4

Concatenation of Lie Algebraic Maps

Liam M. Healy and Alex J. Dragt

ABSTRACT Time evolution in a Hamiltonian system may be represented by a transfer map, which in turn may be represented as a product of Lie transformations factored by order. Two such products in succession may be concatenated into a single product. It is possible to do this even when the Lie transformations include inhomogeneous terms. Rules are given for combining and ordering Lie transformations in the general case through sixth order. These rules are presented in an algorithmic fashion suitable for manipulation by computer.

Such techniques have applications to many Hamiltonian systems, including accelerator beam dynamics and optics. The concatenation could represent, for instance, the combined effects of two successive beamline elements in a particle accelerator. In this case, inhomogeneous terms can arise when there are placement, alignment, or powering errors.

4.1 Introduction

In a Hamiltonian system such as a particle accelerator, it is often important to know where a particle will be at a certain time given its position in phase space at some previous time. The relation that gives this information is called the transfer map. This transfer map can be represented by a sequence of Lie transformations factored by order.

It is often desirable to know to a given order the Lie product corresponding to two successive mappings; mathematically we wish to *concatenate* the two Lie products into one: this is the same as composition of the transfer maps. Treatment of this problem is the primary goal of this paper. One may visualize this, in a particle accelerator, as a means of finding the particular Lie product for the combined effect of two successive beamline elements knowing each separately. This is useful in a variety of calculations. For example, in a single pass system, one can find the combined aberrations of a complete system in terms of the aberrations of its components. Also, when tracking a particle through many turns in a circular accelerator, a great saving in computation time will be realized if many magnetic beamline elements are lumped together as one. Finally, there is an extensive array of analytical tools [8] for analyzing the full one-turn map.

We shall not concern ourselves in this paper with the generation of the original Lie product from the Hamiltonian; that is treated in another paper [12].

4.1.1 Definitions and terminology

It is shown in another paper [12] that the transfer map describing the time evolution of a Hamiltonian system may be represented as a factored product of Lie transformations. Let the transfer map $\mathcal{M}$ send any point in phase space z to another $\bar{z}$ via the relation

$$\bar{z} = \mathcal{M}\ \zeta|_{\zeta=z}, \tag{1.1}$$

where $\zeta = (q_1, p_1, \ldots, q_l, p_l)$ are the $2l$ phase space variables, and z is a $2l$-tuple of values in phase space. Suppose we have a known solution to the time evolution for the orbit through a particular point in phase space. This solution will be called the central orbit. Then, with a suitable choice of coordinates, $\mathcal{M}$ may be represented in the neighborhood of the central orbit in the form

$$\mathcal{M} = e^{:f_1:} e^{:f_2:} e^{:f_3:} \cdots . \tag{1.2}$$

Here, f_n is a polynomial homogeneous of order n in the phase space variables. The *Lie operator* $:f:$ associated with f is defined by its action on another function g on phase space,

$$:f:g \stackrel{\text{def}}{=} [f, g]. \tag{1.3}$$

Here $[,]$ denotes the Poisson bracket. The exponential of a Lie operator is defined by the familiar Taylor series for the exponential,

$$e^{f} = \mathcal{I} + :f: + \frac{:f:^2}{2!} + \frac{:f:^3}{3!} + \cdots . \tag{1.4}$$

The exponential of a Lie operator is called a *Lie transformation*. A Lie transformation $e^{:f_n:}$ corresponding to a homogeneous n^{th}-order polynomial is called an n^{th}-order transformation, and the representation of $\mathcal{M}$ as (1.2) is called the *standard factorization*. As shown in [12], the Lie product (1.2) acting on phase space is equivalent to a Taylor series. It should be pointed out that not all maps can be put in the form (1.2); some will require a second second-order transformation, see [12] or [3]. In general we use (1.2) metaphorically.

A Lie algebra is a vector space with a multiplication operation that satisfies linearity,

$$[af + bg, h] = a[f, h] + b[g, h] \quad \text{and} \quad [f, ag + bh] = a[f, g] + b[f, h], \tag{1.5}$$

antisymmetry,

$$[f, g] = -[g, f], \tag{1.6}$$

and the Jacobi identity,

$$[f,[g,h]]+[g,[h,f]]+[h,[f,g]]=0. \tag{1.7}$$

There are two primary Lie algebras we use. One is the set of functions on phase space S, of which ζ may be thought a member, with the multiplication operation being a Poisson bracket. The other is the set of Lie operators S^* with the multiplication operation being the commutator. There is a *homomorphism*, or multiplication-preserving map, between these two spaces, called 'Ad'. Because of this homomorphism, these spaces may be thought of interchangeably, and the distinction will not be made henceforth.

The product (1.2) maps S into itself in the following fashion. First, note that the order of the Poisson bracket of two homogenous polynomials is the sum of their respective orders, minus two. That is, the Poisson bracket $[f_n, g_m]$ is homogenous of order $n+m-2$. From this we may anticipate the resultant order of applying a Lie transformation of a certain order. The i^{th} term in the series (1.4) is $:f:/i$ applied to the previous term. This means a second-order Lie transformation $e^{:f_2:}$ is a linear transformation because each application of $:f_2:$ leaves the order of its argument unchanged. By contrast, a third-order transformation $e^{:f_3:}$ is non-linear, each application of $:f_3:$ increasing the order of its argument by one; a fourth-order tranformation $e^{:f_4:}$ is also nonlinear, increasing the order of its argument by two at each application of the operator, and so on. A first-order transformation $e^{:f_1:}$, is a special case. Since $:f_1:$ decreases by one the order of its argument, the transformation $e^{:f_1:}$ applied to a polynomial results, in general, in terms of all orders down through a constant. In particular, if it acts on the phase space functions ζ, it will introduce constant terms. These first-order transformations are the result of inhomogeneities in the Hamiltonian, and have the effect of mapping the central orbit onto some other orbit.

4.1.2 The Task of Concatenation

Suppose now that we have two maps $\mathcal{M}$ and $\mathcal{N}$ defined by

$$\begin{aligned}\mathcal{M} &= e^{:f_1:}e^{:f_2:}e^{:f_3:}\dots\\ \mathcal{N} &= e^{:g_1:}e^{:g_2:}e^{:g_3:}\dots,\end{aligned} \tag{1.8}$$

and we wish to join these into a single map $\mathcal{P}$,

$$\mathcal{P}=\mathcal{M}\mathcal{N}. \tag{1.9}$$

We would like to know how to find the factored form of $\mathcal{P}$, *i.e.* the h such that

$$\mathcal{P}=e^{:h_1:}e^{:h_2:}e^{:h_3:}\dots. \tag{1.10}$$

In computing this, the number of terms is important. From a practical standpoint, we must truncate each of the products $\mathcal{M}$ and $\mathcal{N}$ at a certain

order N. It is reasonable to truncate the products $\mathcal{P}$ at that same value of N. The concatenation problem (1.8–1.10) can then be stated compactly: for a given N, and polynomials $f_1, \ldots, f_N$ and $g_1, \ldots, g_N$, we wish to find the polynomials $h_1, \ldots, h_N$ such that

$$:h_1:\,:h_2:\,:h_3:\,\ldots\,:h_N: \cong :f_1:\,:f_2:\,:f_3:\,\ldots\,:f_N:\,:g_1:\,:g_2:\,:g_3:\,\ldots\,:g_N:. \tag{1.11}$$

The symbol '$\cong$' means equivalent through a certain order (in this case N); it will be explained rigorously later. In combining the polynomials, however, higher-order terms will be produced, as we shall see. Furthermore, the presence of first-order terms in the product for one map will combine with any given term of the other product to produce an effect of lower order in the resultant concatenation. These considerations are addressed in the next Section.

4.2 Ideal structure of the Lie algebra

For most realistic maps, the Lie product (1.2) will be infinite. That is, there is no n for which $f_i = 0$ for $i \geq n$. It is of course not practical to deal with an infinite number of terms, so we choose an order at which to truncate the product. This entails making an approximation. This approach is valid in many real physical situations. The Hamiltonian is analytic, so all solutions are analytic in the inital conditions. Therefore, any transfer map is analytic (see [14] chapter 10). We must find a standard by which we can replace a given Poisson bracket by an arbitrary quantity (perhaps zero) if we feel it is "too small," and we must be able to do so consistently. It is crucial not only for derivation of the concatenation formula, but also for calculation of the factored Lie transformation product (1.2) from the Hamiltonian (see [12]). The procedure is divided into two parts: the homogeneous case, where $n \geq 2$ and $m \geq 2$, and the inhomogeneous, where $n = 1$ or $m = 1$. Those not interested in the mathematical underpinnings may without loss of continuity skip to the last paragraph of this Section.

4.2.1 Homogeneous products

The presumption of the Lie transformation product (1.2) is that the corresponding Taylor series is convergent (see [12]). Since we are truncating this product, we want the remainder term that is omitted to be so small that it can be safely dropped. In order to do this, we take the values of each of the phase space variables to be small so that sufficiently high orders may be ignored. Specifically, let each of the phase space variables carry the small factor δ, so that powers of δ count the order of these variables. Polynomials homogeneous of order n have a factor δ^n, and will be said to have *δ-rank n*.

Since δ is small we may, when taking Poisson brackets, neglect terms with δ-rank equal to or greater than some specified value N. For example, if we choose $N = 6$, then the Poisson bracket $[f_4, g_5]$ may be ignored since it is of order 7. We presently shall give this process some rigor; before doing so however, it is necessary to introduce some definitions. In the mathematical definitions that follow, S stands for an arbitrary Lie algebra and not necessarily the Poisson bracket Lie algebra of phase space functions.

We wish to divide up S by order, consistent with the Lie algebra. To do this, we need the concepts of subalgebra and ideal. A subset $S' \subseteq S$ is a *subalgebra* if[1] $[S', S'] \subseteq S'$, where $[S', S']$ is the image of the Poisson bracket restricted to S'. A subset $S' \subseteq S$ is an *ideal* if $[S', S] \subseteq S'$. That is, for $s' \in S'$, $s \in S$, then $[s', s] \in S'$. Clearly, an ideal is also a subalgebra. Ideals will play a critical role in the approximation process by way of quotient algebras.

