The 7th Balkan Conference on Operational Research
"BACOR 05"
Constanta, May 2005, Romania

ABOUT COMPUTATIONS IN LIE ALGEBRAS

CAMELIA CIOBANU
ION COLTESCU
IOAN POPOVICIU
PAUL VASILIU

Naval Academy "Mircea cel Batran", Constanta, Romania

Abstract
In this paper we shall present a reasonable way to compute the nilpotent and the solvable radical of a Lie algebra.

1. INTRODUCTION

We consider some basic algorithmic problems related to finite dimensional associative algebras.

Our starting point is the structure theory of these algebras and we touch upon some applications of the associative decomposition algorithms. These include efficient algorithms for calculating the radical (solvable and nilpotent) of Lie Algebras.

2. BASIC DEFINITIONS AND THEOREMS

First we give some basic definitions related to associative algebras.

A linear space L over the field k is an *algebra* over k if it is equipped with a binary, k – bilinear operation (called multiplication). We denote de product of $x, y \in L$ by *xy*. Multiplication is assured to be associative, i.e. $x(yz) = (xy)z$ for every $x, y, z \in L$.

We shall assume throughout that $dim_k L = n < \infty$. We say that L is a *commutative* algebra if *xy* = *yx* for every $x, y \in L$..

An k subspace S of L is a *subalgebra* of L, if S is closed under multiplication: if $x, y \in S$ then $xy \in S$.

An k subspace I of L is a *left ideal* of L if $yx \in L$ whenever $x \in I$ and $y \in L$. A *right ideal* is defined analogously. An k – subspace I of L is an *ideal* of L if I is both left and right ideal of *L*. If *I* is an ideal in *L*, then we can form the *factor algebra L/I*. The notions of *homomorphism* and *L-module* are used in the standard way.

The algebra *L* is *simple* if it has no ideals except (0) and *L*, and $LL \neq (0)$, where *LL* is the algebra generated by products *ab* with $a,b \in L$. We say that *L* is the *direct sum* of its ideals $L_1,\ldots,L_s$ (written as $L_1 \oplus \ldots \oplus L_s$) if *L* is the direct sum of these linear subspace.

Theorem 2.1 (Representation Theorem) Let L be an algebra over the field k and suppose that $dim_k\ L = n$. Then L is isomorphic to a subalgebra of $\mathrm{M}_{n+1}(k)$ - the algebra of all n + 1 by n + 1 matrices over k.

An element $x \in L$ is called *nilpotent* if $x^p = 0$ for some positive integer *p*. An element *x* is *strongly nilpotent* if *xy* is nilpotent for every $y \in L$.

The *Jacobson radical* $Rad(L)$ of *L* is the set of strongly nilpotent elements of *L*. It is not difficult to see that $Rad(L)$ is an ideal of *L* and that the fact $L/Rad(L)$ has no nonzero strongly nilpotent elements. It can be shown also, that $Rad(L)$, is a *nilpotent ideal*: there exist a positive integer *p* such that $x_1 x_2 \ldots x_p = 0$, for all $x_1, x_2, \ldots, x_p \in Rad(L)$.

An algebra *L* is *semisimple* if $|L| \geq 2$ and $Rad(L) = (0)$.

A characterization of semisimple algebra it was done by next theorem

Theorem 2.2 (Wedderburn's Theorem). If L is a finite dimensional semisimple associative algebra over the field k, then L is expressible as a direct sum

$$L = L_1 \oplus L_2 \oplus \ldots \oplus L_s,$$

where the L_i are exactly the minimal nonzero ideals of L. Moreover, L_i is isomorphic to matrix algebra $\mathrm{M}_{n_i}(k_i)$ where k_i is a possibly noncommutative extension field of k $(1 \leq i \leq s)$.

In the next section we present algorithms for computing the Jacobson radical. Here the interesting case is when *k* (and consequently *L*) is finite. We explain some basic methods for finding in a computationally efficient way the structural ingredients of algebras. These methods are applied in the last section to the computation of the (solvable) radical and the nilradical of Lie algebras.

We recall first some basic facts about Lie algebras. Detailed exposition can be found in Jacobson [10] and Humphreys [9]. A linear space *G* over the field *k* is a *Lie algebra*, if *G* is equipped with a *k* – bilinear binary operation [,] such that $[x,x] = 0$ for every $x \in G$ and $[[x,y],z] + [[y,z],x] + [[z,x],y] = 0$ for every $x,y,z \in G$ (the Jacobi identity).

Just like in the associative case, we have the familiar notions of *subalgebra, ideal, factor algebra* and *homomorphism* for Lie algebras.

The *derived series* of *G* is the collection $G^{(j)}$ of ideals in *G* defined as $G^{(0)} = G$ and $G^{(i+1)} = \left[G^{(i)}, G^{(i)}\right]$ for *i* >0.

