NORTH-HOLLAND
MATHEMATICAL LIBRARY

# Lie Algebras: Theory and Algorithms

W.A. DE GRAAF

North-Holland

# Lie Algebras
# Theory and Algorithms

Willem A. de Graaf
*University of St Andrews*
*Scotland*

N·H

2000
ELSEVIER
Amsterdam – Lausanne – New York – Oxford – Shannon – Singapore – Tokyo

ELSEVIER SCIENCE B.V.
Sara Burgerhartstraat 25
P.O. Box 211, 1000 AE Amsterdam, The Netherlands



First edition 2000

Library of Congress Cataloging in Publication Data
A catalog record from the Library of Congress has been applied for.

ISBN: 0 444 50116 9

⊗The paper used in this publication meets the requirements of ANSI/NISO Z39.48-1992 (Permanence of Paper).

Printed in The Netherlands.

# LIE ALGEBRAS

THEORY AND ALGORITHMS

# North-Holland Mathematical Library



VOLUME 56

ELSEVIER
Amsterdam – Lausanne – New York – Oxford – Shannon – Singapore – Tokyo

# Preface

Lie algebras arise naturally in various areas of mathematics and physics. However, such a Lie algebra is often only known by a presentation such as a multiplication table, a set of generating matrices, or a set of generators and relations. These presentations by themselves do not reveal much of the structure of the Lie algebra. Furthermore, the objects involved (e.g., a multiplication table, a set of generating matrices, an ideal in the free Lie algebra) are often large and complex and it is not easy to see what to do with them. The advent of the computer however, opened up a whole new range of possibilities: it made it possible to work with Lie algebras that are too big to deal with by hand. In the early seventies this moved people to invent and implement algorithms for analyzing the structure of a Lie algebra (see, e.g., [7], [8]). Since then many more algorithms for this purpose have been developed and implemented.

The aim of the present work is two-fold. Firstly it aims at giving an account of many existing algorithms for calculating with finite-dimensional Lie algebras. Secondly, the book provides an introduction into the theory of finite-dimensional Lie algebras. These two subject areas are intimately related. First of all, the algorithmic perspective often invites a different approach to the theoretical material than the one taken in various other monographs (e.g., [42], [48], [77], [86]). Indeed, on various occasions the knowledge of certain algorithms allows us to obtain a straightforward proof of theoretical results (we mention the proof of the Poincaré-Birkhoff-Witt theorem and the proof of Iwasawa's theorem as examples). Also proofs that contain algorithmic constructions are explicitly formulated as algorithms (an example is the isomorphism theorem for semisimple Lie algebras that constructs an isomorphism in case it exists). Secondly, the algorithms can be used to arrive at a better understanding of the theory. Performing the algorithms in concrete examples, calculating with the concepts involved, really brings the theory to life.

The book is roughly organized as follows. Chapter 1 contains a general introduction into the theory of Lie algebras. Many definitions are given that are needed in the rest of the book. Then in Chapters 2 to 5 we explore the structure of Lie algebras. The subject of Chapter 2 is the structure of nilpotent and solvable Lie algebras. Chapter 3 is devoted to Cartan subalgebras. These are immensely powerful tools for investigating the structure of semisimple Lie algebras, which is the subject of Chapters 4 and 5 (which culminate in the classification of the semisimple Lie algebras). Then in Chapter 6 we turn our attention towards universal enveloping algebras. These are of paramount importance in the representation theory of Lie algebras. In Chapter 7 we deal with finite presentations of Lie algebras, which form a very concise way of presenting an often high dimensional Lie algebra. Finally Chapter 8 is devoted to the representation theory of semisimple Lie algebras. Again Cartan subalgebras play a pivotal role, and help to determine the structure of a finite-dimensional module over a semisimple Lie algebra completely. At the end there is an appendix on associative algebras, that contains several facts on associative algebras that are needed in the book.