Approximation is made by doing computations in terms of a *quotient algebra*, which can be defined for any ideal. The ideal plays the role of the remainder, the part that is ignored, and the quotient algebra is the original algebra with "elements approximately the same" identified. Specifically, let I be an ideal in the Lie algebra S. Now define the quotient algebra S/I to be the set of equivalence classes given by the equivalence relation

$$s_1 \cong s_2 \qquad \text{if} \quad s_1 = s_2 + i \quad \text{for some} \quad i \in I. \tag{2.1}$$

We denote these classes by $s+I$, where $s \in S$. Since an ideal is a subalgebra, S/I is a Lie algebra with the rules

$$(s_1 + I) + (s_2 + I) = (s_1 + s_2) + I \tag{2.2}$$

$$c(s_1 + I) = cs_1 + I \quad \text{for} \quad c \in \mathbb{R} \tag{2.3}$$

$$[s_1 + I, s_2 + I] = [s_1, s_2] + I. \tag{2.4}$$

These rules are easily seen to be consistent. If i_1, $i_2 \in I$ are arbitrary, the left side of the first rule (2.2) is

$$(s_1 + i_1) + (s_2 + i_2) = (s_1 + s_2) + (i_1 + i_2) \in (s_1 + s_2) + I, \tag{2.5}$$

since I is a vector subspace of S . A similar argument holds for the second rule. The left side of the third rule (2.4) is

$$[s_1 + i_1, s_2 + i_2] = [s_1, s_2] + [s_1, i_2] + [i_1, s_2] + [i_1, i_2]. \tag{2.6}$$

---

[1]The notation $[X, Y]$ where X and Y are sets means the set

$$\{[x, y] \mid x \in X, y \in Y\}.$$

From the definition of an ideal, we see that the consistency of the third rule is upheld.

Now we address ourselves to the question of the origin of the ideals for the approximation. They come from a *percolation*, a telescoped set of subspaces whose union fill the whole space and whose Lie bracket fulfills a certain property: for each non-negative integer i, there is a subspace $S^{(i)}$ such that

$$S^{(i)} \supseteq S^{(j)} \quad \text{for} \quad i \leq j \tag{2.7}$$

$$S^{(0)} = S \tag{2.8}$$

$$[S^{(i)}, S^{(j)}] \subseteq S^{(i+j)}. \tag{2.9}$$

If S is *graded*, that is, it is the direct sum of subspaces S_i $(i = 0, 1, \ldots, \infty)$ and $[S_i, S_j] \subseteq S_{i+j}$, then it is percolated by the rule

$$S^{(i)} \stackrel{\text{def}}{=} \bigoplus_{j \geq i} S_j. \tag{2.10}$$

In our application, these subspaces will be polynomials of a certain δ-rank or higher.

It is clear from the above definition that each of the members $S^{(i)}$ of a percolation is an ideal. Let $s^{(i)} \in S^{(i)}$, $s \in S = S^{(0)}$. Then $[s, s^{(i)}] \in S^{(i)}$, and also with the arguments in the reverse order, $[s^{(i)}, s] \in S^{(i)}$. Since s and $s^{(i)}$ were arbitrary within their respective sets, $S^{(i)}$ is an ideal in S .

Let us now apply this to our particular problem. Up to now in this Section, we have let S stand for an arbitrary Lie algebra. We now restrict the definition of S so that it is the space of all functions on phase space that have power series expansions and whose power-series expansion has no first-order or constant term. Grade it with subspaces of polynomials homogeneous in a particular order of the phase space variables: let S_i be the set of all homogeneous polynomials of order $i + 2$, for $i > 0$. One may easily verify that this is a grading on S under the Poisson bracket. A particular polynomial that belongs to the subspace S_i has δ-rank $i + 2$.

Given this grading $\{S_i\}$ by polynomial order, we have the corresponding percolation given by (2.10). This gives us a sequencs of ideals $S^{(i)}$, and a sequence of quotient algebras $S/S^{(i)}$. The ideal $S^{(i)}$ consists of all power series with coefficients zero for all terms of order less than $i+2$. The quotient algebra $Q^{(i)} \stackrel{\text{def}}{=} S/S^{(i+1)}$ is a rigorous way of describing the algebra S with the approximation of "neglecting terms of order $i + 3$ and greater."

The computer code MARYLIE 3.0 [2] [6] [7], designed to perform beam dynamics calculations for accelerator physics, has no first-order polynomials and performs computations through fourth order. Speaking in terms of the formal algebra introduced above, it is computing in the Lie algebra $Q^{(2)}$. In this algebra, one must say, for example, $[f_3, g_3] = h_4$ for some specific h_4, but it makes the approximation $[f_3, g_4] \cong 0$. In the algebra $Q^{(2)}$ the values of some Poisson brackets will be in S^3. Since we are working in

the quotient algebra, we may choose any member of S^3. When performing the computation by hand, we will usually choose zero. When working numerically in a computer code (the tracking part of MARYLIE), the choice of some other members may be computationally advantageous.

The homomorphism Ad carries all these definitions to the adjoint algebra S^* . The subalgebras $S^{*(i)}$ are the spaces of all Lie operators that are of order $i+2$, *i.e.*, are images of S_i under Ad, and $S^{*(i)}$ are the direct sum $\oplus_{j\geq i}S_j^*$ or the image of $S^{(i)}$ under Ad. The $S^{*(i)}$ are ideals, so the $Q^{*(i)} = S^*/S^{*(i+1)}$ are quotient algebras. Thus the adjoint algebra has the same ideal and quotient structure as the underlying algebra, as we expect, and commutators of Lie operators are set to zero (or to an arbitrary value) when the Poisson bracket of the corresponding polynomials would be of too high an order.

These quotient Lie algebras give rise to quotient groups in the group of all symplectic maps on phase space. While possibly containing terms of all orders, these maps are accurate only through order $i+1$ for $Q^{(i)}$. Within the group, it will be possible to either truncate and then multiply, or multiply and then truncate, with equal validity. Specifically, let G be the group of symplectic maps on phase space and $G^{(i)}$ $(i = 0, 1, \ldots)$ be the subgroup of these maps that has a power series expansion consisting of terms only order $i+1$ and higher, plus the identity. Then $G^{(i)}$ is a normal or invariant subgroup of G, that is, $ghg^{-1} \in G^{(i)}$, $\forall g \in G, h \in G^{(i)}$. The quotient group $H^{(i)} = G/G^{(i+1)}$ is defined as the equivalence classes given by $g_1 \cong g_2$ if $g_1 g_2^{-1} = h$ where $h \in G^{(i)}$. That is, two elements are equivalent if the power series expansions of their associated symplectic maps differ only by terms of order $i+1$ and higher. That this is in fact a group may be easily verified. These groups $H^{(i)}$ are associated with the algebras $Q^{*(i)}$.

4.2.2 INHOMOGENEOUS PRODUCTS

If a first-order transformation is present in the product (1.2), the truncation by δ-rank given above will not be correct because the δ-rank will be lowered by the first-order term. Suppose we keep terms only through δ-rank 4, and discard anything higher. Then forming, for instance, $[f_1, [g_3, h_4]]$ would yield the wrong answer: we may take $[g_3, h_4]$ as zero because the δ-rank is 5, and so our overall answer would be zero. But this is not correct; even though the inner Poisson bracket is δ-rank 5, the Poisson bracket with f_1 subsequently lowers the δ-rank to an acceptable 4. Clearly, the subspaces described in the last Section are no longer subalgebras when a first-order term is present and we must reformulate their description for correct computations in this case.

The correct treatment of first-order terms becomes evident when we consider their physical origin. As is shown in [12], the first-order transformation is proportional to the first-order term in the Hamiltonian, which in turn is proportional to an error, such as an alignment or powering error in an ac-

celerator beamline element. It is hoped that these are small; consequently, the first-order transformation will also be small. To be specific, say that a factor of the small parameter ϵ multiplies each first-order polynomial. We may now consider how this changes the analysis of the algebra and the corresponding group given in the previous Section.

The space S must be expanded, and the set $\{S_i\}$ given a new member. Let S now stand for the set of all functions on phase space that have power-series expansions; we no longer require that the first-order term be zero. We shall still ignore constant terms since they play no role in the Lie algebra. Let S_{-1} be the space of first-order polynomials. The spaces $\{S_i\}_{i=-1,0,1,.}$ are still a grading. However, we cannot construct a corresponding percolation according to (2.10) because we now have a negative i. Thus the $S^{(i)}$ are no longer ideals and we cannot form the quotient algebra. There is a corresponding destruction of the normal subgroups and quotient groups of symplectic maps.

Instead of using this grading, let us search for another way to create a percolation, and thus obtain a sequence of ideals and a sequence of quotient algebras that will correctly reflect the physics of the perturbation calculation. First, let the *ϵ-rank* of a polynomial mean the lowest order in ϵ for that polynomial. Define a second index j, $j = 0, 1, \ldots$, on the S_i that is equal to the ϵ-rank. Thus S_{ij} is a subset of S that is homogeneous of order $i+2$ in the phase space coordinates, and homogeneous of order j in ϵ. A polynomial has δ-rank i and ϵ-rank j if it belongs to S_{ij}. The only combination of i and j within these ranges that is prohibited is $i = -1$, $j = 0$, the smallness requirement on first-order terms discussed above. The spaces S_{ij} are a finer grading of the spaces S_i graded by δ-rank; it is easy to see that

$$[S_{ij}, S_{kl}] \subseteq S_{i+k,j+l}. \tag{2.11}$$

We now seek a percolation constructed from the S_{ij}.