A Lie algebra is called *solvable* if the derived series reaches (0) in finitely many steps: $G^{(n)} = (0)$ for some natural number n.

Here we consider finite dimensional Lie algebra only. In this case G hase an unique maximal solvable ideal, denoted by $R(G)$, the radical of G.

The *descending central series* of G is the sequence G^j of ideals of G, where $G^0 = G$ and $G^{i+1} = [G, G^i]$ for $i \geq 0$. A Lie algebra G is nilpotent if $G^n = (0)$ for some natural number n. If $dim_k G < \infty$, then G has an unique maximal nilpotent ideal $\mathrm{N}\ (G)$, the *nilradical* of G.

Example: Let L be an associative algebra over k. For two elements $a, b \in L$ we write $[a,b] = ab - ba$ for the additive commutator. It is easy to check this operation satisfies the identities of a Lie-bracket. As a consequence, if a k – subspace S of L is closed with respect to the operation [,], then S ca be considered as a Lie algebra. Particularly important are the Lie subalgebras of this form which are obtained from $G = \mathrm{M}_{\mathrm{p}}(k)$. They are called *linear Lie algebras.*

There is a straightforward analogue of the regular representation for a Lie algebra G. For an $x \in G$, $let\ ad(x) : G \to G$ be the linear map that maps $y \in G$ to $[x, y]$. The map $x \to ad(x)$ is a Lie algebra homomorphism from G to linear Lie algebra $gl(G)$ of all linear transformations of the k space G. Unfortunately, this map is far from being faithful (if G is simple, then this map is faithful). We just remark here that, according to a deep theorem of Ado and Iwasuwa [10], every finite dimensional Lie algebra is actually isomorphic to a linear Lie algebra.

We are interested in exact computations, and k will be either a finite field or on algebraic number field.

We specify now the input of the algorithmic problems addressed. To obtain sufficiently general results, we consider an algebra to be given as a collection of structure constants.

If L is an algebra over the field k and $e_1, e_2, \ldots, e_n$ is a basis of the k – space L, then multiplication is completely described if we express the product $e_i e_j$ as linear combinations of the basis elements:

$$e_i e_j = \gamma_{ij1} e_1 + \ldots + \gamma_{ijn} e_n$$

The coefficients $\gamma_{ijk} \in k$ are called structure constants. When an algebra is given as input, we assume that it is represented as an array of structure constants. Substructures (sack as subalgebras, ideals, subspaces) can then be represented by bases whose elements are linear combinations of basis elements of the ambient structure (algebra).

In our cases k can be viewed as an algebra over its prime field P. (If k is finite then $P = k_p$ for some prime p, if k is number field than $P = Q$).

In these cases k is usually specified by giving the (monic) minimal polynomial f of a single generating element α over the prime field P. This is a special case of the representation with structure constants. The coefficient of f give the structure constants with respect to P – basis $1, \alpha, \alpha^1, \ldots, \alpha^{n-1}$ of k where $n = dim_p\, k$.

Another important way to represent an algebra is in the form of a matrix algebra. In these cases we are given a collection of matrices which generate the algebra. The algorithms described in these notes are applicable in this setting as well. From such a matrix representation one can efficiently find a basis of the algebra and then calculate structure constants with respect to this basis.

We would like to consider algorithms which have a theoretical guarantee for their efficiency. From the perspective of computer science these are the polynomial time algorithms. An algorithm runs in polynomial time if, on inputs of length n the computation requires at most n^c bit operations. Here $c > 0$ is a constant independent of n, and n is a positive integer.

3. COMPUTING THE RADICAL

Suppose L is a finite dimensional algebra over the field k, given as a collection of structure constants. Our objective is to find a basis of $Rad(L)$, the radical of L, in time polynomial in the input size.

If char $k = 0$, then the problem is equivalent to solving a system of linear equations over the ground field as follows from the characterization of the radical by Dickson:

Theorem 3.1 Let L be a finite dimensional algebra of matrices over a field k, and char k = 0. Then

$$\mathrm{Rad(L)=\{x \in L\ /Tr(yx)=0\ for\ every\ y \in L\}}.$$

In fact, if $e_1, e_2, \ldots, e_n$ is a linear basis of L over k, then to find $Rad(L)$, it suffices to solve the linear system $Tr(e_i x) = 0,\ i = 1, \ldots, n,$ where x is an “unknown” element of L.

We now turn to the case where L (and hence $k = k_q$) is finite. We assume that p is a prime, q is a power of p, $k = k_q$ and that L is a subalgebra of $\mathrm{M}_n(k)$.

The statement of Dickson’s Theorem is no longer valid in positive characteristic. There is, however, a more subtle, and useful, description of the radical in this case. We explain this in the sequel.