Along with the theory numerous algorithms are described for calculating with the theoretical concepts. First in Chapter 1 we discuss how to present a Lie algebra on a computer. Of the algorithms that are subsequently given we mention the algorithm for computing a direct sum decomposition of a Lie algebra, algorithms for calculating the nil- and solvable radicals, for calculating a Cartan subalgebra, for calculating a Levi subalgebra, for constructing the simple Lie algebras (in Chapter 5 this is done by directly giving a multiplication table, in Chapter 7 by giving a finite presentation), for calculating Gröbner bases in several settings (in a universal enveloping algebra, and in a free Lie algebra), for calculating a multiplication table of a finitely presented Lie algebra, and several algorithms for calculating combinatorial data concerning representations of semisimple Lie algebras. In Appendix A we briefly discuss several algorithms for associative algebras.

Every chapter ends with a section entitled "Notes", that aims at giving references to places in the literature that are of relevance to the particular chapter. This mainly concerns the algorithms described, and not so much the theoretical results, as there are standard references available for them (e.g., [42], [48], [77], [86]).

I have not carried out any complexity analyses of the algorithms described in this book. The complexity of an algorithm is a function giving an estimate of the number of "primitive operations" (e.g., arithmetical operations) carried out by the algorithm in terms of the size of the input. Now the size of a Lie algebra given by a multiplication table is the sum of the sizes of its structure constants. However, the number of steps performed by an algorithm that operates on a Lie algebra very often depends not only on the size of the input, but also (rather heavily) on certain structural properties of the input Lie algebra (e.g., the length of its derived series). Of course, it is possible to consider only the worst case, i.e., Lie algebras having a structure that poses most difficulties for the algorithm. However, for most algorithms it is far from clear what the worst case is. Secondly, from a practical viewpoint worst case analyses are not very useful since in practice one only very rarely encounters the worst case.

Of the algorithms discussed in this book many have been implemented inside several computer algebra systems. Of the systems that support Lie algebras we mention GAP4 ([31]), LiE ([21]) and Magma ([22]). We refer to the manual of each system for an account of the functions that it contains.

I would like to thank everyone who, directly or indirectly, helped me write this book. In particular I am grateful to Arjeh Cohen, without whose support this book never would have been written, as it was his idea to write it in the first place. I am also grateful to Gábor Ivanyos for his valuable remarks on the appendix. Also I gratefully acknowledge the support of the Dutch Technology Foundation (STW) who financed part of my research.

Willem de Graaf

# Contents

# Chapter 1

# Basic constructions

This chapter serves two purposes. First of all it provides an introduction into the theory of Lie algebras. In the first four sections we define what a Lie algebra is, and we give a number of examples of Lie algebras. In Section 1.5 we discuss some generalities concerning algorithms. Furthermore, we describe our approach to calculating with Lie algebras. We describe how to represent a Lie algebra on a computer (namely by an array of structure constants), and we give two examples of algorithms. In subsequent sections we give several constructions of Lie algebras and objects related to them. In many cases these constructions are accompanied by a an algorithm that performs the construction.

A second purpose of this chapter is to serve as reference for later chapters. This chapter contains most basic constructions used in this book. Therefore it has the nature of a collection of sections, sometimes without clear line of thought connecting them.

## 1.1 Algebras: associative and Lie

**Definition 1.1.1** *An* algebra *is a vector space $A$ over a field $F$ together with a bilinear map $m : A \times A \to A$.*

The bilinear map $m$ of Definition 1.1.1 is called a *multiplication*. If $A$ is an algebra and $x, y \in A$, then we usually write $xy$ instead of $m(x, y)$.

Because an algebra $A$ is a vector space, we can consider subspaces of $A$. A subspace $B \subset A$ is called a *subalgebra* if $xy \in B$ for all $x, y \in B$. It is called an *ideal* if $xy$ and $yx$ lie in $B$ for all $x \in A$ and $y \in B$. Clearly an ideal is also a subalgebra.

Let $A$ and $B$ be two algebras over the field $F$. A linear map $\theta : A \to B$

is called a *morphism of algebras* if $\theta(xy) = \theta(x)\theta(y)$ for all $x, y \in A$ (where the product on the left hand side is taken in $A$ and the product on the right hand side in $B$). The map $\theta$ is an *isomorphism of algebras* if $\theta$ is bijective.