A percolation that satisfies the requirements may be formed by creating a special function of the two indices i and j. Consider the set of allowed index pairs

$$\mathbb{Z}^{2*} \stackrel{\text{def}}{=} \{-1, 0, 1, \ldots\} \times \{0, 1, \ldots\} - \{(-1, 0)\}. \tag{2.12}$$

Let a *truncation criterion* ν be any function into the non-negative integers $\nu : \mathbb{Z}^{2*} \to \mathbb{Z}^{+}$ with the property

$$\nu(i, j) + \nu(k, l) \leq \nu(i + k, j + l). \tag{2.13}$$

Now define the sequence of subspaces $S^{(i)}$, $i \in \mathbb{Z}^{+}$ as the direct sum of all S_{jk} whose value of ν is at least i:

$$S^{(i)} \stackrel{\text{def}}{=} \bigoplus_{\nu(j,k) \geq i} S_{jk}. \tag{2.14}$$

This sequence is a percolation, which may be shown in the following manner. Clearly, the first and second properties of a percolation are satisfied. To show the third (2.9), let $S_{i'j'}$ be an arbitrary element of $S^{(\nu(i,j))}$ and $S_{k'l'}$ be an arbitrary element of $S^{(\nu(k,l))}$. Then

$$S_{i'j'} \in S^{(\nu(i,j))} \quad \Rightarrow \quad \nu(i',j') \geq \nu(i,j) \tag{2.15}$$

$$S_{k'l'} \in S^{(\nu(k,l))} \quad \Rightarrow \quad \nu(k',l') \geq \nu(k,l). \tag{2.16}$$

The Poisson bracket of these two polynomial spaces is

$$\begin{aligned} [S_{i'j'}, S_{k'l'}] &\subseteq S_{i'+k',j'+l'} \subseteq S^{(\nu(i'+k',j'+l'))} \\ &\subseteq S^{(\nu(i',j')+\nu(k',l'))} \subseteq S^{(\nu(i,j)+\nu(k,l))}, \end{aligned} \tag{2.17}$$

by the truncation criterion (2.13) and the first property of a percolation (2.7). Since $S_{i'j'}$ and $S_{k'l'}$ were arbitrary in their respective subspaces, we may conclude the third property of a percolation holds,

$$[S^{(n)}, S^{(m)}] \subseteq S^{(n+m)}, \tag{2.18}$$

for values n, m in the image of ν. By implication, therefore, the $S^{(i)}$ as defined in (2.14) are ideals. This sequence of ideals may be used to define a sequence of quotient algebras.

Note that we have determined a satisfactory set of ideals with ν undetermined except for the condition (2.13). We now must consider specific truncation criteria. Two possibilities are $\nu(i,j) = \min(i,j)$ and $\nu = \alpha i + \beta j$ where $\alpha, \beta \in \mathbb{Z}^+$. The reader may verify that these two indeed satisfy (2.13). The former case corresponds to keeping all terms except those whose δ-rank and whose ϵ-rank each exceed a certain value. The latter excludes those whose weighted sum exceeds a certain value. This form of ν satisfies a stricter condition than (2.13), in fact, $\nu(i,j) + \nu(k,l) = \nu(i+k, j+l)$, and so we have a grading $S_{\nu(i,j)}$, which may form a percolation by (2.10). This percolation is the same as the one defined by (2.14).

Normal subgroups $G^{(\nu(i,j))}$ of the group of inhomogeneous symplectic maps G may be defined by analogy to that of the homogeneous group (see Section 4.2.1). The quotient group $H^{(\nu(i,j))} \stackrel{\text{def}}{=} G/G^{(\nu(i,j))}$ is the group of transformations that will actually be used in computations.

With all the formalism aside, we must choose a particular truncation criterion with which to proceed with the actual computations. The one that makes the most intuitive sense is $\nu(i,j) = i+j$. This will be called the *total rank*. In terms of δ-rank and ϵ-rank, this criterion says that we restrict terms to $O(\delta)+O(\epsilon) \leq N$ for some value of N. Physically, this is a realistic criterion, because it means that the deviation in central orbit caused by the error is of the same order as the perturbation around the central orbit. Thus, we can expect the same accuracy in the result. For example, the misalignment of a magnetic element in an accelerator should not typically

be of greater magnitude than the spread in position of particles within the beam. It is also possible to imagine the case where a different truncation criterion needs to be used; for instance, a weighted sum of the exponents as mentioned above may be appropriate. The calculations in the remainder of this paper, however, are done with the total rank criterion.

4.3 Lie algebraic tools

With the algebraic formalism established, we now turn to the problem of solving the concatentation problem (1.11). There are two important tools used in this calculation that are presented in this Section. At the moment, it is useful to introduce some notation. It will become necessary to label a polynomial by its total rank as well as its δ-rank. In this case, a superscript with the total rank in parenthesis will be placed on the polynomial, *e.g.*, $k_3^{(4)}$. Furthermore, let the function τ give the total rank of a polynomial or operator, so that, *e.g.*, $\tau(k_3^{(4)}) = 4$. In Section 4.4.2 this notation will be generalized slightly; the superscript will identify a family of terms of which at least one is of that total rank, none are lower, and some may be higher.

4.3.1 The exchange rule

The first tool used in the computation is the *exchange rule*. This allows us to rewrite two Lie transformations in succession such that one transformation occurs in the opposite position,

$$e^{:f:}e^{:g:} = e^{:j:}e^{:f:} \tag{3.1}$$

or

$$e^{:f:}e^{:g:} = e^{:g:}e^{:k:}, \tag{3.2}$$

where the polynomials are not necessarily homogeneous. The function j in (3.1) is given by

$$j = e^{:f:}g \tag{3.3}$$

and the function k in (3.2) is given by

$$k = e^{:-g:}f. \tag{3.4}$$

For proofs, see [4] (Theorem 3), [2], or [3] (equation 5.52).

It is also possible to bring a Lie transformation inside a function; that is,

$$e^{:f:}g(\zeta) = g(e^{:f:}\zeta), \tag{3.5}$$

for arbitrary functions f, g. This is proved in [3] and in [2]. In combination with the exchange rule, we have a powerful result because it means that

an operator can be moved to the left or right of another operator by transforming that operator's argument. This is especially useful where the first operator is of δ-rank 2, in which case it produces a linear transformation of the other operator's argument.

4.3.2 Combining transformations and factoring a single transformation

The other useful tool has two parts. First, we shall need to combine two (or more) transformations into a single exponent. This is accomplished using the Baker-Campbell-Hausdorff (BCH) formula. The BCH theorem says that the logarithm of the product of two exponentials is in the Lie algebra; the formula gives the explicit form. That is, if

$$e^{:f:}e^{:g:} = e^{:h:}, \tag{3.6}$$

then h is in the Lie algebra spanned by f and g. The specific formula through three brackets (see [15] or [9]) is

$$h = f + g + \tfrac{1}{2}[f,g] + \tfrac{1}{12}[f,[f,g]] + \tfrac{1}{12}[g,[g,f]] - \tfrac{1}{24}[f,[g,[f,g]]] + \cdots. \tag{3.7}$$

The second part is to rewrite a single transformation as a product of transformations factored in the proper order,

$$e^{:j_1 + j_2 + \cdots + j_n:} = e^{:k_1:}e^{:k_2:}\cdots e^{:k_m:}. \tag{3.8}$$

We do not know yet what m is, except that it will be at least n, and of course it will be no greater than N. This shall be called the *separation procedure*. It consists of two steps; we first repeatedly use the BCH formula to combine the k, represented only symbolically at this point, into a single exponent, then use a general method to find the k in terms of the j.

Consider now the general problem of *Lie function inversion*; we have a set of functions F_i of unknown polynomials k_i; we wish to solve the set k_i in terms of the known values j_i of F_i. That is,

$$j_i = F_i(k_1, \ldots, k_m) \quad \text{for} \quad i = 1, \ldots, N. \tag{3.9}$$

In our particular application, the functions F_i come from multiple application of the BCH formula; we use it to express the j_i in terms of the k_i and their Poisson brackets.

The method of inversion presented here requires only that the functions F_i satisfy two properties. The first is that each function F_i has the term k_i alone, not in a Poisson bracket; for instance $F_3(k_1, \ldots, k_m) = k_3 + [k_1, k_4] + \cdots$. The second is that the ϵ-rank of both j_1 and j_2 be at least one, *i.e.*, $\tau(j_1) \geq 2$ and $\tau(j_2) \geq 3$. The requirement on j_1 is simply the ϵ-rank requirement embodied in (2.12) and insures that the total rank is never

lowered. The requirement on j_2 insures that no equation will be implicit in any of the remainder terms to be introduced, because the only possibility for an operator $:j_i:$ to leave the total rank of its argument unchanged is if the δ-rank is altered, *i.e.*, when $i = 1$. A consequence of these assumptions is that $\tau(k_i) = \tau(j_i)$.

We need keep only terms with a specific number of brackets in the expressions for F_i. If the smallest total rank of the j, $t \stackrel{\text{def}}{=} \min_i\{\tau(j_i)\}$, is at least 3, then the necessity of retaining each term is governed by the total rank criterion. One then keeps only terms of l brackets where

$$l(t-2)+t \leq N. \tag{3.10}$$

On the other hand, if the smallest total rank is 2, the necessity is governed separately by the total rank criterion ($N-3$ brackets) and the δ-rank from applying the $:j_1:$ ($N-1$ brackets), so the condition is

$$l \leq 2N-4. \tag{3.11}$$

The inversion is accomplished iteratively on the total rank. The first step is to assume that k_i is j_i plus a remainder term of the same total rank,

$$k_i \rightarrow j_i + r_i^{(\tau(j_i))} \tag{3.12}$$

where r is a currently-unknown remainder term indexed by the δ-rank and the total rank. Formally, we make this substitution in the expressions (3.9); by the first assumption above about the F_i, both sides of each equation will have a term j_i, which may be eliminated. In the subsequent steps of the iteration, we may solve for the remainder term in the following way. Make a substitution for $r_i^{(t)}$ such that all terms of total rank t are cancelled off by setting it equal to the negative of all the other terms of that total rank in the expression, plus a remainder term of the next higher total rank;

$$r_i^{(t)} \rightarrow -W_i^{(t)} + r_i^{(t+1)}, \tag{3.13}$$

where $W_i^{(t)}$ consists of all terms of total rank t except j_i and the remainder term. Thus all terms of total rank t have been eliminated, and there are only terms of total rank $t+1$ and higher. We continue this iteration until the only terms that remain exceed the total rank cut-off N.

We now may find the answer by going back to the original substitution (3.12) and making the successive replacements (3.13); this gives the answer

$$k_i = j_i + \sum_{t=\tau(j_i)}^{N} -W_i^{(t)}. \tag{3.14}$$

It is tempting say that this can be obtained immediately without iteration, but keep in mind that W is not known at the current step until the last

step has been solved; in general, that last step will have introduced new terms of the current total rank.

