AAECC 8, 291–303 (1997)

AAECC
Applicable Algebra in
Engineering, Communication
and Computing
© Springer-Verlag 1997

Computing Levi Decompositions in Lie algebras

W. A. de Graaf[1], **G. Ivanyos**[2], **A. Küronya**[3], **L. Rónyai**[4]

[1] Department of Mathematics and Computing Science, Technical University of Eindhoven, P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands (e-mail: wdg@win.tue.nl)
[2] Computer and Automation Institute, Hungarian Academy of Sciences, 1111 Budapest, Lágymányosi u. 11, Hungary (e-mail: ig@ilab.sztaki.hu)
[3] Department of Mathematics and Computing Sciences, Faculty of Electrical Engineering and Informatics, Technical University of Budapest, Budapest, Hungary
(e-mail: alex@vma.bme.hu)
[4] Computer and Automation Institute, Hungarian, Academy of Sciences, 1111 Budapest, Lágymányosi u. 11, Hungary (e-mail: lajos@nyest.ilab.sztaki.hu)

Received: October 24, 1995; revised version: May 28, 1996

Abstract. We consider the algorithmic problem of computing Levi decompositions in Lie algebras and Wedderburn–Malcev decompositions in associative algebras over the field of rational numbers. We propose deterministic polynomial time algorithms for both problems. The methods are based on the corresponding classical existence theorems. Computational experiences are discussed at the end of the paper.

Keywords: Lie algebras, Associative algebras, Radical, Decomposition, Symbolic computations, Complexity.

1 Introduction

Due to abounding applications in physics and mathematics, there is a considerable interest in Lie algebra computations. This relatively new field of research is growing rapidly as there is a marked need for efficient Lie algebra algorithms. Yet, in many cases little is known about the algorithmic complexity of the emerging problems and mostly only experimental observations are available on the performance of the methods used.

One of the major motivations is to describe the isomorphism classes of finite dimensional Lie algebras of small dimension. This task is complete at this moment only in the case when the dimension is less than seven in general (see [10]), and up to dimension seven for nilpotent algebras ([14]).

Research supported in part by Hungarian Foundation for Scientific Research grants T016503 and T014546 and by the Dutch Technology Foundation STW

To achieve further advances and reveal more of the structure of Lie algebras, we need algorithms that can be implemented efficiently. In our past work on the subject (cf. [11], [5]) we have given algorithms to compute Cartan subalgebras and the solvable- and nilradicals. The methods run in polynomial time over ground fields admitting efficient symbolic arithmetic. The algorithms are implemented in a general library of Lie algebra algorithms, called ELIAS (for Eindhoven LIe Algebra System) which is built into the computer algebra package GAP. These activities are part of a bigger project, called ACELA, which aims at an interactive book on Lie algebras (cf. [3]).

In this paper we show that a Levi decomposition for Lie algebras over the rational numbers **Q** can be obtained in polynomial time. As a related result we solve the analogous problem of computing Wedderburn–Malcev decompositions in associative algebras. The theorems by Levi and Wedderburn–Malcev that serve as a theoretical basis for the algorithms are important instances of the so-called 'lifting theorems'. They establish the existence of a lifting of a certain substructure from a factor algebra into the original one.

We perform exact (symbolic) computations, therefore we restrict our attention to ground fields admitting efficient symbolic arithmetic: in most cases we work over **Q**, but occasionally some other fields (such as algebraic number fields and finite fields) are also considered.

First we set some conventions about the input of the algorithmic questions we address. An algebra $\mathscr{A}$ (associative or Lie in the paper) over a field $\mathscr{F}$ is considered to be given as a set of structure constants relative to a fixed linear basis over the ground field (this is called the input basis). If $u_1, u_2, \ldots, u_n$ is the input basis, then for any $1 \leqq i, j \leqq n$ we have

$$u_i \cdot u_j = \sum_{k=1}^{n} c_{ij}^{(k)} u_k,$$

where $\cdot$ denotes multiplication in $\mathscr{A}$ and the coefficients $c_{ij}^{(k)} \in \mathscr{F}$ are the structure constants.

We would like to have algorithms with performance guarantees. In computer science terminology, we are primarily interested in algorithms that run in time polynomial in terms of the input size (cf. [6], [12]). For our purposes it suffices to observe that efficient deterministic methods exist for the arithmetical operations and the basic linear algebraic tasks over the fields $\mathscr{F}$ we consider. In most cases $\mathscr{F}$ will be the field of rational numbers **Q**.

Let $\mathscr{A}$ be a finite dimensional algebra over **Q**, and let $u_1, u_2, \ldots, u_n$ be a fixed basis of $\mathscr{A}$, where n denotes the dimension of $\mathscr{A}$ over **Q**. We denote by $c_{ij}^{(k)}$ the structure constants of $\mathscr{A}$ with respect to the basis $\{u_i\}$. Every element $v \in \mathscr{A}$ can be written uniquely in the form $v = \sum_{i=1}^{n} \alpha_i u_i$. This way we identify the **Q**-spaces $\mathscr{A}$ and $\mathbf{Q}^n$: $(\alpha_1, \ldots, \alpha_n) \in \mathbf{Q}^n$ will correspond to $v \in \mathscr{A}$.

