5-2015

Explicit Construction of First Integrals for the Toda Flow on a Classical Simple Lie Algebra

Patrick Seegmiller
Utah State University

EXPLICIT CONSTRUCTION OF FIRST INTEGRALS FOR THE TODA FLOW ON A CLASSICAL SIMPLE LIE ALGEBRA

by

Patrick Seegmiller

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mathematics

Approved:

Zhaohu Nie Major Professor	Ian Anderson Committee Member
Nathan Geer Committee Member	Mark McLellan Vice President for Research Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2015

Abstract

Explicit Construction of First Integrals for the

Toda Flow on a Classical Simple Lie Algebra

by

Patrick Seegmiller, Master of Science

Utah State University, 2015

Major Professor: Dr. Zhaohu Nie
Department: Mathematics and Statistics

The Toda flow is a Hamiltonian system which evolves on the dual of the Borel subalgebra of a complex Lie algebra $\mathfrak{g}$. The dual of the Borel subalgebra can be identified with an affine subspace of its negative plus the element given by the sum of the simple root vectors in $\mathfrak{g}$. The system has been proven completely integrable in the Liouville sense on a generic coadjoint orbit for the Borel subgroup. This paper gives a verification of integrability of the Toda flow on classical simple Lie algebras and describes a method for the construction of a complete collection of integrals of motion for each. After this description, an implementation of the outlined procedures is given in the Maple programming environment, together with explicit examples, demonstrating both the accuracy of the procedure and the efficacy of the Maple programming code.

(96 pages)

Public Abstract

Explicit Construction of First Integrals for the

Toda Flow on a Classical Simple Lie Algebra

by

Patrick Seegmiller, Master of Science

Utah State University, 2015

Major Professor: Dr. Zhaohu Nie
Department: Mathematics and Statistics

The Toda flow is a generalization of a dynamical system describing the interaction of particles in a one-dimensional crystal. The concepts and energy and conservation are prominent in the study of dynamical systems, and quantities which remain the same over the evolution of a system provide valuable insights into the system's behavior. In the realm of mathematics these quantities are called first integrals, or integrals of motion. This paper provides a background for study of the Toda flow, a verification of its integrability, and programming code for finding these quantities which remain unchanged over the evolution of the system.

Acknowledgements

I would like to express my deep gratitude to my supervisory committee for their insights, constructive criticism, and for challenging me. I would especially like to thank my Major Professor Zhaohu Nie for his superhuman level of patience with my inadequacies and idiosyncrasies, for giving guidance without just giving me the answers, and for his words of support and motivation. I want to express gratitude to my parents for always supporting my goals, however crazy or misguided they may be, my in-laws for their unwarranted admiration and assistance, and my two boys for their unconditional love even when “Daddy” is feeling overwhelmed with work. Finally, I want to thank the love of my life, Heather, for her continued support, even when my work kept me from home from the early hours of the morning to the late hours of night on almost a daily basis, for being an inspiration in hard work and patience, for reminding my children that they do in fact have a father, and for every other amazing thing she does for me.

Patrick Seegmiller

Contents

List of Figures

Chapter 1
Introduction

Integrable systems are of interest to both mathematicians and physicists as their study spans a multitude of disciplines ranging from mechanics, both classical and quantum, to algebraic geometry.

An important example of a nonlinear integrable system is called the Toda lattice, a mechanical system which describes the interaction of particles in a one-dimensional crystal. From the problem's introduction by Morikazu Toda in 1967, the Toda lattice and its generalizations have remained a central topic of study, initially by physicists, and more recently by mathematicians.

One such generalization is known as the Toda flow. This system, while lacking any direct physical interpretation, nevertheless is a Hamiltonian system of profound interest, studied in depth in [3, 8, 2]. In fact, the Toda flow has been shown to be completely Liouville integrable on a generic coadjoint orbit.

The central goal of this paper is a verification of complete Liouville integrability of the Toda flow on the dual of the Borel subalgebra of a classical Lie algebra. This will be accomplished by providing a method of construction for a complete set of integrals of motion in involution; a modified version of the method described in [8]. In addition we provide a Maple implementation of the method described herein along with Maple procedures for constructing a specialized basis for the classical Lie algebras of arbitrary rank.

As stated above, study of the Toda flow spans several disciplines, and consequently there are numerous foundational topics which must be introduced to properly explain and understand the Toda flow. To avoid the need for writing an entire textbook, we will assume at least a basic knowledge of some elementary topics from differential geometry and classical mechanics such as differentiable manifolds, the tangent and cotangent bundles, differential forms, the exterior and interior derivatives, Hamilton's equations, and systems of ordinary differential equations. We will not, however, assume experience with Lie algebras, symplectic manifolds, Poisson manifolds, coadjoint orbits, or integrable systems, and we will introduce each of these as the need arises.

As the author has yet to encounter a text which introduces integrable systems without a considerable body of prerequisite material (even taking the assumptions above into consideration), a secondary goal of this paper, which will ultimately succeed or fail with the reception and understanding of individual readers, is to provide a resource which introduces the study of the Toda flow in a manner more accessible to one not yet schooled in the aforementioned topics. With these goals in mind we will proceed by providing a brief outline of the material to follow in subsequent chapters.

Chapter 2 will constitute our attempt at reaching our secondary goal. Therein we will provide the necessary background for our study of Toda flow. We will begin by introducing Lie algebras, Lie subalgebras and in particular the Borel subalgebras, and the classification of simple Lie algebras by Dynkin diagrams. Following this, we introduce Hamiltonian systems and their integrability in the context of Poisson structures, while briefly exploring the underlying symplectic geometry. Next, we return to the dual of a Lie algebra to explore the connection between symplectic manifolds and coadjoint orbits, introducing the Lie-Poisson bracket. Finally we mention some of the structural properties for semisimple Lie algebras which allow simplification of the Lie-Poisson bracket.

In Chapter 3 we introduce the Toda lattice as well as some of the generalizations it has inspired, and introduce the Toda flow. We briefly examine integrability of each as we proceed. This process requires that we introduce the concept of a Lax pair, as well as its relationship to Liouville integrability. We demonstrate the utility of the symmetric form of the Lax pair for the Toda lattice, and introduce the asymmetric form given by Kostant in [10]. Finally we introduce the Toda flow, whose integrability will be established in the chapter which follows.

Chapter 4 contains the central focus of the paper, the algorithm for verification of integrability of the Toda flow for classical simple Lie algebras. To accomplish this, we employ a combinatorial argument which allows us to justify integrability for classical Lie algebras of arbitrary rank. This abstract formulation will be followed by an exhibition and explanation of the author's Maple implementation of the method of constructing first integrals, after which examples are given to demonstrate the efficacy of the implementation.

In Chapter 5 we conclude our study by identifying the primary difficulties encountered, any unanswered questions, and giving suggestions of areas for further exploration.

Chapter 2

Notation and Preliminary Constructions

What follows constitutes, in the opinion of the author, the minimal prerequisite material for understanding the Toda flow.

We begin with an overview of elementary Lie theory, along with an explanation of the classification of the complex simple Lie algebras by Dynkin diagrams. Examples are provided along the way to illustrate some of the less obvious definitions and properties, but proofs of elementary results are not provided as they are available in any introductory text on the topic (such as [4, 7, 14]).

Hamiltonian systems are first introduced informally. The natural structure of phase space as a symplectic manifold is introduced, and the natural Poisson structure found in every symplectic manifold is given. This is followed by an explanation of a Poisson structure on the dual of a Lie algebra. Coadjoint orbits, while unnecessary for any practical purposes in this paper, are defined as they possess a natural symplectic structure and are foundational to the mathematics underlying the Toda flow.

2.1 Lie Algebras

A **Lie algebra** is a vector space $\mathfrak{g}$ over a field $\mathbb{K}$ closed under an operation $[-,-] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$, called a **Lie bracket**, such that for all $x, y, z \in \mathfrak{g}$ and $a, b \in \mathbb{K}$ we have:

$$\begin{aligned} [ax + by, z] &= a[x, z] + b[y, z], \\ [x, ay + bz] &= a[x, y] + b[x, z], \\ [x, x] &= 0, \\ [x, [y, z]] + [y, [z, x]] &+ [z, [x, y]] = 0. \end{aligned} \tag{2.1}$$

It is worth noting that (2.1) implies

$$\begin{aligned}0 &= [x+y, x+y] = [x+y, x] + [x+y, y] = [x,x] + [y,x] + [x,y] + [y,y] \\ &= 0 + [y,x] + [x,y] + 0 = [y,x] + [x,y]\end{aligned}$$

which makes $[x, y] = -[y, x]$ (i.e. $[-,-]$ is skew-symmetric). Now we will introduce some of the substructures of Lie algebras.

A **Lie subalgebra** is a vector subspace $\mathfrak{l}$ of $\mathfrak{g}$ such that if $x, y \in \mathfrak{l}$ then $[x, y] \in \mathfrak{l}$. If for each $x \in \mathfrak{l}$ and $y \in \mathfrak{g}$, $[x, y] \in \mathfrak{l}$, then we say that $\mathfrak{l}$ is an **ideal**.

Note that each of these substructures' definitions depend directly on the bracket operation. Perhaps unsurprisingly, a rough classification of Lie algebras can be made solely by examining the algebra's bracket operation. For example, if $[-,-] \equiv 0$ we call $\mathfrak{g}$ Abelian. We say that $\mathfrak{g}$ is simple whenever it is non-Abelian and contains no ideals other than 0 and itself. Furthermore, if $\mathfrak{g}$ is a direct sum of simple Lie algebras, it is called semisimple. Unless otherwise specified, all Lie algebras hereafter are assumed complex simple or split-real simple.

Let's look at an example to illustrate our definition of a Lie algebra. Define the matrices $X, Y,$ and H as follows:

$$X = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}; \quad Y = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}; \quad H = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

We claim that these matrices form a basis for a Lie algebra, where the bracket operation $[-,-]$ is given by the matrix commutator $[A, B] = AB - BA$. To show that this space does in fact constitute a Lie algebra, we need to verify the properties of the bracket operation defined above. In fact, we can prove the more general result that for any vector space V of matrices closed under the bracket $[-,-]$, defined by the matrix commutator, is a Lie algebra. Let $A, B,$ and C be matrices in $\text{span}\{X, Y, H\}$, and

$a, b, c \in \mathbb{C}$. Then we have:

$$\begin{aligned}
[aA + bB, C] &= (aA + bB)C - C(aA + bB) = aAC + bBC - CaA - CbB \\
&= aAC - aCA + bBC - bCB = a(AC - CA) + b(BC - CB) \\
&= a[A, C] + b[B, C]. \\
[A, bB + cC] &= A(bB + cC) - (bB - cC)A = AbB + AcC - bBA - cCA \\
&= bAB - bBA + cAC - cCA = b(AB - BA) + c(AC - CA) \\
&= b[A, B] + c[A, C]. \\
[A, A] &= AA - AA = 0. \\
[A, [B, C]] &+ [B, [C, A]] + [C, [A, B]] \\
&= [A, BC - CB] + [B, CA - AC] + [C, AB - BA] \\
&= A(BC - CB) - (BC - CB)A + B(CA - AC) - (CA - AC)B \\
&\quad + C(AB - BA) - (AB - BA)C \\
&= ABC - ACB - BCA + CBA + BCA - BAC - CAB + ACB \\
&\quad + CAB - CBA - ABC + BAC \\
&= (ABC - ABC) + (ACB - ACB) + (BCA - BCA) \\
&\quad + (CBA - CBA) + (BAC - BAC) + (CAB - CAB) \\
&= 0.
\end{aligned}$$

It is important to note that while we have shown that the matrix commutator is a Lie bracket, we have not shown whether span$\{X, Y, H\}$ is closed under this Lie bracket. It is clear upon inspection that $\{X, Y, H\}$ forms a basis for the space of 2×2 trace-free matrices. To know more about the structure of this space as a Lie algebra we will need to examine their brackets one with another.

$$
\begin{aligned}
[X,Y] &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = H \\
[H,X] &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix} = 2X \\
[H,Y] &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ -2 & 0 \end{pmatrix} = -2Y
\end{aligned}
$$

With these calculations alone, all possible products are accounted for by skew-symmetry of the Lie bracket. Thus the multiplication table for the Lie algebra $\mathfrak{g} = \text{span}\{X, Y, H\}$ can be written as follows:

	H	X	Y
$H\vert$	0	$2X$	$-2Y$
$X\vert$	$-2X$	0	H
$Y\vert$	$2Y$	$-H$	0

Therefore we can conclude that $\{X, Y, H\}$ does in fact form a basis for the Lie algebra of 2×2 trace-free matrices. There is more we can extrapolate from our multiplication table. For example, the Lie algebra contains no nontrivial ideals nor is it Abelian, allowing us to conclude that it is a simple Lie algebra. As a matter of fact, $\{X, Y, H\}$ is a basis for the complex simple Lie algebra $\mathfrak{sl}(2, \mathbb{C})$.

One class of Lie algebras of particular interest for our groundwork are those called **nilpotent**. To understand this concept we will construct a series of Lie algebras. First we examine a particular Lie subalgebra called the **derived Lie algebra**, which is defined by $\mathfrak{g}_1 = [\mathfrak{g}, \mathfrak{g}]$, the span of all the brackets of a basis for $\mathfrak{g}$. Now we can give a recursive definition for the subsequent terms in our series. We define $\mathfrak{g}_{n+1} = [\mathfrak{g}_{\mathfrak{n}}, \mathfrak{g}]$. This gives us a descending chain of Lie algebras, known as the **lower central series**:

$$\mathfrak{g} \supseteq \mathfrak{g}_1 \supseteq \mathfrak{g}_2 \supseteq \dots \supseteq \mathfrak{g}_k \supseteq \dots$$

Note that by our definition for a semisimple Lie algebra $\mathfrak{g}$ we have $\mathfrak{g} = [\mathfrak{g}, \mathfrak{g}]$, making the series constant. With this new implement we can now define a nilpotent Lie algebra. We say a Lie algebra $\mathfrak{g}$ is nilpotent if there exists an $n \in \mathbb{N}$ such that $\mathfrak{g}_n = 0$. This definition can seem difficult to work with. How exactly can we use this idea? It turns out that there is an alternate formulation of a nilpotent Lie algebra. This will require a small degree of representation theory.

2.1.1 Lie Algebra Representations

Let V be a vector space and $\mathfrak{g}$ be a Lie algebra. A **Lie algebra representation** is a map $\rho : \mathfrak{g} \to \mathfrak{gl}(V)$ which takes every $x \in \mathfrak{g}$ to a linear transformation which is denoted by ρ_x. We require ρ to have the property that for every $x, y \in \mathfrak{g}$ we have $\rho_{[x,y]} = [\rho_x, \rho_y]$, where the bracket operation on elements of $\mathfrak{gl}(V)$ for finite dimensional V is the matrix commutator. Therefore $[\rho_x, \rho_y] = \rho_x\rho_y - \rho_y\rho_x$. With this property we call ρ a **Lie algebra homomorphism**. It is important to note that the notation is often abused and instead of referring to a map ρ, V itself is referred to as the representation of $\mathfrak{g}$.

A representation is a powerful map. It allows us to change from an abstract setting to the concrete and well-understood setting of linear algebra. One representation will be particularly useful for us later on, as well as in our current attempt at understanding nilpotent Lie algebras. Since each Lie algebra has an underlying vector space structure, we can consider a representation where V from the definition above is equal to $\mathfrak{g}$. This representation, which we generally express as $\mathrm{ad} : \mathfrak{g} \to \mathfrak{gl}(\mathfrak{g})$ is called the **adjoint representation**. To be more precise, $x \mapsto \mathrm{ad}_x$, where $\mathrm{ad}_x : \mathfrak{g} \to \mathfrak{g}$ is defined by $\mathrm{ad}_x(y) = \mathrm{ad}(x)(y) = [x, y]$ for all $y \in \mathfrak{g}$. With this definition under our belt, we can return to the topic of nilpotency.

We have reached a point where we can think of nilpotent Lie algebras in a more familiar and understandable way. For a nilpotent Lie algebra under the adjoint representation there exists a number k such that for every $x \in \mathfrak{g}$, $\mathrm{ad}_x^k = 0$. That is, ad_x is a nilpotent linear transformation. Thus, nilpotency of a Lie algebra can be understood in a linear algebraic sense. Let's look at an example of a nilpotent Lie algebra.

Let $\mathfrak{g}$ be the space of strictly upper triangular 3×3 matrices over the field of real numbers. An obvious choice of basis is $\{X_1, X_2, X_3\}$, where:

$$X_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad X_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad X_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

This is called the Heisenberg algebra $H_3(\mathbb{R})$. Let's look at its lower central series. The brackets are easily calculated to be:

$$[X_1, X_2] = 0, \quad [X_2, X_3] = 0, \quad [X_1, X_3] = X_2.$$

Therefore if we let $H_3(\mathbb{R})_1 = \text{span}\{X_2\}$, then by skew-symmetry of the Lie bracket, the lower central series is easily seen to be

$$H_3(\mathbb{R}) \supseteq H_3(\mathbb{R})_1 \supseteq 0.$$

Thus we can conclude that the $H_3(\mathbb{R})$ is a nilpotent Lie algebra. Alternatively we could examine $H_3(\mathbb{R})$ under the adjoint representation:

$$\text{ad}_{X_1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \text{ad}_{X_2} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \text{ad}_{X_3} = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Note that the brackets are preserved:

$$[\text{ad}_{X_1}, \text{ad}_{X_2}] = 0, \quad [\text{ad}_{X_2}, \text{ad}_{X_3}] = 0, \quad [\text{ad}_{X_1}, \text{ad}_{X_3}] = \text{ad}_{X_2}.$$

Furthermore, we have that $\text{ad}^2_{X_i} = 0$ for each $i \in \{1, 2, 3\}$, verifying nilpotency of $H_3(\mathbb{R})$ using our linear algebraic characterization.

2.1.2 Cartan Subalgebras

For us, the primary importance of nilpotent Lie algebras is this: in every semisimple Lie algebra $\mathfrak{g}$ there exists a subalgebra $\mathfrak{h}$ which has the property that if $[x, y] \in \mathfrak{h}$ for every $x \in \mathfrak{h}$, then $y \in \mathfrak{h}$. Any subalgebra $\mathfrak{h}$ with this property is called **self-normalizing**.

When such an $\mathfrak{h}$ is also nilpotent, then $\mathfrak{h}$ is called a **Cartan subalgebra**. The dimension of $\mathfrak{h}$ is called the **rank** of $\mathfrak{g}$. Such a subalgebra exists for any finite dimensional Lie algebra over an infinite field [7].

Since every Lie algebra $\mathfrak{g}$ we are studying is finite dimensional over an algebraically closed field of characteristic zero, all Cartan subalgebras are conjugate under automorphisms of $\mathfrak{g}$, making every Cartan subalgebra $\mathfrak{h}$ isomorphic. It is also worth noting that since all of the Lie algebras we are studying in this paper are linear (that is to say they are Lie subalgebras of the Lie algebra of endomorphisms of the underlying vector space allowing us to view them as matrix algebras), any Cartan subalgebra corresponds to elements which are diagonalizable as endomorphisms of $\mathfrak{g}$ [4, 7]. Thus, since elements of a Cartan subalgebra $\mathfrak{h}$ are simultaneously diagonalizable, we have $[x, y] = 0$ for every $x, y \in \mathfrak{h}$ making $\mathfrak{h}$ abelian.

2.1.3 Toward a Classification of Complex Semisimple Lie Algebras

The purpose of introducing this material is to briefly explain the background for understanding the Dynkin diagram and the extended Dynkin diagram, the latter of which is useful in the study of the Toda flow. As the utility of these topics, for our purposes, lies primarily in the end result (the extended Dynkin diagrams) we will move through the classification rather quickly.

Let $\mathfrak{g}$ be a Lie algebra over a field $\mathbb{K}$. Define a map $B : \mathfrak{g} \times \mathfrak{g} \to \mathbb{K}$ so that for $x, y \in \mathfrak{g}$:

$$B(x, y) = \mathrm{tr}(\mathrm{ad}(x)\mathrm{ad}(y)) \tag{2.2}$$

This definition gives a symmetric bilinear form on $\mathfrak{g}$ known as the **Killing form**.

The Killing form has some interesting properties: B is an invariant form, meaning that $B([x, y], z) = B(x, [y, z])$, that for $\mathfrak{g}$ simple, any invariant symmetric bilinear form is a scalar multiple of B, and B is invariant under automorphisms ϕ of $\mathfrak{g}$. That is $B(\phi(x), \phi(y)) = B(x, y)$.

Two other important facts to note about B are that a Lie algebra $\mathfrak{g}$ is semisimple if and only if B is non-degenerate, and for $\mathfrak{g}$ nilpotent, B is zero [7]. This is known as **Cartan's criterion**.

To illustrate this idea, let's turn back to the example of $\mathfrak{sl}(2, \mathbb{C})$ given above, and calculate the matrix for its Killing form. First we need to calculate the adjoint matrices for X, Y and H. With our multiplication table we can literally read off the entries.

$$\mathrm{ad}_H = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{pmatrix}, \quad \mathrm{ad}_X = \begin{pmatrix} 0 & 0 & 1 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathrm{ad}_Y = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix}.$$

Now let's examine the products of these matrices. Then all that will remain is to calculate the trace of all of our results.

$$\begin{aligned} \mathrm{ad}_X \mathrm{ad}_Y &= \begin{pmatrix} 0 & 0 & 1 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \\ \mathrm{ad}_H \mathrm{ad}_X &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 \\ -4 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \\ \mathrm{ad}_H \mathrm{ad}_Y &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -4 & 0 & 0 \end{pmatrix}. \end{aligned}$$

We have only three more calculations to perform by commutativity of the trace of a product of matrices.

$$\begin{aligned}
\mathrm{ad}_X^2 &= \begin{pmatrix} 0 & 0 & 1 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix} \\
\mathrm{ad}_Y^2 &= \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -2 & 0 \end{pmatrix}. \\
\mathrm{ad}_H^2 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -2 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix}.
\end{aligned}$$

Now we can write down a matrix B for the Killing form on $\mathfrak{sl}(2, \mathbb{C})$ by looking at the trace for all of our results.

$$B = \begin{pmatrix} 8 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 4 & 0 \end{pmatrix}$$

Note that $\det(B) = -128 \neq 0$. Thus B has full rank and the Killing form on $\mathfrak{sl}(2, \mathbb{C})$ is nondegenerate, giving an alternate verification of semisimplicity.

2.1.4 The Root Space Decomposition

Let $\mathfrak{g}$ be a Lie algebra with a Cartan subalgebra $\mathfrak{h}$. Suppose $\mathfrak{h}$ acts on a representation V of $\mathfrak{g}$. The eigenvalue $\alpha \in \mathfrak{h}^*$ of such an action is called a **weight of the representation** V. Define a subspace V_λ of V by

$$V_\lambda := \{v \in V : \forall h \in \mathfrak{h}, xv = \lambda(h)v\} \tag{2.3}$$

where hv denotes the action of $h \in \mathfrak{h}$ on $v \in V$. This subspace V_λ is called the **weight space of** V **with weight** λ. When we mention a weight of a representation, we will understand the corresponding weight space to be nonzero. The elements of V_λ not equal to zero are called **weight vectors**. Notice that by definition $\mathfrak{h} = \mathfrak{g}_0$.