As it happens, the first remainder solution $r_i^{(i)}$ will yield the trivial answer $W_i^{(i)} = 0$ unless $\epsilon = 0$ for at least two non-zero terms j_m, j_n, with $m, n \geq 3$. This result is obtained in the exchange-and-seperate method for moving the first-order term (Section 4.4.2), where all j at each step have the same total rank. On the other hand, the solution of the subsequent remainders $r_i^{(t)}, t > i$, will yield the trivial answer $W_i^{(t)} = 0$ unless $\epsilon \geq 1$ for some j. This is obtained in moving higher-order terms (Section 4.4.4).

The solution to (3.8) may be obtained once and for all for a given N using this method by assuming the lowest possible total ranks on each of the j. Then at any step, one may use this result, discarding the terms whose total rank exceeds N or which involve a j that is zero. This has been done for $\tau(j) \geq 3$ and $N = 6$; the result is given in Appendix 4.C. Examples of the computation itself for less general cases are given in Sections 4.4.2 and 4.4.4.

4.4 Computation method for the concatenation formula

Selective use of the tools given will now allow us to acheive our goal of solving (1.11). The general idea is to move to the left all terms of lower δ-rank than their immediate left-hand neighbor. By "move to the left," we really mean to rewrite a pair of transformations in the standard factorization:

$$e^{:a_n:} e^{:b_m:} = e^{:k_1:} e^{:k_2:} e^{:k_3:} \cdots e^{:k_N:}, \tag{4.1}$$

where $n \geq m$. Initially on the right side of (1.11) there is only one pair of transformations that satisfies this condition, $e^{:f_N:} e^{:g_1:}$. Once this is put into the standard factorization, other wrongly-ordered transformation pairs will occur, and these must be treated in a similar fashion, which will in turn give rise to more such pairs and so on.

The overall plan of attack is organized as follows. Initially, the first-order transformation will be moved to the left. This will leave behind many second and higher-order transformations, not at all in a suitable order. However, all the first-order transformations will be at the extreme left except for the intervening $e^{:f_2:}$, and they may be combined in a simple fashion. Now, the second-order transformations may moved to the left and $e^{:f_2:}$ to the right, leaving third and higher order transformations behind. Then the third-order ones may be moved; but, when they get to the left they must be combined into a single exponent and then this exponent split up by order. Fourth and higher orders are treated in a similar way. When this has been done through order $N - 1$, the Lie transformations will be in the proper order.

The basic methods for treating pairs of transformations (4.1) are presented in the next Section, and the actual procedure for each order m in the succeeding Sections. As an alternative to the pairwise treatment of terms described above and elaborated below, one may try to combine all the terms of the concatenation problem (1.11) except $e^{:f_2:}$ and $e^{:g_2:}$ at once using an expanded version of the combine-and-separate method described in the next Section. This has a certain conceptual simplicity, but it is not practical as it would generate an inordinate number of terms for even small N.

4.4.1 Putting a pair of transformations into standard factorization

For the task of putting two successive transformations into standard factorization (4.1), there are three possibilities. If $m = 2$ and $\tau(b_2) = 2$, there is no choice but to use the exchange method. If this is not the case, there are two choices.

The *exchange-and-separate* method uses the exchange rule to write

$$e^{:a_n:}e^{:b_m:} = e^{:b_m:}e^{:e^{:-b_m:}a_n:} = e^{:b_m:}e^{:a_n+:-b_m:a_n+\cdots:}, \tag{4.2}$$

and then application of the separation procedure on the second transformation. Because the transformation $e^{:b_m:}$ remains separate on the left, this method is not useful for the case $m = n$. It is advantageous for $m = 1$ because all terms in the second transformation have total rank at least 3, reducing the number of Poisson brackets required.

The *combine-and-separate* method uses the BCH formula to combine the terms into a single exponent,

$$e^{:a_n:}e^{:b_m:} = e^{:a_n + b_m + \frac{1}{2}[a_n, b_m] + \cdots:}, \tag{4.3}$$

and then application of the separation procedure on this transformation. This method is preferable when $m > 1$ and necessary when $m = n$. Note that this incarnation of the BCH formula is distinct from that of the separation procedure that produces the functions F_i in (3.9); this is the combination of only two terms and changes at each step.

4.4.2 Moving the first-order term

We concentrate initially on the terms f_3 through g_1 of (1.11) and temporarily ignore the other Lie transformations. That is, we wish to put

$$e^{:f_1:}e^{:f_2:}e^{:f_3:}\ldots e^{:f_N:}e^{:g_1:} \tag{4.4}$$

into standard factorization. This is done successively on each transformation; we start by putting the right-hand pair with f_N and g_1 into standard

factorization. Then, with a first-order term on the left, we put the next pair with f_{N-1} into standard factorization, and so on until all the first-order terms are to the left of all terms of third order and higher. At each step, we have a pair of terms involving an f_n and some first-order term that will be rewritten in the standard factorization,

$$e^{:f_n:}e^{:b_1:} = e^{:k_1:}e^{:k_2:}e^{:k_3:}\cdots. \tag{4.5}$$

Although it is possible to use the combine-and-separate method instead of the exchange-and-separate, it is not practical. With $\tau(b_1) = 2$, there are many more terms in the BCH series that need to be retained, because no total rank 2 term occurs in the latter method; it is exactly the condition (3.11) versus (3.10). For large N, there are an enormous number of terms, so it is more practical to use the exchange-and-separate method for the first-order transformations.

In order to solve (4.5) for $n \geq 3$, we first make the first-order term specific and use the exchange rule to write

$$\begin{aligned} e^{:f_n:}e^{:b_1:} &= e^{:b_1:}e^{:e^{:-b_1:}f_n:} \\ &= e^{:b_1:}e^{:f_n + [-b_1, f_n] + \cdots + :-b_1:^{n-1} f_n/(n-1)!:}. \end{aligned} \tag{4.6}$$

Note that this second transformation has only a finite number of terms in it, and of each order from n down through 1, because of the order-reducing property of a first-order Lie operator. Rewriting each of these terms as $j_{n-i} \stackrel{\text{def}}{=} \frac{1}{i!}:-b_1:^i f_n$ for $0 \leq i \leq n-1$, we have

$$e^{:f_n:}e^{:b_1:} = e^{:b_1:}e^{:j_1 + j_2 + \cdots + j_n:}. \tag{4.7}$$

We may now apply the separation procedure to put the right-hand side of this equation into the standard factorization (3.8), which we can then substitute into (4.7) to get

$$e^{:f_n:}e^{:b_1:} = e^{:b_1:}e^{:k_1:}e^{:k_2:}\cdots e^{:k_m:}. \tag{4.8}$$

The next step is to combine the two adjacent first-order transformations into a single exponent. This is a trivial application of the BCH formula; because they are first-order terms, their Poisson brackets are constants and may therefore be discarded. Thus we simply put the sum of the two first-order terms into a single exponent:

$$e^{:f_n:}e^{:b_1:} = e^{:b_1 + k_1:}e^{:k_2:}\cdots e^{:k_m:}. \tag{4.9}$$

Each of the Lie transformation pairs that is put into standard factorization according to (4.5) produces a set of polynomials $\{k_i^{(t)}\}$ which we label

with a superscript that superficially indicates the total rank. If $i \leq t$, it is the total rank. If $i > t$, the total rank is at least i, but the term was produced in the same step as the total rank t terms. At the first step involving f_N, all the polynomials produced will be of total rank N, at the next they will be of total rank $N-1$, and at the step i mostly of total rank $N-i+1$ but some may be higher.

The first-order terms are accumulated at each step according to (4.9). On the first step, $b_1 = g_1$, and at each step the particular k_1 is added. Thus the final first-order term is g_1 plus the sum of all the k_1 terms,

$$G_1 \stackrel{\text{def}}{=} g_1 + \sum_t k_1^{(t)}. \tag{4.10}$$

When all the terms of the type (4.5) in (4.4) have been treated in succession we will end up with the result

$$\begin{aligned} e^{:f_1:} e^{:f_2:} e^{:f_3:} \cdots e^{:f_N:} e^{:g_1:} \;=\; & e^{:f_1:} e^{:f_2:} e^{:G_1:} \\ & \times e^{:k_2^{(3)}:} e^{:k_3^{(3)}:} \cdots e^{:k_{N-1}^{(3)}:} \cdots \\ & \times e^{:k_2^{(N-1)}:} \cdots e^{:k_{N-1}^{(N-1)}:} e^{:k_2^{(N)}:} \cdots \\ & \times e^{:k_{N-1}^{(N)}:} e^{:k_N^{(N)}:}, \end{aligned} \tag{4.11}$$

with the first-order terms to the left of all third- and higher-order terms. It should be noted that there will be no $k_N^{(l)}$ term for $l \leq N-1$ because such a term can be formed only by taking nested Poisson brackets solely of $j_l^{(l)}$, up to the maximum number of brackets that the total rank criterion will allow; the Poisson bracket $[j_l^{(l)}, j_l^{(l)}]$ though, is zero.

In order to put the right-hand side of (4.11) into standard factorization, we need to move $e^{:f_2:}$ to the right of $e^{:G_1:}$. For this we just use the exchange rule as described in the next Section, and then the two adjacent first-order transformations may be combined by adding their exponents. The higher-order terms may be put into standard factorization by the procedure described in Section 4.4.4. Because the second-order terms $e^{:k_2^{(t)}:}$ have total rank higher than two, it is possible to move and combine them using the rules for higher-order transformations instead of using the exchange rule.