We can assume without loss of generality that the structure constants $c_{ij}^{(k)}$ are rational integers (if not, then we multiply the basis elements with the least common denominator of the structure constants; this ensures that the new structure constants will all be integers). We put $c = \max_{i,j,k} |c_{ij}^{(k)}|$. Via the identification above we introduce two quantities to measure the size of an element $v \in \mathscr{A}$. We put

$$M(v) = \max_i |\alpha_i|$$

and

$$D(v) = \text{the least common multiple of the denominators of the } \alpha_i\text{'s},$$

where the α_i's are the coordinates of v in the basis $\{u_i\}$. We extend this measure of complexity to subspaces of $\mathbf{Q}^n$ (or $\mathscr{A}$) as follows: If a subspace $\mathscr{W} \subseteq \mathbf{Q}^n$ is given by a basis $x_1, x_2, \ldots, x_k$, with $k = \dim_{\mathbf{Q}} \mathscr{W}$, then we set

$$M(\mathscr{W}) = \max_i M(x_i)$$

$$D(\mathscr{W}) = \prod_i D(x_i).$$

Note that this is a slight abuse of language because the quantities M and D depend on the basis $x_1, x_2, \ldots, x_k$ rather than just the subspace $\mathscr{W}$. As the basis we work with will always be clear from the context, this causes no ambiguity. In this setting a statement saying that a substructure $\mathscr{W} \subseteq \mathscr{A}$ is small means that we can efficiently compute a basis of $\mathscr{W}$ which yields small values of $M(\mathscr{W})$ and $D(\mathscr{W})$.

In complexity theory it is customary to measure the size of an object in the number of bits used for a standard description of the object. To clarify the relation between the bit-size above and our measures M and D, we note that an element $x \in \mathscr{A}$ has bit-size $O(n(1 + \log\lceil M(x)\rceil + \log\lceil D(x)\rceil))$. Similarly, a subspace $\mathscr{K}$ can be described in $O(n \cdot \dim \mathscr{K}(1 + \log M(\mathscr{K})) + \log D(\mathscr{K}))$ bits.

When working with a Lie algebra $\mathscr{L}$, it is often important to understand the way it is built up from the solvable radical and the semisimple part. A beautiful theorem by Levi states that the latter can actually be identified as a subalgebra of $\mathscr{L}$:

Theorem 1.1 (Levi) *Let $\mathscr{L}$ be a finite dimensional Lie algebra over a field of characteristic zero, $\mathscr{R}$ be its solvable radical. Then there exists a semisimple subalgebra $\mathscr{S}$ of $\mathscr{L}$ for which $\mathscr{L} = \mathscr{R} \oplus \mathscr{S}$. In particular, $\mathscr{S}$ is isomorphic to $\mathscr{L}/\mathscr{R}$.*

The semisimple complement $\mathscr{S}$ is not unique, but it is determined in a very strong sense: according to the Malcev–Harish–Chandra theorem, if $\mathscr{S}_1$ and $\mathscr{S}_2$ are two Levi-complements, then there exists an automorphism α of $\mathscr{L}$ for which $\alpha(\mathscr{S}_1) = \mathscr{S}_2$ (cf. [8]).

There is a decomposition theorem for associative algebras which is very similar in nature. This is due to Wedderburn and Malcev and is a significant result in the theory of associative algebras. We state here a version for finite dimensional algebras. We denote by $\mathscr{R} = Rad(\mathscr{A})$ the Jacobson radical of $\mathscr{A}$. Then the Wedderburn–Malcev theorem reads as follows (see [9]):

Theorem 1.2 *Let $\mathscr{A}$ be a finite dimensional associative $\mathscr{F}$-algebra for which $\mathscr{A}/Rad(\mathscr{A})$ is separable.*

Then there is a subalgebra $\mathscr{B}$ of $\mathscr{A}$ such that $\mathscr{A} = \mathscr{B} \oplus Rad(\mathscr{A})$ as $\mathscr{F}$-spaces and $\mathscr{B} \cong \mathscr{A}/Rad(\mathscr{A})$ as $\mathscr{F}$-algebras. Moreover, for every two such complements $\mathscr{B}$, $\mathscr{B}'$ there exists an element $x \in Rad(\mathscr{A})$ for which $\mathscr{B}' = (1 - x)^{-1}\mathscr{B}(1 - x)$.

The objective of this paper is to give deterministic polynomial time algorithms for finding decompositions whose existence is stated in the preceding theorems.

More precisely we give algorithms for following two problems:

- Computing a Levi decomposition of a Lie algebra over **Q**;
- Determining a Wedderburn–Malcev decomposition for an associative algebra over a finite field or over **Q**.

Our algorithms rely heavily on the corresponding existence theorems above. The lifting of the semisimple factor will be carried out along a suitable chain of ideals in the radical.

In the second section of the paper we give an algorithm for computing a Levi decomposition in a Lie algebra. The third section describes the algorithm that produces a Wedderburn–Malcev decomposition in an associative algebra. In the last section we discuss computational experiences.

2 Computing a Levi Decomposition

2.1 The Algorithm

We describe here a polynomial time algorithm that finds a Levi decomposition in a finite dimensional Lie algebra $\mathscr{L}$ given by structure constants over **Q**. We note here that the method can be extended very easily to the case when the ground field $\mathscr{F}$ is an algebraic number field. Indeed, $\mathscr{L}$ can be considered as an algebra over **Q**, and a Levi decomposition over **Q** will serve as a Levi decomposition over $\mathscr{F}$. The algorithm still runs in polynomial time, provided that $\mathscr{F}$ is given in the usual way by an irreducible polynomial $f \in \mathbf{Z}[x]$ such that $\mathscr{F} = \mathbf{Q}(\alpha)$, where $f(\alpha) = 0$.