In the adjoint representation the language is further specialized. If both $\lambda \in \mathfrak{h}^*$ and $\mathfrak{g}_\lambda$ are nonzero, we call λ a **root** and $\mathfrak{g}_\lambda$ its corresponding **root space**. Furthermore, it can be shown that for complex simple Lie algebras, all root spaces $\mathfrak{g}_\lambda$ are one-dimensional.

It is now classical that V, as a representation of a Cartan subalgebra $\mathfrak{h}$, can be written as a direct sum of its weight spaces. If we let Φ be the set of all roots, then we can decompose the Lie algebra $\mathfrak{g}$, expressing it in a form called the **root space decomposition** of $\mathfrak{g}$:

$$\mathfrak{g} = \mathfrak{h} \oplus \bigoplus_{\alpha \in \Phi} \mathfrak{g}_\alpha. \tag{2.4}$$

Let's examine some of the properties of the Lie bracket and the Killing form associated with root spaces.

If $x_\alpha \in \mathfrak{g}_\alpha$ and $x_\beta \in \mathfrak{g}_\beta$ with $\alpha \neq \beta$, and $h \in \mathfrak{h}$, then we have by (2.1):

$$\begin{aligned}
[h, [x_\alpha, x_\beta]] &= [[h, x_\alpha], x_\beta] + [x_\alpha, [h, x_\beta]] \\
&= \alpha(h)[x_\alpha, x_\beta] + \beta(h)[x_\alpha, x_\beta] \\
&= (\alpha + \beta)(h)[x_\alpha, x_\beta].
\end{aligned}$$

This implies that $[\mathfrak{g}_\alpha, \mathfrak{g}_\beta] \subseteq \mathfrak{g}_{\alpha+\beta}$, which also implies that $\mathfrak{g}_{\alpha+\beta} \neq 0$ when $\alpha + \beta$ is a root of $\mathfrak{g}$.

If we suppose that $\alpha + \beta$ is a root. Then for some $h \in \mathfrak{h}$ we have $(\alpha+\beta)(h) \neq 0$. Therefore by the properties of the Killing form we have

$$\begin{aligned}\alpha(h)B(x_\alpha, x_\beta) &= B([h, x_\alpha,], x_\beta)\\ &= -B(x_\alpha, [h, x_\beta])\\ &= -\beta(h)B(x_\alpha, x_\beta)\end{aligned}$$

This implies that

$$\begin{aligned}\alpha(h)B(x_\alpha, x_\beta) + \beta(h)B(x_\alpha, x_\beta) &= (\alpha+\beta)(h)B(x_\alpha, x_\beta) = 0\\ \implies B(x_\alpha, x_\beta) &= 0.\end{aligned}$$

This property will be important and should be kept in mind.

2.1.5 An Inner Product on a Cartan Subalgebra

If we suppose that the restriction of the Killing form to a fixed Cartan subalgebra $\mathfrak{h} \subset \mathfrak{g}$ is degenerate, then for some $h_0 \neq 0 \in \mathfrak{h}$ we have $B(h_0, h) = 0$ for all $h \in \mathfrak{h}$. By the root space decomposition we can write every element $x \in \mathfrak{g}$ as $x = h \oplus \bigoplus_{\alpha\in\Phi} x_\alpha$, where $x_\alpha \in \mathfrak{g}_\alpha$ and $h \in \mathfrak{f}$. Then by (2.60), we must conclude that $B(h_0, x) = 0$ for every $x \in \mathfrak{g}$ contradicting nondegeneracy of the Killing form on a simple Lie algebra. This allows us to conclude that the restriction of the Killing form to a Cartan subalgebra is nondegenerate.

By bilinearity of B the map $B_x : \mathfrak{g} \to \mathbb{C}$ defined by $B_x(y) = B(x, y)$ is linear. Therefore by definition of $\mathfrak{h}^*$ and the nondegeneracy of B, for every $h \in \mathfrak{h}$, there exists a unique $\alpha_h \in \mathfrak{h}^*$ such that $\alpha_h(x) = B(h, x)$. This allows the us to create a well-defined map $\xi : \mathfrak{h}^* \to \mathfrak{h}$ given by $\alpha \mapsto h_\alpha$ where $\alpha(x) = B(h_\alpha, x)$. Recall that the elements of a Cartan subalgebra $\mathfrak{h}$ of a linear Lie algebra $\mathfrak{g} \subset \mathfrak{gl}_n$, over an algebraically closed field (e.g. $\mathbb{C}$) are simultaneously diagonalizable. If we pick a basis $\{h_1, \ldots, h_n\}$ for $\mathfrak{h}$ of diagonal elements we have the following:

$$B(h_i, h_j) = \sum_{\alpha\in\Phi} \alpha(h_i)\alpha(h_j) = \sum_{\alpha\in\Phi} B(h_\alpha, h_i)B(h_\alpha, h_j).$$

We can define a real subspace $\mathfrak{h}_\mathbb{R}$ of $\mathfrak{h}$ by considering only the span of those elements in

$\mathfrak{h}$ on which the roots (which we recall span $\mathfrak{h}^*$) are real-valued. Then on this subspace we have the following:

$$B(h_i, h_i) = \sum_{\alpha \in \Phi} B^2(h_\alpha, h_i) \geq 0.$$

That is, the Killing form restricted to $\mathfrak{h}_{\mathbb{R}}$ is positive-definite, and consequently an inner product on this space [7]. This inner product induces an inner product on the real vector space spanned by the roots in $\mathfrak{h}^*$, which we denote by $\mathfrak{h}^*_{\mathbb{R}}$, given by:

$$\begin{aligned} \langle -, - \rangle : \mathfrak{h}^*_{\mathbb{R}} \times \mathfrak{h}^*_{\mathbb{R}} \to \mathbb{F} \\ \langle \alpha, \beta \rangle = B(h_\alpha, h_\beta), \alpha, \beta \in \mathfrak{h}^*_{\mathbb{R}}. \end{aligned} \tag{2.5}$$

If the dimension of $\mathfrak{h}$ is n, we can construct an isometry between $\mathfrak{h}^*_{\mathbb{R}}$ and $\mathbb{R}^n$ by picking a basis of $\mathfrak{h}^*_{\mathbb{R}}$ which is orthonormal with respect to the induced inner product $\langle -, - \rangle$ and mapping it onto the standard basis of $\mathbb{R}^n$. This identification of spaces allows us to think of linear maps on $\mathfrak{h}_{\mathbb{R}}$ as vectors in $\mathbb{R}^n$. From this point on we can think of roots of a Lie algebra as vectors in Euclidean space, and in place of the induced inner product on $\mathfrak{h}^*_{\mathbb{R}}$ we can use the standard inner product on $\mathbb{R}^n$, significantly simplifying calculations. In fact, this is the approach we will take as it has some very useful applications. Hereafter we will use the notation $(-, -)$ in place of $\langle -, - \rangle$ for the inner product of two roots of a Lie algebra [7].

2.1.6 Root Systems

Suppose Φ is a set of roots for a complex simple Lie algebra $\mathfrak{g}$ of rank n, and that the following properties hold: Φ spans all of $\mathbb{R}^n$, if $\alpha \in \Phi$ then the only other scalar multiple of α in Φ is $-\alpha$, for any $\alpha, \beta \in \Phi$ the element $\beta - 2\frac{(\alpha,\beta)}{(\alpha,\alpha)}\alpha \in \Phi$, and the number $\langle \beta, \alpha \rangle = 2\frac{(\alpha,\beta)}{(\alpha,\alpha)} \in \mathbb{Z}$. Then we call Φ the **root system** of the Lie algebra $\mathfrak{g}$.

Let Φ^+ be a subset of Φ such that for each $\alpha \in \Phi$, exactly one of the roots $\alpha, -\alpha \in \Phi^+$, and for any two distinct $\alpha, \beta \in \Phi^+$ where $\alpha + \beta \in \Phi$, then $\alpha + \beta \in \Phi^+$. We call Φ^+ a set of **positive roots**. Once such a subset is chosen, the elements of the set $-\Phi^+ = \Phi^-$ are called **negative roots**. A **simple root** is an element of Φ^+ which cannot be written as a sum of two elements of Φ^+. Simple roots $\{\alpha_i\}_{i=1}^n$ of a root system for a Lie algebra of rank n span $\mathbb{R}^n$ and every other root in Φ can be written as a linear combination of these simple α_i, with either all non-negative or all non-positive coefficients. We define the **height of a root** is equal to the sum of the coefficients of this linear combination. It is common to let Δ^+ denote the set of simple roots of a root system, a convention we will adopt from now on. We say that a root system Φ is **irreducible** if it cannot be partitioned into disjoint subsets $\Phi = \Phi_1 \cup \Phi_2$ such that for each $\phi_1 \in \Phi_1$ and $\phi_2 \in \Phi_2$, $(\phi_1, \phi_2) = 0$. Every simple Lie algebra $\mathfrak{g}$ has an irreducible root system Φ [4,7].

Given two roots $\alpha, \beta \in \Phi$, we can put fairly strict limitation on the possible angles between roots by considering the product $\langle \beta, \alpha \rangle \langle \alpha, \beta \rangle$:

$$\begin{aligned}
\langle \beta, \alpha \rangle \langle \alpha, \beta \rangle &= 2\frac{(\alpha, \beta)}{(\alpha, \alpha)} 2\frac{(\beta, \alpha)}{(\beta, \beta)} \\
&= 4\frac{(\alpha, \beta)^2}{|\alpha|^2 |\beta|^2} \\
&= 4\cos^2(\theta) \\
&= (2\cos(\theta))^2.
\end{aligned}$$

For Φ to be a root system we required that this result be integer-valued. Note that the range of $2\cos(\theta)$ is $[-2, 2]$. Therefore, the only possible values for $\cos(\theta)$ are $0, \pm\frac{1}{2}, \pm\frac{\sqrt{2}}{2}, \pm\frac{\sqrt{3}}{2}, \pm\frac{\sqrt{4}}{2} = \pm 1$. Furthermore we required that the only integer multiples of a root α which can be in Φ are α and $-\alpha$. This implies that we can restrict these even further by throwing out ± 1 as they correspond to $\pm\alpha$. The remaining values correspond to the angles $\frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{3\pi}{4}, \frac{5\pi}{6}$.

Now let $\mathfrak{g}$ be a complex simple Lie algebra with irreducible root system Φ, and let Δ^+ be a set of simple roots. We will construct a graph based on the angles between these simple roots. For each simple root α_i we draw a vertex, and we draw a number of edges between vertices corresponding to nonorthogonal roots based on the angle between them. Let θ be the angle between simple roots α_i and α_j. If $\theta = \frac{2\pi}{3}$ then we draw one undirected edge, if $\theta = \frac{3\pi}{4}$ then we draw two edges together with an angle sign pointing to the vertex of the shorter root, and if $\theta = \frac{5\pi}{6}$ then we draw three edges together with an angle sign, again pointing to the vertex of the shorter root. Such a graph is called a **Dynkin diagram**.

What exactly does this give us aside from further abstraction? The utility comes from the fact that all of the important structural information of a simple (or semisimple) Lie algebra can be reconstructed from the corresponding Dynkin diagram(s). Even more astounding is the fact that isomorphic Lie algebras have the diagrams which are equivalent under an isometry (as with B_2 and C_2), and that every permissible diagram has a corresponding Lie algebra. So a classification of complex simple Lie algebras reduces to a classification of permissible Dynkin diagrams. Note that Dynkin diagrams for semisimple Lie algebras will just have multiple components, each one corresponding to a simple direct summand of the Lie algebra. Thus all semisimple Lie algebras are classified by Dynkin diagrams. In Figure 2.1, we provide all permissible Dynkin diagrams associated with the complex simple Lie algebras.

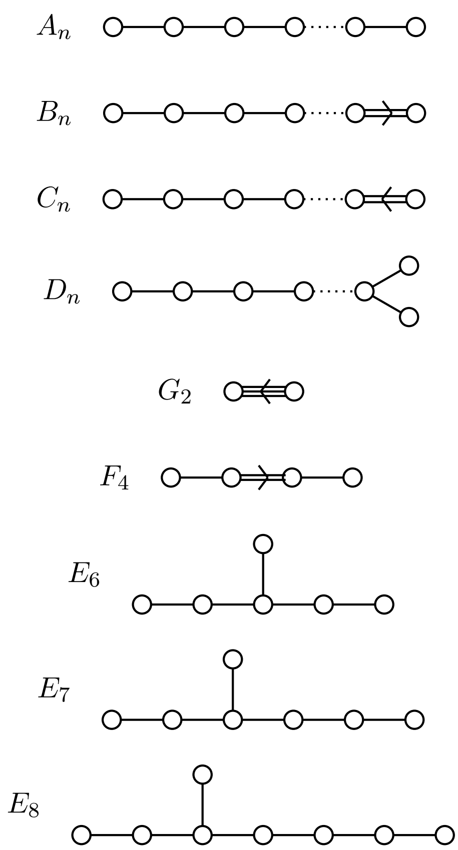

Figure 2.1: Dynkin Diagrams for Complex Simple Lie Algebras

The Lie algebras of interest for our study are those of types A_n, B_n, C_n, and D_n, which are commonly called the **classical Lie algebras**. For our purposes, each of these algebras will be explored concretely by viewing them as matrix algebras, which allows us to characterize Lie algebraic properties in a linear algebraic fashion, providing us with both familiar and powerful tools. As stated previously, we are studying the split real forms of these Lie algebras. Thus we view Lie algebras of type A_n as $\mathfrak{sl}_{n+1}(\mathbb{R})$, B_n as $\mathfrak{so}_{n,n+1}(\mathbb{R})$, C_n as $\mathfrak{sp}_{2n}(\mathbb{R})$, and D_n as $\mathfrak{so}_{2n}(\mathbb{R})$.

In addition to a classification of complex semisimple Lie algebras by Dynkin diagrams, a method of classifying certain extensions of these algebras (called Affine Lie algebras) was developed, building upon the idea of Dynkin diagrams. The diagrams associated with these superstructures are called extended Dynkin diagrams.

While an introduction to the actual structures represented by these diagrams is unnecessary for our study, the diagrams themselves provide useful tools for studying the Toda flow.

The construction of an extended Dynkin diagram begins with a Dynkin diagram for a complex simple Lie algebra. A single vertex is added which corresponds to the negative of the highest root of the original Lie algebra. Edges are then added in the same way as for regular Dynkin diagrams. For reference, all extended Dynkin diagrams for complex simple Lie algebras are given in Figure 2.2. The extra nodes are shaded.

In a treatment as cursory as this, the advantages of this transition from the complex and abstract (the Lie algebra $\mathfrak{g}$) to the concrete and simple (the corresponding Dynkin diagram(s)) cannot be adequately appreciated without working with them directly. That being said, providing additional depth in our introduction would not be beneficial in our current study.

Before leaving the realm of Lie theory for a time, we need to briefly mention an important type of Lie subalgebra, one which we avoided defining earlier as we lacked some useful notational conventions from the study of roots and root systems.

2.1.7 The Borel Subalgebra

Our last object of study in our crash course in Lie theory is the Borel subalgebra. While it may seem like an odd place to include this topic (as we defined subalgebras near the beginning of the chapter), the importance of this structure must be impressed upon the mind of the reader as it will play a central role.

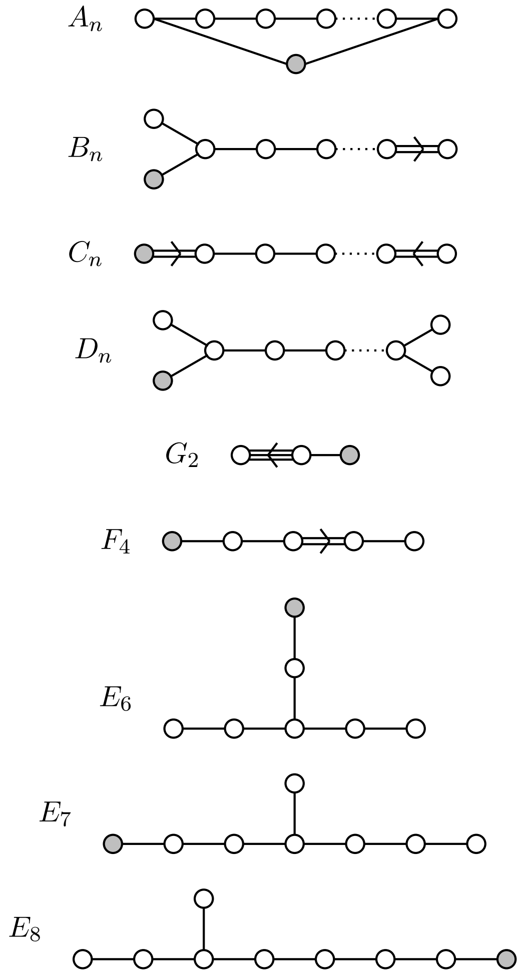

FIGURE 2.2: The Extended Dynkin Diagrams

Let $\mathfrak{g}$ be complex simple Lie algebra with Cartan subalgebra $\mathfrak{h}$ and root system Φ. The Borel subalgebra $\mathfrak{b}_+$ of $\mathfrak{g}$ is defined as follows:

$$\mathfrak{b}_+ = \mathfrak{h} \oplus \bigoplus_{\alpha \in \Phi^+} \mathfrak{g}_\alpha, \tag{2.6}$$

where Φ^+ denotes a choice of positive roots. In our matrix algebras the Borel subalgebra consists of the upper triangular elements, with the Cartan subalgebra being the diagonal elements.

The importance of this structure in the study of the Toda flow is enormous, but as of yet we lack the rest of the foundation necessary for explaining why this is so. For the time being, we need to set it aside and move elsewhere.

We have given an extremely short (and dense) introduction to Lie theory, the details of which can be found in any introductory text on the topic [4, 7, 14]. The underlying framework has been defined, but only in the Lie theoretic sense. The Toda lattice, from which the Toda flow was generalized, requires some material from the realm of Hamiltonian systems.

2.2 Hamiltonian Systems and Poisson Structures

The foundations of the Toda flow hail from the realm of Hamiltonian mechanics. Thus, even though our work will ultimately share few (superficial) similarities with mechanical systems, to truly understand the Toda flow, we need to understand the Toda lattice which in turn requires a basic understanding of Hamiltonian mechanics. The topics that follow may seem disjoint at first glance, but each will be integrated one with another as new tools and topics are introduced. The ultimate synthesis of these topics is beautiful mathematics.

While references to (perhaps) more sophisticated topics may appear from time to time, their purpose is entirely for theoretical considerations and consequently are unnecessary for understanding the Toda flow. As a final note, all manifolds are assumed smooth.

2.2.1 Hamiltonian Systems

Recall from mechanics that the state $\boldsymbol{x}$ of dynamical system can be represented as vector in phase space $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{p})$, where $\boldsymbol{q}$ and $\boldsymbol{p}$ are vectors of the same dimension, denoting position and momentum, respectively.

A **Hamiltonian system** is a dynamical system which is completely described by a function $H(\boldsymbol{x}, t) = H(\boldsymbol{q}, \boldsymbol{p}, t)$, called a **Hamiltonian**. The evolution of this system is governed by **Hamilton's equations**:

$$\begin{aligned} \frac{d\boldsymbol{p}}{dt} &= -\frac{\partial H}{\partial \boldsymbol{q}} \\ \frac{d\boldsymbol{q}}{dt} &= \frac{\partial H}{\partial \boldsymbol{p}}. \end{aligned} \tag{2.7}$$

To understand the benefit of describing a system in this way, we must better understand some of the underlying geometry of the phase space for this system.

We will assume that the configuration space of a dynamical system is a smooth manifold, so we may refer to the **configuration manifold** of a Hamiltonian system. If we let M be the configuration manifold of a Hamiltonian system, then the space of possible velocities at a point $\boldsymbol{q} \in M$ is naturally the tangent space of M at $\boldsymbol{q}$, $T_{\boldsymbol{q}}M$. With this perspective, momentum vectors can be understood as linear maps on the tangent space $T_{\boldsymbol{q}}M$, that is, elements of the cotangent space $T^*_{\boldsymbol{q}}M$. Therefore the space of all positions and momenta, our **phase space**, can be viewed as the cotangent bundle of M, T^*M.

We have just introduced a fairly general definition of phase space, but we need to extend our definition much further. This is possible because the underlying structure of phase space for any Hamiltonian system is far more rich than a cursory glance might suggest. As we move toward a more general phase space we will require some elementary symplectic geometry.

A **symplectic manifold** is a differentiable manifold M together with a differential 2-form ω, which is both closed and nondegenerate. That is $d\omega = 0$, where d is the exterior derivative, and for every $p \in M$, if there exists $X \in T_pM$ such that $\omega(X, Y) = 0$ for every $Y \in T_pM$, then $X = 0$. Generally we write a symplectic manifold as a pair (M, ω), and say that M has a **symplectic structure**.

Any smooth (real-valued) function H on a symplectic manifold induces a unique vector field X_H on M called the **Hamiltonian vector field** with Hamiltonian H. This is guaranteed by requiring that $dH = \omega(X_H, -)$ Note that uniqueness follows directly from the nondegeneracy of ω.

An important example of a symplectic manifold is $(\mathbb{R}^{2n}, \omega)$, where, given coordinates $(q_1, ..., q_n, p_1, ..., p_n)$, ω is defined by $\omega(q_i, p_j) = -\omega(p_j, q_i) = \delta_{ij}$ and $\omega(q_i, q_j) = \omega(p_i, p_j) = 0$. Therefore, in coordinates we can write $\omega = \sum_{j=1}^{n}(dq_j \wedge dp_j)$, and the matrix Ω for ω is given by

$$\Omega = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}$$

What needs to be verified in order to conclude that we do indeed have a symplectic manifold is to show that ω is both nondegenerate and closed. Recall that a bilinear form is nondegenerate if and only if its corresponding matrix is invertible. Therefore it suffices to show that Ω is invertible and that $d\omega = 0$. Consider the product

$$\begin{aligned}
\Omega\Omega^T &= \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix} \begin{pmatrix} 0 & -I_n \\ I_n & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 + I_n^2 & 0 \\ 0 & (-I_n)^2 + 0 \end{pmatrix} \\
&= \begin{pmatrix} I_n & 0 \\ 0 & I_n \end{pmatrix} \\
&= I_{2n}
\end{aligned}$$

Therefore Ω is invertible with $\Omega^{-1} = \Omega^T$.

Since Ω is invertible we can conclude that ω is nondegenerate. To show that ω is closed we just note that as ω has constant coefficients, $d\omega = 0$ is trivial, proving closure. Alternatively, we could introduce what is commonly referred to as the **tautological 1-form** $\theta = \sum_{j=1}^{n} p_j dq_j$ in the canonical coordinates $(q_1, ..., q_n, p_1, ..., p_n)$. Recall that $d^2 = 0$ (where d is the exterior derivative). As d is an antiderivation of degree one, we have:

$$\begin{aligned}
\theta &= \sum_{j=1}^{n} p_j dq_j \\
d\theta &= d\sum_{j=1}^{n} p_j dq_j = \sum_{j=1}^{n} d(p_j dq_j) = \sum_{j=1}^{n} (dp_j \wedge dq_j + (-1)^0 p_j \wedge d(dq_j)) \\
&= \sum_{j=1}^{n} (dp_j \wedge dq_j + p_j \wedge 0) = \sum_{j=1}^{n} (dp_j \wedge dq_j + 0) = \sum_{j=1}^{n} (dp_j \wedge dq_j) \\
&= -\sum_{j=1}^{n} (dq_j \wedge dp_j) = -\omega.
\end{aligned}$$

Therefore, since $\omega = -d\theta$ we can conclude, yet again, that $d\omega = d(-d\theta) = 0$ and that ω is closed. This shows that $(\mathbb{R}^{2n}, \omega)$ is indeed a symplectic manifold. We can see directly from Ω that if H is a Hamiltonian, then the Hamiltonian vector field with Hamiltonian H is given in local coordinates by $X_H = (\frac{\partial H}{\partial \boldsymbol{p}}, -\frac{\partial H}{\partial \boldsymbol{q}})$.