To see how this works in practice, let us look at a simple example. Suppose we take $N = 4$, and consider the part of the problem where we will move the first-order term to the left of the third-order term. That is, we will write

$$e^{:f_3:} e^{:g_1:} = e^{:k_1:} e^{:k_2:} e^{:k_3:} e^{:k_4:}, \tag{4.12}$$

and determine the polynomials k_n. Following the procedure above, we write

$$e^{:f_3:} e^{:g_1:} = e^{:g_1:} e^{:j_1^{(3)} + j_2^{(3)} + j_3^{(3)}:}, \tag{4.13}$$

with the $j_n^{(3)}$ given by the exchange rule,

$$\begin{aligned} j_1^{(3)} &= \tfrac{1}{2}[g_1,[g_1,f_3]] \\ j_2^{(3)} &= -[g_1,f_3] \\ j_3^{(3)} &= f_3. \end{aligned} \tag{4.14}$$

We now break apart the exponent into separate transformations. The answer may be obtained from Appendix 4.C, or we may perform the separation procedure ourselves. Since each of these terms is total rank 3 and we are keeping through $N = 4$, we will need the BCH formula (3.7) only through the one-Poisson-bracket term, that is, equation (3.9) is explicitly in this case,

$$\begin{aligned} j_1^{(3)} &= F_1(k_1^{(3)},k_2^{(3)},k_3^{(3)},k_4^{(3)}) = k_1^{(3)} + \tfrac{1}{2}[k_1^{(3)},k_2^{(3)}] \\ j_2^{(3)} &= F_2(k_1^{(3)},k_2^{(3)},k_3^{(3)},k_4^{(3)}) = k_2^{(3)} + \tfrac{1}{2}[k_1^{(3)},k_3^{(3)}] \\ j_3^{(3)} &= F_3(k_1^{(3)},k_2^{(3)},k_3^{(3)},k_4^{(3)}) = k_3^{(3)} + \tfrac{1}{2}[k_2^{(3)},k_3^{(3)}] + \tfrac{1}{2}[k_1^{(3)},k_4^{(3)}] \\ 0 &= F_4(k_1^{(3)},k_2^{(3)},k_3^{(3)},k_4^{(3)}) = k_4^{(3)} + \tfrac{1}{2}[k_2^{(3)},k_4^{(3)}]. \end{aligned} \tag{4.15}$$

Making the initial substitution $k_i^{(3)} \to j_i^{(3)} + r_i^{(3)}$ (3.12) and then the next substitution $r_i^{(3)} \to -W_i^{(3)} + r_i^{(4)}$ (3.13) yields the result $W^{(3)} = 0$ (because all Poisson bracket terms are of higher order than 3) and the residual equations

$$\begin{aligned} 0 &= r_1^{(4)} + \tfrac{1}{2}[j_1^{(3)},j_2^{(3)}] + \tfrac{1}{2}[r_1^{(4)},j_2^{(3)}] + \tfrac{1}{2}[j_1^{(3)},r_2^{(4)}] + \tfrac{1}{2}[r_1^{(4)},r_2^{(4)}] \\ 0 &= r_2^{(4)} + \tfrac{1}{2}[j_1^{(3)},j_3^{(3)}] + \tfrac{1}{2}[r_1^{(4)},j_3^{(3)}] + \tfrac{1}{2}[j_1^{(3)},r_3^{(4)}] + \tfrac{1}{2}[r_1^{(4)},r_3^{(4)}] \\ 0 &= r_3^{(4)} + \tfrac{1}{2}[j_2^{(3)},j_3^{(3)}] + \tfrac{1}{2}[r_2^{(4)},j_3^{(3)}] + \tfrac{1}{2}[j_2^{(3)},r_3^{(4)}] + \tfrac{1}{2}[r_2^{(4)},r_3^{(4)}] \\ &\quad + \tfrac{1}{2}[j_1^{(3)},r_4^{(4)}] + \tfrac{1}{2}[r_1^{(4)},r_4^{(4)}] \\ 0 &= r_4^{(4)} + \tfrac{1}{2}[j_2^{(3)},r_4^{(4)}] + \tfrac{1}{2}[r_2^{(4)},r_4^{(4)}], \end{aligned} \tag{4.16}$$

after cancelling off the $j_n^{(3)}$. We have made the substitution $k_4^{(3)} \to r_4^{(4)}$ because there is no $j_4^{(3)}$.

Looking only at terms of total rank 4, and solving for $r_i^{(4)}$ according to (3.13), we find

$$\begin{aligned} r_1^{(4)} &= -\tfrac{1}{2}[j_1^{(3)},j_2^{(3)}] + r_1^{(5)} \\ r_2^{(4)} &= -\tfrac{1}{2}[j_1^{(3)},j_3^{(3)}] + r_2^{(5)} \\ r_3^{(4)} &= -\tfrac{1}{2}[j_2^{(3)},j_3^{(3)}] + r_3^{(5)} \\ r_4^{(4)} &= 0. \end{aligned} \tag{4.17}$$

The new remainder term and the three remaining Poisson brackets that occur in each of the equations (4.16) are of total rank at least 5. Since

our truncation criterion is to drop terms total rank 5 and greater, we may discard them. We are thus at the end of the iteration, and we find the answer by adding up all the remainders according to (3.14). There was only one remainder in this example, so

$$\begin{aligned} k_1^{(3)} &= j_1^{(3)} - \tfrac{1}{2}[j_1^{(3)}, j_2^{(3)}] \\ k_2^{(3)} &= j_2^{(3)} - \tfrac{1}{2}[j_1^{(3)}, j_3^{(3)}] \\ k_3^{(3)} &= j_3^{(3)} - \tfrac{1}{2}[j_2^{(3)}, j_3^{(3)}] \\ k_4^{(3)} &= 0. \end{aligned} \tag{4.18}$$

Substituting back into (4.13), we conclude

$$\begin{aligned} e^{:f_3:}e^{:g_1:} &= e^{:g_1:}e^{:j_1^{(3)} - \frac{1}{2}[j_1^{(3)}, j_2^{(3)}]:} \\ &\quad \times e^{:j_2^{(3)} - \frac{1}{2}[j_1^{(3)}, j_3^{(3)}]:}e^{:j_3^{(3)} - \frac{1}{2}[j_2^{(3)}, j_3^{(3)}]:} \\ &= e^{:g_1:}e^{:\frac{1}{2}[g_1,[g_1,f_3]] - \frac{1}{2}[\frac{1}{2}[g_1,[g_1,f_3]],-[g_1,f_3]]:} \\ &\quad \times e^{:-[g_1,f_3] - \frac{1}{2}[\frac{1}{2}[g_1,[g_1,f_3]],f_3]:}e^{:f_3 - \frac{1}{2}[-[g_1,f_3],f_3]:}. \end{aligned} \tag{4.19}$$

It is then possible to simplify the nested Poisson brackets.

This is a very simple example, and it may seem that it is easily solved intuitively, without the formalism. This is so, but for higher orders the formalism is necessary. Appendix 4.A describes some of the Lie algebraic manipulation that can keep the calculation under control. This example is unfortunately too simple to show that terms $k_i^{(n)}$ with $i > n$ do occur, which only happens for $N \geq 5$.

4.4.3 Moving second-order terms

Moving the second-order terms in (4.11) to the right of the first-order terms or moving them to the left of the higher-order terms is a pure application of the exchange rule. Since a second-order transformation is just a linear transformation on phase space, one can effectively rewrite a pair of Lie transformations involving a second-order transformation as

$$e^{:a(\zeta):}e^{:b_2:} = e^{:b_2:}e^{:e^{:-b_2:}a(\zeta):} = e^{:b_2:}e^{:a(e^{:-b_2:}\zeta):} \tag{4.20}$$

Therefore, one would move the second-order terms by transforming the intervening operators' arguments. For instance, the first three terms on the right-hand side of (4.11) are rewritten

$$e^{:f_1:}e^{:e^{:f_2:}g_1:}e^{:f_2:} = e^{:f_1 + e^{:f_2:}g_1:}e^{:f_2:} = e^{:f_1 + g_1^T:}e^{:f_2:}, \tag{4.21}$$

where here g_1^T is defined as the transformed polynomial

$$g_1^T(\zeta) = e^{:f_2:} g_1(\zeta) = g_1(e^{:f_2:}\zeta). \tag{4.22}$$

By so moving all second-order transformations to the right of first-order transformations and to the left of all higher-order ones, we will then be left with all second order terms together in the proper δ-rank sequence. From an aesthetic point of view, it is now desirable to combine these second-order terms into a single second-order term. From a practical point of view, however, this is not necessary because the linear transformations are all kept as matrices when doing numerical computations and matrix multiplication is computationally simple. This is a good thing, because it is not possible in general to combine all these terms. There are two terms, $e^{:f_2:}$ and $e^{:g_2:}$, that are of total rank 2, which means that any BCH series involving it does not terminate. On the other hand, all the other second-order transformations are of total rank 3 and higher, so that application of the BCH formula with appropriate truncation according to the total rank criterion will yield an answer with a finite number of terms.

4.4.4 Moving Higher-Order Terms

Moving transformations of order 3 and higher is essentially the same problem mathematically as that of the first-order terms, but because it is simpler, and because we need to combine terms of the same order, we use the combine-and-separate method rather than the exchange-and-separate method. Ignoring the first- and second-order terms, which now have been pushed to the left, we may look at the concatenation problem as reformulating a product of transformations each of third or higher order into the standard factorization. Specifically, we look at the problem as putting

$$e^{:f_3:} e^{:f_4:} \cdots e^{:f_N:}\ e^{:g_3:} e^{:g_4:} \cdots e^{:g_N:} \tag{4.23}$$

into standard factorization. Although there are actually many terms of each order 3 and higher resulting from the first-order calculation, knowing this solution will allow us to apply it repeatedly to obtain the general solution. As with moving the first-order term, we start by moving the term $e^{:g_3:}$ leftward, ignoring all the other g transformations, then proceed to move the term $e^{:g_4:}$, and so on. This process gets successively easier.

The elemental task in this process is to re-express the two adjacent terms $e^{:a_n:} e^{:b_m:}$ in the standard factorization. Because now $n \geq 3$ and $m \geq 3$, there is no problem with first- or second-order terms recurring. Using the combine-and-separate method, the BCH formula is applied to obtain

$$e^{:a_n:} e^{:b_m:} = e^{:a_n + b_m + \frac{1}{2}[a_n, b_m] + \cdots:}. \tag{4.24}$$

The combined terms can be any δ-rank from m through N, although by the nature of the BCH formula, only certain ones will appear. After applying

the separation procedure, this exponent will be expressed as a product of transformations in the standard factorization. In this case, we are solving equation (3.8), but with no first- or second-order terms, and perhaps some other terms missing as well.

After a transformation of a given-order is moved to the left, it must be combined with the other term of the same δ-rank. In this respect, the problem is very different from the first-order case, because there the combination of the terms of like rank was trivial. Here, the combination once again produces a whole menagerie of terms and must be treated in the same way as when any other term is encountered.