Rand, Winternitz and Zassenhaus [10] have given an elegant method to compute Levi decompositions. Their iterative approach, however, does not seem to provide satisfactory bounds for the size of the coefficients obtained during the computation. We propose here a modified and simplified approach; instead of the derived series we work with the lower central series of the radical. This change renders the algorithm simpler and, at the same time, allows us to obtain polynomial bounds on the size of the resulting decomposition as well as the intermediate objects. This way we obtain a deterministic polynomial time algorithm.

We assume that a Lie algebra $\mathscr{L}$ is given as a collection of structure constants $c_{ij}^{(k)} \in \mathbf{Z}$ with respect to the input basis $u_1, u_2, \ldots, u_n$ over **Q**.

We intend to compute a new basis $r_1, \ldots, r_m, v_1, \ldots, v_k$ of $\mathscr{L}$ such that the r_i's form a basis of the radical $\mathscr{R}$ and the elements v_j constitute a basis of a semisimple complement $\mathscr{S}$ in $\mathscr{L}$. In particular, we have $m + k = n$. The first part of this task is easy: one can identify the solvable radical in polynomial time (cf. [11]).

The problem of finding a Levi complement can be reduced to the case when $\mathscr{L}$ possesses a nilpotent radical. This is a consequence of the statement below (see [8], Section 9, Chapter III for a proof).

Lemma 2.1 *Let $\mathscr{S}_1$ be the inverse image of a Levi complement of $\mathscr{L}/Rad(\mathscr{L})^2$ in $\mathscr{L}$, and let a Levi decomposition of $\mathscr{S}_1$ be $\mathscr{S}_1 = Rad(\mathscr{S}_1) \oplus \mathscr{S}$. Then a Levi decomposition of $\mathscr{L}$ is $\mathscr{L} = Rad(\mathscr{L}) \oplus \mathscr{S}$.* □

Indeed, to obtain a Levi complement in $\mathscr{L}$ amounts to finding a Levi complement in $\mathscr{L}/Rad(\mathscr{L})^2$ and then in $\mathscr{S}_1$. Moreover, the solvable radicals of these algebras are clearly nilpotent.

From now on we assume that $\mathscr{R} = Rad(\mathscr{L})$ is a nilpotent ideal in $\mathscr{L}$. As noted before, $\mathscr{R}$ and the lower central series $\mathscr{R} \supseteq \mathscr{R}^2 \supseteq \cdots \supseteq \mathscr{R}^l = 0$ of $\mathscr{R}$ can be computed efficiently.

Our approach to finding a Levi complement in $\mathscr{L}$ is as follows. We start up with the (semisimple) algebra $\mathscr{L}/\mathscr{R}$ and then proceed to obtain a Levi complement $\mathscr{S}_i$ in $\mathscr{L}/\mathscr{R}^i$ for $i = 2, \ldots, l$. The algebra $\mathscr{S}_l$ obtained at the end of this procedure will be a Levi complement in $\mathscr{L}/\mathscr{R}^l = \mathscr{L}$. Our method follows the general pattern of lifting procedures. We refine a basis of $\mathscr{S}_i$ into a basis of $\mathscr{S}_{i+1}$ by working in the factor space $\mathscr{R}^i/\mathscr{R}^{i+1}$.

After possibly renumbering the input basis $u_1, \ldots, u_n$ we can suppose that the images $\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_k$ of $u_1, \ldots, u_k$ under the natural map $\mathscr{L} \to \mathscr{L}/\mathscr{R}$ form a basis of $\mathscr{L}/\mathscr{R}$.

We can also find (bases of) subspaces V_i *in* $\mathscr{R}^i$ complementary to $\mathscr{R}^{i+1}$ (i.e. we have $\mathscr{R}^i = \mathscr{R}^{i+1} \oplus V_i$). We write $\dim V_i = d_i$ and $\dim \mathscr{L}/\mathscr{R}^t = n_t$. We denote by $w_j^{(i)}, j = 1, \ldots, d_i$ the basis of V_i, we work with. To lighten notation, we assume also, that the vectors $w_j^{(i)}$ all belong to the set $\{u_1, u_2, \ldots, u_n\}$. (This can be achieved after possibly computing structure constants with respect to a new basis, a task clearly feasible in polynomial time.)

Now we describe an iteration that stops in $l \leqq n$ rounds (here $n = \dim \mathscr{L}$) which transforms the u_i's into a set of elements $v_1, v_2, \ldots, v_k$ for which $\mathscr{S} = Span(v_1, v_2, \ldots, v_k)$ is a Lie subalgebra of $\mathscr{L}$ isomorphic to $\mathscr{L}/\mathscr{R}$. Then $\mathscr{S}$ is obviously a Levi complement in $\mathscr{L}$.

Let $v_i^{(t)}$ denote the temporary basis obtained in the tth round. Initially we have $v_i^{(1)} = u_i$, and upon termination $v_i = v_i^{(l)}$. As a loop-invariant, we assume that after the tth round we have

$$[v_i^{(t)}, v_j^{(t)}] \equiv \sum_{s=1}^{k} c_{ij}^{(s)} v_s^{(t)} \pmod{\mathscr{R}^t}. \tag{1}$$

Condition (1) is equivalent to saying that the elements $v_i^{(t)}$ span a Levi complement modulo $\mathscr{R}^t$. This is clearly true for $t = 1$.