It turns out that every Hamiltonian system has as natural symplectic structure. Given a Hamiltonian H and, in canonical coordinates, a state $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{p})$ in some phase space T^*M, where the length of $\boldsymbol{q}$ and $\boldsymbol{p}$ are both n, we will write:

$$\nabla_{\boldsymbol{x}} H = \begin{pmatrix} \frac{\partial H}{\partial \boldsymbol{q}} \\ \frac{\partial H}{\partial \boldsymbol{p}} \end{pmatrix}.$$

Then, by Hamilton's equations we have

$$\begin{aligned} \frac{d\boldsymbol{x}}{dt} &= \begin{pmatrix} \frac{\partial H}{\partial \boldsymbol{p}} \\ -\frac{\partial H}{\partial \boldsymbol{q}} \end{pmatrix} \\ &= \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial H}{\partial \boldsymbol{q}} \\ \frac{\partial H}{\partial \boldsymbol{p}} \end{pmatrix} \\ &= \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix} \nabla_{\boldsymbol{x}} H \\ &= \Omega \nabla_{\boldsymbol{x}} H. \end{aligned}$$

Thus we see that the symplectic form ω is intrinsic to any Hamiltonian system. In our most recent definition of a Hamiltonian system we generalized the idea of phase space to the cotangent space T^*M of a smooth manifold M. Our retrieval of ω from an arbitrary phase space T^*M shows that (T^*M, ω) a symplectic manifold. This suggests that a further generalization of a Hamiltonian system and phase space may be possible. Before attempting to find such a generalization, the possible advantages of a symplectic structure in the study of Hamiltonian systems need to be mentioned.

The interest lies in an object we have yet to explain beyond its definition: the Hamiltonian vector field. It turns out that the integral curves of the Hamiltonian vector field with Hamiltonian H are exactly solutions to Hamilton's equations, each with a different initial condition. Therefore, instead of thinking of Hamiltonian systems as necessarily describing a mechanical system we can treat them as problems in geometry. This allows for a far more general definition of a Hamiltonian and Hamiltonian system once we recall an important result from geometry: Darboux's theorem, which implies that (locally) every symplectic manifold looks like exactly $(\mathbb{R}^{2n}, \omega)$, our introductory example. The reason this is such a valuable property is that, locally, we can always pick coordinates so that ω has the matrix Ω.

Now we can define a **Hamiltonian system** as a triple (M, ω, H), where M is a smooth manifold with a symplectic structure imbued by ω, and where $H(\boldsymbol{q}, \boldsymbol{p})$ is a smooth real-valued function on (M, ω) which we call a Hamiltonian. The Hamiltonian H induces a Hamiltonian vector field, whose integral curves $\boldsymbol{x}(t)$ are solutions to the initial value problem

$$\begin{aligned} \dot{q}_i &= \frac{\partial H}{\partial p_i} \\ \dot{p}_i &= -\frac{\partial H}{\partial q_i} \\ \mathbf{x}_0 &= \mathbf{x}(t_0) \in M. \end{aligned} \tag{2.8}$$

It may seem unlikely that a further generalization of the concept of a Hamiltonian system is possible, but we can abstract the problem one step further.

2.2.2 Poisson Manifolds

Let M be a smooth manifold, and let $C^\infty(M)$ be the space of smooth functions on M. Let $\{-,-\} : C^\infty(M) \times C^\infty(M) \to C^\infty(M)$ be a bilinear map such that for each $f, g, h \in C^\infty(M)$ we have

$$\begin{aligned} \{f, g\} &= -\{g, f\} \\ \{f, \{g, h\}\} + \{g, \{h, f\}\} + \{h, \{f, g\}\} &= 0 \\ \{fg, h\} &= f\{g, h\} + g\{f, h\} \end{aligned} \tag{2.9}$$

This map is called a **Poisson bracket** on M and M equipped with such a map is said to have a **Poisson structure**. A smooth manifold with a Poisson structure is called a **Poisson manifold**. In working with Poisson manifolds it can be highly beneficial to study the geometry of one Poisson manifold in relation to another. In fact, the Toda lattice was proved to be completely integrable using this very concept. To make such a study possible, one must produce a map between the manifolds which preserves the Poisson structure.

More formally, given two Poisson manifolds, $(M_1, \{-,-\}_{M_1})$ and $(M_2, \{-,-\}_{M_2})$, we can consider mappings between them. Let $\phi : M_1 \to M_2$ be a smooth map such that for each $x \in M$ and $f, g \in C^\infty(M_2)$ we have that

$$\{f, g\}_{M_2}(\phi(x)) = \{f \circ \phi, g \circ \phi\}_{M_1}(x)$$

Such a ϕ is called a **Poisson map**. This is a very useful kind of map as it allows us to use a familiar manifold to study a foreign one. Now let's return to Poisson structures.

To better understand the idea of a Poisson manifold, let's look at an example. Let $M = \mathbb{R}^{2n}$, using the canonical coordinates from Hamiltonian systems $(q_1, \ldots, q_n, p_1, \ldots, p_n)$, and define $\{-,-\} : C^\infty(\mathbb{R}^{2n}) \times C^\infty(\mathbb{R}^{2n}) \to C^\infty(\mathbb{R}^{2n})$ by

$$\{f, g\} = \sum_{j=1}^{n} \Big(\frac{\partial f}{\partial q_j} \frac{\partial g}{\partial p_j} - \frac{\partial f}{\partial p_j} \frac{\partial g}{\partial q_j} \Big). \tag{2.10}$$

With this definition, we claim the pair $(\mathbb{R}^{2n}, \{-,-\})$, as defined, is a Poisson manifold. This is fairly easy (albeit tedious) to check. Let $f, g, h \in C^\infty(\mathbb{R}^{2n})$. First we will verify linearity in the first argument and skew-symmetry. Note that these properties together imply linearity in the second argument:

$$\begin{aligned}
\{f + g, h\} &= \sum_{j=1}^{n} \Big(\frac{\partial (f+g)}{\partial q_j} \frac{\partial h}{\partial p_j} - \frac{\partial (f+g)}{\partial p_j} \frac{\partial h}{\partial q_j} \Big) \\
&= \sum_{j=1}^{n} \Big(\Big(\frac{\partial f}{\partial q_j} + \frac{\partial g}{\partial q_j} \Big) \frac{\partial h}{\partial p_j} - \Big(\frac{\partial f}{\partial p_j} + \frac{\partial g}{\partial p_j} \Big) \frac{\partial h}{\partial q_j} \Big) \\
&= \sum_{j=1}^{n} \Big(\frac{\partial f}{\partial q_j} \frac{\partial h}{\partial p_j} + \frac{\partial g}{\partial q_j} \frac{\partial h}{\partial p_j} - \frac{\partial f}{\partial p_j} \frac{\partial h}{\partial q_j} - \frac{\partial g}{\partial p_j} \frac{\partial h}{\partial q_j} \Big) \\
&= \sum_{j=1}^{n} \Big(\frac{\partial f}{\partial q_j} \frac{\partial h}{\partial p_j} - \frac{\partial f}{\partial p_j} \frac{\partial h}{\partial q_j} \Big) + \Big(\frac{\partial g}{\partial q_j} \frac{\partial h}{\partial p_j} - \frac{\partial g}{\partial p_j} \frac{\partial h}{\partial q_j} \Big) \\
&= \sum_{j=1}^{n} \Big(\frac{\partial f}{\partial q_j} \frac{\partial h}{\partial p_j} - \frac{\partial f}{\partial p_j} \frac{\partial h}{\partial q_j} \Big) + \sum_{j=1}^{n} \Big(\frac{\partial g}{\partial q_j} \frac{\partial h}{\partial p_j} - \frac{\partial g}{\partial p_j} \frac{\partial h}{\partial q_j} \Big) \\
&= \{f, h\} + \{g, h\}.
\end{aligned}$$

$$\begin{aligned}\{f,g\} &= \sum_{j=1}^{n}\Big(\frac{\partial f}{\partial q_j}\frac{\partial g}{\partial p_j} - \frac{\partial f}{\partial p_j}\frac{\partial g}{\partial q_j}\Big)\\ &= \sum_{j=1}^{n}-\Big(\frac{\partial f}{\partial p_j}\frac{\partial g}{\partial q_j} - \frac{\partial f}{\partial q_j}\frac{\partial g}{\partial p_j}\Big)\\ &= \sum_{j=1}^{n}-\Big(\frac{\partial g}{\partial q_j}\frac{\partial f}{\partial p_j} - \frac{\partial g}{\partial p_j}\frac{\partial f}{\partial q_j}\Big)\\ &= -\{g,f\}.\end{aligned}$$

The next property we will verify is (2.15), which is commonly referred to as the **Leibniz rule**. We verify this now.

$$\begin{aligned}\{fg,h\} &= \sum_{l=1}^{n}\Big(\frac{\partial(fg)}{\partial q_l}\frac{\partial h}{\partial p_l} - \frac{\partial(fg)}{\partial p_l}\frac{\partial h}{\partial q_l}\Big)\\ &= \sum_{l=1}^{n}\Big(\Big(\frac{\partial f}{\partial q_l}g + f\frac{\partial g}{\partial q_l}\Big)\frac{\partial h}{\partial p_l} - \Big(\frac{\partial f}{\partial p_l}g + f\frac{\partial g}{\partial p_l}\Big)\frac{\partial h}{\partial q_l}\Big)\\ &= \sum_{l=1}^{n}\Big(\frac{\partial f}{\partial q_l}g\frac{\partial h}{\partial p_l} + f\frac{\partial g}{\partial q_l}\frac{\partial h}{\partial p_l} - \frac{\partial f}{\partial p_l}g\frac{\partial h}{\partial q_l} - f\frac{\partial g}{\partial p_l}\frac{\partial h}{\partial q_l}\Big)\\ &= f\sum_{l=1}^{n}\Big(\frac{\partial g}{\partial q_l}\frac{\partial h}{\partial p_l} - \frac{\partial g}{\partial p_l}\frac{\partial h}{\partial q_l}\Big) + g\sum_{l=1}^{n}\Big(\frac{\partial f}{\partial q_l}\frac{\partial h}{\partial p_l} - \frac{\partial f}{\partial p_l}\frac{\partial h}{\partial q_l}\Big)\\ &= f\{g,h\} + g\{f,h\}.\end{aligned}$$

Lastly we will verify (2.14), which is commonly referred to as the **Jacobi identity**. In order to simplify later calculations we will first show that for every $x_i \in \{q_1,..,q_n,p_1,...p_n\}$ we have $\frac{\partial}{\partial x_i}\{f,g\} = \{\frac{\partial f}{\partial x_i}, g\} + \{f, \frac{\partial g}{\partial x_i}\}$:

$$\begin{aligned}\frac{\partial}{\partial x_i}\{f,g\} &= \frac{\partial}{\partial x_i}\sum_{j=1}^{n}\Big(\frac{\partial f}{\partial q_j}\frac{\partial g}{\partial p_j} - \frac{\partial f}{\partial p_j}\frac{\partial g}{\partial q_j}\Big)\}\\ &= \sum_{j=1}^{n}\frac{\partial}{\partial x_i}\Big(\frac{\partial f}{\partial q_j}\frac{\partial g}{\partial p_j} - \frac{\partial f}{\partial p_j}\frac{\partial g}{\partial q_j}\Big)\\ &= \sum_{j=1}^{n}\Big(\frac{\partial}{\partial x_i}\Big(\frac{\partial f}{\partial q_j}\frac{\partial g}{\partial p_j}\Big) - \frac{\partial}{\partial x_i}\Big(\frac{\partial f}{\partial p_j}\frac{\partial g}{\partial q_j}\Big)\Big)\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{n} \Big(\frac{\partial^2 f}{\partial x_i \partial q_j} \frac{\partial g}{\partial p_j} + \frac{\partial f}{\partial q_j} \frac{\partial^2 g}{\partial x_i \partial p_j} \\
&\quad - \frac{\partial^2 f}{\partial x_i \partial p_j} \frac{\partial g}{\partial q_j} - \frac{\partial f}{\partial p_j} \frac{\partial^2 g}{\partial x_i \partial q_j} \Big) \\
&= \sum_{j=1}^{n} \Big(\frac{\partial^2 f}{\partial q_j \partial x_i} \frac{\partial g}{\partial p_j} - \frac{\partial^2 f}{\partial p_j \partial x_i} \frac{\partial g}{\partial q_j} \Big) \\
&\quad + \sum_{j=1}^{n} \Big(\frac{\partial f}{\partial q_j} \frac{\partial^2 g}{\partial p_j \partial x_i} - \frac{\partial f}{\partial p_j} \frac{\partial^2 g}{\partial q_j \partial x_i} \Big) \\
&= \{\frac{\partial f}{\partial x_i}, g\} + \{f, \frac{\partial g}{\partial x_i}\}.
\end{aligned}
$$

Next we calculate $\{f, \{g, h\}\}$. By applying linearity and the Leibniz rule we can write:

$$
\begin{aligned}
&\{f, \{g, h\}\} \\
&\quad = \{f, \sum_{i=1}^{n} \Big(\frac{\partial g}{\partial q_i} \frac{\partial h}{\partial p_i} - \frac{\partial g}{\partial p_i} \frac{\partial h}{\partial q_i} \Big) \\
&\quad = \sum_{i=1}^{n} \Big(\{f, \frac{\partial g}{\partial q_i} \frac{\partial h}{\partial p_i}\} - \{f, \frac{\partial g}{\partial p_i} \frac{\partial h}{\partial q_i}\} \Big) \\
&\quad = \sum_{i=1}^{n} \Big(\frac{\partial g}{\partial q_i} \{f, \frac{\partial h}{\partial p_i}\} + \frac{\partial h}{\partial p_i} \{f, \frac{\partial g}{\partial q_i}\} - \frac{\partial g}{\partial p_i} \{f, \frac{\partial h}{\partial q_i}\} - \frac{\partial h}{\partial q_i} \{f, \frac{\partial g}{\partial p_i}\} \Big)
\end{aligned}
$$

Therefore $\{f, \{g, h\}\} + \{g, \{h, f\}\} + \{h, \{f, g\}\}$ can be written, using our new result from (2.34), as:

$$
\begin{aligned}
&\sum_{i=1}^{n} \Big(\frac{\partial g}{\partial q_i} \{f, \frac{\partial h}{\partial p_i}\} + \frac{\partial h}{\partial p_i} \{f, \frac{\partial g}{\partial q_i}\} - \frac{\partial g}{\partial p_i} \{f, \frac{\partial h}{\partial q_i}\} - \frac{\partial h}{\partial q_i} \{f, \frac{\partial g}{\partial p_i}\} \\
&\quad + \frac{\partial h}{\partial q_i} \{g, \frac{\partial f}{\partial p_i}\} + \frac{\partial f}{\partial p_i} \{g, \frac{\partial h}{\partial q_i}\} - \frac{\partial h}{\partial p_i} \{g, \frac{\partial f}{\partial q_i}\} - \frac{\partial f}{\partial q_i} \{g, \frac{\partial h}{\partial p_i}\} \\
&\quad + \frac{\partial f}{\partial q_i} \{h, \frac{\partial g}{\partial p_i}\} + \frac{\partial g}{\partial p_i} \{h, \frac{\partial f}{\partial q_i}\} - \frac{\partial f}{\partial p_i} \{h, \frac{\partial g}{\partial q_i}\} - \frac{\partial g}{\partial q_i} \{h, \frac{\partial f}{\partial p_i}\} \Big) \\
= &\sum_{i=1}^{n} \Big(\Big(\{\frac{\partial g}{\partial q_i}, h\} + \{g, \frac{\partial h}{\partial q_i}\} \Big) \frac{\partial f}{\partial p_i} - \Big(\{\frac{\partial g}{\partial p_i}, h\} + \{g, \frac{\partial h}{\partial p_i}\} \Big) \frac{\partial f}{\partial q_i} \\
&\quad + \Big(\{\frac{\partial h}{\partial q_i}, f\} + \{h, \frac{\partial f}{\partial q_i}\} \Big) \frac{\partial g}{\partial p_i} - \Big(\{\frac{\partial h}{\partial p_i}, f\} + \{h, \frac{\partial f}{\partial p_i}\} \Big) \frac{\partial g}{\partial q_i} \\
&\quad + \Big(\{\frac{\partial f}{\partial q_i}, g\} + \{f, \frac{\partial g}{\partial q_i}\} \Big) \frac{\partial h}{\partial p_i} - \Big(\{\frac{\partial f}{\partial p_i}, g\} + \{f, \frac{\partial g}{\partial p_i}\} \Big) \frac{\partial h}{\partial q_i} \Big)
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{i=1}^{n}\Big(\frac{\partial}{\partial q_i}\{g,h\}, \frac{\partial f}{\partial p_i}\} - \frac{\partial}{\partial p_i}\{g,h\}, \frac{\partial f}{\partial q_i}\}\Big) \\
&\quad + \sum_{i=1}^{n}\Big(\frac{\partial}{\partial q_i}\{h,f\}, \frac{\partial g}{\partial p_i}\} - \frac{\partial}{\partial p_i}\{h,f\}, \frac{\partial g}{\partial q_i}\}\Big) \\
&\quad + \sum_{i=1}^{n}\Big(\frac{\partial}{\partial q_i}\{f,g\}, \frac{\partial h}{\partial p_i}\} - \frac{\partial}{\partial p_i}\{f,g\}, \frac{\partial h}{\partial q_i}\}\Big) \\
&\quad = \{\{g,h\},f\} + \{\{h,f\},g\} + \{\{f,g\},h\}.
\end{aligned}
$$

That is,

$$
\{f,\{g,h\}\} + \{g,\{h,f\}\} + \{h,\{f,g\}\} = \{\{g,h\},f\} + \{\{h,f\},g\} + \{\{f,g\},h\}.
$$

Skew-symmetry implies that the right hand side is equal to

$$
-\{f,\{g,h\}\} - \{g,\{h,f\}\} - \{h,\{f,g\}\}
$$

Therefore we have

$$
\begin{aligned}
2\{f,\{g,h\}\} + 2\{g,\{h,f\}\} + 2\{h,\{f,g\}\} &= 0 \\
\{f,\{g,h\}\} + \{g,\{h,f\}\} + \{h,\{f,g\}\} &= 0.
\end{aligned}
$$

The bracket we defined does indeed obey the Jacobi identity.

Therefore we conclude that $(\mathbb{R}^{2n}, \{-,-\})$, as defined, is in fact a Poisson manifold. We have already seen that $(\mathbb{R}^{2n}, \omega)$ is a symplectic manifold. How does the Poisson structure of $(\mathbb{R}^{2n}, \{-,-\})$ relate to the symplectic structure of $(\mathbb{R}^{2n}, \omega)$?

We will briefly return to symplectic manifolds to obtain an important result. Let (M,ω) be a symplectic manifold and define a map $\{-,-\} : C^\infty(M)\times C^\infty(M) \to C^\infty(M)$ by $\{f,g\} = \omega(X_f, X_g)$, where $f,g \in C^\infty$ and X_f and X_g are the vector fields on M induced by f and g, respectively. The map $\{-,-\}$ gives (M,ω) a Poisson structure [12]. Since $\{-,-\}$ gives (M,ω) a Poisson structure, we can think of (M,ω) as a Poisson manifold [12]. Note, however, that while we have determined that every symplectic manifold is a Poisson manifold, not every Poisson manifold is a symplectic manifold. This is because the definition of a Poisson bracket does not require nondegeneracy. It is possible, however, to restrict ourselves to certain submanifolds as every Poisson manifold can be partitioned into symplectic "leaves" of even dimension (not all necessarily of the same dimension)[8, 10, 12].

The question that remains is whether the Poisson structure provides us with any advantages for studying Hamiltonian systems aside from more notation for describing the same systems. At first glance it might be seen as an unnecessary complication.

Let (M, ω, H) be a Hamiltonian system, and $\{-,-\}$ the Poisson structure on M induced by ω. Pick the local canonical coordinates $(q_1, ..., q_n, p_1, ..., p_n)$ so that ω takes the form $\omega = \sum_{j=1}^{n}(dq_j \wedge dp_j)$. Then the Poisson bracket $\{-,-\}$ takes the form

$$\{f, g\} = \sum_{j=1}^{n} \Big(\frac{\partial f}{\partial q_j}\frac{\partial g}{\partial p_j} - \frac{\partial f}{\partial p_j}\frac{\partial g}{\partial q_j}\Big),$$

and Hamilton's equations can be written as

$$\dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}} \end{pmatrix} = \begin{pmatrix} \frac{\partial H}{\partial \boldsymbol{p}} \\ -\frac{\partial H}{\partial \boldsymbol{q}} \end{pmatrix} = \begin{pmatrix} \{\boldsymbol{q}, H\} \\ \{\boldsymbol{p}, H\}. \end{pmatrix}$$

Here we understand $\{\boldsymbol{x}, H\}$ to be a column vector where the ith component is $\{x_i, H\}$.

Therefore, for any smooth function $f(\boldsymbol{q}, \boldsymbol{p}, t)$ we have by substitution

$$\begin{aligned} \frac{d}{dt}f(\boldsymbol{q}, \boldsymbol{p}, t) &= \sum_{i=1}^{n} \Big(\frac{\partial f}{\partial q_i}\dot{q}_i + \frac{\partial f}{\partial p_i}\dot{p}_i\Big) + \frac{\partial f}{\partial t} \\ &= \sum_{i=1}^{n} \Big(\frac{\partial f}{\partial q_i}\frac{\partial H}{\partial p_i} - \frac{\partial f}{\partial p_i}\frac{\partial H}{\partial q_i}\Big) + \frac{\partial f}{\partial t} \\ &= \{f, H\} + \frac{\partial f}{\partial t}. \end{aligned} \tag{2.11}$$

Note then that for Hamilton's equations we have n pairs of equations with the ith and $(i+n)$th equations given by

$$\begin{aligned} \dot{q}_i &= \frac{\partial H}{\partial p_i} = \{q_i, H\} \\ \dot{p}_i &= -\frac{\partial H}{\partial q_i} = -\{p_i, H\}. \end{aligned}$$

The real utility of the Poisson bracket is found in the search for constants of motion, or more particularly, integrals of motion for the Hamiltonian system. A degree of complexity is added, in that on the entire space the Poisson bracket may be degenerate undermining its utility for the search for integrals of motion, but on symplectic submanifolds on which the Poisson bracket is nondegenerate we have a workable problem.

2.3 Integrability of a Hamiltonian System

Recall that a **constant of motion** is a quantity which remains constant through the evolution of a dynamical system, and that an **integral of motion** (or **first integral**) is a constant of motion with no explicit time-dependency. In terms of the Poisson bracket we can say that if F is a constant of motion for a Hamiltonian system with Hamiltonian H, then its time derivative should be zero. Therefore by (2.11), we obtain what is called **Liouville's equation**:

$$0 = \frac{d}{dt}F = \{F, H\} + \frac{\partial F}{\partial t} \implies -\frac{\partial F}{\partial t} = \{F, H\}. \tag{2.12}$$

If F is an integral of motion the result is further simplified to

$$\{F, H\} = -\{H, F\} = 0.$$

That is, the Poisson bracket of an integral of motion F and the Hamiltonian function H vanishes. Note that this implies that a time-independent Hamiltonian itself is an integral of motion for the system. If F, G are both constants of motion for a Hamiltonian system and $\{F, G\} = 0$ then we say that F and G are in **involution**.