Let us consider specifically one of these elemental tasks with $N = 6$, in fact one of combining terms of like δ-rank, that of finding the standard factorization for $e^{:f_3:}e^{:g_3:}$. Using the BCH formula, we write

$$\begin{aligned} e^{:f_3:}e^{:g_3:} &= \exp\bigl(f_3 + g_3 + \tfrac{1}{2}[f_3, g_3] + \tfrac{1}{12}[f_3, [f_3, g_3]] \\ &\qquad + \tfrac{1}{12}[g_3, [g_3, f_3]] - \tfrac{1}{24}[f_3, [g_3, [f_3, g_3]]]\bigr) \\ &= e^{:k_3:}e^{:k_4:}e^{:k_5:}e^{:k_6:}, \end{aligned} \tag{4.25}$$

and solve for k. Order by order, the terms combined in the exponent are

$$\begin{aligned} j_3 &= f_3 + g_3 \\ j_4 &= \tfrac{1}{2}[f_3, g_3] \\ j_5 &= \tfrac{1}{12}[f_3, [f_3, g_3]] + \tfrac{1}{12}[g_3, [g_3, f_3]] \\ j_6 &= -\tfrac{1}{24}[f_3, [g_3, [f_3, g_3]]]. \end{aligned} \tag{4.26}$$

We now apply the separation procedure. As in Section 4.4.2, we may consult Appendix 4.C for the general solution and apply it to this case, or we may perform the separation procedure ourselves. Doing the latter, we use the BCH formula to find the functions F_i of (3.9),

$$\begin{aligned} &e^{:k_3:}e^{:k_4:}e^{:k_5:}e^{:k_6:} \\ &= e^{:k_3 + k_4 + k_5 + k_6 + \frac{1}{2}[k_3, k_4] + \frac{1}{2}[k_3, k_5] + \frac{1}{12}[k_3, [k_3, k_4]]:}, \end{aligned} \tag{4.27}$$

$$\begin{aligned} j_3 &= F_3(k_3, k_4, k_5, k_6) = k_3 \\ j_4 &= F_4(k_3, k_4, k_5, k_6) = k_4 \\ j_5 &= F_5(k_3, k_4, k_5, k_6) = k_5 + \tfrac{1}{2}[k_3, k_4] \\ j_6 &= F_6(k_3, k_4, k_5, k_6) = k_6 + \tfrac{1}{2}[k_3, k_5] + \tfrac{1}{12}[k_3, [k_3, k_4]]. \end{aligned} \tag{4.28}$$

We make the substitutions

$$\begin{aligned} k_3 &\rightarrow j_3 + r_3^{(3)} \\ k_4 &\rightarrow j_4 + r_4^{(4)} \\ k_5 &\rightarrow j_5 + r_5^{(5)} \\ k_6 &\rightarrow j_6 + r_6^{(6)}. \end{aligned} \tag{4.29}$$

and get, after cancelling off the j terms,

$$\begin{aligned} 0 &= r_3^{(3)} \\ 0 &= r_4^{(4)} \\ 0 &= r_5^{(5)} + \tfrac{1}{2}[j_3 + r_3^{(3)}, j_4 + r_4^{(4)}] \\ 0 &= r_6^{(6)} + \tfrac{1}{2}[j_3 + r_3^{(3)}, j_5 + r_5^{(5)}] \\ &\quad + \tfrac{1}{12}[j_3 + r_3^{(3)}, [j_3 + r_3^{(3)}, j_4 + r_4^{(4)}]]. \end{aligned} \tag{4.30}$$

Solving for the r and substituting in (4.29) gives

$$\begin{aligned} k_3 &= j_3 = f_3 + g_3 \\ k_4 &= j_4 = \tfrac{1}{2}[f_3, g_3] \\ k_5 &= j_5 - \tfrac{1}{2}[j_3, j_4] \\ &= \tfrac{1}{12}[f_3, [f_3, g_3]] + \tfrac{1}{12}[g_3, [g_3, f_3]] - \tfrac{1}{2}[f_3 + g_3, \tfrac{1}{2}[f_3, g_3]] \\ k_6 &= j_6 - \tfrac{1}{2}[j_3, j_5] - \tfrac{1}{12}[j_3, [j_3, j_4]] \\ &= -\tfrac{1}{24}[f_3, [g_3, [f_3, g_3]]] - \tfrac{1}{2}[f_3 + g_3, \tfrac{1}{12}[f_3, [f_3, g_3]] \\ &\quad + \tfrac{1}{12}[g_3, [g_3, f_3]]] - \tfrac{1}{12}[f_3 + g_3, [f_3 + g_3, \tfrac{1}{2}[f_3, g_3]]. \end{aligned} \tag{4.31}$$

This is already the complete solution; there is no further need to iterate because the total rank of each term equals the δ-rank.

4.5 Summary

The transfer map arising from a Hamiltonian flow may be represented, including inhomogeneities, through a given order by a product of Lie transformations on phase space factored by order. Two such maps may be combined into a single map using the concatenation formula whose derivation is outlined above. In order to calculate the concatenation formula and the Lie products, it is necessary that the inhomogeneity be small. How this smallness relates to the smallness of the phase space perturbation is imposed in a general way by the algebra; one is then free to choose specifically the criterion to be in accord with physical considerations. Here, the criterion choosen is that these two quantities count equally and no term beyond a certain order N is retained.

The concatenation formula is accurate through the order computed without approximation, including feed-down effects from the inhomogeneity. It will not produce the same result as when the two transformations are computed in succession because of effects beyond order N that are lost in the concatenation. Thus if these effects are important concatenation is not suitable.

The concatenation formula have been computed and checked through $N = 6$. The result and the details of the symbolic manipulation used are given in the Appendices.

4.A A basis for the Lie algebra

The computations described above involve a considerable amount of algebraic manipulation, particularly for computations of higher N. For $N \gtrsim 5$ it becomes unfeasible to do these computations by hand, and more practical to automate this process. There are several stages where it is useful to use symbolic computation. First of all, the BCH formula (3.7) may be obtained from references (for example [15]) for the first few terms, but for more terms, it may be necessary to compute them. This can be done symbolically using a recursive procedure derived from the formula in [9], or by other methods. Second, the separation procedure may be implemented automatically to take any set of functions F_i (3.9), and invert them. Finally, throughout all these computations it is vital, particularly for large N, to keep the expressions to a manageable size. This is done by reducing the terms to a minimal independent set, *i.e.*, a basis. The last two tasks have been implemented as a program in the symbolic manipulation program *SMP* [13]. Interested readers should contact the first author for details.

In this Appendix we present the last of these tasks; a basis for the Lie algebra of up to four quantities, some of which may be duplicated. A reduction to a basis is possible because of the axioms of a Lie algebra (1.5–1.7). This builds on the work of Dragt and Forest [5], who presented a basis for an arbitrary number of distinct quantities, but the orientation here is algorithmic, *i.e.*, the way a symbolic manipulation package would project a term onto the basis.

To aid in this effort, we shall need some notation and terminology. The term *atom* will mean a single symbol that represents something in the Lie algebra. A *bracket* is a less fundamental quantity, being a bracket in the Lie algebra of either atoms or brackets. Everything in the Lie algebra is either an atom or a bracket, or both. Since we are dealing with formal symbol manipulation, the distinction between atoms and brackets is arbitrary and has no mathematical meaning. In fact, it will prove convenient to temporarily manipulate some brackets as atoms. Since they are just symbols, atoms will be thought of as having no other property than their lexical order with respect to other atoms. When a bracket is to be thought of as an atom, its lexical position will be given explicitly. The symbols 'a', 'b', 'c' and 'd' used in this Appendix are not the elements of the Lie algebra themselves, but stand for these elements considered as symbols, either atoms or brackets. For example, $[a, b]$ may be lexically ordered or not depending on what symbols a and b stand for.

A *nest* is a special kind of bracket that contains an atom in the first position and a nest or an atom in the second. We shall denote a nest by leaving off all but the outer pair of brackets, *e.g.*, $[f, g, f, g]$ means $[f, [g, [f, g]]]$. Second, we shall classify the repetition of atoms in a term by numerical counts of each different one. For example, $[f, g, f, g]$ is in the class 2×2 and $[f, g, f, h]$ is in the class $2 \times 1 \times 1$. This terminology is not restricted to nests.

Let us see how to rewrite a term in the basis for n different atoms, or $1 \times 1 \times \cdots \times 1$, as derived from Dragt and Forest [5]. Any member of the Lie algebra of this type may be reduced to a nest with the lexically first (or any particular) atom last, as may be shown by induction. First, suppose that the number of atoms is two. Then the term is automatically a nest. If the lexically first atom is not last, we use antisymmetry to rewrite the term so that it is last.

Suppose we have a term of brackets that is not a nest; $[a_1, a_2, \ldots, a_n]$, where any term except the last may be some bracket. Let us look at one specifically, and say that $a_i = [c, d]$. Then the term is $[a_1, \ldots, a_{i-1}, [c, d], a_{i+1}, \ldots, a_n]$. Using the Jacobi identity, we may rewrite this as

$$\begin{aligned}[a_1, a_2, \ldots, a_n] &= [a_1, \ldots, a_{i-1}, c, d, a_{i+1}, \ldots, a_n] \\ &\quad -[a_1, \ldots, a_{i-1}, d, c, a_{i+1}, \ldots, a_n]. \qquad \text{(A.1)}\end{aligned}$$

Now each of c and d may themselves be brackets, but they will have fewer total atoms than $[c, d]$, so if we apply this transformation repeatedly, we will eventually have only single atoms, and therefore a nest.

If the lexically first atom is not last in this nest, we must rewrite the nest so that it is. Let us suppose in $[a_1, a_2, \ldots, a_n]$ that each of the a_j stands for an atom and that a_i is the lexically first atom. If $i = n - 1$, then we apply antisymmetry and we are done. If $i < n - 1$, call $b \stackrel{\text{def}}{=} [a_{i+2}, a_{i+3}, \ldots, a_n]$. Consider now the term written supposing that b stands for an atom and with the last two brackets explicit; then apply the Jacobi identity and antisymmetry:

$$\begin{aligned}[a_1, \ldots, a_n] &= [a_1, \ldots, a_{i-1}, a_i, a_{i+1}, b] = [a_1, \ldots, a_{i-1}, [a_i, [a_{i+1}, b]]] \\ &= -[a_1, \ldots, a_{i-1}, [a_{i+1}, [b, a_i]]] + [a_1, \ldots, a_{i-1}, [b, [a_{i+1}, a_i]]] \\ &= -[a_1, \ldots, a_{i-1}, a_{i+1}, b, a_i] + [a_1, \ldots, a_{i-1}, b, a_{i+1}, a_i]. \\ &\qquad \text{(A.2)}\end{aligned}$$

Each of these terms has the a_i in the final position. Of course they may not be nests, because b can be a bracket, but by reapplying the nesting algorithm (A.1), which does not alter the position of the last atom, we will have expressed the original term as the sum of nests that have the lexically first atom last.