To proceed from t to $t + 1$ we look for elements $\delta_i^{(t)} \in V_t$ such that if we choose $v_i^{(t+1)} = v_i^{(t)} + \delta_i^{(t)}$ (where $\delta_i^{(t)} = \sum_{s=1}^{d_t} \alpha_i^{(s)} w_s^{(t)}$ with $\alpha_i^{(s)} \in \mathbf{Q}$) then we obtain

$$[v_i^{(t+1)}, v_j^{(t+1)}] \equiv \sum_{s=1}^{k} c_{ij}^{(s)} v_s^{(t+1)} \pmod{\mathscr{R}^{t+1}}. \tag{2}$$

Let us expand this formula a bit:

$$[v_i^{(t)}, \delta_j^{(t)}] + [\delta_i^{(t)}, v_j^{(t)}] + [\delta_i^{(t)}, \delta_j^{(t)}] \equiv$$
$$\sum_{s=1}^{k} c_{ij}^{(s)} v_s^{(t)} + \sum_{s=1}^{k} c_{ij}^{(s)} \delta_s^{(t)} - [v_i^{(t)}, v_j^{(t)}] \pmod{\mathscr{R}^{t+1}}.$$

We observe that

$$[\delta_i^{(t)}, \delta_j^{(t)}] \equiv 0 \pmod{\mathscr{R}^{t+1}}$$

and

$$[v_i^{(t)}, \delta_j^{(t)}] \equiv [u_i, \delta_j^{(t)}] \pmod{\mathscr{R}^{t+1}}$$

which follow immediately from the relations $[\delta_i^{(s)}, \delta_j^{(r)}] \in \mathscr{R}^{t+1}$ for all $t+1 \leqq r+s$ $(1 \leqq r, s)$. We infer that

$$[u_i, \delta_j^{(t)}] + [\delta_i^{(t)}, u_j] - \sum_{s=1}^{k} c_{ij}^{(s)} \delta_s^{(t)} \equiv \sum_{s=1}^{k} c_{ij}^{(s)} v_s^{(t)} - [v_i^{(t)}, v_j^{(t)}] \quad (\mathrm{mod}\, \mathscr{R}^{t+1}). \tag{3}$$

This is a system of linear equations for the unknown elements $\delta_j^{(t)} \in V_t$. More precisely we have equations for the unknown coefficients $\alpha_i^{(s)}$. We have $d_t k$ unknowns and a linear equation for every triplet (i, j, r) such that $1 \leqq i < j \leqq k$ and $1 \leqq r \leqq d_t$. Here r corresponds to the d_t coordinates in V_t. To verify this latter point, we observe that the elements on both sides of (3) are in V_t modulo $\mathscr{R}^{t+1}$. This is immediate by inspection of the left-hand side. For the right hand side the loop-invariant (1) gives the same conclusion. The number of equations is therefore $\binom{k}{2} d_t$. A system with these parameters is in general overdetermined and has no solution. In our case however, Levi's theorem applied for $\mathscr{L}/\mathscr{R}^{t+1}$ ensures that the system has a solution. Conversely, a solution of (3) leads to a solution of (2) and hence to a Levi complement in $\mathscr{L}/\mathscr{R}^{t+1}$. These considerations imply that in round t we can complete the task by solving a system of linear equations, hence the whole iteration can be carried out at the expense of a polynomial number of *arithmetical operations* in $\mathbf{Q}$. To obtain a polynomial time algorithm, we have to establish bounds on the numbers encountered in the course of the computation. This will be done in the next subsection. For later use we record here an equivalent version of (3). By substituting $v_i^{(t)} = u_i + \sum_{p=1}^{t-1} \delta_i^{(p)}$ into (3) we obtain

$$[u_i, \delta_j^{(t)}] + [\delta_i^{(t)}, u_j] - \sum_{s=1}^{k} c_{ij}^{(s)} \delta_s^{(t)} \equiv \sum_{s=1}^{k} c_{ij}^{(s)} \left(u_s + \sum_{r=1}^{t-1} \delta_s^{(r)} \right) - \left[u_i + \sum_{p=1}^{t-1} \delta_i^{(p)}, u_j + \sum_{q=1}^{t-1} \delta_j^{(q)} \right] \quad (\mathrm{mod}\, \mathscr{R}^{t+1}). \tag{4}$$

Remarks.

1. We note here the interesting fact that in round t only the coordinates in V_t impose nontrivial conditions on $\delta_j^{(t)} \in V_t$.
2. Our procedure can be described in a slightly more transparent way. Starting from the Lie algebra $\mathscr{L}/\mathscr{R}$, we are actually looking for a set of representatives $x_1, \ldots, x_k$ of the cosets $\bar{u}_1, \ldots, \bar{u}_k$ of $\mathscr{L}/\mathscr{R}$ (recall that the elements $\bar{u}_i$ are a basis for $\mathscr{L}/\mathscr{R}$) such that $Span(x_1, \ldots, x_k)$ is a subalgebra in $\mathscr{L}$.

More formally we seek a $\mathbf{Q}$-linear injective map $\sigma\colon \mathscr{L}/\mathscr{R} \to \mathscr{L}$ for which

(a) $\overline{\sigma(\bar{x})} = \bar{x}$ for any $x \in \mathscr{L}$, and
(b) the subspace $\sigma(\mathscr{L}/\mathscr{R})$ is a subalgebra.

We start out with σ_0 given by $\sigma_0(\bar{u}_i) = u_i$. This obviously satisfies (a) above. Next we look for a linear correction map $\Delta\colon \mathscr{L}/\mathscr{R} \to \mathscr{R}$ such that $\sigma_0 + \Delta$ maps $\mathscr{L}/\mathscr{R}$ bijectively onto a subalgebra of $\mathscr{L}$ which is then bound to be semisimple and isomorphic to $\mathscr{L}/\mathscr{R}$. The existence of such a correction map (for any choice of coset representatives) is the actual content of Levi's theorem.