Let (M, ω, H) be a Hamiltonian system in n dimensions. We say that (M, ω, H) is **completely integrable** (in the Liouville sense) if there exist n integrals of motion in involution. If we are working with a Poisson manifold, we can consider complete integrability on its symplectic submanifolds. At this time we will not give an example of a completely integrable system, as the Toda lattice, which we will examine in detail in the next chapter, is one such system.

2.3.1 Hamiltonian Systems on Coadjoint Orbits

The most important Poisson manifold for us to understand (hence its need for its own section) is $\mathfrak{g}^*$, where $\mathfrak{g}$ is a simple Lie algebra. Perhaps surprisingly, we don't use the natural Poisson structure for studying Hamiltonian systems on this space. Let $f, g \in C^\infty(\mathfrak{g}^*)$ and $\alpha \in \mathfrak{g}^*$. The **Lie-Poisson bracket**, which was studied in depth in

[10], is defined by

$$\{f, g\}(\alpha) = \langle \alpha, [df_\alpha, dg_\alpha] \rangle.$$

where $\langle \alpha, x \rangle$ is $\alpha(x)$, and where df_α and dg_α are regarded as elements in $\mathfrak{g}$. This is fairly opaque, but the fact that we are looking specifically at a simple Lie algebra allows for some simplification. To simplify this map, we need to use the Killing form B (which we again remind the reader is nondegenerate for $\mathfrak{g}$ simple). For the sake of brevity, from this point on we will follow the more common convention and suppress notation, writing (x, y) in place of $B(x, y)$.

Let $\phi : \mathfrak{g} \to \mathfrak{g}^*$ be given by $x \mapsto \phi_x$, where $\phi_x(y) = (x, y)$. By bilinearity of the Killing form we can see both that ϕ_x is linear, making ϕ well-defined, and that ϕ itself is linear. Nondegeneracy of the Killing form makes ϕ injective, and finite dimensionality (implying equal dimension for $\mathfrak{g}$ and $\mathfrak{g}^*$) makes it surjective. This allows us to identify $\mathfrak{g}$ and $\mathfrak{g}^*$. Furthermore, we note that because $\mathfrak{g}$ can be (canonically) identified with $(\mathfrak{g}^*)^*$, allowing us to see $\mathfrak{g}$ as naturally sitting inside of $C^\infty(\mathfrak{g}^*)$. Thus, we obtain a Poisson structure on $\mathfrak{g}$ by

$$\{f, g\}(x) = (x, [\nabla f(x), \nabla g(x)]), \quad \text{with} \quad x \in \mathfrak{g}$$

where $f, g \in C^\infty(\mathfrak{g})$, and $\nabla f(x) \in \mathfrak{g}$ is defined by the property

$$(\nabla f(x), y) = \lim_{t \to 0} \frac{f(x + ty) - f(x)}{t} \quad x, y \in \mathfrak{g}.$$

How does this relate to Hamiltonian systems? We have a Poisson manifold, but how exactly is it partitioned into symplectic spaces? The answer lies in a particular group action. Let $\mathfrak{g}$ be a simple Lie algebra with Lie group G and dual $\mathfrak{g}^*$. Then G acts on $\mathfrak{g}$ by $g \cdot x = \mathrm{Ad}_g(x)$ where Ad_g is the map to which g is sent in the adjoint representation of G. Furthermore G acts on $\mathfrak{g}^*$ by $(g \cdot \alpha)(x) = \alpha(Ad_{g^{-1}}(x))$. This group action is called the **coadjoint action**. This particular group action is important because it is the coadjiont action of the Borel subgroup B of G which partitions $\mathfrak{b}^*$ into symplectic spaces, wherein the Lie-Poisson bracket is nondegenerate [10]. This allows us to consider integrability of Hamiltonian systems on coadjoint orbits in $\mathfrak{b}^*$. In fact, it is on coadjoint orbits that the Toda flow evolves.

Chapter 3
From the Toda Lattice to the Toda Flow

In the sections that follow, we will introduce the Toda flow by way of an examination of some of is predecessors, specifically the Toda lattice and the Kostant-Toda lattice.

3.1 The Toda Lattice

In 1967, Morikazu Toda published a paper in which he introduced a Hamiltonian system describing particles on a line with exponential interaction between closest neighbors. In his honor, this system was named the Toda lattice. Since its introduction into the literature, many variations of the system have been studied and generalized. To name a few, there is an infinite lattice, a finite periodic lattice (where the particles on the ends are considered neighbors), and a finite non-periodic lattice. The last of these is the lattice of interest to us. The system has been studied in depth (see [1, 5, 6, 10, 11, 12, 15] for a few noteworthy studies). Even still, an understanding of the original Toda lattice will help us as we move to the far more general Toda flow, justifying an examination of its Liouville integrability.

The Hamiltonian for the finite non-periodic Toda lattice with n particles can be written as

$$H(q,p) = \frac{1}{2}\sum_{j=1}^{n} p_j^2 + \sum_{j=1}^{n-1} e^{(q_j - q_{j+1})}. \tag{3.1}$$

Phase space in this case is the Poisson manifold $(\mathbb{R}^{2n}, \{-,-\})$, with the standard Poisson bracket from (2.93).

Recalling Hamilton's equations $\dot{q_j} = \frac{\partial H}{\partial p_j}$ and $\dot{p_j} = -\frac{\partial H}{\partial q_j}$, we can calculate the equations of motion:

$$\begin{aligned}
\dot{q_j} &= p_j, \quad j = 1, ..., n, \\
\dot{p_1} &= e^{(q_1 - q_2)}, \\
\dot{p_j} &= -e^{(q_j - q_{j+1})} + e^{(q_{j-1} - q_j)}, \quad j = 2, ..., n-1, \\
\dot{p_n} &= e^{(q_{n-1} - q_n)}
\end{aligned} \tag{3.2}$$

where the position (displacement from rest) and momentum of the jth particle in the system is given by q_j and p_j, respectively. Let's take a brief look at how the Toda lattice was shown to be completely integrable.

3.1.1 The Toda Lattice is Completely Integrable

Let $a_i = \frac{1}{2} e^{\frac{1}{2}(q_i - q_{i+1})}$, and let $b_i = -\frac{1}{2} p_i$. The new variables are referred to as **Flaschka's variables** for mathematical physicist Hermann Flashcka, who first used the variable substitution in his study of the Toda lattice [5, 6]. With this substitution our equations of motion take the form:

$$\begin{aligned}
\dot{a_j} &= a_j(b_{j+1} - b_j), \quad j = 0, \ldots, n-1, \\
\dot{b_j} &= 2(a_j^2 - a_{j-1}^2), \quad j = 1, \ldots, n
\end{aligned}$$

with the assumption that $a_0 = a_n = 0$. In Flaschka's variables, our system is equivalent to the equation:

$$\dot{L} = [M, L] \tag{3.3}$$

where L is symmetric and tridiagonal, M is skew-symmetric and tridiagonal with the same first superdiagonal as L, and $[M, L]$ is given by the matrix commutator $ML - LM$ [11]. This equation is called **Lax's equation** and was first introduced by Peter Lax[10, 11]. The importance of a **Lax form** for a Hamiltonian system (L and M satisfying (3.8) are called a **Lax pair**) is that the eigenvalues, or equivalently the trace powers $I_k = \frac{1}{k}\mathrm{tr}(L^k)$ of L, are integrals of motion for the original system [11]. We should point out that the symmetry of L and the skew-symmetry of M are only the case for this specific Lax form. We will see another form below.

Finding a Lax pair may seem like a difficult problem. It can be, in practice, extremely difficult. For the Toda lattice, Flaschka also gave us the **symmetric Lax form** for the Toda lattice (in Flaschka's variables):

$$L = \begin{pmatrix} b_1 & a_1 & & & 0 \\ a_1 & b_2 & a_2 & & \\ & a_2 & \ddots & \ddots & \\ & & \ddots & b_{n-1} & a_{n-1} \\ 0 & & & a_{n-1} & b_n \end{pmatrix}$$

$$M = \begin{pmatrix} 0 & a_1 & & & 0 \\ -a_1 & 0 & a_2 & & \\ & -a_2 & \ddots & \ddots & \\ & & \ddots & 0 & a_{n-1} \\ 0 & & & -a_{n-1} & 0 \end{pmatrix}.$$

In the center of momentum frame of reference we have $p_1 + \ldots + p_n = 0$. Therefore, by substitution, the sum of the diagonal elements of L is equal to zero as well. Perhaps the most important thing to note at this time is that in matrix form we can think of the Toda lattice as an evolution on tridiagonal matrices in the space of trace-free matrices, the Lie algebra $\mathfrak{sl}_n$. This gives us a hint about how the Toda lattice might be generalized.

But how can we be sure that important structural properties of the Toda lattice are preserved when we move from $\mathbb{R}^{2n}$ to the tridiagonal elements of $\mathfrak{sl}_n$? The answer is the existence of a Poisson map.

Let $\phi : (\mathbb{R}^{2n}, \{-,-\}) \to (\mathfrak{sl}_n, (-, [-,-]))$, where $\{-,-\}$ is the standard Poisson structure on $\mathbb{R}^{2n}$ and $(-, [-,-])$ is the Lie-Poisson bracket. If we let E_{ij} be the matrix with 1 in the ij position then we can define the map by

$$q_i \mapsto e^{(q_i - q_{i+1})}(E_{i+1,i} + E_{i,i+1}), i = 1, \ldots, n-1$$
$$p_i \mapsto p_i E_{ii}, \quad i = 1, \ldots, n.$$

Flaschka showed that this map, when restricted to the tridiagonal elements of $\mathfrak{sl}_n$, is Poisson [5, 6]. Therefore we can work in either space and obtain equivalent results. One reason this is possible is because the Lie-Poisson structure restricted to the tridiagonal elements is nondegenerate [10].

Before going on, we need to show that L and M so defined do indeed give a Lax pair for the Toda lattice, that is, we need to demonstrate that these matrices do indeed satisfy Lax's equation (3.8). First lets examine $\dot{L}$. By (3.2) - (3.5) we obtain

$$\dot{L} = \begin{pmatrix} \dot{b_1} & \dot{a_1} & & & 0 \\ \dot{a_1} & \dot{b_2} & \dot{a_2} & & \\ & \dot{a_2} & \ddots & \ddots & \\ & & \ddots & \dot{b_{n-1}} & \dot{a_{n-1}} \\ 0 & & & \dot{a_{n-1}} & \dot{b_n} \end{pmatrix}$$

$$= \begin{pmatrix} 2a_1^2 & a_1(b_2-b_1) & & & 0 \\ a_1(b_2-b_1) & 2(a_2^2-a_1^2) & a_2(b_3-b_2) & & \\ & a_2(b_3-b_2) & \ddots & \ddots & \\ & & \ddots & 2(a_{n-1}^2-a_{n-2}^2) & a_{n-1}(b_n-b_{n-1}) \\ 0 & & & a_{n-1}(b_n-b_{n-1}) & 2(a_n^2-a_{n-1}^2) \end{pmatrix}$$

Now we will calculate $[M, L] = ML - LM$.

$$ML = \begin{pmatrix} a_1^2 & a_1b_2 & a_1a_2 & & 0 \\ -a_1b_1 & a_2^2-a_1^2 & a_2b_3 & \ddots & \\ -a_1a_2 & -a_2b_2 & \ddots & \ddots & a_{n-2}a_{n-1} \\ & \ddots & \ddots & a_{n-1}^2-a_{n-2}^2 & a_{n-1}b_n \\ 0 & & -a_{n-2}a_{n-1} & -a_{n-1}b_{n-1} & a_n^2-a_{n-1}^2 \end{pmatrix}$$

$$LM = \begin{pmatrix} -a_1^2 & a_1b_1 & a_1a_2 & & 0 \\ -a_1b_2 & -a_2^2+a_1^2 & a_2b_2 & \ddots & \\ -a_1a_2 & -a_2b_3 & \ddots & \ddots & a_{n-2}a_{n-1} \\ & \ddots & \ddots & -a_{n-1}^2+a_{n-2}^2 & a_{n-1}b_{n-1} \\ 0 & & -a_{n-2}a_{n-1} & -a_{n-1}b_n & -a_n^2+a_{n-1}^2 \end{pmatrix}$$

$$\begin{aligned} ML - LM &= \begin{pmatrix} 2a_1^2 & a_1(b_2-b_1) & & & 0 \\ a_1(b_2-b_1) & 2(a_2^2-a_1^2) & a_2(b_3-b_2) & & \\ & a_2(b_3-b_2) & \ddots & \ddots & \\ & & \ddots & 2(a_{n-1}^2-a_{n-2}^2) & a_{n-1}(b_n-b_{n-1}) \\ 0 & & & a_{n-1}(b_n-b_{n-1}) & 2(a_n^2-a_{n-1}^2) \end{pmatrix} \\ &= \dot{L}. \end{aligned}$$

Thus $\dot{L} = [L, M]$ as claimed. Therefore we can conclude that L and M constitute a Lax pair. Consequently, we can conclude that the Toda lattice is completely integrable, with first integrals obtained as eigenvalues or trace of powers of L. We need to examine how to obtain the first integrals with more depth.

3.1.2 Obtaining First Integrals for the Toda Lattice

We already mentioned that one can calculate either the eigenvalues of L or trace powers of L to obtain n independent first integrals. To briefly demonstrate this idea we will recover the Hamiltonian function, an integral of motion, by calculating $\frac{1}{2}\mathrm{tr}(L^2)$:

$$L^2 = \begin{pmatrix} a_1^2+b_1^2 & a_1(b_1+b_2) & & & 0 \\ a_1(b_1+b_2) & a_1^2+a_2^2+b_2^2 & a_2(b_2+b_3) & & \\ & a_2(b_2+b_3) & \ddots & \ddots & \\ & & \ddots & a_{n-2}^2+a_{n-1}^2+b_{n-1}^2 & a_{n-1}(b_{n-1}+b_n) \\ 0 & & & a_{n-1}(b_{n-1}+b_n) & a_{n-1}^2+b_n^2 \end{pmatrix}$$

Now we can calculate $\frac{1}{2}\mathrm{tr}(L^2)$:

$$\begin{aligned}
\frac{1}{2}\mathrm{tr}(L^2) &= \frac{1}{2}(\sum_{j=1}^{n} b_j^2 + 2\sum_{j=1}^{n-1} a_j^2) \\
&= \frac{1}{2}\sum_{j=1}^{n} b_j^2 + \sum_{j=1}^{n-1} a_j^2 \\
&= \frac{1}{2}\sum_{j=1}^{n}(-\frac{1}{2}p_j)^2 + \sum_{j=1}^{n-1}(\frac{1}{2}e^{\frac{1}{2}(q_j-q_{j+1})})^2 \\
&= \frac{1}{2}\sum_{j=1}^{n}\frac{1}{4}p_j^2 + \sum_{j=1}^{n-1}\frac{1}{4}e^{(q_j-q_{j+1})} \\
&= \frac{1}{4}(\frac{1}{2}\sum_{j=1}^{n}p_j^2 + \sum_{j=1}^{n-1}e^{(q_j-q_{j+1})}) \\
&= \frac{1}{4}H(q,p)
\end{aligned}$$

The process can be repeated with higher powers of L to obtain a sufficient number of first integrals. We will forgo doing so at this point as the computations are not informative and because we will calculate Poisson-commuting integrals of motion for the Toda flow at a later time. For now, let's think about how we might generalize this problem. The first object to consider would probably be the Hamiltonian itself.

3.1.3 Lie Algebras and the Toda Lattice

If we write Δ^+ to denote a set of simple roots for $\mathfrak{sl}_n$, then in place of (3.1) we can write

$$H(q,p) = \frac{1}{2}\sum_{j=1}^{n}p_j^2 + \sum_{\alpha_j\in\Delta^+} e^{(\alpha_j,\mathbf{q})}. \tag{3.4}$$

Suppose we want to consider a more general case, perhaps where Δ^+ is a set of simple roots for any complex simple Lie algebra. This problem was introduced by Bogoyavlensky in [1]. The Hamiltonian system with Hamiltonian

$$H(q,p) = \frac{1}{2}\sum_{j=1}^{n}p_j^2 + \sum_{\alpha_j\in\Delta^+} e^{(\alpha_j,\mathbf{q})}$$

is sometimes called the Bogoyavlensky-Toda lattice (where Δ^+ is a set of simple roots for any simple Lie algebra $\mathfrak{g}$). The system is also completely integrable and the Lax pair is built in a way analogous to the pair given above. The difference being that instead of trace-free matrices, the matrix L reflects the structure of the matrix Lie algebra with the given simple roots [10].

3.2 The Kostant-Toda Lattice

Bertram Kostant, in his extensive study, introduced a new Lax form which can be obtained be conjugating L by a diagonal matrix. The resultant matrix, which by a slight abuse of notation we also call L and with a minor alteration to the definition of a_i and b_j, has the form of a tridiagonal lower Hessenburg matrix [10]. For the case of the original Toda lattice we have

$$L = \begin{pmatrix} b_1 & 1 & & & 0 \\ a_1 & b_2 & 1 & & \\ & a_2 & b_3 & \ddots & \\ & & \ddots & \ddots & 1 \\ 0 & & & a_{n-1} & b_n \end{pmatrix}.$$

In this case M is just the projection of L onto its strictly lower triangular part. With this Lax form, Lax's equation

$$\dot{L} = [L, M]$$

is called the **Kostant-Toda equation**, and the system is referred to as the **Kostant-Toda lattice**.

This does in fact constitute a Lax pair, which can be shown by calculations nearly identical to those performed for the symmetric case. In addition to being an another Lax form for the Toda lattice (and its generalization to the Bogoyavlensky-Toda lattice), Kostant's Lax form allows for an easier transition to the even more general Toda flow.

For our purposes, we write L in the form where the distance from the diagonal is representative of the structure of the corresponding root system [16]. An example of this formulation, for $\mathfrak{g} = C_3$, is as follows:

$$L = \begin{pmatrix} b_1 & 1 & 0 & 0 & 0 & 0 \\ a_1 & b_2 & 1 & 0 & 0 & 0 \\ 0 & a_2 & b_3 & 1 & 0 & 0 \\ 0 & 0 & a_3 & -b_3 & -1 & 0 \\ 0 & 0 & 0 & -a_2 & -b_2 & -1 \\ 0 & 0 & 0 & 0 & -a_1 & -b_1 \end{pmatrix}$$

Even in this form, where the height of the roots is reflected in the structure of the matrix, the pair L and M (again with M as the projection onto the strictly lower triangular part of L) constitute a Lax pair[10].

Further generalizations have been studied. For example, the case where L is a generic symmetric matrix can be found in [12]. Before we introduce the generalization to the problem central to our study, something must be said about the apparent retreat from the context of Hamiltonian systems.

3.2.1 Hamiltonian Systems in the Background

As is the case with many physical systems, we can generalize to such a degree that the original problem can become completely hidden. This is certainly true for the Toda lattice. As a matter of fact, the practice of starting with a Hamiltonian and trying to construct a Lax pair is no longer central to study of the Kostant-Toda lattice and the Toda flow. The current approach to studying the Toda lattice in almost any of its forms begins and ends with little to no mention of its beginnings as a Hamiltonian system.

The approach found among contemporary researchers of Toda-like systems is to begin with a generic element X in a coadjoint orbit of the dual of a Lie algebra. Rarely is a Hamiltonian function mentioned, but the convention has become to define a Hamiltonian to be $\frac{1}{2}\mathrm{tr}(X^2)$. It is the integrability of this system that is studied, which results in attempting to construct, or prove the existence of, sufficient integrals of motion in involution. In essence, we have moved from particles to studying evolutions of matrices with the understanding that, if we so desire, we can view the problem through the lens of Hamiltonian systems. With this in mind we now introduce the Toda flow.

3.3 The Toda Flow

Recall that the trace of a product of $n \times n$ matrices A and B is commutative, in that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$. Furthermore, for any $m \times n$ matrix A with complex entries we have $\mathrm{tr}(A^*A) \geq 0$ (where A^* denotes the conjugate transpose of A), with $\mathrm{tr}(A^*A) = 0$ if and only if $A = 0$. This allows us to define an inner product on a space square of matrices, which we call the **trace form**, by defining

$$\langle A, B \rangle = \mathrm{tr}(B^*A). \tag{3.5}$$

Kostant showed that if $\mathfrak{b}_+^*$ is the dual of the Borel subalgebra for a simple Lie algebra, we can use the trace form to identify $\mathfrak{b}_+^*$ with the affine subspace $\epsilon + \mathfrak{b}_-$, where $\epsilon = \sum_{\alpha \in \Delta^+} e_\alpha$[10]. Thus, we can, and will, consider Hamiltonian systems on $\mathfrak{b}_+^*$ as evolutions on matrices in $\epsilon + \mathfrak{b}_-$. In fact, we will incorporate this identification into our definition of the Toda flow.

Let $\mathfrak{g}$ be simple linear Lie algebra and let $\epsilon + \mathfrak{b}_-$ be the affine subspace where $\mathfrak{b}_-$ is the negative of the Borel subalgebra and where ϵ is the sum of the simple root vectors of $\mathfrak{g}$. Let $X \in \epsilon + \mathfrak{b}_-$ and let PX denote the projection of X onto $\mathfrak{n}_-$, the negative of the nilpotent subalgebra of strictly upper triangular elements $\mathfrak{n}_+$. Then the **Toda flow** (sometimes called the **full Kostant-Toda lattice**) is a Hamiltonian system governed by the equation

$$\dot{X} = [PX, X]. \tag{3.6}$$

We say that the system is completely integrable if there exist integrals of motion equal to half the dimension of the symplectic leaves of $\mathfrak{b}_+$. So the superficial difference between X and L from the Kostant-Toda lattice is that we completely fill in the lower triangular part of L. The differences of the system, however are more profound. For example, the trace of powers of L, while integrals of motion for the system, are insufficient in number to prove Liouville integrability. Thus, a Lax form, in contrast with the Toda lattice and Kostant-Toda lattice, does not guarantee integrability of the system. The system, however, is in fact integrable, but the method for obtaining a complete set of integrals of motion in involution is somewhat more complex than for the Kostant-Toda lattice.

To find our integrals of motion for the Toda flow we need to recall the coadjoint action. If we let G be the Lie group of $\mathfrak{g}$, then G acts on $\mathfrak{g}$ by the adjoint action, and $\mathfrak{g}^*$ by the coadjoint action. That is

$$
\begin{aligned}
gx &= \mathrm{Ad}_g(x), \quad x \in \mathfrak{g}, g \in G \qquad (3.7)\\
(g\alpha)(x) &= \alpha(\mathrm{Ad}_{g^{-1}}(x)), \quad x \in \mathfrak{g}, \alpha \in \mathfrak{g}^*, g \in G.
\end{aligned}
$$

In fact, G acts on the subalgebra $\mathfrak{P}(\mathfrak{g}) \subset C^\infty(\mathfrak{g})$, of polynomials in $\mathfrak{g}$, by the same action[10]. This polynomial subalgebra was studied in detail by Chevalley, where he showed that if $\mathfrak{g}$ has rank n, and if we let $\mathfrak{P}^I(\mathfrak{g})$ be the space of polynomials invariant under the coadjoint action, then $\mathfrak{P}^I(\mathfrak{g}) = \mathbb{C}[I_1, \ldots, I_n]$, the space of polynomials with coefficients in $\mathbb{C}$ generated by the functions $\{I_1, \ldots, I_n\}$ [10]. The functions $I_1, \ldots, I_n$ are called **primitive invariant functions** (sometimes **invariant functions** or **Chevalley invariants**), and Kostant showed that the primitive invariant functions give a complete set of integrals of motion for the Kostant-Toda lattice. For some classical Lie algebras a full set primitive invariant functions for the Kostant-Toda lattice can be obtained from the coefficients of the characteristic polynomial for the matrix L from the Lax pair [10].

