - \tfrac{23}{604800}\, [x, [x, [[x, [x, y]], [x, [[[x, y], y], y]]]]] \\
&- \tfrac{1}{75600}\, [x, [x, [[x, [x, y]], [[x, [x, y]], [x, y]]]]] + \tfrac{11}{241920}\, [x, [x, [[[x, y], y], [[[[x, y], y], y], y]]]] \\
&- \tfrac{1}{50400}\, [x, [x, [[x, [x, [[x, y], y]]], [x, [x, y]]]]] + \tfrac{1}{30240}\, [x, [x, [[x, [[[x, y], y], y]], [[x, y], y]]]] \\
&+ \tfrac{1}{40320}\, [x, [x, [[[x, y], [[x, y], y]], [[x, y], y]]]] - \tfrac{1}{67200}\, [x, [x, [[x, [[x, y], [[x, y], y]]], [x, y]]]] \\
&+ \tfrac{1}{15120}\, [x, [x, [[x, [[[[x, y], y], y], y]], [x, y]]]] + \tfrac{17}{201600}\, [x, [x, [[[x, [[x, y], y]], [x, y]], [x, y]]]] \\
&- \tfrac{1}{302400}\, [x, [x, [[[[[[[x, y], y], y], y], y], y], y]]] - \tfrac{1}{20160}\, [x, [[x, y], [[x, y], [[x, y], [[x, y], y]]]]] \\
&+ \tfrac{1}{15120}\, [x, [[x, y], [[x, y], [[[[x, y], y], y], y]]]] + \tfrac{1}{30240}\, [x, [[x, y], [[[x, y], y], [[[x, y], y], y]]]] \\
&- \tfrac{1}{86400}\, [x, [[x, y], [[[[[[x, y], y], y], y], y], y]]] - \tfrac{1}{50400}\, [x, [[x, [x, y]], [x, [[x, y], [[x, y], y]]]]] \\
&+ \tfrac{1}{34560}\, [x, [[x, [x, y]], [x, [[[[x, y], y], y], y]]]] + \tfrac{1}{16800}\, [x, [[x, [x, y]], [[x, [[x, y], y]], [x, y]]]] \\
&- \tfrac{1}{67200}\, [x, [[[x, y], y], [[[[[x, y], y], y], y], y]]] - \tfrac{1}{86400}\, [x, [[x, [x, [x, y]]], [x, [x, [[x, y], y]]]]] \\
&- \tfrac{1}{120960}\, [x, [[x, [[x, y], y]], [x, [[[x, y], y], y]]]] - \tfrac{1}{40320}\, [x, [[x, [[x, y], y]], [[x, y], [[x, y], y]]]] \\
&- \tfrac{1}{120960}\, [x, [[[[x, y], y], y], [[[[x, y], y], y], y]]] - \tfrac{1}{604800}\, [x, [[x, [x, [x, [x, y]]]], [x, [x, [x, y]]]]] \\
&- \tfrac{1}{134400}\, [x, [[x, [x, [[x, y], y]]], [x, [[x, y], y]]]] - \tfrac{1}{181440}\, [x, [[x, [[[x, y], y], y]], [[[x, y], y], y]]] \\
&+ \tfrac{1}{50400}\, [x, [[[x, [x, y]], [x, y]], [x, [[x, y], y]]]] + \tfrac{1}{43200}\, [x, [[x, [x, [[[x, y], y], y]]], [x, [x, y]]]] \\
&+ \tfrac{1}{60480}\, [x, [[x, [[x, [x, y]], [x, y]]], [x, [x, y]]]] - \tfrac{1}{120960}\, [x, [[x, [[[[x, y], y], y], y]], [[x, y], y]]] \\
&- \tfrac{1}{30240}\, [x, [[[x, y], [[[x, y], y], y]], [[x, y], y]]] + \tfrac{1}{20160}\, [x, [[x, [[x, y], [[[x, y], y], y]]], [x, y]]] \\
&- \tfrac{1}{43200}\, [x, [[x, [[[[[x, y], y], y], y], y]], [x, y]]] - \tfrac{1}{40320}\, [x, [[[x, [[x, y], y]], [[x, y], y]], [x, y]]] \\
&- \tfrac{1}{20160}\, [x, [[[x, [[[x, y], y], y]], [x, y]], [x, y]]] - \tfrac{1}{100800}\, [x, [[[[x, [x, y]], [x, y]], [x, y]], [x, y]]] \\
&+ \tfrac{1}{2419200}\, [x, [[[[[[[[x, y], y], y], y], y], y], y], y]]
\end{aligned}
$$