In the tth step of the algorithm we construct a correction map $\Delta^{(t)}$ that works $(\mathrm{mod}\, \mathscr{R}^{t+1})$. The procedure stops with a correction map $\Delta = \Delta^{(l)}$ such that

$\sigma = \sigma_0 + \Delta$ satisfies conditions (a) and (b). We can describe the correction map $\Delta^{(t)}$ quite explicitly. If $\bar{x} = \sum_{s=1}^{k} \alpha_s \bar{u}_s$ with $\alpha_s \in \mathbf{Q}$ then

$$\Delta^{(t)}(\bar{x}) = \sum_{l=1}^{t} \sum_{s=1}^{k} \alpha_s \delta_s^{(l)}. \tag{5}$$

3. The algorithm that is given in [10] can be rephrased in our language as follows. Let $\mathcal{R} \supseteq \mathcal{R}_2 \supseteq \cdots \supseteq \mathcal{R}_m = 0$ be the derived series of the radical $\mathcal{R}$. Then as before we have $\mathcal{R}_i = \mathcal{R}_{i+1} \oplus V_i$. In a similar way we construct a sequence of elements $v_i^{(t)}$ such that the $v_i^{(t)}$ span a Levi subalgebra modulo $\mathcal{R}_t$. Then by analogous arguments we arrive at the equations

$$[v_i^{(t)}, \delta_j^{(t)}] + [\delta_i^{(t)}, v_j^{(t)}] + \sum_{s=1}^{k} c_{ij}^{(s)} \delta_s^{(t)} \equiv \sum_{s=1}^{k} c_{ij}^{(s)} v_s^{(t)} - [v_i^{(t)}, v_j^{(t)}] \quad (\mathrm{mod}\, \mathcal{R}_{t+1}).$$

We think that this is a more transparent way of formulating the method. Furthermore it leads to a faster implementation than the formulation in [10].

2.2 Bounding the Size

We have seen that a Levi complement can be obtained by solving at most n systems of linear equations over $\mathbf{Q}$. A system has at most n^2 variables and at most n^3 equations. Such systems can be solved by using a polynomial number of arithmetical operations. To conclude that the algorithm runs in deterministic polynomial time we prove bounds on the coefficients $\alpha_i^{(s)}$. More precisely we show that we can efficiently find a solution of (4) (and hence of (2)) where the numbers $\alpha_i^{(s)}$ have moderate size. The difficulty is that the system we consider in round t depends on the solutions we calculated earlier. We overcome this by a careful analysis of (4).

We write $M_t = M(Span(v_1^{(t)}, v_2^{(t)}, \ldots, v_k^{(t)}))$ and $D_t = D(Span(v_1^{(t)}, v_2^{(t)}, \ldots, v_k^{(t)}))$. We intend to prove that the quantities $\log M_t$ and $\log D_t$ are bounded by a polynomial of the input size n and $\log c$. The following lemma is a direct consequence of Cramer's rule.

Lemma 2.2 *Let* $\mathbf{Ax} = \mathbf{b}$ *be a system of linear equations where* $\mathbf{A}$ *is a* $k \times m$ *matrix with integer entries,* $\mathbf{b}$ *is a rational vector from the column space of* $\mathbf{A}$, *rank*$(\mathbf{A}) = d$, $\alpha = \max_{i,j} |(\mathbf{A})_{ij}|$. *Then we can always find a solution* $\mathbf{x}$ *in polynomial time for which*

$$M(\mathbf{x}) \leqq d! \alpha^d M(\mathbf{b}),$$

and

$$D(\mathbf{x}) \leqq d! \alpha^d D(\mathbf{b}). \qquad \square$$

We intend to apply this to the linear system below (see (4)) which describes a Levi complement in $\mathcal{L}/\mathcal{R}^{t+1}$. Let us consider (4) again:

$$\begin{aligned}[u_i, \delta_j^{(t)}] + [\delta_i^{(t)}, u_j] - \sum_{s=1}^{k} c_{ij}^{(s)} \delta_s^{(t)} &\equiv \sum_{s=1}^{k} c_{ij}^{(s)} \left(u_s + \sum_{r=1}^{t-1} \delta_s^{(r)} \right) \\ &\quad - \left[u_i + \sum_{p=1}^{t-1} \delta_i^{(p)}, u_j + \sum_{q=1}^{t-1} \delta_j^{(q)} \right] \quad (\mathrm{mod}\, \mathcal{R}^{t+1}).\end{aligned} \tag{6}$$

As noted earlier, this is a system of linear equations for the unknown coefficients $\alpha_i^{(s)}$ of $\delta_j^{(t)} \in V_t$. We have $kd_t \leqq n^2$ unknowns and a linear equation for every triplet (i, j, r) such that $1 \leqq i < j \leqq k$ and $1 \leqq r \leqq d_t$. Here r corresponds to the d_t coordinates in V_t. We have thus $\binom{k}{2} d_t \leqq n^3$ equations. Thus, the matrix $\mathbf{A}$ of the system has at most n^3 rows and n^2 columns. In particular we have $\operatorname{rank}(\mathbf{A}) \leqq n^2$.