3.3.1 Primitive Invariant Functions and the Toda Flow

In the case of the Toda flow, while the primitive invariant functions are still integrals of motion, they are insufficient in number to prove complete integrability.

If we introduce some additional structure into our matrix X, we can again utilize coefficients of a characteristic polynomial to obtain a full set of integrals of motion. Furthermore, the coefficients in the characteristic polynomial that we need coincide exactly with those for obtaining the primitive invariant functions for the Kostant-Toda lattice, and we will still be able to extract the primitive invariants themselves. So before we proceed further into the realm of the Toda flow, lets say a bit more about these functions.

It happens that while the primitive invariant functions I_k are not unique, the degrees $\deg(I_k) = d_k$ are invariant, as are $m_k = d_k - 1$, which are often called the **exponents of** $\mathfrak{g}$. Knowing the degrees of the primitive invariant functions allows us to determine which of the coefficients of the characteristic polynomial we need. The exponents for the classical Lie algebras are now well known (see [9]):

$\underline{\mathfrak{g}}$	$\underline{\text{Exponents}}$
A_n	$1, 2, 3, \ldots, n$
B_n	$1, 3, 5, \ldots, 2n-1$
C_n	$1, 3, 5, \ldots, 2n-1$
D_n	$1, 3, 5, \ldots, 2n-3; n-1$

Therefore, we know that the degrees of the primitive invariant functions are

$\underline{\mathfrak{g}}$	Degrees of Primitive Invariant Functions
A_n	$2, 3, 4, \ldots, n+1$
B_n	$2, 4, 6, \ldots, 2n$
C_n	$2, 4, 6, \ldots, 2n$
D_n	$2, 4, 6, \ldots, 2n-2; n$

(3.8)

This allows us to extrapolate the correct coefficients of the characteristic polynomial of X we need to extract. The constant term of the characteristic polynomial will always correspond with the invariant function of highest degree, as it is the determinant of X, and consequently the highest degree polynomial in the entries of X by the principal minor definition of the characteristic polynomial. From there, we simple move up through powers of λ by the difference of the degrees (with an exception in the case of D_n which we will address later) to arrive at the position corresponding to the next highest degree invariant function.

For example, suppose we wish to extract from the characteristic polynomial of the matrix L, from the Kostant-Toda lattice on the Lie algebra A_7, a primitive invariant function of degree 3. Then starting at 8, the highest degree primitive invariant by the table above, we count down to 3, all the while counting up from 0 in exponents of λ. Therefore we see the desired coefficient is on λ^5. If we wish to extract the primitive invariant function of degree 16 on the Lie algebra D_{12}, then starting with 22, the highest degree invariant, we count down, this time by twos (the difference of the degrees), to 16, and count up by twos from 2 (as the term constant with respect to λ has degree 24) in powers of λ. Therefore the desired coefficient is on λ^8.

The case with the Toda flow is somewhat more complex. The coefficients extracted in this manner will not be primitive invariant functions nor integrals of motion for the Toda flow. The primitive invariants and all other integrals of motion for the Toda flow exist inside these coefficients and require extraction. In the case of a generic X for the Toda flow, the same coefficients in λ used for obtaining the primitive invariants for the Kostant-Toda lattice are needed for obtaining a sufficient number of integrals of motion for the Toda flow, but these coefficients require further manipulation. We will explain this more fully and examine each of the classical Lie algebras individually.

Chapter 4

The Toda Flow on a Classical Lie Algebra is Completely Integrable

We have arrived at the primary focus of this paper, the integrability of the Toda flow on a classical simple Lie algebra. In particular, we will present algorithms for explicit construction of the integrals of motion. Our method expands upon the abstract formulation used by Gekhtman and Shapiro in their proof of complete integrability of the Toda flow on a semisimple Lie algebra [8]. While their proof utilizes many new mathematical topics beyond what has been presented thus far, our procedure for explicit construction will be possible after the introduction of relatively few new ideas.

We will begin by outlining our general procedure, introducing some of the constructions used in [8] as the need arises. After outlining the procedure in general, we will present the specific method for each of the simple classical Lie algebras individually. Once we have explained how to obtain the first integrals for each of the classical simple Lie algebras, we will present an implementation of the described methods in the Maple programming language. Examples of input and output will then be provided to demonstrate both the process of construction as well as the efficacy of our Maple implementation.

4.1 A Brief Overview of Construction of Integrals of Motion

Let $\mathfrak{g}$ be a simple Lie algebra, let e_{m_0} denote the root vector in $\mathfrak{g}$ corresponding to the highest root m_0 in Φ^+, and let $\mathfrak{g}' = \ker(\mathrm{ad}_{e_{m_0}}|\mathfrak{g})$, the Lie subalgebra of elements that commute with e_{m_0}. The resultant $\mathfrak{g}'$ is also a semisimple classical Lie subalgebra [8]. Now we recursively define a descending chain of simple Lie subalgebras by letting $\mathfrak{g}_i = (\mathfrak{g}_{i-1})'$. This results in the finite nested chain of simple classical Lie algebras:

$$\mathfrak{g} \supseteq \mathfrak{g}' = \mathfrak{g}_1 \supseteq \mathfrak{g}_2 \supseteq \dots \supseteq \mathfrak{g}_l.$$

Following the order of the Lie algebras in the descending chain we let $\mathfrak{M} = \{m_0, m_1, m_2, \ldots, m_l\}$ be the set of highest roots for the Lie algebras in order of the descending chain shown above. That is, m_0 is the highest root of $\mathfrak{g}$, m_1 is the highest root in $\mathfrak{g}_1$, and so forth. After [8], we call $\mathfrak{M}$ a set of **strongly orthogonal roots** of the Lie algebra $\mathfrak{g}$.

Creating this chain computationally by repeated use of the Lie bracket, while simple enough for a computer to do, is unnecessary. Its mention here is primarily for abstract considerations.

In [8], Gekhtman and Shapiro point out that the nested chain of simple Lie algebras can be computed by examining the extended Dynkin diagram for the given Lie algebra. The next Lie algebra in the chain can be found by deleting the extra root, along with any roots to which it is connected, from the extended Dynkin diagram. This will result in the Dynkin diagram of the next Lie algebra in the sequence. This process is then repeated by performing the same action on the extended Dynkin diagram of the resultant Lie algebra, until a terminating sequence is obtained. Here we provide the chain for each of the simple classical Lie algebras.

Examination of Figure 2.2 reveals that we have the following chains for the classical Lie algebras:

$$
\begin{aligned}
&A_n \supseteq A_{n-2} \supseteq A_{n-4} \supseteq \ldots \\
&B_n \supseteq B_{n-2} \times A_1 \supseteq B_{n-4} \times A_1 \supseteq \ldots \\
&C_n \supseteq C_{n-1} \supseteq C_{n-2} \supseteq \ldots \\
&D_n \supseteq D_{n-2} \times A_1 \supseteq D_{n-4} \times A_1 \supseteq \ldots .
\end{aligned}
$$

As $B_2 \cong C_2$ (by their Dynkin diagrams) we require $n \geq 3$ for B_n, and $n \geq 4$ for D_n. Note that in the case of D_n, if n is even, the last two Lie algebras of the sequence are D_4 followed by $A_1 \times A_1 \times A_1$, and if n is odd the last three Lie algebras in the sequence are D_5, $A_3 \times A_1$, and A_1. This property will be important to remember for when we establish a count of integrals of motion.

We have explained how to abstractly produce the nested chain of Lie algebras, but how in practice do we identify which of the root vectors corresponds to the sequence of highest roots in $\mathfrak{M}$? The answer is found in how we have chosen to represent elements of $\epsilon + \mathfrak{b}_-$.

In our realization of the simple classical Lie algebras, the sequence of highest root vectors all lie on the antidiagonal in the case of types A_n and C_n, or they are skew-symmetric with respect to the antidiagonal, with nonzero elements in the first super- and sub-antidiagonals in the case of types B_n and D_n. An additional advantage of our realization is that we can count the number of highest root vectors for a simple classical Lie algebra by examining the size of the matrix together with the algebra's descending chain of Lie algebras.

4.1.1 Constructing a Sufficient Number of Integrals of Motion

In their paper, Gekhtman and Shapiro constructed a Poisson map between successive Lie subalgebras in the chain (4.2) called a $1-$**chop**. Multiple $1-$chops can be composed to give a Poisson map called a $k-$**chop**. Then they showed by induction that one can obtain integrals of motion for the Lie algebra $\mathfrak{g}$ by applying a $1-$chop and lifting the family of Poisson-commuting functions from the subsequent Lie algebra in the descending chain $\mathfrak{g}'$, and, in the more complex cases of B_n and D_n, by constructing functions invariant under the adjoint action of parabolic subgroups which stabilize the highest root of $\mathfrak{g}$ from the primitive invariant functions of $\mathfrak{g}$[8].

The method employed in this paper makes concrete and simplifies (in the cases of B_n and D_n) the work of of Gekhtman and Shapiro in [8], and was developed by myself and Zhaohu Nie over the course of numerous discussions and concrete explorations in Maple. It, in essence, performs all necessary "chops" simultaneously, after which the explicit construction of all integrals of motion is reduced to the extraction of coefficients from a polynomial (given some a scaling factor). Furthermore, this procedure results in a complete family of commuting integrals of motion without having to perform separate computations for the functions invariant under that adjoint action of the parabolic subgroups mentioned above. This method results in simple algorithms for the explicit construction of all necessary integrals of motion to prove Liouville integrability.

Given $X \in \epsilon + \mathfrak{b}_-$ in a classical simple Lie algebra $\mathfrak{g}$, together with its corresponding set of strongly orthogonal roots $\mathfrak{M}$, we obtain integrals of motion sufficient to prove complete integrability of the Toda flow (which we saw in Section 3.3 is half the dimension of the generic symplectic leaves of $\epsilon+\mathfrak{b}_-$) on $\epsilon+\mathfrak{b}_-$ from the coefficients of the polynomial

$$\det(X + \lambda I + \sum_{i=1}^{l} \mu_i e_{m_i}). \tag{4.1}$$

Which coefficients in λ are utilized depends on which of the classical Lie algebras X belongs to (see Section 3.31). In every case the coefficients extracted can be thought of as polynomials in the variables μ_i. The coefficients of interest in the μ_i are given in a sequence dependent on the order of the sequence of strongly orthogonal roots $\mathfrak{M}$.

To be more specific, let $\{\mu_1, \ldots, \mu_l\}$ be our sequence of highest roots, and let $I_k, k \in \{1, \ldots, n\}$ be the coefficients corresponding to the primitive invariant functions ordered by degree from least to greatest, excluding the degree n invariant for D_n which we mentioned defies the regular convention above (e.g. $\deg(I_2) = 3$ for type A_n and $\deg(I_2) = 4$ for types B_n, C_n, and D_n). For Lie algebras of types A_n and C_n, we find the coefficients of the polynomials I_k of interest are on the terms

$$\mu^0, \mu_1, \mu_1\mu_2, \mu_1\mu_2\mu_3, \ldots, \mu_1\mu_2 \ldots \mu_l.$$

For Lie algebras of types B_n and D_n, we find the coefficients of the polynomials I_k of interest are on the terms

$$\mu^0, \mu_1, \mu_1^2, \mu_1^2\mu_2, \mu_1^2\mu_2^2, \ldots, \mu_1^2\mu_2^2 \ldots \mu_k, \mu_1^2\mu_2^2 \ldots \mu_l^2.$$

The nonzero functions obtained in this way are then scaled by being divided by the leading coefficient with respect to the Cartan subalgebra. That is, if we label our basis elements of the Cartan subalgebra as h_i and all other basis elements in $\epsilon + \mathfrak{b}_-$ as x_i, and we view the nonzero functions as polynomials in the Cartan elements h_i, then the coefficient on the term with the most h_i is the factor by which we scale.

Which of the scaled functions is non-constant obviously depends on $\deg(I_k)$. Note that for each I_k, the term of highest degree in x_i and h_i corresponds to the term constant with respect to μ_i. Corresponding to an increase in powers of μ_i is a decrease in powers of x_i and h_i. This implies that to be of any use, $\deg(I_k)$ must always be at least 2, as otherwise the scaled functions would be constant (and consequently useless as they Poisson-commute with everything). Furthermore, this implies that if $\deg(I_k) = d$, then the coefficients on μ_i for powers in μ_i greater than $d-2$ cannot be integrals of motion. This, together with the necessity of following the order of the sequence of highest roots tells us which of the results will give non-constant functions. Let's look at a few concrete cases to illustrate this idea.

If we consider the coefficients corresponding to a primitive invariant of degree 2 for any of the classical Lie algebras, which according to our labeling convention is always I_1, we cannot use any terms of I_1 aside from the term which is constant with respect to the μ_i. For the coefficients in I_k corresponding to a primitive invariant of degree 6, any coefficients on terms with more than 4 μ_i (counting duplicates) cannot be integrals of motion.

In this overview we have of necessity spoken informally about the procedure. We will now provide specific, and more detailed, analyses for each of the classical simple Lie algebras. I remind the reader that we understand that the Lie algebras should be viewed as split-real.

4.2 Lie Algebras of Type A_n: $\mathfrak{sl}_{n+1}(\mathbb{R})$

Let $\mathfrak{g}$ be a Lie algebra of type A with rank n. We produce an arbitrary element X of $\epsilon + \mathfrak{b}_-$ in the realization described in section (3.2). Interestingly, the basis orthonormal with respect to the trace form for a classical simple Lie algebra of type A lacks much of the symmetry which appears for types B, C, and D. This results in linear combinations of the basis elements h_i of the Cartan subalgebra in the diagonal entries of X:

$$X = \begin{pmatrix} \sum_{j=1}^n a_{1j}h_j & 1 & & & 0 \\ x_1 & \sum_{j=1}^n a_{2j}h_j & 1 & & \\ x_{n+1} & x_2 & \sum_{j=1}^n a_{3j}h_j & * & \\ * & * & * & * & 1 \\ x_{\frac{n(n+1)}{2}} & * & x_{2n-1} & x_n & \sum_{j=1}^n a_{(n+1)j}h_j \end{pmatrix}$$

Inspection suggest that the dimension of $\epsilon + \mathfrak{b}_-$ is

$$n + n + (n-1) + \ldots + 3 + 2 + 1 = n + \frac{n(n+1)}{2} = \frac{n^2+3n}{2}.$$

When restricted to a generic coadjoint orbit, the entries x_i of X can be restricted as well. For Lie algebras of type A, restrictions result in the need to subtract $\lfloor \frac{n}{2} \rfloor$ from the apparent dimension in (4.10)[8]. Therefore the dimension of $\epsilon + \mathfrak{b}_-$ for n even is $\frac{n^2+3n}{2} - \frac{n}{2} = \frac{n^2+2n}{2}$, and for n odd is $\frac{n^2+3n}{2} - \frac{n-1}{2} = \frac{n^2+2n+1}{2}$. Thus, to verify complete integrability of the Toda flow on a Lie algebra of type A we need to produce $\frac{n^2+2n}{4}$ first integrals when n is even, and $\frac{n^2+2n+1}{4}$ first integrals when n is odd.

Recall that in our realization of $\epsilon + \mathfrak{b}_-$, the strongly orthogonal root vectors lie on the antidiagonal. Since the matrix is $(n+1) \times (n+1)$ we will have $\lfloor \frac{n}{2} \rfloor$ strongly orthogonal roots (and consequently $\lfloor \frac{n}{2} \rfloor$ root vectors as well). For reference, we write our sequence of strongly orthogonal roots as $\mathfrak{M} = \{m_1, \ldots, m_{\lfloor \frac{n}{2} \rfloor}\}$. Now we multiply the corresponding root vectors e_{m_j} by μ_j, in order from 1 to $\lfloor \frac{n}{2} \rfloor$, and add this result to our arbitrary element X:

$$X + \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \mu_i e_{m_i} = \begin{pmatrix} \sum_{j=1}^{n-1} a_{1j} h_j & 1 & & & \mu_1 \\ x_1 & \sum_{j=1}^{n-1} a_{2j} h_j & 1 & \iddots & \\ x_n & x_2 & \sum_{j=1}^{n-1} a_{3j} h_j & * & \\ * & * & * & * & 1 \\ x_{\frac{n(n+1)}{2}} & * & x_{2n-1} & x_{n-1} & \sum_{j=1}^{n-1} a_{nj} h_j \end{pmatrix}$$

Next we calculate $\det(X + \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \mu_i e_{m_i} + \lambda I)$. The primitive invariant functions for a Lie algebra of type A_n have degrees $\{2, 3, \ldots, n+1\}$ (Section 3.3.1), and consequently we need to extract the coefficients on λ^{n-1} down through λ^0. Let I_k be the coefficient on λ^{n-k} for $k \in \{1, 2, \ldots, n\}$. This labeling orders the I_k by degree from lowest to highest. Each coefficient I_k is now thought of as a polynomial in μ_j. To differentiate between the terms in μ_j we let $I_{k,[a_1,\ldots,a_{\lfloor \frac{n}{2} \rfloor}]}$ be the coefficient of I_k on $\mu_1^{a_1} \mu_2^{a_2} \ldots \mu_{\lfloor \frac{n}{2} \rfloor}^{a_{\lfloor \frac{n}{2} \rfloor}}$. Following the order of the sequence of highest roots we obtain the following integrals of motion:

$$\begin{gathered} I_{1,[0,\ldots,0]} \\ I_{2,[0,\ldots,0]}, I_{2,[1,0,\ldots,0]} \\ I_{3,[0,\ldots,0]}, I_{3,[1,0,\ldots,0]} \\ \vdots \end{gathered}$$

If n is odd:

$$I_{n-1,[0,\ldots,0]}, I_{n-1,[1,0,\ldots,0]}, I_{n-1,[1,1,0,\ldots,0]}, \ldots, I_{n-1,[1,1,\ldots,1,1]}$$

$$I_{n,[0,\ldots,0]}, I_{n,[1,0,\ldots,0]}, I_{n,[1,1,0,\ldots,0]}, \ldots I_{n,[1,1,\ldots,1,0]}, I_{n,[1,1,\ldots,1,1]}$$

If n is even:

$$I_{n-1,[0,\ldots,0]}, I_{n-1,[1,0,\ldots,0]}, I_{n-1,[1,1,0,\ldots,0]}, \ldots, I_{n-1,[1,1,\ldots,1,0]}$$
$$I_{n,[0,\ldots,0]}, I_{n,[1,0,\ldots,0]}, I_{n,[1,1,0,\ldots,0]}, \ldots I_{n,[1,1,\ldots,1,0]}, I_{n,[1,1,\ldots,1,1]}$$

Noting that we have $\lfloor \frac{n}{2} \rfloor$ indices, we can add up the number of functions:

If n is odd:

$$2 + 4 + \ldots + (n-1) + n = \frac{n^2 + 4n - 1}{4}$$

If n is even:

$$1 + 3 + \ldots + (n-1) + n = \frac{n^2 + 4n}{4}$$

Something appears to be wrong, as we have the wrong number of functions. However, we can rest easy because of the existence of certain special functions called **Casimirs**, which are found among our functions and which must be excluded from our count. These functions' distinguishing characteristic is they Poisson commute with everything in $C^\infty(\mathfrak{g})$ which, while interesting, is not strict enough a property to be useful for our purposes. The number of Casimirs is equal to $\lfloor \frac{n}{2} \rfloor$. Thus, for the case when n is odd we actually have $\frac{n^2+4n-1}{4} - \frac{n-1}{2} = \frac{n^2+2n+1}{4}$ integrals of motion, and when n is even we have $\frac{n^2+4n}{4} - \frac{n}{2} = \frac{n^2+2n}{4}$. Thus we have the desired number of first integrals, as well as complete integrability.

4.3 Lie Algebras of Type B_n: $\mathfrak{so}_{n,n+1}(\mathbb{R})$

If $\mathfrak{g}$ is a Lie algebra of type B_n with rank n, then an arbitrary element X in $\epsilon + \mathfrak{b}_-$ in our realization is a matrix of size $(2n+1) \times (2n+1)$ of the form

$$X = \begin{pmatrix} h_1 & 1 & & & & & 0 \\ x_1 & h_2 & 1 & & & & \\ x_{n+1} & x_2 & h_3 & * & & & \\ * & * & * & * & * & & \\ -x_{n^2+n-1} & -x_{n^2+n-2} & 0 & * & -h_3 & -1 & \\ -x_{n^2+n} & 0 & x_{n^2+n-2} & * & -x_2 & -h_2 & -1 \\ 0 & x_{n^2+n} & x_{n^2+n-1} & * & -x_{n+1} & -x_1 & -h_1 \end{pmatrix}$$

We note that $\dim(\mathfrak{b}_-) = n^2 + n$. Therefore we need a total of $\frac{n^2+n}{2} = \frac{n(n+1)}{2}$ integrals of motion to verify complete integrability.

We obtain our sequence of highest roots $\mathfrak{M} = \{m_1, \ldots, m_{\lfloor \frac{n}{2} \rfloor}\}$ and obtain the element

$$X + \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \mu_i e_{m_i} = \begin{pmatrix} h_1 & 1 & & & & -\mu_1 & 0 \\ x_1 & h_2 & 1 & & * & & \mu_1 \\ x_{n+1} & x_2 & h_3 & * & & * & \\ * & * & * & * & * & & \\ -x_{n^2+n-1} & -x_{n^2+n-2} & 0 & * & -h_3 & -1 & \\ -x_{n^2+n} & 0 & x_{n^2+n-2} & * & -x_2 & -h_2 & -1 \\ 0 & x_{n^2+n} & x_{n^2+n-1} & * & -x_{n+1} & -x_1 & -h_1 \end{pmatrix}$$

Let I_k be the coefficients on $\det(X + \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \mu_i e_{m_i} + \lambda I)$ corresponding to the primitive invariant functions in order of ascending degree. Just as before, we let $I_{k,[a_1,\ldots,a_{\lfloor \frac{n}{2} \rfloor}]}$ be the coefficient in I_k on $\mu_1^{a_1} \mu_2^{a_2} \ldots \mu_{\lfloor \frac{n}{2} \rfloor}^{a_{\lfloor \frac{n}{2} \rfloor}}$. Then following the order of the roots in $\mathfrak{M}$, we can obtain a complete set of integrals of motion.

$$I_{1,[0,\ldots,0]}$$

$$I_{2,[0,\ldots,0]}, I_{2,[1,0,\ldots,0]}$$

$$I_{3,[0,\ldots,0]}, I_{3,[1,0,\ldots,0]}, I_{3,[2,0,\ldots,0]}$$

$$\vdots$$

If n is odd:

$$I_{n-1,[0,\ldots,0]}, I_{n-1,[1,0\ldots,0]}, I_{n-1,[2,0\ldots,0]}, I_{n-1,[2,1,0,\ldots,0]}, \cdots, I_{n-1,[2,2,\ldots,2,1]}$$

$$I_{n,[0,\ldots,0]}, I_{n,[1,0\ldots,0]}, I_{n,[2,0\ldots,0]}, I_{n,[2,1,0,\ldots,0]}, \cdots, I_{n,[2,2,\ldots,2,2]}$$

If n is even:

$$I_{n-1,[0,\ldots,0]}, I_{n-1,[1,0\ldots,0]}, I_{n-1,[2,0\ldots,0]}, I_{n-1,[2,1,0,\ldots,0]}, \cdots, I_{n-1,[2,2,\ldots,2,0]}$$

$$I_{n,[0,\ldots,0]}, I_{n,[1,0\ldots,0]}, I_{n,[2,0\ldots,0]}, I_{n,[2,1,0,\ldots,0]}, \cdots, I_{n,[2,2,\ldots,2,1]}$$

In the odd case, the final row has $\frac{n-1}{2} \cdot 2 + 1 = n$ integrals of motion. In the even case, the final row has $(\frac{n}{2} - 1) \cdot 2 + 2 = n$ functions as well. In either case we obtain functions equal to

$$1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}.$$

Therefore we conclude that the Toda flow on Lie algebras of type B_n is completely integrable.