C.2. Relations Between the Transformation Generators

C.2.1. Compositions of Transformations

In table 4, we give the number of terms and computation time for terms of successive order, for the computation of the composition of transformations. We give, for each order n, the dimension of the module $\tilde{L}_n(X)$, where $X = \{u_1, v_1, \ldots, u_n, v_n, \ldots\}$ and, for any $i \in \mathbb{N}$, we have $|u_i|_* = |v_i|_* = i$, $u_i < v_i < u_{i+1} < v_{i+1}$.

Order	1	2	3	4	5	6	7	8	9	10
Dim. of the Space	2	3	8	18	48	116	312	810	2184	5880
w_i as a function of the u_i, and the v_i in $\cdots\exp(\varepsilon^n L_{w_n})\cdots\exp(\varepsilon L_{w_1}) = [\cdots\exp(\varepsilon^n L_{u_n})\cdots\exp(\varepsilon L_{u_1})]\,[\cdots\exp(\varepsilon^n L_{v_n})\cdots\exp(\varepsilon L_{v_1})]$										
Nb. of Monomials	2	2	4	7	12	27	48	113	229	539
Computation Time		0.21	0.07	0.19	0.55	2.27	9.53	40.54	180.15	798.61
w_i as a function of the u_i and the v_i in $\exp(\varepsilon L_{w_1} + \cdots + \varepsilon^n L_{w_n} + \cdots) = \exp(\varepsilon L_{u_1} + \cdots + \varepsilon^n L_{u_n} + \cdots)\exp(\varepsilon L_{v_1} + \cdots + \varepsilon^n L_{v_n} + \cdots)$										
Nb. of monomials	2	2	4	7	15	33	75	174	413	987
Computation Time		0.10	0.07	0.18	0.74	4.21	13.98	66.95	328.38	1745.93

Table 4

C.2.2. Change in Representation

In table 5, we give the computation times, as well as the number of terms in the computation of the generators for the various transformations. For each order, we give the dimension of $\tilde{L}_n(X)$ where $wX = \{u_1, \ldots, u_n, \ldots\}$, where $|u_i|_* = i$ such that, for any i, $u_i < u_{i+1}$.

Order	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Dimension of the Space	**1**	**1**	**2**	**3**	**6**	**9**	**18**	**30**	**56**	**99**
g_i as a function of the k_i in $\cdots\exp(\varepsilon^n L_{g_n})\cdots\exp(\varepsilon L_{g_1}) = \exp(\varepsilon L_{k_1} + \cdots + \varepsilon^n L_{k_n} + \cdots)$										
Number of monomials	1	1	2	3	6	9	18	29	56	95
Computation Time		0.21	0.04	0.07	0.14	0.31	0.71	1.82	5.25	16.31
w_i as a function of the k_i in $T_{\varepsilon w_1+\cdots\varepsilon^n w_n+\cdots} = \exp(\varepsilon L_{k_1} + \cdots + \varepsilon^n L_{k_n} + \cdots)$										
Number of monomials	1	1	2	3	6	8	18	29	53	90
Computation Time		0.06	0.05	0.10	0.27	0.78	2.28	8.21	20.93	68.88
w_i as a function of the g_i in $T_{\varepsilon w_1+\cdots\varepsilon^n w_n+\cdots} = \cdots\exp(\varepsilon^n L_{g_n})\cdots\exp(\varepsilon L_{g_1})$										
Number of monomials	1	1	2	2	4	5	8	12	19	28
Computation Time		0.04	0.05	0.10	0.26	0.69	3.28	5.93	17.74	56.83
k_i as a function of the g_i in $\exp(\varepsilon L_{k_1} + \cdots + \varepsilon^n L_{k_n} + \cdots) = \cdots\exp(\varepsilon^n L_{g_n})\cdots\exp(\varepsilon L_{g_1})$										
Number of monomials	1	1	2	3	5	7	11	16	23	33
Computation Time		0.04	0.05	0.07	0.13	0.29	0.89	3.83	8.34	27.35
k_i as a function of the w_i in $\exp(\varepsilon L_{k_1} + \cdots + \varepsilon^n L_{k_n} + \cdots) = T_{\varepsilon w_1+\cdots\varepsilon^n w_n+\cdots}$										
Number of monomials	1	1	2	3	6	8	18	26	55	95
Computation Time		0.04	0.03	0.06	0.14	0.30	0.97	4.27	9.86	34.34
g_i as a function of the w_i in $\cdots\exp(\varepsilon^n L_{g_n})\cdots\exp(\varepsilon L_{g_1}) = T_{\varepsilon w_1+\cdots\varepsilon^n w_n+\cdots}$										
Number of monomials	1	1	2	3	6	8	18	28	55	95
Computation Time		0.02	0.04	0.08	0.15	0.29	0.69	1.80	5.16	16.29