Next we observe the coefficients of the unknowns in the terms of the left-hand side of (6) are actually structure constants:

$$\alpha = \max_{l,m} |(\mathbf{A})_{lm}| \leqq 3 \max_{i,j,l} |c_{ij}^{(l)}| = 3c. \tag{7}$$

To apply the Lemma, we bound the vector $\mathbf{b}$ on the right-hand side of (6). (We need bounds for the coordinates with respect to the basis $\{u_i\}$). We write this inhomogeneous part into a form that makes it easier to estimate.

$$\sum_{s=1}^{k} c_{ij}^{(s)} u_s + \sum_{s=1}^{k} \sum_{r=1}^{t-1} c_{ij}^{(s)} \delta_s^{(r)} + [u_i, u_j] + \sum_{q=1}^{t-1} [u_i, \delta_j^{(q)}] \\ + \sum_{p=1}^{t-1} [\delta_i^{(p)}, u_j] + \sum_{p=1}^{t-1} \sum_{q=1}^{t-1} [\delta_i^{(p)}, \delta_j^{(q)}] \tag{8}$$

As before, let $M(x)$ stand for the maximal coefficient of the vector $x \in \mathscr{L}$, and let M_r denote the maximum among 1 and the coefficients occurring up until the rth step (that is, among the coefficients of the $\delta_i^{(p)}$'s, where $1 \leqq p \leqq r$ and $1 \leqq i \leqq k$). Obviously, we have $M_r \leqq M_q$ for $r \leqq q$ and $1 \leqq M_1$. One easily computes that if $x, y \in \mathscr{L}$ then $M([x, y]) \leqq cn^2 M(x) M(y)$. We use also the congruences $[\delta_i^{(p)}, \delta_j^{(q)}] \equiv 0 \pmod{\mathscr{R}^{t+1}}$ for $p + q > t$. For $n > 1$ the greatest coefficient $\tilde{M}_t$ of $\mathbf{b}$ in the tth system of linear equations can be bounded as

$$\begin{aligned} \tilde{M}_t &\leqq c + kc \sum_{r=1}^{t-1} M_r + c + cn^2 \sum_{q=1}^{t-1} M_q + cn^2 \sum_{p=1}^{t-1} M_p + cn^2 \sum_{l=1}^{t} \sum_{r=1}^{l} M_r M_{l-r} \\ &\leqq 3cn^2 \sum_{p=1}^{t-1} M_p + cn^3 \sum_{l=1}^{t} M_l M_{t-l} \\ &\leqq 4cn^3 \sum_{l=0}^{t} M_l M_{t-l}. \end{aligned} \tag{9}$$

We infer that $M(\mathbf{b}) = \tilde{M}_t \leqq 4cn^3 \sum_{l=0}^{t} M_l M_{t-l}$. Using also $\operatorname{rank}(\mathbf{A}) \leqq n^2$ and substituting $X = 4(n^2)!(3c)^{n^2} cn^3$, the Lemma gives

$$M_{t+1} \leqq M(\mathbf{x}) \leqq 4(n^2)!(3c)^{n^2} cn^3 \sum_{l=0}^{t} M_l M_{t-l} = X \sum_{l=0}^{t} M_l M_{t-l}.$$

For the common denominator D_t one can readily prove the following upper bound by induction:

$$D_t \leqq (\varDelta_1 \varDelta_2 \ldots \varDelta_t)^t, \tag{10}$$

where $\varDelta_l$ is the determinant of a maximal minor of the matrix $\mathbf{A} = \mathbf{A}_l$ of the lth system. Clearly we have $\varDelta_l \leqq (d_l)! c^{d_l} \leqq n! c^n$.

We know that $D_0 = 1$ and $M_0 = c$. An easy induction on t shows that

$$M_t \leqq t!(Xc)^t.$$

Expanding the above expression and writing out D_t explicitly we obtain the following estimates:

$$M_t \leqq t!(4(n^2)!(3c)^{n^2+2}n^3)^t \tag{11}$$

$$D_t \leqq (n!)^{t^2} c^{nt^2}. \tag{12}$$

Taking into consideration $t \leqq n$, we have

$$M_t \leqq M_n \leqq n!(4(n^2)!(3c)^{n^2+2}n^3)^n \tag{13}$$

$$D_t \leqq D_n \leqq (n!)^{n^2} c^{n^3}. \tag{14}$$

This means that the size measures $\log M_t$, $\log D_t$ of the intermediate results as well as of the ultimate solution $\log M_n$ and $\log D_n$ are bounded by a polynomial in n and $\log c$. The number of arithmetical operations is polynomial in n, therefore we have the following:

Theorem 2.3 *Let $\mathscr{L}$ be a Lie algebra over* **Q** *given by structure constants. Then we can compute a Levi decomposition for $\mathscr{L}$ in deterministic polynomial time.*

3 The Wedderburn–Malcev Decomposition for Associative Algebras

Here we outline a deterministic polynomial time procedure for computing Wedderburn–Malcev complements in finite dimensional associative algebras. The method works over finite fields and over **Q**. Note that the separability-assumption holds in these cases.

As in the Lie case, we assume that the associative algebra $\mathscr{A}$ is given by a collection of structure constants $c_{ij}^{(s)} \in \mathscr{F}$ relative to a fixed basis. Here $\mathscr{F}$ is either a finite field, or $\mathscr{F} = \mathbf{Q}$. We intend to compute a new basis $r_1, r_2, \ldots, r_k$, $s_1, s_2, \ldots, s_m$ of $\mathscr{A}$ over $\mathscr{F}$, where the s_i's are a basis of the Jacobson radical $\mathscr{R}$ and the r_j's form a basis of a semisimple complement $\mathscr{B} \leqq \mathscr{A}$.

We start with a collection of elements $a_1, a_2, \ldots, a_k \in \mathscr{A}$ such that the images $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_k$ of $a_1, a_2, \ldots, a_k$ under the natural map $\mathscr{A} \to \mathscr{A}/\mathscr{R}$ form a basis of $\mathscr{A}/\mathscr{R}$.