4.4 Lie Algebras of Type C_n: $\mathfrak{sp}_n(\mathbb{R})$

If $\mathfrak{g}$ is of type C_n, then in our realization a general element $X \in \epsilon + \mathfrak{b}_-$ is given by a $2n \times 2n$ matrix of the form

$$X = \begin{pmatrix} h_1 & 1 & & & & & 0 \\ x_1 & h_2 & 1 & & & & \\ x_n & x_2 & h_3 & * & & & \\ * & * & * & * & * & & \\ x_{n^2+n-3} & x_{n^2+n-4} & \sqrt{2}x_{n^2+n-5} & * & -h_3 & -1 & \\ x_{n^2+n-1} & \sqrt{2}x_{n^2+n-2} & x_{n^2+n-4} & * & -x_2 & -h_2 & -1 \\ \sqrt{2}x_{n^2+n} & x_{n^2+n-1} & x_{n^2+n-3} & * & -x_n & -x_1 & -h_1 \end{pmatrix}$$

First we note that $\dim(\epsilon + \mathfrak{b}_-)$ has dimension $n^2 + n$. Therefore we need $\frac{n^2+n}{2} = \frac{n(n+1)}{2}$ integrals of motion to verify complete integrability of the Toda flow.

Once again, we obtain our sequence of highest roots $\mathfrak{M} = \{m_1, \ldots, m_{n-1}\}$ and construct our element

$$X + \sum_{i=1}^{n-1} \mu_i e_{m_i} = \begin{pmatrix} h_1 & 1 & & & & & \mu_1 \\ x_1 & h_2 & 1 & & & * & \\ x_n & x_2 & h_3 & * & * & & \\ * & * & * & * & * & & \\ x_{n^2+n-3} & x_{n^2+n-4} & \sqrt{2}x_{n^2+n-5} & * & -h_3 & -1 & \\ x_{n^2+n-1} & \sqrt{2}x_{n^2+n-2} & x_{n^2+n-4} & * & -x_2 & -h_2 & -1 \\ \sqrt{2}x_{n^2+n} & x_{n^2+n-1} & x_{n^2+n-3} & * & -x_n & -x_1 & -h_1 \end{pmatrix}$$

Let I_k be the coefficients on $\det(X + \sum_{i=1}^{n-1} \mu_i e_{m_i} + \lambda I)$ corresponding to the primitive invariant functions in ascending order of degree. We let $I_{k,[a_1,\ldots,a_{n-1}]}$ be the coefficient in I_k on $\mu_1^{a_1}\mu_2^{a_2}\ldots\mu_{n-1}^{a_{n-1}}$. Then following the order of the roots in $\mathfrak{M}$ we have the following nonconstant functions

$$
\begin{aligned}
&I_{1,[0,\ldots,0]}\\
&I_{2,[0,\ldots,0]}, I_{2,[1,0,\ldots,0]}\\
&I_{3,[0,\ldots,0]}, I_{3,[1,0,\ldots,0]}, I_{3,[1,1,0,\ldots,0]}\\
&\qquad\qquad \vdots\\
&I_{n-1,[0,\ldots,0]}, I_{n-1,[1,0,\ldots,0]}, I_{n-1,[1,1,0,\ldots,0]}, \ldots, I_{n-1,[1,1,\ldots,1,0]}\\
&I_{n,[0,\ldots,0]}, I_{n,[1,0,\ldots,0]}, I_{n,[1,1,0,\ldots,0]}, \ldots, I_{n,[1,1,\ldots,1,0]}, I_{n,[1,1,\ldots,1,1]}.
\end{aligned}
$$

If we count the number of integrals of motion, we have

$$1 + 2 + 3 + \ldots + n - 1 + n = \frac{n(n+1)}{2}.$$

Therefore we can conclude that the Toda flow on a Lie algebra of type C_n is completely integrable.

4.5 Lie Algebras of Type D_n: $\mathfrak{so}_{n,n}(\mathbb{R})$

Finally, if $\mathfrak{g}$ is a Lie algebra of type D_n, then in our realization an arbitrary element $X \in \epsilon + \mathfrak{b}_-$ is a $2n \times 2n$ matrix of the form

$$
X = \begin{pmatrix}
h_1 & 1 & & & & & & 0\\
x_1 & h_2 & 1 & & & & & \\
x_n & x_2 & h_3 & * & -1 & & & \\
* & * & * & * & 0 & 1 & & \\
* & * & * & * & * & * & & \\
-x_{n^2-1} & -x_{n^2-2} & 0 & * & * & -h_3 & -1 & \\
-x_{n^2} & 0 & x_{n^2-2} & * & * & -x_2 & -h_2 & -1\\
0 & x_{n^2} & x_{n^2-1} & * & * & -x_n & -x_1 & -h_1
\end{pmatrix}
$$

To clarify, the last simple root vector lies on the second superdiagonal, thus for the case of D_n, our realization doesn't completely represent the root vectors according to the heights of the corresponding roots. Furthermore we note that the dimension of $\epsilon + \mathfrak{b}_-$ is n^2. To verify complete integrability we need $\lfloor \frac{n^2}{2} \rfloor$ integrals of motion.

Next we obtain the sequence of simple roots $\mathfrak{M} = \{m_1, \ldots, m_{\lfloor \frac{n-1}{2} \rfloor}\}$ and construct

$$X + \sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} \mu_i e_{m_i} = \begin{pmatrix} h_1 & 1 & & & & & -\mu_1 & 0 \\ x_1 & h_2 & 1 & & & * & & \mu_1 \\ x_n & x_2 & h_3 & * & -1 & & * & \\ * & * & * & * & 0 & 1 & & \\ * & * & * & * & * & * & & \\ -x_{n^2-1} & -x_{n^2-2} & 0 & * & * & -h_3 & -1 & \\ -x_{n^2} & 0 & x_{n^2-2} & * & * & -x_2 & -h_2 & -1 \\ 0 & x_{n^2} & x_{n^2-1} & * & * & -x_n & -x_1 & -h_1 \end{pmatrix}$$

Let I_k be the coefficients on $\det(X + \sum_{i=1}^{\lfloor \frac{n-1}{2} \rfloor} \mu_i e_{m_i} + \lambda I)$ corresponding to the primitive invariant functions in ascending order of degree, and let $I_{k,[a_1,\ldots,a_{n-1}]}$ be the coefficient in I_k on $\mu_1^{a_1}\mu_2^{a_2}\ldots\mu_{\lfloor \frac{n-1}{2} \rfloor}^{a_{\lfloor \frac{n-1}{2} \rfloor}}$. Then following the sequence of strongly orthogonal roots we have the following nonconstant functions

$$\begin{aligned} &I_{1,[0,\ldots,0]} \\ &I_{2,[0,\ldots,0]}, I_{2,[1,0,\ldots,0]} \\ &I_{3,[0,\ldots,0]}, I_{3,[1,0,\ldots,0]}, I_{3,[2,0,\ldots,0]} \\ &\qquad\vdots \end{aligned}$$

If n is even:

$$I_{n-1,[0,\ldots,0]}, I_{n-1,[1,0,\ldots,0]}, I_{n-1,[2,0,\ldots,0]}, \ldots, I_{n-1,[2,2,\ldots,2,1]}, I_{n-1,[2,2,\ldots,2,2]}$$

If n is odd:

$$I_{n-1,[0,\ldots,0]}, I_{n-1,[1,0,\ldots,0]}, I_{n-1,[2,0,\ldots,0]}, \ldots, I_{n-1,[2,2,\ldots,2,1]}, I_{n-1,[2,2,\ldots,2,1]}$$

Note that the number of functions in the bottom row in the even and odd cases have $(\frac{n}{2} - 1) \cdot 2 + 1 = n - 1$ and $(\frac{n-1}{2} - 1) \cdot 2 + 2 = n - 1$ functions, respectively.

Thus, if we count all of these functions we have

$$1 + 2 + 3 + \ldots + (n-1) = \frac{n(n-1)}{2}.$$

You may have noticed that we have not obtained a number of integrals of motion equal to half the dimension of $\mathfrak{b}_-$. It turns out, there is a primitive invariant function which we never used for anything. This is the primitive invariant function of degree n that we noted earlier has special circumstances. This function is called the Pfaffian.

Let $A = \{a_{ij}\}$ be a $2n \times 2n$ skew-symmetric matrix. The Pfaffian pf(A) of A is defined by

$$\mathrm{pf}(A) = \frac{1}{2^n n!} \sum_{\sigma \in S_{2n}} \mathrm{sgn}(\sigma) \prod_{i=1}^{n} a_{\sigma(2i-1),\sigma(2i)}.$$

Note that S_{2n} is the symmetric group and sgn is the sign of the permutation σ. A recursive, and computationally more useful definition, is

$$\mathrm{pf}(A) = \sum_{i=2}^{2n} (-1)^i a_{1i} \mathrm{pf}(A_{\hat{1}\hat{i}}). \tag{4.2}$$

Here, $A_{\hat{1}\hat{i}}$ is the matrix obtained by deleting the first and ith rows and columns of A, and we take the convention of having the Pfaffian of a 0×0 matrix to be 1.

Note that the definition requires that we have a skew-symmetric along the diagonal to calculate a Pfaffian. This requires us to conjugate our matrix X into a skew-symmetric form X_S, allowing us to calculate pf(X_S).

Once we have calculated $I_n = \mathrm{pf}(X_S)$, we again need to look at the coefficients on μ_i. The nonconstant coefficients are

$$I_{n,[0,\ldots,0]}, I_{n,[1,0,\ldots,0]}, I_{n,[1,1,0,\ldots,0]}, \ldots, I_{n,[1,1,\ldots,1,0]}, I_{n,[1,1,\ldots,1,1]}.$$

We mentioned previously that the final two Lie algebras in the descending chain for finding the strongly orthogonal roots would be important. In the case where n is odd, we saw that we move from D_5 to A_3. This implies that if n is odd we will have a Casimir function, as the case for A_3 has a single Casimir, which happens to be the function $I_{n,[1,1,\ldots,1,1]}$.

Therefore, the total number integrals of motion for n even is

$$\frac{n}{2} + \frac{n(n-1)}{2} = \frac{n^2}{2} = \lfloor \frac{n^2}{2} \rfloor.$$

and the total number in for n odd is

$$\frac{n-1}{2} + \frac{n(n-1)}{2} = \frac{n^2-1}{2} = \lfloor \frac{n^2}{2} \rfloor.$$

Therefore we can conclude that the Toda flow on a classical Lie algebra of type D_n is completely integrable. We have determined that the Toda flow on any classical simple Lie algebra is completely integrable in the Liouville sense. Now we will provide an implementation of the described methods in Maple, followed by examples of the process of construction using our Maple code.

4.6 A Maple Program for Constructing Integrals of Motion

Now that we have given an overview of the method for calculating a complete set of integrals of motion to prove integrability of the Toda flow, let's examine an implementation of this method programmed in Maple. While libraries for working with Lie algebras already exist for Maple, the specialized bases resulted in the need to create custom procedures.

4.6.1 Creating the Right Basis

For each of the classical Lie algebras, we construct a basis that coincides with the description in Section 3.2. The following procedures (An, Bn, Cn, and Dn) take the rank of the desired Lie algebra as input and return a list containing the specialized basis.

```
An:=proc(n)
        return [seq(seq(Matrix(n+1,{(i,j)=1}),j=i+1..n+1),i=1..n+1)
                ,seq(seq(Matrix(n+1,{(i,j)=1}),i=j+1..n+1),j=1..n+1)
                ,seq(Matrix(n+1,{(i,i)=1})-Matrix(n+1,{(i+1,i+1)=1})
                        ,i=1..n)];
end proc:

Bn:=proc(n)
        return DifferentialGeometry:-DGbasis([seq(seq(Matrix(2*n+1,
                {(i,j)=1})-LinearAlgebra:-Transpose(Matrix(2*n+1,
                {(2*n+2-i,2*n+2-j)=1})),j=1..n+1),i=1..n+1),
                seq(seq(-Matrix(2*n+1,{(i,j)=1}) +
```

```
                 LinearAlgebra:-Transpose(Matrix(2*n+1,
                {(2*n+2-i,2*n+2-j)=1})),j=n+2..2*n+1),
                i=1..2*n+1),seq(seq(-Matrix(2*n+1,{(i,j)=1}) +
                LinearAlgebra:-Transpose(Matrix(2*n+1,
                {(2*n+2-i,2*n+2-j)=1})),j=1..2*n+1),
                        i=n+2..2*n+1)]);
end proc:

Cn:=proc(n)
        local A,B,k,i,j,l;
        k := 0;
        A := DifferentialGeometry:-DGbasis([seq(seq(Matrix(2*n,
                {(i,j)=1})-LinearAlgebra:-Transpose(Matrix(2*n,
                {(2*n-i+1,2*n-j+1)=1})),j=1..n),i=1..n),
                seq(seq(Matrix(2*n,{(i,j)=1})+
                LinearAlgebra:-Transpose(Matrix(2*n,
                {(2*n-i+1,2*n-j+1)=1})),i=1..n),j=n+1..2*n),
                seq(seq(Matrix(2*n,{(i,j)=1})+
                LinearAlgebra:-Transpose(Matrix(2*n,
                {(2*n-i+1,2*n-j+1)=1})),i=n+1..2*n),j=1..n)]);
end proc:

Dn:=proc(n)
        return DifferentialGeometry:-DGbasis([seq(seq(
                Matrix(2*n,{(i,j)=1})-LinearAlgebra:-Transpose(
                Matrix(2*n,{(2*n+1-i,2*n+1-j)=1})),j=1..n),
                i=1..n),seq(seq(-Matrix(2*n,{(i,j)=1}) +
                 LinearAlgebra:-Transpose(Matrix(2*n,
                {(2*n+1-i,2*n+1-j)=1})),j=n+1..2*n),i=1..2*n),
                seq(seq(-Matrix(2*n,{(i,j)=1}) +
                 LinearAlgebra:-Transpose(Matrix(2*n,
                {(2*n+1-i,2*n+1-j)=1})),j=1..2*n),i=n+1..2*n)]);
end proc:
```

For example, list returned by Bn(2) is:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

It is important to note that these procedures do not return an orthogonal basis which has been normalized with respect to the trace form. The procedures which return the orthogonal normalized (to length 2) basis (OAn, OBn, OCn, ODn) rely on both a normalization procedure, NormalizeBasis, and a sorting procedure, BasisSort. The sorting simplifies the work for several other procedures which take a basis as input.

```
OAn:=proc(n)
        local A,SA;
        A := An(n);
        SA := BasisSort(A,n,"A");
        return NormalizeBasis(SA,n);
end proc:
```

```
OBn := proc(n)
        local B,SB;
        B := Bn(n);
        SB := BasisSort(B,n,"B");
        return NormalizeBasis(SB,n);
end proc:

OCn := proc(n)
        local C,SC;
        C := Cn(n);
        SC := BasisSort(C,n,"C");
        return NormalizeBasis(SC,n);
end proc:

ODn := proc(n)
        local E,SE;
        E := Dn(n);
        SE := BasisSort(E,n,"D");
        return NormalizeBasis(SE,n);
end proc:
```

Before we explain these new procedures lets look at the basis returned by OBn(2) for comparison with the output of Bn(2) above. As the elements are already orthogonal, the only changes are from Bn(2) are the order of the elements:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.3}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \tag{4.4}$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.5}$$

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.6}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.7}$$

BasisSort, as the name suggests, is a procedure which sorts the basis elements it receives as input. The method of sorting depends on which type of Lie algebra is being worked with, which is why it takes a type as input. The size, which is assigned based on the type of Lie algebra is used in the custom procedures internal to BasisSort, namely, LMatrixCompare and UMatrixCompare. These take two matrices as input and determine which of the two should be first, for lower and upper triangular matrices respectively. Based upon the output received from these procedures, BasisSort, which stores the basis in a list called B, either swaps the position of the two matrices or leaves them. More precisely, BasisSort uses a simple BubbleSort algorithm. Finally, BasisSort places the negative root vectors first, followed by the positive root vectors, with the elements of the Cartan subalgebra stored at the end.

```
BasisSort:=proc(A,rank,t)
        local i,B,C,size,n,newn;
        if t = "A" then
                size := rank+1;
        elif t = "B" then
                size := 2*rank + 1;
        elif t = "C" then
                size := 2*rank;
        elif t = "D" then
                size := 2*rank;
        else
                return -1;
        end if;
        B := A;
        i := 0;
        n := nops(B);
        while n <> 0 do
                newn := 0;
                for i from 1 to n - 1 do
                        if ArrayTools:-IsEqual(B[i],(B[i])^+)
                                and evalb(ArrayTools:-IsEqual(B[i+1]
                                                                ,(B[i])^+)=false) then
```

```
                C := B[i+1];
                B[i+1] := B[i];
                B[i] := C;
                newn := i;
            end if;
        end do;
        n := newn;
end do;
n := nops(B) - rank;
while n <> 0 do
        newn := 0;
        for i from 1 to n-1 do
            if LMatrixCompare(B[i],B[i+1],size) then
                C := B[i+1];
                B[i+1] := B[i];
                B[i] := C;
                newn := i;
            end if;
        end do;
        n := newn;
end do;
n := nops(B) - rank;
while n <> 0 do
        newn := 0;
        for i from (nops(B) - rank)/2 + 1 to n-1 do
            if UMatrixCompare(B[i],B[i+1],size) then
                C := B[i+1];
                B[i+1] := B[i];
                B[i] := C;
                newn := i;
            end if;
        end do;
        n := newn;
end do;
```

```
        n := nops(B);
        for i from 0 to floor(rank/2) - 1 do
                C := B[n - i];
                B[n - i] := B[n - rank + 1 + i];
                B[n - rank + 1 + i] := C;
        end do;
        return B;
end proc:
```

The procedures for comparing two matrices are separate according to whether they are positive (upper triangular) or negative (lower triangular) root vectors. The procedure for comparing two positive root vectors is UMatrixCompare, and for the negative LMatrixCompare. The way matrices in a basis are compared is by using the distance of a nonzero entry from both the diagonal and antidiagonal.

```
UMatrixCompare:=proc(a,b,size)
        local i,j,step;
        step:=1;
        while step < size do
                for i from step + 1 to size do
                        if abs(a[i-step,i]) < abs(b[i-step,i]) then
                                return true;
                        elif abs(a[i-step,i]) > abs(b[i-step,i]) then
                                return false;
                        end if;
                end do;
                step := step + 1;
        end do;
        return false;
end proc:

LMatrixCompare:=proc(a,b,size)
        local i,j,step;
        step := 1;
```

```
        while step < size do
                for i from step + 1 to size do
                        if abs(a[i,i-step]) < abs(b[i,i-step]) then
                                return true;
                        elif abs(a[i,i-step]) > abs(b[i,i-step]) then
                                return false;
                        end if;
                end do;
                step := step + 1;
        end do;
        return false;
end proc:
```

Note how the output for OBn(2) has the lower triangular elements first, followed by the upper triangular and finally diagonal elements. In addition, the elements for each of these three subsets is organized by the distance of nonzero elements from both the diagonal and antidiagonal, as explained above.