**Definition 1.1.2** *An algebra $A$ is said to be* associative *if for all elements $x, y, z \in A$ we have*

$$(xy)z = x(yz) \text{ (associative law).}$$

**Definition 1.1.3** *An algebra $L$ is said to be a* Lie algebra *if its multiplication has the following properties:*

*($L_1$) $xx = 0$ for all $x \in L$,*

*($L_2$) $x(yz) + y(zx) + z(xy) = 0$ for all $x, y, z \in L$ (Jacobi identity).*

Let $L$ be a Lie algebra and let $x, y \in L$. Then $0 = (x+y)(x+y) = xx + xy + yx + yy = xy + yx$. So condition ($L_1$) implies

$$xy = -yx \text{ for all } x, y \in L. \tag{1.1}$$

On the other hand (1.1) implies $xx = -xx$, or $2xx = 0$ for all $x \in L$. The conclusion is that if the characteristic of the ground field is not 2, then ($L_1$) is equivalent to (1.1). Using (1.1) we see that the Jacobi identity is equivalent to $(xy)z + (yz)x + (zx)y = 0$ for all $x, y, z \in L$.

**Example 1.1.4** Let $V$ be an $n$-dimensional vector space over the field $F$. Here we consider the vector space $\mathrm{End}(V)$ of all linear maps from $V$ to $V$. If $a, b \in \mathrm{End}(V)$ then their product is defined by

$$ab(v) = a(b(v)) \text{ for all } v \in V.$$

This multiplication makes $\mathrm{End}(V)$ into an associative algebra.

For $a, b \in \mathrm{End}(V)$ we set $[a, b] = ab - ba$ The bilinear map $(a, b) \to [a, b]$ is called the *commutator*, or *Lie bracket*. We verify the requirements ($L_1$) and ($L_2$) for the Lie bracket. First we have $[a, a] = aa - aa = 0$ so that ($L_1$) is satisfied. Secondly,

$$\begin{aligned}[a,[b,c]] + [b,[c,a]] + [c,[a,b]] = \\ &a(bc - cb) - (bc - cb)a + b(ca - ac) \\ &- (ca - ac)b + c(ab - ba) - (ab - ba)c = 0.\end{aligned}$$

Hence also ($L_2$) holds for the commutator. It follows that the space of linear maps from $V$ to $V$ together with the commutator is a Lie algebra. We denote it by $\mathfrak{gl}(V)$.

Now fix a basis $\{v_1, \dots, v_n\}$ of $V$. Relative to this basis every linear transformation can be represented by a matrix. Let $M_n(F)$ be the vector space of all $n \times n$ matrices over $F$. The usual matrix multiplication makes $M_n(F)$ into an associative algebra. It is isomorphic to the algebra $\text{End}(V)$, the isomorphism being the map that sends a linear transformation to its matrix with respect to the fixed basis. Analogously we let $\mathfrak{gl}_n(F)$ be the Lie algebra of all $n \times n$ matrices with coefficients in $F$. It is equipped with the product $(a, b) \to [a, b] = ab - ba$ for $a, b \in \mathfrak{gl}_n(F)$. The map that sends a linear transformation to its matrix relative to a fixed basis is an isomorphism of $\mathfrak{gl}(V)$ onto $\mathfrak{gl}_n(F)$.

Let $A$ be an algebra and let $B \subset A$ be a subalgebra. Then $B$ is an algebra in its own right, inheriting the multiplication from its "parent" $A$. Furthermore, if $A$ is a Lie algebra then clearly $B$ is also a Lie algebra, and likewise if $A$ is associative. If $B$ happens to be an ideal, then, by the following proposition, we can give the quotient space $A/B$ an algebra structure. The algebra $A/B$ is called the *quotient algebra* of $A$ and $B$.