As in the Lie case, we work with subspaces V_i in $\mathscr{R}^i$ complementary to $\mathscr{R}^{i+1}$ (i.e. we have $\mathscr{R}^i = \mathscr{R}^{i+1} \oplus V_i$). We transform the elements a_i into a set of elements $b_1, b_2, \ldots, b_k$ such that $\mathscr{B} = Span(b_1, b_2, \ldots, b_k)$ is a subalgebra in $\mathscr{A}$ and $\mathscr{B} \cong \mathscr{A}/\mathscr{R}$. This means that $\mathscr{B}$ is a Wedderburn–Malcev complement in $\mathscr{A}$. We apply the iterative method that proved to be successful for Lie algebras.

Let $b_i^{(t)}$ denote the temporary result obtained in the tth step, $b_i^{(0)} = a_i$ and $b_i = b_i^{(last)}$. Our assumption is that

$$b_i^{(t)} b_j^{(t)} \equiv \sum_{s=1}^{k} c_{ij}^{(s)} b_s^{(t)} \pmod{\mathscr{R}^t} \tag{15}$$

holds after the tth round of the iteration for $1 \leqq i, j \leqq k$.

Then we look for elements $\delta_i^{(t)} \in V_t$ such that with $b_i^{(t+1)} = b_i^{(t)} + \delta_i^{(t)}$ we have

$$b_i^{(t+1)} b_j^{(t+1)} \equiv \sum_{s=1}^{k} c_{ij}^{(s)} b_s^{(t+1)} \pmod{\mathscr{R}^{t+1}} \tag{16}$$

for $1 \leqq i, j \leqq k$. After expanding (16) and using the relations $\delta_i^{(t)}\delta_j^{(t)} \equiv 0 \pmod{\mathscr{R}^t}$, we obtain

$$b_i\delta_j^{(t)} + \delta_i^{(t)} b_j - \sum_{s=1}^{k} c_{ij}^{(s)} \delta_s^{(t)} \equiv \sum_{s=1}^{k} c_{ij}^{(s)} b_s^{(t)} - b_i^{(t)} b_j^{(t)} \pmod{\mathscr{R}^{t+1}}. \tag{17}$$

The Wedderburn–Malcev theorem applied to $\mathscr{A}/\mathscr{R}^{t+1}$ implies that there always exist such elements $\delta_i^{(t)}$; and conversely, a solution to this linear system gives a Wedderburn–Malcev complement in $\mathscr{A}/\mathscr{R}^{t+1}$.

Just like in the Lie case, we infer immediately that a Wedderburn–Malcev complement can be found by using a polynomial (in $\dim \mathscr{A}$) number of arithmetical operations over $\mathscr{F}$. This gives a polynomial bound on the running time at once if $\mathscr{F}$ is finite. Over $\mathbf{Q}$ one has to establish polynomial bounds on the size of the coefficients of the elements $b_i^{(t+1)}$. This can be done using essentially the same argument as with the Levi decomposition. This is possible because (17) has the same transparent structure as (3). We have the following:

Theorem 3.1 *Let $\mathscr{A}$ be a finite dimensional associative algebra over the field $\mathscr{F}$ ($\mathscr{F}$ is either finite or $\mathscr{F} = \mathbf{Q}$), given by structure constants. Then we can compute a Wedderburn–Malcev decomposition of $\mathscr{A}$ in deterministic polynomial time.*

4 Computational Experiences

We considered the following two methods:

1. (nil-levi) a method that uses the upper central series of the radical,
2. (solv-levi) a method that proceeds along the derived series of the radical.

As test problems we worked with the Lie algebras described below. Let n be an even positive integer and consider $n \times n$ matrices (over $\mathbf{Q}$) that are composed of two by two blocks satisfying the following conditions:

1. all blocks under the main diagonal are 0,
2. all blocks on a line parallel to the main diagonal are identical,
3. the blocks on the diagonal are elements from sl_2,
4. the blocks on the other diagonals are from gl_2.

Let L_n be the Lie algebra determined by these matrices. Then the dimension of L_n is $2n - 1$. It has a basis of the following form:

$$\{h, x, y, z_1, \ldots, z_{2n-4}\}.$$

Here $\{h, x, y\}$ is the standard basis of sl_2 and for $i = 1, \ldots, n/2 - 1$ the elements $z_{4i-3}, z_{4i-2}, z_{4i-1}, z_{4i}$ are from gl_2. Since the above basis already contains a Levi subalgebra, we use a different input basis, namely

$$\left\{z_1, \ldots, z_{2n-4}, h + \sum_{i=1}^{n/2-1} z_{4i-3}, x + \sum_{i=1}^{n/2-1} z_{4i-2}, y + \sum_{i=1}^{n/2-1} z_{4i-1}\right\}.$$

For $n = 20, 22, \ldots, 34$ we computed a Levi subalgebra of L_n. The results are summarized in Figure 1.

Since the scale in Figure 1 is logarithmic, the points should lie on a straight line with slope equal to the order of the method. From this we conclude that the order of nil-levi is about 3.9 whereas the order of solv-levi is about 3.25.

The figure shows that on these inputs solv-levi is faster than nil-levi. This is (for the greater part) explained by the fact that the derived series turns out to be easier to calculate than the lower central series, as is seen from Table 1.

From this table it is apparent that the time taken for solving the systems of linear equations (i.e., the "difference") is comparable for the methods. This was to be expected since the total number of equations to be processed is essentially the same (namely $\binom{k}{2}\dim L_n$).