Now we can examine the procedure for normalizing the ordered basis. The idea is actually quite simple, it is an implementation of the Gram-Schmidt procedure, which uses the trace form (3.31) as inner product.

```
NormalizeBasis := proc(matrices,rank)
        local A,i,j,k,dim;
        dim:=nops(matrices);
        A:=[seq(0,i=1..dim)];
        for i from 1 to dim do
                A[i] := matrices[i];
        end do;
        for i from 1 to (dim - rank) do
                A[i] := A[i]/sqrt(LinearAlgebra:-Trace(A[i].A[i]^+));
        end do;
        for j from (dim - rank)+1 to dim do
                A[j] := A[j]/sqrt((LinearAlgebra:-Trace(A[j].A[j])));
```

```
                for k from j+1 to dim do
                        A[k] := A[k] - ((LinearAlgebra:-Trace(A[j].A[k]))
                                        /(LinearAlgebra:-Trace(A[j].A[j])))
                                        .A[j];
                end do;
        end do;
        return A;
end proc:
```

4.6.2 Building the Right Element

After we have constructed an orthonormal basis we need to build a generic element $X \in \epsilon + \mathfrak{b}_{-}$. The procedure GeneralElement does this. The code makes an exception in its method for Lie algebras of type D_n as the structure is slightly different.

```
GeneralElement:=proc(A,rank,t)
        local i, x,dim;
        x:=0;
        dim:=nops(A);
        for i from 1 to (dim - rank)/2 do
                x := x+x||i*A[i];
        end do;
        if t = "A" or t = "B" or t = "C" then
                for i from (dim - rank)/2 + 1 to (dim - rank)/2 + rank do
                        x := x + A[i];
                end do;
        elif t = "D" then
                for i from (dim - rank)/2 + 1 to (dim - rank)/2 + rank - 1 do
                        x := x + A[i];
                end do;
                x := x + A[(dim - rank)/2 + 2 * rank - 2];
        else
                return -1;
        end if;
        for i from 1 to rank do
```

```
            x := x + h||i * A[dim-rank + i];
        end do;
        return x;
end proc:
```

Here we have the matrix built by GeneralElement(D4,4,“D”) (where D4 is the list returned by ODn(4)):

$$\begin{pmatrix} h_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_1 & h_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ x_4 & x_2 & h_3 & 1 & 1 & 0 & 0 & 0 \\ x_7 & x_5 & x_3 & h_4 & 0 & -1 & 0 & 0 \\ -x_9 & -x_8 & -x_6 & 0 & -h_4 & -1 & 0 & 0 \\ -x_{11} & -x_{10} & 0 & x_6 & -x_3 & -h_3 & -1 & 0 \\ -x_{12} & 0 & x_{10} & x_8 & -x_5 & -x_2 & -h_2 & -1 \\ 0 & x_{12} & x_{11} & x_9 & -x_7 & -x_4 & -x_1 & -h_1 \end{pmatrix} \tag{4.8}$$

The next step is to obtain the sequence of highest root vectors. The way we have constructed our Lie algebras makes this fairly simple as we know exactly what they look like. For example, with Lie algebras of type A_n or C_n, the sequence of highest root vectors is given by the positive root vectors whose nonzero element lies on the antidiagonal, beginning from the top right corner to get the proper order. This allows us to just check whether the elements on the upper antidiagonal are nonzero. A similar process is used for Lie algebras of types B_n and D_n, where symmetry across the antidiagonal is employed. The procedure which constructs the list of highest root vectors is HighestRoots, and it takes a list of orthonormal basis vectors, the rank, and the type of the Lie algebra as input.

```
HighestRoots:=proc(A,rank,t)
        local size,l,i,j,k,C,temp;
        k := 1;
        if t = "A" then
                size := rank + 1;
                C := [seq(0,m=1..ceil(size/2)-1)];
```

```
        for i from (nops(A) - rank)/2 + 1 to nops(A) - rank do
                for j from 1 to ceil(size/2)-1 do
                        if A[i][j,size + 1 -j] <> 0 then
                                C[k] := A[i];
                                k := k + 1;
                        end if;
                end do;
        end do;
elif t = "B" then
        size := 2*rank + 1;
        C := [seq(0,m=1..floor(rank/2))];
        for i from (nops(A) - rank)/2 + 1 to nops(A) - rank do
                for j from 1 to floor(rank/2) do
                        if A[i][j,size-j] = -A[i][j+1,size + 1 -j]
                                and A[i][j,size - j] <> 0 then
                                C[k] := A[i];
                                k := k + 1;
                        end if;
                end do;
        end do;
elif t = "C" then
        size := 2*rank;
        C := [seq(0,m=1..rank-1)];

        for i from (nops(A) - rank)/2 + 1 to nops(A) - rank do
                for j from 1 to rank - 1 do
                        if A[i][j,size + 1 -j] <> 0 then
                                C[k] := A[i];
                                k := k + 1;
                        end if;
                end do;
        end do;
elif t = "D" then
        size := 2*rank;
```

```
                C := [seq(0,m=1..floor((rank-1)/2))];
                for i from (nops(A) - rank)/2 + 1 to nops(A) - rank do
                        for j from 1 to floor((rank-1)/2) do
                                if A[i][j,size-j] = -A[i][j+1,size + 1 -j]
                                        and A[i][j,size - j] <> 0 then
                                        C[k] := A[i];
                                        k := k + 1;
                                end if;
                        end do;
                end do;
        else
                return -1;
        end if;
        for i from 1 to floor(nops(C)/2) do
                temp := C[i];
                C[i] := C[nops(C)-i+1];
                C[nops(C)-i+1] := temp;
        end do;
        return C;
end proc:
```

With the general element X and the list of highest root vectors e_{m_i} the next step is to calculate $\det(X + \lambda I + \sum_{i=1}^{l} \mu_i e_{m_i})$. The procedure for this purpose is called TodaFlowDet and it takes the general element and list of highest root vectors as input.

```
TodaFlowDet:=proc(x,ems)
        return LinearAlgebra:-Determinant(x + lambda
                * Matrix(LinearAlgebra:-RowDimension(x),shape=identity)
                + add(mu||i * ems[i],i=1..nops(ems)));
end proc:
```

4.6.3 Extracting the Right Coefficients

The final step is to extract the appropriate coefficients and divide by the leading coefficient with respect to the Cartan subalgebra as explained previously. ConstructFirstIntegrals does this by iterating over all of the important coefficients in λ and each of the μ_i. As explained above, the coefficients depend on which of the classical Lie algebras we are working with. Note also, that this procedure utilizes another procedure called Pfaffian for Lie algebras of type D_n, which is an implementation of the Pfaffian as defined above. Furthermore, every case utilizes the LeadingCoeff procedure for scaling of the retrieved coefficients. The LeadingCoeff procedure just returns the leading coefficient of the determinant with respect to the Cartan subalgebra

```
ConstructInvariantFunctions:=proc(det,x,rank,ems,t)
        local A,B,C,E,i,j,k,l,s,degs,temp,temp1,temp2;
        A := PolynomialTools:-CoefficientList(det,lambda);
        degs := EvaluateDegrees(rank,t);
        B := [seq(0,i=1..rank)];
        C := [seq(0,i=1..rank+3*rank*nops(ems))];
        s := LinearAlgebra:-RowDimension(x);
        k := 1;
        if t = "A" then
                for i from 1 to rank do
                        B[i] := A[degs[i] - 1];
                end do;
                for i from 1 to rank do
                        temp := B[i];
                        C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],temp);
                        k := k + 1;
                        for j from 1 to nops(ems) do
                                temp := coeff(temp,mu||j);
                                C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],
                                                temp);
                                k := k + 1;
                        end do;
                end do;
```

```
elif t = "B" then
        for i from 1 to rank do
                B[i] := A[degs[i] - 1];
        end do;
        for i from 1 to rank do
                temp := B[i];
                C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],temp);
                k := k + 1;
                for j from 1 to nops(ems) do
                        temp1 := coeff(temp,mu||j);
                        temp2 := coeff(temp,(mu||j)^2);
                        temp := coeff(temp,(mu||j)^2);
                        C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],
                                        temp1);
                        k := k + 1;
                        C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],
                                        temp2);
                        k := k + 1;
                end do;
        end do;
elif t = "C" then
        for i from 1 to rank do
                B[i] := A[degs[i] - 1];
        end do;
        for i from 1 to rank do
                temp1 := B[i];
                C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],
                                temp1);
                k := k + 1;
                for j from 1 to nops(ems) do
                        temp1 := coeff(temp1,mu||j);
                        C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],
                                temp1);
                        k := k + 1;
```

```
                end do;
        end do;
elif t = "D" then
        for i from 1 to rank-1 do
                B[i] := A[degs[i] + 1];
        end do;
        for i from 1 to rank-1 do
                temp := B[i];
                C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],temp);
                k := k + 1;
                for j from 1 to nops(ems) do
                        temp1 := coeff(temp,mu||j);
                        temp2 := coeff(temp,(mu||j)^2);
                        temp := coeff(temp,(mu||j)^2);
                        C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],
                                        temp1);
                        k := k + 1;
                        C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],
                                        temp2);
                        k := k + 1;
                end do;
        end do;
        temp1 := Pfaffian(add(seq(Matrix(s,s,{(i,s-i+1)=1}),
                        i=1..s)[j],j=1..s).(x+add(mu||i * ems[i],
                        i=1..nops(ems))));
        C[k] := subs([seq(mu||i = 0, i = 1..nops(ems))],temp1);
        k := k + 1;
        for j from 1 to nops(ems) do
                temp1 := coeff(temp1,mu||j);
                C[k] := subs([seq(mu||i = 0,i=1..nops(ems))],temp1);
                k := k + 1;
        end do;
else
        return -1;
```

```
        end if;
        E := [];
        for i from 1 to nops(C) do
                if C[i] <> 0 then
                        C[i] := simplify(expand(C[i]/LeadingCoeff(C[i],rank)));
                        if C[i] <> 1 then
                                E := [op(E),C[i]];
                        end if;
                end if;
        end do;
        return E;
end proc:
```

```
LeadingCoeff := proc(p,rank)
        local s,C;
        s := subs([seq(h||i = h, i = 1..rank)],p);
        C := PolynomialTools:-CoefficientList(s,h);
        return C[nops(C)];
end proc:
```

As the name suggests, the Pfaffian procedure implements the recursive definition above in (4.62).

```
Pfaffian := proc(A)
        local i;
        if LinearAlgebra:-RowDimension(A) < 2 then
                return 1;
        else
                return add((-1)^(i+1) * A[1,i+1]*Pfaffian(LinearAlgebra:
                        -SubMatrix(A,[2..i,i+2..LinearAlgebra:
                        -RowDimension(A)],[2..i,i+2..LinearAlgebra:
                        -ColumnDimension(A)])),i=1..LinearAlgebra:
                        -RowDimension(A)-1);
```

```
        end if
end proc:
```

4.6.4 Checking Our Results

We have not yet accounted for possible Casimir functions among our results. This is easily remedied with the RemoveCasimirs procedure, which takes a list of functions as input, and returns a list without any Casimir elements. The procedure works by calculating the Lie Poisson bracket of the functions with an arbitrary function of the basis elements x_i and h_j of the given Lie algebra. The LiePoissonBracket procedure is an implementation of the Lie-Poisson bracket described in Section 2.3.1. LiePoissonBracket uses the Grad procedure which is an implementation of the gradient from the definition of the Lie-Poisson bracket.

```
RemoveCasimirs := proc(E,x,Basis,rank)
        local i,j,C;
        C:=[];
        for i from 1 to nops(E) do
                if simplify(expand(LiePoissonBracket(E[i],
                          f(seq(x||j, j=1..(1/2)(nops(Basis)-rank)),
                          seq(h||j, j=1..rank)), x, Basis, rank))) <> 0 then
                        C := [op(C), E[i]];
                end if;
        end do;
        return C;
end proc;

LiePoissonBracket:=proc(A,B,X,Basis,rank)
        return LinearAlgebra:-Trace(X.(Grad(A,Basis,nops(Basis),rank)
                        .Grad(B,Basis,nops(Basis),rank)
                        - Grad(B,Basis,nops(Basis),rank)
                        .Grad(A,Basis,nops(Basis),rank)));
end proc:

Grad:=proc(f,Basis,dim,rank)
```

```
        return add(diff(f,x||i).(Basis[i])^+,i=1..(dim-rank)/2)
                +add(diff(f,h||i).(Basis[nops(Basis)-rank+i]),i=1..rank);
end proc:
```

After retrieving the integrals of motion we check their Lie-Poisson brackets to verify involution and prove integrability of the system. We use the CheckLiePoisson procedure to calculate the Lie Poisson brackets for each of the obtained integrals of motion returning true or false depending on whether the functions' Poisson brackets vanish.

```
CheckLiePoisson := proc(E,x,Basis,rank)
        local i,j;
        for i from 1 to nops(E) do
                for j from i+1 to nops(E) do
                        if simplify(expand(LiePoissonBracket(E[i],E[j],
                                        x,Basis,rank))) <> 0 then
                                return false;
                        end if;
                end do;
        end do;
        return true;
end proc:
```

4.7 Examples

To truly understand what we are doing here we need to work through some concrete examples. We will take one from each of the four types of Lie algebras mentioned above. We will begin with some simple examples from smaller spaces.

4.7.1 $\mathfrak{sp}_4(\mathbb{R})$

An arbitrary element ox $\epsilon + \mathfrak{b}_-$ for the case with C_2 is:

$$\begin{pmatrix} h_1 & 1 & 0 & 0 \\ x_1 & h_2 & \sqrt{2} & 0 \\ x_3 & \sqrt{2}x_2 & -h_2 & -1 \\ \sqrt{2}x_4 & x_3 & -x_1 & -h_1 \end{pmatrix}$$

Now we find the highest root vector:

$$\begin{pmatrix} 0 & 0 & 0 & \sqrt{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The determinant we extract our functions from is calculated to be:

$$\begin{aligned} &2h_2^2\mu_1x_4 - 2\lambda^2\mu_1x_4 + 4\mu_1x_2x_4 - 2\sqrt{2}h_2\mu_1x_1x_3 + h_1^2h_2^2 - h_1^2\lambda^2 - h_2^2\lambda^2 \\ &+ \lambda^4 + 2\mu_1x_1^2x_2 + 2x_4 - 2\sqrt{2}h_1x_3 + 2h_1^2x_2 - 2h_1h_2x_1 - 2\lambda^2x_1 - 2\lambda^2x_2 \\ &- 2\mu_1x_3^2 + x_1^2 \end{aligned}$$

The primitive invariant functions for a simple classical Lie algebra of type C with rank 2 have degrees 2 and 4. Therefore we will examine the coefficients of the above polynomial on λ^2 and λ^0.

By inspection, and using our notational convention, we write the coefficient on λ^2 as:

$$I_1 = 2\mu_1x_4 - h_1^2 - h_2^2 - 2x_1 - 2x_2$$

and the coefficient on λ^0 (the term constant with respect to λ) as:

$$\begin{aligned} I_2 = {} &2h_2^2\mu_1x_4 + 4\mu_1x_2x_4 - 2\sqrt{2}h_2\mu_1x_1x_3 + h_1^2h_2^2 + 2\mu_1x_1^2x_2 + 2x_4 - 2\sqrt{2}h_1x_3 \\ &+ 2h_1^2x_2 - 2h_1h_2x_1 - 2\mu_1x_3^2 + x_1^2 \end{aligned}$$

The integrals of motion will be the functions $I_{1,[0]}, I_{2,[0]}$, and $I_{2,[1]}$. The function $I_{1,[0]}$ is the scaled (see Section 4.1) coefficient of I_1 on μ_1^0 (the term constant with respect to μ_1):

$$I_{1,[0]} = \frac{1}{-1}(-h_1^2 - h_2^2 - 2x_1 - 2x_2) = h_1^2 + h_2^2 + 2x_1 + 2x_2.$$

Likewise, the function $I_{2,[0]}$ is the scaled coefficient of I_2 on μ_1^0:

$$\begin{aligned} I_{2,[0]} &= \frac{1}{1}(h_1^2h_2^2 + 2x_4 - 2\sqrt{2}h_1x_3 + 2h_1^2x_2 - 2h_1h_2x_1 + x_1^2) \\ &= h_1^2h_2^2 + 2x_4 - 2\sqrt{2}h_1x_3 + 2h_1^2x_2 - 2h_1h_2x_1 + x_1^2 \end{aligned}$$

Finally, the function $I_{2,[1]}$ is the scaled coefficient of I_2 on μ_1^1:

$$\begin{aligned} I_{2,[1]} &= \frac{1}{2x_4}(2h_2^2x_4 + 4x_2x_4 - 2\sqrt{2}h_2x_1x_3 + 2x_1^2x_2 - 2x_3^2) \\ &= h_2^2 + 2x_2 - \sqrt{2}\frac{x_1x_3}{x_4} + \frac{x_1^2x_2}{x_4} - \frac{x_3^2}{x_4} \end{aligned}$$

4.7.2 $\mathfrak{sl}_3(\mathbb{R})$

We will now look at another fairly simple example, that of A_2, which as a case where our method extracts a Casimir function. Note that in the case of A_n, the root vectors are normalized to length 1. A general element in A_2 is:

$$\begin{pmatrix} \frac{\sqrt{2}}{2}h_1 + \frac{\sqrt{6}}{6}h_2 & 1 & 0 \\ x_1 & -\frac{\sqrt{2}}{2}h_1 + \frac{\sqrt{6}}{6}h_2 & 1 \\ x_3 & x_2 & -\frac{\sqrt{6}}{3}h_2 \end{pmatrix}$$

The highest root vector we find is:

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Now we calculate $\det(\lambda I + X + \mu_1 e_{\mu_1})$:

$$\begin{aligned} &\frac{\sqrt{6}}{6}h_1^2h_2 - \frac{1}{2}h_1^2\lambda - \frac{\sqrt{2}}{2}h_1x_2 - \frac{\sqrt{6}}{18}h_2^3 - \frac{1}{2}h_2^2\lambda \\ &- \frac{\sqrt{6}}{6}h_2x_2 + \lambda^3 - x_2\lambda + \frac{\sqrt{6}}{3}x_1h_2 - \lambda x_1 + x_2\mu_1x_1 + x_3 \\ &+ \frac{\sqrt{2}}{2}x_3\mu_1h_1 - \frac{\sqrt{6}}{6}x_3\mu_1h_2 - x_3\mu_1\lambda \end{aligned}$$

The degrees of the primitive invariant functions for A_2 are 2 and 3. Therefore we need the coefficients on λ^1 and λ^0. We obtain the functions:

$$
\begin{aligned}
I_1 &= -\frac{1}{2}h_1^2 - \frac{1}{2}h_2^2 - x_2 - x_1 - x_3\mu_1 \\
I_2 &= \frac{\sqrt{6}}{6}h_1^2h_2 - \frac{\sqrt{2}}{2}h_1x_2 - \frac{\sqrt{6}}{18}h_2^3 - \frac{\sqrt{6}}{6}h_2x_2 + \frac{\sqrt{6}}{3}x_1h_2 \\
&\quad + x_2\mu_1x_1 + x_3 + \frac{\sqrt{2}}{2}x_3\mu_1h_1 - \frac{\sqrt{6}}{6}x_3\mu_1h_2
\end{aligned}
$$

To obtain our first integrals we create the functions $I_{1,[0]}$, $I_{2,[0]}$, and $I_{2,[1]}$ by scaling (see Section 4.1) the coefficients of I_1 on μ_1^0, I_2 on μ_1^0, and I_2 on μ_1^1, respectively:

$$
\begin{aligned}
I_{1,[0]} &= -2(-\frac{1}{2}h_1^2 - \frac{1}{2}h_2^2 - x_1 - x_2) \\
I_{2,[0]} &= \frac{-9}{\sqrt{6}}(\frac{\sqrt{6}}{6}h_1^2h_2 - \frac{\sqrt{2}}{2}h_1x_2 - \frac{\sqrt{6}}{18}h_2^3 - \frac{\sqrt{6}}{6}h_2x_2 \\
&\quad + \frac{\sqrt{6}}{3}x_1h_2 + x_3) \\
I_{2,[1]} &= \frac{6}{x_3(3\sqrt{2} - \sqrt{6})}(x_2x_1 + \frac{\sqrt{2}}{2}x_3h_1 - \frac{\sqrt{6}}{6}x_3h_2)
\end{aligned}
$$

The last function is actually a Casimir, so we have the expected number of first integrals:

$$
\begin{aligned}
I_{1,[0]} &= -2(-\frac{1}{2}h_1^2 - \frac{1}{2}h_2^2 - x_1 - x_2) \\
I_{2,[0]} &= \frac{-9}{\sqrt{6}}(\frac{\sqrt{6}}{6}h_1^2h_2 - \frac{\sqrt{2}}{2}h_1x_2 - \frac{\sqrt{6}}{18}h_2^3 - \frac{\sqrt{6}}{6}h_2x_2 \\
&\quad + \frac{\sqrt{6}}{3}x_1h_2 + x_3)
\end{aligned}
$$

4.7.3 $\mathfrak{sl}_4(\mathbb{R})$

The benefit of this method can only be truly appreciated when working with larger Lie algebras, where a computer becomes a necessity. Let's examine the case for $\mathfrak{g} = A_3$, seemingly only a small step up from A_2. We will quickly see that the added complexity is substantial. We create our general element:

$$X = \begin{pmatrix} \frac{\sqrt{2}}{2}h_1 + \frac{\sqrt{6}}{6}h_2 + \frac{\sqrt{3}}{6}h_3 & 1 & 0 & 0 \\ x_1 & -\frac{\sqrt{2}}{2}h_1 + \frac{\sqrt{6}}{6}h_2 + \frac{\sqrt{3}}{6}h_3 & 1 & 0 \\ x_4 & x_2 & -\frac{\sqrt{6}}{3}h_2 + \frac{\sqrt{3}}{6}h_3 & 1 \\ x_6 & x_5 & x_3 & -\frac{\sqrt{3}}{2}h_3 \end{pmatrix}$$

Now we obtain the sequence of highest root vectors(there is only one in this case):

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

With these we can calculate $\det(\lambda I + X + \mu_1 e_{\mu_1})$:

$$\begin{aligned} \det = & -x_1\mu_1x_2x_3 + x_1x_5\mu_1\lambda + x_4\mu_1x_3\lambda - \frac{\sqrt{2}}{2}h_1x_2\lambda + \frac{\sqrt{3}}{6}h_2^2\lambda \\ & - \frac{\sqrt{3}}{3}h_3\lambda x_3 + \frac{\sqrt{3}}{3}h_3x_2\lambda + \frac{\sqrt{3}}{3}x_1h_3\lambda + \frac{\sqrt{3}}{6}h_1^2h_3\lambda - \frac{\sqrt{3}}{2}x_4h_3 + \frac{\sqrt{2}}{12}h_2^2h_3 \\ & - \frac{1}{2}h_1^2\lambda^2 + \frac{\sqrt{2}}{2}h_1x_5 + \frac{\sqrt{3}}{6}h_3x_5 + \lambda^4 - \frac{1}{2}h_3^2\lambda^2 - \lambda^2x_1 + \frac{\sqrt{6}}{12}x_6\mu_1h_1h_3 \\ & - \frac{\sqrt{6}}{18}h_2^3\lambda - \frac{\sqrt{2}}{4}h_1^2h_2h_3 - \frac{\sqrt{2}}{2}x_1h_2h_3 + \frac{\sqrt{2}}{4}h_2x_2h_3 - \frac{\sqrt{2}}{6}h_2h_3x_3 - \frac{1}{48}h_3^4 \\ & + \frac{1}{4}x_1h_3^2 + x_1x_3 + \frac{1}{8}h_1^2h_3^2 + \frac{1}{2}h_1^2x_3 + \frac{1}{8}h_2^2h_3^2 - \frac{1}{6}h_2^2x_3 - \frac{1}{12}h_3^2x_3 + \frac{1}{4}h_3^2x_2 \\ & + \frac{\sqrt{6}}{4}h_1x_2h_3 - x_6 - x_4x_5\mu_1 + \frac{1}{3}x_6\mu_1h_2^2 - \frac{1}{12}x_6\mu_1h_3^2 - x_6\lambda^2\mu_1 + x_6\mu_1x_2 \\ & - \frac{\sqrt{3}}{9}h_3^3\lambda - \frac{\sqrt{6}}{3}x_1x_5\mu_1h_2 + \frac{\sqrt{6}}{6}x_4\mu_1x_3h_2 + \frac{\sqrt{6}}{6}x_6\mu_1h_2\lambda - \frac{1}{2}h_2^2\lambda^2 \\ & - \lambda^2x_3 + \lambda x_5 + x_4\lambda - \frac{\sqrt{6}}{3}h_2\lambda x_3 - \frac{\sqrt{6}}{6}h_2x_2\lambda + \frac{\sqrt{6}}{3}x_1h_2\lambda + \frac{\sqrt{6}}{6}h_1^2h_2\lambda \\ & - \frac{\sqrt{3}}{3}x_6\mu_1h_1h_2 + \frac{\sqrt{2}}{12}x_6\mu_1h_2h_3 - \lambda^2x_2 + \frac{\sqrt{6}}{6}h_2x_5 + \frac{\sqrt{3}}{6}x_1x_5\mu_1h_3 \end{aligned}$$