Table 5

Résumé : Le but premier de cette thèse est de comparer les deux méthodes de Lie utilisées en théorie des perturbations en mécanique hamiltonienne : la méthode de Deprit et la méthode de Dragt-Finn. Toutes deux reposent sur l'utilisation de transformations canoniques (symplectiques) formelles. On étudie en particulier les liens entre ces deux approches pour conclure qu'elles conduisent, dans les cas non-résonants, à la construction des mêmes objets (forme normale et intégrales premières formelles). On propose, par la suite, des méthodes de construction de tels objets et on observe que la méthode de Dragt-Finn est moins coûteuse.

La connaissance des premiers termes de ces séries de Lie peut fournir des informations sur la stabilité de systèmes hamiltoniens, au voisinage de points d'équilibre elliptiques. On propose, à l'instar de ce qui a été fait avec la transformation de Lie, des estimations pour celle de Dragt-Finn.

Pour étudier les relations entre les différentes transformations canoniques, on se place dans une algèbre de Lie libre pour étudier des identités exponentielles. On donne alors des méthodes générales pour l'obtention de telles formules (dont la formule de Baker-Campbell-Hausdorff), en utilisant des propriétés de la base de Lyndon.

Ces identités explicites et leur méthode de calcul permettent alors d'obtenir des intégrateurs symplectiques. On retrouve alors les intégrateurs connus et on en donne une liste exhaustive pour les petits ordres, montrant que les différentes méthodes de recherches sont complètement équivalentes.

On décrit l'implantation de ces algorithmes dans le système de Calcul Formel AXIOM et des exemples de calcul sont donnés en appendice.

Mots-clefs : méthodes de Lie, méthodes de perturbations, algèbres de Lie libres, transformations canoniques, calcul formel, intégrateurs symplectiques.

Abstract: Our first goal in this report is the comparison of the two Lie perturbation methods used in hamiltonian mechanics. The Deprit as well as the Dragt-Finn method are based on the use of canonical formal transformations. We particularly study the relations between these two transformations and conclude that they produce the same objects, for instance the normal form and formal first integrals. Therefore, we give methods to built such invariants and show that the Dragt-Finn method is less expensive, with regard to the computational cost.

The knowledge of the first terms of these series may provide results for the stability of hamiltonian systems near an elliptic equilibrium point. We give some estimates for the Dragt-Finn method, like it has been given for the Deprit transform.

In order to study the relations between several canonical transform, we work in a free Lie algebra and study exponential identities. We give also general methods to compute these relations (including the Baker-Campbell-Hausdorff formula) using the Lyndon basis.

These explicit identities and their methods of computation allow us to obtain symplectic integrators. We list all the integrators of low order and prove that the several methods of investigation are totally equivalent.

The algorithms have been implemented using AXIOM and some examples are given at the appendix.

Key words: Lie methods, perturbation methods, free Lie algebras, canonical transformations, computer algebra, symplectic integrators.

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
TECHNICAL INFORMATION DEPARTMENT
BERKELEY, CALIFORNIA 94720