In this first example the input was "nice" (i.e., the structure constants of the Lie algebra were all small numbers and many of them were 0). In the next, more realistic, example we select a basis of L with random coefficients. The results are displayed in Table 2.

This table supports the claim that the two methods are comparable in speed on more complex inputs (nil-levi is even somewhat faster). It is also clear that the calculation of the series still dominates the running times. Our experiences show that there is no significant difference in the practical performance of nil-levi and

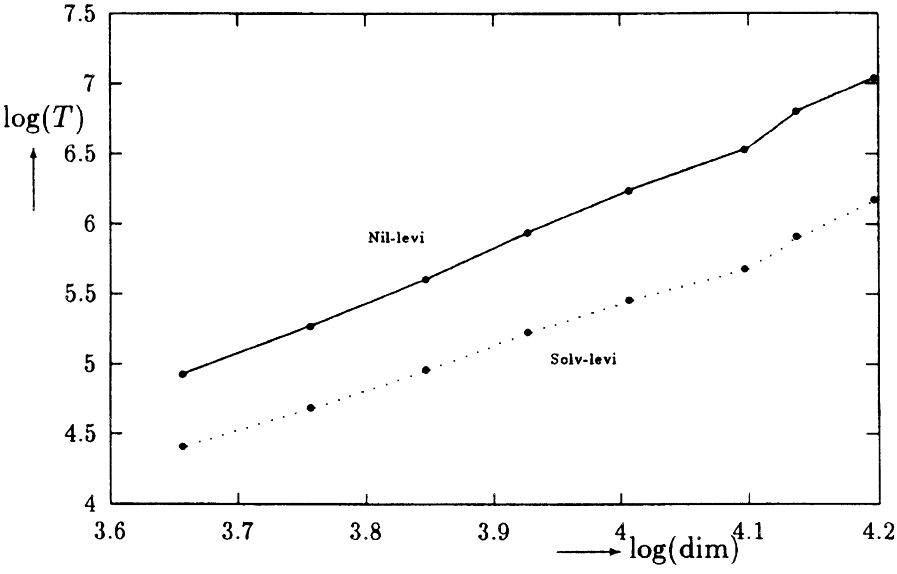

Fig. 1. Running times (in seconds) of the calculation of a Levi complement in L_n for $n = 20, 22, \ldots, 34$

Table 1. Running times (in seconds) of the calculation of the lower central series and the derived series of the radical of L_n. The third and fifth columns display the difference of the total computing time and the time consumed by the computation of these chains of ideals

n	Lower central series	Difference	Derived series	Difference
26	282	102	95	91
28	395	128	123	115
30	535	154	158	136
32	714	191	206	164
34	925	231	257	221

Table 2. Running times (in seconds) of the methods nil-levi and solv-levi on a random input basis

n	nil-levi	Lower central series	solv-levi	Derived series
6	18	12	18	10
8	126	98	120	85
10	2126	1847	2210	1832

solv-levi. For nil-levi, however, we have a theoretical guarantee of good performance, even on numerically complex inputs. The experimental results give some evidence that a suitable version of solv-levi may also provide a polynomial time method. We have not been able to prove this as yet.

Remark. In all cases the output basis provided by both methods turned out to be the same. This fact seems also to support our conjecture on solv-levi.

References

1. Beck, R. E., Kolman, B., Stewart, I. N.: Computing the structure of a Lie algebra. In: Computers in Non-associative Rings and Algebras. New York: Academic Press 1977
2. Belinfante, J. G. F., Kolman, B.: A Survey of Lie Groups and Lie Algebras with Applications and Computational Methods. SIAM 1990
3. Cohen, A. M., Meertens, L.: The ACELA project: Aims and Plans. To appear in: Kailer, N. (ed.) Human Interaction for Symbolic Computation. Texts and Monographs in Symbolic Computation. Vienna: Springer 1995
4. Friedl, K., Rónyai, L.: Polynomial time solutions of some problems in computational algebra. In: Proceedings of the 17th ACM STOC, Providence, Rhode Island, pp. 153–162 (1985)
5. de Graaf, W. A., Ivanyos, G., Rónyai, L.: Computing Cartan subalgebras in Lie algebras. Applicable Algebra in Engineering. Communication and Computing (to appear)
6. Hopcroft, J. E., Ullman, J. D.: Introduction to automata theory, languages and computation. London: Addison-Wesley 1979
7. Humphreys, J. E.: Introduction to Lie Algebras and Representation Theory. New York Heidelberg Berlin: Springer 1972

8. Jacobson, N.: Lie Algebras. New York: Dover 1979
9. Pierce, R. S.: Associative Algebras. New York Heidelberg Berlin: Springer 1982
10. Rand, D., Winternitz, P., Zassenhaus, H.: On the Identification of a Lie Algebra Given by its Structure Constants. I. Direct Decompositions, Levy Decompositions, and Nilradicals. Linear Algebra Appl. **109**, 197–246 (1988)
11. Rónyai, L.: Computing the structure of finite algebras. J. Symbolic Computation **9**, 355–373 (1990)
12. Rónyai, L.: Computations in Associative Algebras. In: Groups and Computation, DIMACS Series, 11, Am. Math. Soc., pp. 221–243 (1993)
13. Schwartz, J. T.: Fast probabilistic algorithms for verification of polynomial identities. J. ACM **27**, 701–717 (1980)
14. Seeley, C.: 7-Dimensional Nilpotent Lie Algebras. Trans. Am. Math. Soc. **335**, 479–496 (1993)
15. Winter, D. J.: Abstract Lie Algebras. The MIT Press (1972)