What remains is to extract and scale the appropriate coefficients and λ and μ_1. We obtain the following functions:

$$
\begin{aligned}
I_{1,[0]} &= \frac{2}{3}x_2 + \frac{1}{3}h_2^2 + \frac{1}{3}h_3^2 + \frac{2}{3}x_3 + \frac{1}{3}h_1^2 + \frac{2}{3}x_1 \\
I_{2,[0]} &= \frac{\sqrt{3}}{6\sqrt{2}+12}(3\sqrt{6}h_1^2h_2 - \sqrt{6}h_2^3 + 6\sqrt{6}x_1h_2 - 3\sqrt{6}h_2x_2 - 6\sqrt{6}h_2x_3 \\
&\quad + 3\sqrt{3}h_1^2h_3 + 3\sqrt{3}h_2^2h_3 - 2\sqrt{3}h_3^3 - 9\sqrt{2}h_1x_2 + 6\sqrt{3}x_1h_3 + 6\sqrt{3}x_2h_3 \\
&\quad - 6\sqrt{3}h_3x_3 + 18x_4 + 18x_5) \\
I_{2,[1]} &= \frac{\sqrt{6}x_6h_2 + 3\sqrt{2}x_6h_1 - 2\sqrt{3}x_6h_3 + 6x_1x_5 + 6x_3x_4}{x_6(\sqrt{6}+3\sqrt{2}-2\sqrt{3})}
\end{aligned}
$$

$$
\begin{aligned}
I_{3,[0]} &= \frac{-1}{8\sqrt{2}-11}(12\sqrt{6}h_1x_2h_3 - 12\sqrt{2}h_1^2h_2h_3 + 4\sqrt{2}h_2^3h_3 + 8\sqrt{6}h_2x_5 \\
&\quad - 24\sqrt{2}x_1h_2h_3 + 12\sqrt{2}h_2x_2h_3 - 8\sqrt{2}h_2h_3x_3 + 6h_1^2h_3^2 + 6h_2^2h_3^2 - h_3^4 \\
&\quad + 24\sqrt{2}h_1x_5 - 24\sqrt{3}x_4h_3 + 8\sqrt{3}h_3x_5 + 24h_1^2x_3 - 8h_2^2x_3 + 12x_1h_3^2 + 12h_3^2x_2 \\
&\quad - 4h_3^2x_3 + 48x_1x_3 - 48x_6) \\
I_{3,[1]} &= \frac{1}{x_6(\sqrt{6}+\sqrt{2}-4\sqrt{3}+3)}(\sqrt{6}x_6h_1h_3 - 4\sqrt{6}x_1x_5h_2 + 2\sqrt{6}x_4x_3h_2 \\
&\quad - 6\sqrt{2}x_4x_3h_1 + \sqrt{2}x_6h_2h_3 - 4\sqrt{3}x_6h_1h_2 + 2\sqrt{3}x_1x_5h_3 + 2\sqrt{3}x_4x_3h_3 \\
&\quad + 4x_6h_2^2 - x_6h_3^2 - 12x_1x_2x_3 + 12x_2x_6 - 12x_5x_4)
\end{aligned}
$$

According to our discussion in Section 4.2 $I_{2,[1]}$ is a Casimir function. We will use RemoveCasimirs(E,X,A3,3) to check. As expected, we are given the following functions:

$$
\begin{aligned}
I_{1,[0]} &= \frac{2}{3}x_2 + \frac{1}{3}h_2^2 + \frac{1}{3}h_3^2 + \frac{2}{3}x_3 + \frac{1}{3}h_1^2 + \frac{2}{3}x_1 \\
I_{2,[0]} &= \frac{\sqrt{3}}{6\sqrt{2}+12}(3\sqrt{6}h_1^2h_2 - \sqrt{6}h_2^3 + 6\sqrt{6}x_1h_2 - 3\sqrt{6}h_2x_2 - 6\sqrt{6}h_2x_3 \\
&\quad + 3\sqrt{3}h_1^2h_3 + 3\sqrt{3}h_2^2h_3 - 2\sqrt{3}h_3^3 - 9\sqrt{2}h_1x_2 + 6\sqrt{3}x_1h_3 + 6\sqrt{3}x_2h_3 \\
&\quad - 6\sqrt{3}h_3x_3 + 18x_4 + 18x_5)
\end{aligned}
$$

$$
\begin{aligned}
I_{3,[0]} = {} & \frac{-1}{8\sqrt{2}-11}(12\sqrt{6}h_1x_2h_3 - 12\sqrt{2}h_1^2h_2h_3 + 4\sqrt{2}h_2^3h_3 + 8\sqrt{6}h_2x_5 \\
& - 24\sqrt{2}x_1h_2h_3 + 12\sqrt{2}h_2x_2h_3 - 8\sqrt{2}h_2h_3x_3 + 6h_1^2h_3^2 + 6h_2^2h_3^2 - h_3^4 \\
& + 24\sqrt{2}h_1x_5 - 24\sqrt{3}x_4h_3 + 8\sqrt{3}h_3x_5 + 24h_1^2x_3 - 8h_2^2x_3 + 12x_1h_3^2 + 12h_3^2x_2 \\
& - 4h_3^2x_3 + 48x_1x_3 - 48x_6) \\
I_{3,[1]} = {} & \frac{1}{x_6(\sqrt{6}+\sqrt{2}-4\sqrt{3}+3)}(\sqrt{6}x_6h_1h_3 - 4\sqrt{6}x_1x_5h_2 + 2\sqrt{6}x_4x_3h_2 \\
& - 6\sqrt{2}x_4x_3h_1 + \sqrt{2}x_6h_2h_3 - 4\sqrt{3}x_6h_1h_2 + 2\sqrt{3}x_1x_5h_3 + 2\sqrt{3}x_4x_3h_3 \\
& + 4x_6h_2^2 - x_6h_3^2 - 12x_1x_2x_3 + 12x_2x_6 - 12x_5x_4)
\end{aligned}
$$

It is likely apparent by now why a computer is necessary for working with any larger spaces.

4.7.4 $\mathfrak{so}_{3,4}(\mathbb{R})$

Let's look at a larger space, B_3. A general element in B_3 takes the form:

$$
\begin{pmatrix}
h_1 & 1 & 0 & 0 & 0 & 0 & 0 \\
x_1 & h_2 & 1 & 0 & 0 & 0 & 0 \\
x_4 & x_2 & h_3 & 1 & 0 & 0 & 0 \\
x_6 & x_5 & x_3 & 0 & -1 & 0 & 0 \\
-x_8 & -x_7 & 0 & -x_3 & -h_3 & -1 & 0 \\
-x_9 & 0 & x_7 & -x_5 & -x_2 & -h_2 & -1 \\
0 & x_9 & x_8 & -x_6 & -x_4 & -x_1 & -h_1
\end{pmatrix}
$$

The one and only highest root vector we need is:

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Now we calculate $\det(\lambda I + X + \mu_1 e_{m_1})$:

$$
\begin{aligned}
&-\lambda(h_1^2h_2^2h_3^2 - h_12h_2^2\lambda^2 - h_1^2h_3^2\lambda^2 + h_1^2\lambda^4 + 2h_1h_2h_3^2\mu_1x_9 - 2h_1h_2h_3\mu_1x_2x_8\\
&-2h_1h_2h_3\mu_1x_4x_7 - 2h_1h_2\lambda^2\mu_1x_9 + 2h_1h_2\mu_1x_2x_3x_6 - 2h_1h_2\mu_1x_3x_4x_5\\
&-2h_1h_3^2\mu_1x_5x_6 + 2h_1h_3\mu_1x_2x_3x_6 + 2h_1h_3\mu_1x_3x_4x_5 - 2h_1\lambda^2\mu_1x_2x_8\\
&-2h_1\lambda^2\mu_1x_4x_7 + 2h_1\lambda^2\mu_1x_5x_6 - 2h_1\mu_1x_2x_3^2x_4 - h_2^2h_3^2\lambda^2 + h_2^2\lambda^4\\
&+2h_2h_3^2\mu_1x_5x_6 - 2h_2h_3\mu_1x_2x_3x_6 - 2h_2h_3\mu_1x_2x_4x_5 + h_2\lambda^2\mu_1x_2x_8\\
&-2h_2\lambda^2\mu_1x_4x_7 - 2h_2\lambda^2\mu_1x_5x_6 + 2h_2\mu_1x_2x_3^2x_4 + h_3^2\lambda^4 - 2h_3^2\lambda^2\mu_1x_9\\
&+h_3^2\mu_1^2x_9^2 - 2h_3^2\mu_1x_1x_5^2 + 2h_3\lambda^2\mu_1x_2x_8 - 2h_3\lambda^2x_4x_7 - 2h_3\mu_1^2x_2x_8x_9\\
&+2h_3\mu_1^2x_4x_7x_9 + 4h_3\mu_1x_1x_2x_3x_5 - \lambda^6 + 2\lambda^4\mu_1x_9 - \lambda^2\mu_1^2x_9^2 - r\lambda^2\mu_1x_1x_2x_7\\
&+2\lambda^2\mu_1x_1x_5^2 - 2\lambda^2\mu_1x_2x_3x_6 + 2\lambda^2\mu_1x_3x_4x_5 + \mu_1^2x_2^2x_8^2 + 2\mu_1^2x_2x_3x_6x_9\\
&-2\mu_1^2x_2x_4x_7x_8 - 2\mu_1^2x_2x_5x_6x_8 + 2\mu_1^2x_2x_6^2x_7 - 2\mu_1^2x_3x_4x_5x_9 + \mu_1^2x_4^2x_7^2\\
&+2\mu_1^2x_4x_5^2x_8 - 2\mu_1^2x_4x_5x_6x_7 - 2\mu_1x_1x_2^2x_3^2 + 2h_1^2h_2^2x_3 - 2h_1^2h_2h_3x_2 - 2h_1^2\lambda^2x_2\\
&-2h_1^2\lambda^2x_3 - 2h_1h_2h_3^2x_1 + 2h_1h_2\lambda^2x_1 + 4h_1h_2\mu_1x_3x_9 - 2h_1h_2\mu_1x_5x_8\\
&+2h_1h_2\mu_1x_6x_7 - 2h_1h_3\mu_1x_2x_9 - 2h_1h_3\mu_1x_5x_8 - 2h_1h_3\mu_1x_6x_7 + 2h_1\mu_1x_2^2x_8\\
&+2h_1\mu_1x_2x_3x_8 - 2h_1\mu_1x_2x_4x_7 - 2h_1\mu_1x_2x_5x_6 + 2h_1\mu_1x_3x_4x_7\\
&+3h_1\mu_1x_4x_5^2 - 2h_2^2\lambda^2x_3 + 2h_2h_3\lambda^2x_2 + 2h_2h_3\mu_1x_5x_8 + 2h_2h_3\mu_1x_6x_7\\
&-2h_2\mu_1x_2x_3x_8 - 2h_2\mu_1x_3x_4x_7 - 2h_3^2\lambda^2x_1 - 2h_3^2\mu_2x_1x_9 + 2h_3^2\mu_1x_6^2\\
&+2h_3\mu_1x_1x_2x_8 - 2h_3\mu_1x_1x_4x_7 - 4h_3\mu_1x_1x_5x_7 - 2h_3\mu_1x_2x_5x_6 - 4h_3\mu_1x_3x_4x_6\\
&+2h_3\mu_1x_4x_5^2 + 2\lambda^4x_1 + 2\lambda^4x_2 + 2\lambda^4x_3 + 2\lambda^2\mu_1x_1x_9\\
&-2\lambda^2\mu_1x_2x_9 - 4\lambda^2\mu_1x_3x_9 + 4\lambda^2\mu_1x_4x_8 + 2\lambda^2\mu_1x_5x_8 - 2\lambda^2\mu_1x_6^2
\end{aligned}
$$

$$-2\lambda^2\mu_1x_6x_7+2\mu_1^2x_3x_9^2-2\mu_1^2x_5x_8x_9+2\mu_1^2x_6x_7x_9-2\mu_1x_1x_2x_3x_6$$
$$+4\mu_1x_1x_2x_3x_7+2\mu_1x_1x_3x_4x_5+2\mu_1x_2^2x_3x_6-2\mu_1x_2x_3x_4x_5+2\mu_1x_3^2x_4^2$$
$$-2h_1^2h_2x_5-2h_1^2h_3x_5+h_1^2x_2^2+2h_1^2x_2x_3+2h_1h_2h_3x_4-4h_1h_2x_1x_3$$
$$+2h_1h_3x_1x_2+2h_1\lambda^2x_4-2h_1\mu_1x_5x_9-2h_1\mu_1x_7x_8+2h_2\lambda^2x_4+2h_2\lambda^2x_5$$
$$+2h_2\mu_1x_7x_8+h_3^2x_1^2+2h_3\lambda^2x_4+2h_3\lambda^2x_5+2h_3\mu_1x_4x_9-2h_3\mu_1x_5x_9$$
$$+4h_3\mu_1x_6x_8-\lambda^2x_1^2-2\lambda^2x_1x_2-4\lambda^2x_1x_3-\lambda^2x_2^2-2\lambda^2x_2x_3-4\mu_1x_1x_3x_9$$
$$+2\mu_1x_1x_5x_8-2\mu_1x_1x_6x_7-2\mu_1x_1x_7^2+2\mu_1x_2x_3x_9-2\mu_1x_2x_4x_8+2\mu_1x_2x_6^2$$
$$-2\mu_1x_2x_6x_7-4\mu_1x_3x_4x_8+2\mu_1x_4^2x_7-2\mu_1x_4x_5x_6+2\mu_1x_4x_5x_7-2h_1^2x_7$$
$$+2h_1h_2x_6+2h_1h_3x_6+2h_1x_1x_5-2h_1x_2x_4-2h_1x_3x_4+2h_1h_3x_6$$
$$-2h_2x_3x_4-2h_3x_1x_4-2h_3x_1x_5+2\lambda^2x_6+2\lambda^2x_7+2\mu_1x_6x_9-2\mu_1x_7x_9$$
$$+2\mu_1x_8^2+2x_1^2x_3+2x_1x_2x_3+2h_1x_8+2h_2x_8-2x_1x_6-2x_1x_7-2x_2x_6$$
$$+x_4^2+2x_4x_5-2x_9)$$

We obtain the following functions:

$$I_{1,[0]}=\frac{2}{3}x_2+\frac{1}{3}h_2^2+\frac{1}{3}h_3^2+\frac{2}{3}x_3+\frac{1}{3}h_1^2+\frac{2}{3}x_1$$
$$\begin{aligned}I_{2,[0]}=&\frac{1}{3}h_1^2h_2^2+\frac{1}{3}h_1^2h_3^2+\frac{2}{3}h_2^2h_3^2+\frac{2}{3}h_1^2x_2-\frac{2}{3}h_1h_2x_1\\&+\frac{2}{3}h_2^2x_3-\frac{2}{3}h_2h_3x_2+\frac{2}{3}x_1h_3^2-\frac{2}{3}h_1x_4-\frac{2}{3}x_4h_2\\&-\frac{2}{3}h_2x_5-\frac{2}{3}h_3x_4-\frac{2}{3}h_3x_5+\frac{1}{3}x_1^2+\frac{2}{3}x_2x_1+\frac{4}{3}x_1x_3\\&+\frac{1}{3}x_2^2+\frac{2}{3}x_2x_3-\frac{2}{3}x_6-\frac{2}{3}x_7\end{aligned}$$
$$\begin{aligned}I_{2,[1]}=&\frac{1}{2x_9}(h_1h_2x_9+h_1x_2x_8+h_1x_4x_7-h_1x_5x_6-h_2x_2x_8-h_2x_4x_7\\&+h_2x_5x_6+h_3^2x_9-h_3x_2x_8+h_3x_4x_7+2x_1x_2x_7-x_2x_5^2x_2x_3x_6\\&-x_3x_4x_5-x_1x_9+x_2x_9+2x_3x_9-2x_4x_8-x_5x_8+x_6^2+x_6x_7)\end{aligned}$$
$$\begin{aligned}I_{3,[0]}=&h_1^2h_2^2h_3^2+2h_1^2h_2^2x_3-2h_1^2h_2h_3x_2-2h_1h_2h_3^2x_1-2h_1^2h_2x_5\\&-2h_1^2h_3x_5+h_1^2x_2^2+2h_1^2x_2x_3+2h_1h_2h_3x_4-4h_1h_2x_1x_32h_1h_3x_1x_2\\&+h_3^2x_1^2-2h_1^2x_6+2h_1h_2x_6+2h_1h_3x_6+2h_1x_1x_5-2h_1x_2x_4\\&-2h_1x_3x_4+2h_2h_3x_6-2h_2x_3x_4-2h_3x_1x_4-2h_2x_1x_5+2x_1^2x_3\\&+2x_1x_2x_3+2h_1x_8+2h_2x_8-2x_1x_6-2x_1x_7-2x_2x_6+x_4^2+2x_4x_5-2x_9\end{aligned}$$

$$
\begin{aligned}
I_{3,[1]} = \frac{1}{x_9}(&h_1h_2h_3^2x_9 - h_1h_2h_3x_2x_8 + h_1h_2h_3x_4x_7 + h_1h_2x_2x_3x_6 - h_1h_2x_3x_4x_5 \\
&- h_{!}h_3^2x_5x_6 + h_1h_3x_2x_3x_6 + h_1h_3x_3x_4x_5 - h_1x_2x_3^2x_4 + h_2h_3^2x_5x_6 \\
&- h_2h_3x_2x_3x_6 - h_2h_3x_3x_4x_5 + h_2x_2x_3^2x_4 - h_3^2x_1x_5^2 + 2h_3x_1x_2x_3x_5 \\
&- x_1x_2^2x_3^2 + 2h_1h_2x_3x_9 - h_1h_2x_5x_8 + h_1h_2x_6x_7 - h_1h_3x_2x_9 \\
&- h_1h_3x_5x_8 - h_1h_3x_6x_7 + h_1x_2^2x_8 + h_1x_2x_3x_8 - h_1x_2x_4x_7 - h_1x_2x_5x_6 \\
&h_1x_3x_4x_7 + h_1x_4x_5^2 + h_2h_3x_5x_8 + h_2h_3x_6x_7 - h_2x_2x_3x_8 - h_2x_3x_4x_7 \\
&- h_3^2x_1x_9 + h_3^2x_6^2 + h_3x_1x_2x_8 - h_3x_1x_4x_7 - 2h_3x_1x_5x_7 - h_3x_2x_5x_6 \\
&- 2h_3x_3x_4x_6 + h_3x_4x_5^2 - x_1x_2x_3x_6 + 2x_1x_2x_4x_7 + x_1x_3x_4x_5 + x_2^2x_3x_6 \\
&- x_2x_3x_4x_5 + x_3^2x_4^2 - h_1x_5x_9 - h_1x_7x_8 + h_2x_7x_8 + h_3x_4x_9 - h_3x_5x_9 \\
&+ 2h_3x_6x_8 - 2x_1x_3x_9 + x_1x_5x_8 - x_1x_6x_7 - x_1x_7^2 + x_2x_3x_9 - x_2x_4x_8 \\
&+ x_2x_6^2 - x_2x_6x_7 - 2x_3x_4x_8 + x_4^2x_7 - x_4x_5x_6 + x_4x_5x_7 + x_6x_9 - x_7x_9 + x_8^2) \\
I_{3,[2]} = \frac{1}{x_9^2}(&h_3^2x_9^2 - 2h_3x_2x_8x_9 + 2h_3x_4x_7x_9 + x_2^2x_8^2 + 2x_3x_3x_6x_9 - 2x_2x_4x_7x_8 \\
&- 2x_2x_5x_6x_8 + 2x_2x_6^2x_7 - 2x_3x_4x_5x_9 + x_4^2x_7^2 + 2x_4x_5^2x_8 - 2x_4x_5x_6x_7 \\
&+ 2x_3x_9^2 - 2x_5x_8x_9 + 2x_6x_7x_9)
\end{aligned}
$$

There are no Casimirs in this case, so the RemoveCasimirs function returns the list unchanged. We have $\frac{3(3+1)}{2} = 6$ functions, as expected.

We will forgo an example from Lie algebras of type D as even in the case of D_4 several of the integrals of motion we obtain require several pages of space to write down. Suffice it to say that explicit construction of first integrals for the Toda flow on a simple classical Lie algebra of higher rank is finally possible with this method.

Chapter 5
Conclusion

The Toda flow provides a phenomenal framework for the introduction and study of truly fascinating and beautiful mathematics. We have seen how an interweaving of Lie theory and Hamiltonian systems results in a rich mathematical framework, and how linear algebra provides some powerful tools for the study of problems in this framework. We have provided usable programming code which can be reworked in any language that allows for abstract computation and possesses a library of linear algebraic functions. Examples have shown how these tools work together to construct the desired integrals of motion.

Despite all of the things we have done, we mentioned little of the challenges faced in working with the Toda flow.

5.1 The Challenge of the Toda Flow

The most obvious challenge, at least to the mind of the uninitiated, is the depth and breadth of prerequisite material required to study the Toda flow. What is less obvious is that this challenge is exacerbated by the complete lack of material intended for the non-professional researcher. While the Toda lattice has a fairly large presence in literature intended as an introduction, introductory literature pertaining to the study of the Toda flow is all but non-existent (with a possible exception being this paper). All papers either assumed a working knowledge of the mathematical foundations of classical mechanics, Hamiltonian systems, or algebraic geometry, to name a few. As a student of pure math without experience in any of these areas, this proved a significant challenge.

Despite the difficulty experienced, it would be untrue to claim that no benefit was found as well. The lack of study material required a level of determination and self-reliance never before expected of the author. This resulted in the development of a sense of confidence in the author's ability to learn new and challenging things independently. Furthermore, the author now understands both the benefits and challenges of independent research; that freedom requires self-discipline.

Regardless, there are other areas which could be improved upon, that time will not allow.

5.2 Further Directions for Study

The first, and most obvious addition to this work would be a formal treatment of the abstract explanation of the procedure for each of the different types of Lie algebras studied above. Aside from this, an extension or adaptation of this method to the exceptional simple Lie algebras would be an important next step. The difficulty here is that the matrices for the exceptional Lie algebras get as large as 248×248, resulting in an extremely computationally heavy result. While a supercomputer could certainly obtain the desired results, another possibility is an implementation of the method utilizing parallel processing and distributed computing. One other possible improvement would be to optimize the algorithm and the programming code provided.

It is clear that there are many ways in which this work could be improved and expanded, and time will likely see them actualized.

Bibliography

[1] Bogoyavlensky O. I. On perturbations of the periodic Toda lattices. *Commun. Math. Phys.* **51** (1976), 201-209.

[2] Bloch, A. M.; Gekhtman, M. I. Lie algebraic aspects of the finite nonperiodic Toda flows. *J. Comp. Appl. Math.* **202** (2007), 3-25.

[3] Deift, P.; Li, L. C.; Nanda, T.; Tomei, C. The Toda flow on a generic orbit is integrable. *Comm. Pure Appl. Math.* **39** (1989), no. 4, 443-521.

[4] Erdmann, K.; Wildon, M. J. *Introduction to Lie Algebras.* London: Springer, (2006).

[5] Flaschka, H. The Toda lattice I. Existence of integrals. *Phys. Rev.* **B9** (1974), 1924-1925.

[6] Flaschka, H. The Toda lattice II. Inverse scattering solution. *Progr. Theor. Phys.* **51** (1974), 703-716.

[7] Fulton, W.; Harris, J. *Representation Theory: A First Course.* New York: Springer-Verlag, (1991).

[8] Gekhtman, M. I.; Shapiro, M. Z. Noncommutative and Commutative Integrability of Generic Toda Flows in Simple Lie Algebras *Comm. Pure Appl. Math.* **52** (1999) no. 1, 53-84.

[9] Kostant, B. The principal three-dimensional subgroup and the Betti numbers of a complex simple Lie group. *Amer. J. Math.* **81** (1959), 973-1032.

[10] Kostant, B. The solution to a generalized Toda lattice and representation theory. *Adv. in Math.* **34** (1979), no. 3, 195-338.

[11] Lax, P. Integrals of nonlinear equations of evolution and solitary waves. *Comm. Pure Appl. Math.* **21** (1968), no. 5, 467-490.

[12] Perelomov, A. M. *Integrable Systems of Classical Mechanics and Lie Algebras.* Basel: Birckhäuser Verlag, (1990).

[13] Shipman, B.A. The Geometry of the full Kostant-Toda lattice of $sl(4, \mathbb{C})$. *J. Geom. Phys.* **33** (2000), 295-325.

[14] Tapp, K. *Matrix Groups for Undergraduates.* Providence, R.I.: American Mathematical Society, (2005).

[15] Toda, M. Vibration of a chain with a non-linear interaction. *J. Phys. Soc. Japan.* **22** (1967), 431-436.

[16] Yamaguchi, K. Differential Systems Associated with Simple Graded Lie Algebras. *Adv. Stu. Pure Math.* **22** (1993), 413-494.