We now need to consider the case of duplicate atoms. Of course, all the reductions made for the case of distinct atoms apply here, but there will be further reductions that can be made if some of the quantities are the same. The general technique is to start with the Jacobi identity on the outer nest, then successively apply antisymmetry and the inner Jacobi identities to find identities. Unfortunately, a general treatment for arbitrary numbers of atoms is more involved than in the case of distinct atoms. Here, we will consider the particular terms that arise in the calculation for $N = 6$. Different aspects of the general problem are addressed in more detail in [1].

For the calculation through total order $N = 6$, no term of more than four atoms will occur. Therefore, let us consider the possibilities for three and four atoms. For three atoms, the only case is 2×1. There is clearly only one term: $[f, g, f]$ if it is the lexically first term that is duplicated, or $[g, g, f]$ if not.

For four atoms, there are three possible situations, $2 \times 1 \times 1$, 3×1, and 2×2 . There are respectively three, one, and one term in the bases for each. To analyze these three cases, one should start with the six terms arising when all atoms are different:

$$[k, h, g, f], \quad [k, g, h, f], \quad [h, k, g, f], \quad [h, g, k, f], \quad [g, h, k, f], \quad [g, k, h, f]. \tag{A.3}$$

Now we may identify k with h to find three different terms:

$$[h, h, g, f], \quad [h, g, h, f], \quad [g, h, h, f]. \tag{A.4}$$

For duplication of the lexically first atom, one must make use of the Jacobi identity on the outer bracket, to reduce the number of terms from four to three:

$$[k, f, h, f], \quad [h, f, k, f], \quad [f, k, h, f]. \tag{A.5}$$

We see easily that for 3×1 there is one term: $[g, g, g, f]$ (or $[f, f, g, f]$).

Now consider the 2×2 case. At first glance, it looks like there are two terms in the basis, $[g, f, g, f]$ and $[f, g, g, f]$. However, these can be shown to be equal by applying the Jacobi identity on the outer brackets, and antisymmetry:

$$\begin{aligned} [f, g, g, f] + [g, [g, f], f] + [[g, f], f, g] &= 0 \\ [f, g, g, f] - [g, f, g, f] &= 0. \end{aligned} \tag{A.6}$$

We have now shown that any member of the Lie algebra of atoms can be reexpressed as the linear combination of nests of these atoms with the lexically first atom last. Accounting for duplicate atoms, this set can be made smaller to form a basis. This is not necessarily the representation in the fewest number of terms, nor is it the most natural way of expressing the equations for efficient numerical computation. For either of these alternate goals, it is not obvious which way to express the equations. In fact one must try many possiblities, guided by intuition, to improve the expression.

4.B Concatenation formulae for $N = 6$

In this Appendix, the results for the computation described above for $N = 6$ are presented for reference.

First, we consider the problem of concatenating just the first-order term. The tables 4.1-4.3 at the end of this Appendix give the solution for the terms

on the right side of (4.11). In these tables, the k are described in terms of j. Since these j always have the same superscript as the k, it has been left off. Note in the tables the manifestation of the condition determining the number of brackets required (3.10): for the $k^{(3)}$ expressions, there are at most 3 brackets in a term, for the $k^{(4)}$ calculations, at most 1, and for $k^{(5)}$ and $k^{(6)}$, no brackets. The expressions for k may be obtained from Appendix 4.C by keeping only these number of brackets and setting the appropriate j to zero.

As mentioned in Section 4.4.2, it is possible to apply the concatenation rules for second- and higher-order to rewrite (4.11) in the standard factorization. In this case,

$$e^{:}f_1^{:}e^{:}f_2^{:}e^{:}f_3^{:}e^{:}f_4^{:}e^{:}f_5^{:}e^{:}f_6^{:}e^{:}g_1^{:} = e^{:}h_1^{:}e^{:}f_2^{:}e^{:}h_2^{:}e^{:}h_3^{:}e^{:}h_4^{:}e^{:}h_5^{:}e^{:}h_6^{:} \tag{B.1}$$

where

$$\begin{aligned}
h_1 &= f_1 + e^{:}f_2^{:}(g_1 + k_1^{(3)} + k_1^{(4)} + k_1^{(5)} + k_1^{(6)}) \\
h_2 &= k_2^{(3)} + k_2^{(4)} + k_2^{(5)} + k_2^{(6)} + \tfrac{1}{2}[k_2^{(3)}, k_2^{(4)} + k_2^{(5)}] + \tfrac{1}{12}[k_2^{(3)}, k_2^{(3)}, k_2^{(4)}] \\
h_3 &= k_3^{(3)} + k_3^{(4)} + k_3^{(5)} + k_3^{(6)} - [k_2^{(4)} + k_2^{(5)}, k_3^{(3)}] \\
h_4 &= k_4^{(3)} + k_4^{(4)} + k_4^{(5)} + k_4^{(6)} + \tfrac{1}{2}[k_3^{(3)}, k_3^{(4)} + k_3^{(5)}] \\
h_5 &= k_5^{(3)} + k_5^{(4)} + k_5^{(5)} + k_5^{(6)} - \tfrac{1}{6}[k_3^{(3)}, k_3^{(3)}, k_3^{(4)}] \\
h_6 &= k_6^{(6)}.
\end{aligned} \tag{B.2}$$

The formula for concatenation of the higher order terms, as described in Section 4.4.4, is fortunately much simpler. We know first of all that second-order transformations are handled using the exchange rule. We may thus consider the problem to be

$$e^{:}f_3^{:}e^{:}f_4^{:}e^{:}f_5^{:}e^{:}f_6^{:}\, e^{:}g_3^{:}e^{:}g_4^{:}e^{:}g_5^{:}e^{:}g_6^{:} = e^{:}h_3^{:}e^{:}h_4^{:}e^{:}h_5^{:}e^{:}h_6^{:}. \tag{B.3}$$

The solution is

$$\begin{aligned}
h_3 &= f_3 + g_3 \\
h_4 &= f_4 + g_4 + \tfrac{1}{2}[f_3, g_3] \\
h_5 &= f_5 + g_5 - [g_3, f_4] - \tfrac{1}{6}{:}f_3{:}^2\, g_3 + \tfrac{1}{3}{:}g_3{:}^2\, f_3 \\
h_6 &= f_6 + g_6 - [g_3, f_5] + \tfrac{1}{2}{:}g_3{:}^2\, f_4 + \tfrac{1}{2}[f_4, g_4] - \tfrac{1}{4}[f_4, f_3, g_3] \\
&\quad -\tfrac{1}{4}[g_4, f_3, g_3] + \tfrac{1}{24}{:}f_3{:}^3\, g_3 - \tfrac{1}{8}{:}g_3{:}^3\, f_3 + \tfrac{1}{8}[f_3, g_3, f_3, g_3].
\end{aligned} \tag{B.4}$$

Again, the reader is reminded that this is not necessarily the most compact or computationally efficient form. For purposes of numerical coding, it is possible to rearrange the terms so that quantities already calculated can be reused to maximum benefit. In such a code, there are three basic routines, one that moves the first-order transformation according to (B.2),

one that does the linear transformations on the polynomials according to the exchange rule, and a third that moves the higher-order terms according to (B.4).

$k_1^{(3)}$	$\tau = 3$	$j_1 + \frac{1}{2}[j_2, j_1] - \frac{1}{6}[j_1, j_3, j_1] + \frac{1}{6}[j_2, j_2, j_1]$ $-\frac{1}{8}[j_1, j_3, j_2, j_1] - \frac{1}{24}[j_2, j_1, j_3, j_1] + \frac{1}{24}[j_2, j_2, j_2, j_1]$
$k_2^{(3)}$	$\tau = 3$	$j_2 + \frac{1}{2}[j_3, j_1] - \frac{1}{12}[j_2, j_3, j_1] + \frac{1}{6}[j_3, j_2, j_1]$ $-\frac{1}{24}[j_2, j_3, j_2, j_1] - \frac{1}{24}[j_3, j_1, j_3, j_1] + \frac{1}{24}[j_3, j_2, j_2, j_1]$
$k_3^{(3)}$	$\tau = 3$	$j_3 + \frac{1}{2}[j_3, j_2] - \frac{1}{6}[j_2, j_3, j_2] + \frac{1}{6}[j_3, j_3, j_1] + \frac{1}{24}[j_2, j_2, j_3, j_2]$ $-\frac{1}{8}[j_2, j_3, j_3, j_1] + \frac{1}{24}[j_3, j_2, j_3, j_1] + \frac{1}{24}[j_3, j_3, j_2, j_1]$
$k_4^{(3)}$	$\tau = 5$	$-\frac{1}{12}[j_3, j_3, j_2] + \frac{1}{24}[j_3, j_2, j_3, j_2] - \frac{1}{24}[j_3, j_3, j_3, j_1]$
$k_5^{(3)}$	$\tau = 6$	$\frac{1}{24}[j_3, j_3, j_3, j_2]$

$j_1^{(3)}$	$\frac{1}{2}{:}{-}g_1{:}^2 f_3$
$j_2^{(3)}$	$:{-}g_1{:} f_3$
$j_3^{(3)}$	f_3

TABLE 4.1. The polynomials $k^{(3)}$.

$k_1^{(4)}$	$\tau = 4$	$j_1 + \frac{1}{2}[j_2, j_1]$
$k_2^{(4)}$	$\tau = 4$	$j_2 + \frac{1}{2}[j_3, j_1]$
$k_3^{(4)}$	$\tau = 4$	$j_3 + \frac{1}{2}[j_3, j_2] + \frac{1}{2}[j_4, j_1]$
$k_4^{(4)}$	$\tau = 4$	$j_4 + \frac{1}{2}[j_4, j_2]$
$k_5^{(4)}$	$\tau = 6$	$\frac{1}{2}[j_4, j_3]$

$j_1^{(4)}$	$\frac{1}{6}{:}{-}g_1{:}^3 f_4$
$j_2^{(4)}$	$\frac{1}{2}{:}{-}g_1{:}^2 f_4$
$j_3^{(4)}$	$:{-}g_1{:} f_4$
$j_4^{(4)}$	f_4

TABLE 4.2. The polynomials $k^{(4)}$.