**Proposition 1.1.5** *Let $A$ be an algebra and let $B \subset A$ be an ideal. Let $A/B$ denote the quotient space. Then the multiplication on $A$ induces a multiplication on $A/B$ by $\bar{x}\bar{y} = \overline{xy}$ (where $\bar{x}$ denotes the coset of $x \in A$ in $A/B$). Furthermore, if $A$ is a Lie algebra, then so is $A/B$ (and likewise if $A$ is an associative algebra).*

**Proof.** First of all we check that the multiplication on $A/B$ is well defined. So let $x, y \in A$ and $b_1, b_2 \in B$. Then $\bar{x} = \overline{x + b_1}$ and $\bar{y} = \overline{y + b_2}$ and hence

$$\bar{x}\bar{y} = \overline{(x + b_1)}\ \overline{(y + b_2)} = \overline{(x + b_1)(y + b_2)} = \overline{xy + xb_2 + b_1y + b_1b_2} = \overline{xy}.$$

Consequently the product $\bar{x}\bar{y}$ is independent of the particular representatives of $\bar{x}$ and $\bar{y}$ chosen.

The fact that the Lie (respectively associative) structure is carried over to $A/B$ is immediate. □

Associative algebras and Lie algebras are intimately related in the sense that given an associative algebra we can construct a related Lie algebra and the other way round. First let $A$ be an associative algebra. The commutator yields a bilinear operation on $A$, i.e., $[x, y] = xy - yx$ for all $x, y \in A$, where the products on the right are the associative products of $A$. Let $A_{Lie}$ be the underlying vector space of $A$ together with the product [ , ]. It is straightforward to check that $A_{Lie}$ is a Lie algebra (cf. Example 1.1.4). In Chapter 6 we will show that every Lie algebra occurs as a subalgebra of a Lie algebra of the form $A_{Lie}$ where $A$ is an associative algebra. (This is the content of the theorems of Ado and Iwasawa.) For this reason we will use square brackets to denote the product of any Lie algebra.

From a Lie algebra we can construct an associative algebra. Let $L$ be a Lie algebra over the field $F$. For $x \in L$ we define a linear map

$$\mathrm{ad}_L x : L \longrightarrow L$$

by $\mathrm{ad}_L x(y) = [x, y]$ for $y \in L$. This map is called the *adjoint map* determined by $x$. If there can be no confusion about the Lie algebra to which $x$ belongs, we also write $\mathrm{ad}x$ in place of $\mathrm{ad}_L x$. We consider the subalgebra of $\mathrm{End}(L)$ generated by the identity mapping together with $\{\mathrm{ad}x \mid x \in L\}$ (i.e., the smallest subalgebra of $\mathrm{End}(L)$ containing 1 and this set). This associative algebra is denoted by $(\mathrm{ad}L)^*$.

Since $\mathrm{ad}x$ is the left multiplication by $x$, the adjoint map encodes parts of the multiplicative structure of $L$. We will often study the structure of a Lie algebra $L$ by investigating its adjoint map. This will allow us to use the tools of linear algebra (matrices, eigenspaces and so on). Furthermore, as will be seen, the associative algebra $(\mathrm{ad}L)^*$ can be used to obtain valuable information about the structure of $L$ (see, e.g., Section 2.2).

## 1.2 Linear Lie algebras

In Example 1.1.4 we encountered the Lie algebra $\mathfrak{gl}_n(F)$ consisting of all $n \times n$ matrices over the field $F$. By $E_{ij}^n$ we will denote the $n \times n$ matrix with a 1 on position $(i, j)$ and zeros elsewhere. If it is clear from the context which $n$ we mean, then we will often omit it and write $E_{ij}$ in place of $E_{ij}^n$. So a basis of $\mathfrak{gl}_n(F)$ is formed by all $E_{ij}$ for $1 \leq i, j \leq n$.

Subalgebras of $\mathfrak{gl}_n(F)$ are called *linear Lie algebras*. In this section we construct several linear Lie algebras.