$k_1^{(5)} = j_1^{(5)}$	$\frac{1}{24}{:}{-}g_1{:}^4 f_5$
$k_2^{(5)} = j_2^{(5)}$	$\frac{1}{6}{:}{-}g_1{:}^3 f_5$
$k_3^{(5)} = j_3^{(5)}$	$\frac{1}{2}{:}{-}g_1{:}^2 f_5$
$k_4^{(5)} = j_4^{(5)}$	$:{-}g_1{:} f_5$
$k_5^{(5)} = j_5^{(5)}$	f_5

$k_1^{(6)} = j_1^{(6)}$	$\frac{1}{120}{:}{-}g_1{:}^5 f_6$
$k_2^{(6)} = j_2^{(6)}$	$\frac{1}{24}{:}{-}g_1{:}^4 f_6$
$k_3^{(6)} = j_3^{(6)}$	$\frac{1}{6}{:}{-}g_1{:}^3 f_6$
$k_4^{(6)} = j_4^{(6)}$	$\frac{1}{2}{:}{-}g_1{:}^2 f_6$
$k_5^{(6)} = j_5^{(6)}$	$:{-}g_1{:} f_6$
$k_6^{(6)} = j_6^{(6)}$	f_6

TABLE 4.3. The polynomials $k^{(5)}$ and $k^{(6)}$.

4.C The results of the separation procedure for $N = 6$

In this Appendix the solution to equation (3.8) is presented for all j terms of total rank at least 3 through a cutoff $N = 6$. This is sufficient to calculate all the results of Appendix 4.B using the exchange-and-separate method on the first-order transformations. By condition (3.10), we need at most 3 Poisson brackets in the BCH formula; this is as given explicitly in (3.7). Therefore, by combining the multiple exponents, we have

$$\begin{aligned}
j_1 &= k_1 - \tfrac{1}{2}[k_2,k_1] - \tfrac{1}{12}[k_1,k_3,k_1] + \tfrac{1}{12}[k_2,k_2,k_1] + \tfrac{1}{12}[k_1,k_3,k_2,k_1] \\
j_2 &= k_2 - \tfrac{1}{2}[k_3,k_1] - \tfrac{1}{12}[k_1,k_4,k_1] - \tfrac{1}{6}[k_2,k_3,k_1] + \tfrac{1}{3}[k_3,k_2,k_1] \\
&\quad + \tfrac{1}{12}[k_2,k_3,k_2,k_1] + \tfrac{1}{24}[k_3,k_1,k_3,k_1] - \tfrac{1}{12}[k_3,k_2,k_2,k_1] \\
j_3 &= k_3 - \tfrac{1}{2}[k_3,k_2] - \tfrac{1}{2}[k_4,k_1] - \tfrac{1}{12}[k_2,k_3,k_2] - \tfrac{1}{6}[k_2,k_4,k_1] \\
&\quad + \tfrac{1}{12}[k_3,k_3,k_1] + \tfrac{1}{3}[k_4,k_2,k_1] + \tfrac{1}{12}[k_3,k_2,k_3,k_1] - \tfrac{1}{12}[k_3,k_3,k_2,k_1] \\
j_4 &= k_4 - \tfrac{1}{2}[k_4,k_2] - \tfrac{1}{2}[k_5,k_1] - \tfrac{1}{12}[k_2,k_4,k_2] + \tfrac{1}{12}[k_3,k_3,k_2] \\
&\quad - \tfrac{1}{6}[k_3,k_4,k_1] + \tfrac{1}{3}[k_4,k_3,k_1] + \tfrac{1}{24}[k_3,k_2,k_3,k_2] \\
j_5 &= k_5 - \tfrac{1}{2}[k_4,k_3] - \tfrac{1}{2}[k_5,k_2] - \tfrac{1}{6}[k_3,k_4,k_2] + \tfrac{1}{3}[k_4,k_3,k_2] \\
j_6 &= k_6 - \tfrac{1}{2}[k_5,k_3] - \tfrac{1}{12}[k_3,k_4,k_3].
\end{aligned} \tag{C.1}$$

When inverted, we have the results for k in terms of j, through total rank six:

$$\begin{aligned}
k_1 &= j_1 + \tfrac{1}{2}[j_2,j_1] - \tfrac{1}{6}[j_1,j_3,j_1] + \tfrac{1}{6}[j_2,j_2,j_1] \\
&\quad - \tfrac{1}{8}[j_1,j_3,j_2,j_1] - \tfrac{1}{24}[j_2,j_1,j_3,j_1] + \tfrac{1}{24}[j_2,j_2,j_2,j_1] \\
k_2 &= j_2 + \tfrac{1}{2}[j_3,j_1] - \tfrac{1}{6}[j_1,j_4,j_1] - \tfrac{1}{12}[j_2,j_3,j_1] + \tfrac{1}{6}[j_3,j_2,j_1] \\
&\quad - \tfrac{1}{24}[j_2,j_3,j_2,j_1] - \tfrac{1}{24}[j_3,j_1,j_3,j_1] + \tfrac{1}{24}[j_3,j_2,j_2,j_1] \\
k_3 &= j_3 + \tfrac{1}{2}[j_3,j_2] + \tfrac{1}{2}[j_4,j_1] - \tfrac{1}{6}[j_2,j_3,j_2] - \tfrac{1}{3}[j_2,j_4,j_1] \\
&\quad + \tfrac{1}{6}[j_3,j_3,j_1] + \tfrac{1}{6}[j_4,j_2,j_1] + \tfrac{1}{24}[j_2,j_2,j_3,j_2] \\
&\quad - \tfrac{1}{8}[j_2,j_3,j_3,j_1] + \tfrac{1}{24}[j_3,j_2,j_3,j_1] + \tfrac{1}{24}[j_3,j_3,j_2,j_1] \\
k_4 &= j_4 + \tfrac{1}{2}[j_4,j_2] + \tfrac{1}{2}[j_5,j_1] - \tfrac{1}{6}[j_2,j_4,j_2] - \tfrac{1}{12}[j_3,j_3,j_2] + \tfrac{1}{6}[j_3,j_4,j_1] \\
&\quad - \tfrac{1}{12}[j_4,j_3,j_1] + \tfrac{1}{24}[j_3,j_2,j_3,j_2] - \tfrac{1}{24}[j_3,j_3,j_3,j_1] \\
k_5 &= j_5 + \tfrac{1}{2}[j_5,j_2] + \tfrac{1}{2}[j_4,j_3] - \tfrac{1}{12}[j_3,j_4,j_2] \\
&\quad - \tfrac{1}{12}[j_4,j_3,j_2] + \tfrac{1}{24}[j_3,j_3,j_3,j_2] \\
k_6 &= j_6 + \tfrac{1}{2}[j_5,j_3] + \tfrac{1}{12}[j_3,j_4,j_3].
\end{aligned} \tag{C.2}$$

The results in the tables of Appendix 4.B, as well as the examples of Sections 4.4.2 and 4.4.4, may be checked against this by setting the appropriate j terms to zero and discarding the Poisson brackets whose total rank exceeds six.

4.6 Acknowledgements

This paper is based on the dissertation [10] of one of the authors. We would also like to thank the Inference Corporation, who provided much assistance in understanding and applying *SMP*. This work was supported by U.S. Department of Energy contract number DE-AS05-80ER-10666.

4.7 References

[1] N. Bourbaki, *Elements of Mathematics, Lie Groups and Lie Algebras* (Addison-Wesley, Reading, Massachusetts, 1971); Part I: Chapters 1–3.

[2] D. R. Douglas, Ph.D. dissertation, University of Maryland Physics Department Technical Report (1982).

[3] A.J. Dragt, Lectures on Nonlinear Orbit Dynamics, in: *Physics of High Energy Particle Accelerators*, Proceedings of the 1981 Summer School on High Energy Particle Accelerators, New York, R.A. Carrigan, F.R. Huson, and M. Month, eds., American Institute of Physics, Conference Proceedings Number 87, 1982, p. 147.

[4] A.J. Dragt, and J.M. Finn, Lie series and invariant functions for analytic symplectic maps, *J. Math. Phys.* **17**, 2215 (1976).

[5] A.J. Dragt, and E. Forest, Computation of nonlinear behavior of Hamiltonian systems using Lie algebraic methods, *J. Math. Phys.* **24**, 2734 (1983).

[6] A.J. Dragt *et al.*, MARYLIE 3.0 manual, University of Maryland Technical Report (1988).

[7] A.J. Dragt, L.M. Healy, F.Neri, R.D. Ryne, D.R. Douglas, and E.Forest, MARYLIE 3.0 —A Program for Nonlinear Analysis of Accelerators and Beamline Lattices, *IEEE Trans. Nucl. Sci.* **NS-32**, 2311 (1985).

[8] A.J. Dragt, F. Neri, G. Rangarajan, D. Douglas, L.M. Healy, and R.D. Ryne, Lie algebraic treatment of linear and nonlinear beam dynamics, *Ann. Rev. Nucl. Part. Sci.* **38**, 455–496 (1988).

[9] M. Hausner and J. Schwartz, *Lie Groups Lie Algebras* (Gordon and Breach, New York, 1968).

[10] L.M. Healy, Ph.D. dissertation, University of Maryland Physics Department Technical Report (1986).

[11] L.M. Healy and A.J. Dragt, Lie algebraic methods for treating lattice parameter errors in accelerators. In: Proceedings of the 1987 IEEE Particle Accelerator Conference Vol. 2, (1987), p. 1060 *et seq.*

[12] L.M. Healy and A.J. Dragt, Computation of nonlinear behavior of inhomogeneous Hamiltonian systems using Lie algebraic methods (in preparation).

[13] *SMP Manual*, Inference Corporation, 1985.

[14] F.J. Murray and K.S. Miller, *Existence Theorems* (New York University Press, New York, 1954).

[15] V.S. Varadarajan, *Lie Groups, Lie Algebras, and their Representations* (Springer-Verlag, New York, 1984).