**Example 1.2.1** For a matrix $a$ let $\mathrm{Tr}(a)$ denote the trace of $a$. Let $a, b \in \mathfrak{gl}_n(F)$, then

$$\mathrm{Tr}([a, b]) = \mathrm{Tr}(ab - ba) = \mathrm{Tr}(ab) - \mathrm{Tr}(ba) = \mathrm{Tr}(ab) - \mathrm{Tr}(ab) = 0. \quad (1.2)$$

Set $\mathfrak{sl}_n(F) = \{a \in \mathfrak{gl}_n(F) \mid \mathrm{Tr}(a) = 0\}$, then, since the trace is a linear function, $\mathfrak{sl}_n(F)$ is a linear subspace of $\mathfrak{gl}_n(F)$. Moreover, by (1.2) we see that $[a, b] \in \mathfrak{sl}_n(F)$ if $a, b \in \mathfrak{sl}_n(F)$. Hence $\mathfrak{sl}_n(F)$ is a subalgebra of $\mathfrak{gl}_n(F)$. It is called the *special linear Lie algebra*. The Lie algebra $\mathfrak{sl}_n(F)$ is spanned by all $E_{ij}$ for $i \neq j$ together with the diagonal matrices $E_{ii} - E_{i+1,i+1}$ for $1 \leq i \leq n-1$. Hence the dimension of $\mathfrak{sl}_n(F)$ is $n^2 - 1$.

Let $V$ be an $n$-dimensional vector space over $F$. We recall that a *bilinear form* $f$ on $V$ is a bilinear function $f : V \times V \to F$. It is *symmetric* if $f(v, w) = f(w, v)$ and *skew symmetric* if $f(v, w) = -f(w, v)$ for all $v, w \in V$. Furthermore, $f$ is said to be *non-degenerate* if $f(v, w) = 0$ for all $w \in V$ implies $v = 0$. For a bilinear form $f$ on $V$ we set

$$L_f = \{a \in \mathfrak{gl}(V) \mid f(av, w) = -f(v, aw) \text{ for all } v, w \in V\}, \tag{1.3}$$

which is a linear subspace of $\mathfrak{gl}(V)$.

**Lemma 1.2.2** *Let $f$ be a bilinear form on the $n$-dimensional vector space $V$. Then $L_f$ is a subalgebra of $\mathfrak{gl}(V)$.*

**Proof.** For $a, b \in L_f$ we calculate

$$\begin{aligned} f([a,b]v, w) &= f((ab - ba)v, w) = f(abv, w) - f(bav, w) \\ &= -f(bv, aw) + f(av, bw) = f(v, baw) - f(v, abw) = -f(v, [a,b]w). \end{aligned}$$

So for each pair of elements $a, b \in L_f$ we have that $[a, b] \in L_f$ and hence $L_f$ is a subalgebra of $\mathfrak{gl}(V)$. □

Now we fix a basis $\{v_1, \ldots, v_n\}$ of $V$. This allows us to identify $V$ with the vector space $F^n$ of vectors of length $n$. Also, as pointed out in Example 1.1.4, we can identify $\mathfrak{gl}(V)$ with $\mathfrak{gl}_n(F)$, the Lie algebra of all $n \times n$ matrices over $F$. We show how to identify $L_f$ as a subalgebra of $\mathfrak{gl}_n(F)$. Let $M_f$ be the $n \times n$ matrix with on position $(i, j)$ the element $f(v_i, v_j)$. Then a straightforward calculation shows that $f(v, w) = v^t M_f w$ (where $v^t$ denotes the transpose of $v$). The condition for $a \in \mathfrak{gl}_n(F)$ to be an element of $L_f$ translates to $v^t a^t M_f w = -v^t M_f a w$ which must hold for all $v, w \in F^n$. It follows that $a \in L_f$ if and only if $a^t M_f = -M_f a$.

The next three examples are all of the form of $L_f$ for some non-degenerate bilinear form $f$.

**Example 1.2.3** Let $f$ be a non-degenerate skew symmetric bilinear form with matrix

$$M_f = \begin{pmatrix} 0 & I_l \\ -I_l & 0 \end{pmatrix},$$