We define the natural number l by the following inequalities: $p^l \leq n < p^{l+1}$. Let M denote the set of matrices $L \bigcup \{I\}$ where I is the identity element of $\mathrm{M}_n(k)$.. Let a $\in \mathrm{M}_n(k)$ be a matrix. It will be convenient to work with the following variant of the characteristic polynomial of a: $\varphi_a(X) = \det(Xa + I) \in k[X]$.

Consider the expansion of $\overline{\varphi_a}(X)$ as a polynomial in the variable X:

$$\overline{\varphi}_a(X) = 1 + \sum_{i=1}^{n} c_{a,i} X^i$$

The indices of the form $i = p^j,\ j = 0, \ldots, l$ play a key role in the following arguments. For $j = 0, \ldots, l$ we define the “trace functions” T_j by $T_j(a) := c_{a,p^j}$. Obviously, $T_0(a) = Tr(a)$ is the trace of the matrix a.

We also define a sequence $L : R_0 \supseteq R_1 \supseteq \ldots \supseteq R_{l+1}$ of subsets of L as

$$R_j := \{a \in L \mid T_i(ma) = 0 \ \ for\ every\ m \in M \ \ and\ 0 \le i < j\} \quad (1 \le j \le l+1).$$

Alternatively, for every $0 \le j \le l,$ we have

$$R_{j+1} := \{a \in R_j \mid T_j(ma) = 0 \ \ for\ every\ \ m \in M\}$$

At this point we can formulate a characterization of $Rad(L)$ which is the main result of this section, it is useful and reads as follows:

Theorem 3.2 Let $L \le \mathrm{M}_n(k)$ be an algebra of matrices over the finite field k of characteristic p. Put $l = \lfloor log_p\, n \rfloor,$ and let $R_0, R_1, \cdots, R_{l+1}$ be as defined above. Then:

1. $R_0, R_1, \ldots, R_{l+1}$ are ideals of L:
2. $R_{l+1} = Rad(L):$
3. For every $j \in \{0, \ldots, l\}$ the function T_j is p^j-semilinear on R_j, i.e.

$$T_j(\alpha a + \beta b) = \alpha^{p^j} T_j(a) + \beta^{p^j} T_j(b)$$

for every $\alpha, \beta \in k\ \ and\ a, b \in R_j.$

Property 3 implies that we can obtain a basis of R_{j+1} from a basis R_j by solving a system of linear equations over *k*. Indeed set $a_0 = I,$ ant let $a_1, \ldots, a_s$ be a basis of L over *k*. Suppose that we have a basis $\{b_1, b_2, \ldots, b_r\}$ of R_j over *k*, and we are looking for a basis of R_{j+1}. Semilinearity implies that an element $a \in R_j, a = \sum_{i=1}^{r} \lambda_i b_i$ is in R_{j+1} if and only if $\sum_{i=1}^{r} T_j(a_t b_i) \lambda_i^{p^j} = 0, (t = 0, \ldots, s).$

The inverse of the automorphism $\lambda \to \lambda^{p^s}$ of the finite field $k = k_q$ can be computed efficiently, hence the system above can be translated into

$$\sum_{i=1}^{r} T_j(a_t b_i)^{\frac{1}{p^s}} \lambda_i = 0 \quad (t = 0, \ldots, s)$$

This latter is a system of linear equation in the variables $\lambda_1, \lambda_2, \ldots, \lambda_r.$ Thus we start with $R_0 = L$ and then in turn proceed to compute $R_l, \ldots, R_{l+1}.$

From a basis of R_i we obtain a basis of R_{i+1} by solving a system of linear equation of *k*. The number of equations and the number of variables is at most n^2, hence the system can be solved in time $(n + log\, q)^{O(1)}$. We obtain a basis of $Rad(L)$ in $l + 1 = 0(log\, n)$ such rounds; there fore the overall cost of the computation is $(n + log\, q)^{0(1)}$ bit operations. Below we give a formal description of the algorithm.

```
Radical $(L):=$
  $A := \{I\} \cup$ basis of $L$;
  $B :=$ basis of $L$;
```

```
for $j$ from 1 to $\lfloor \log_p n \rfloor + 1 do$
  if $B \neq \phi\ then$
    $C := \left( T_j(ab)^{p^{\frac{1}{j}}} \right) a \in A, b \in B$
    $\Lambda := a$ basis of $KerC$ :
    $B := \{ \lambda_1 b_1 + \ldots + \lambda_r b_r \ | (\lambda_1, \ldots, \lambda_r) \in \Lambda \}$
  fi
od
return B.
```

The proof of Theorem 3.2 is immediately with the next sequence of lemmas. The statement of the first lemma can be considered as a special case of the theorem, where the underlying module is simple.

Lemma 3.3 Let S be a simple algebra over the finite field k and U be a simple S – module. Then there exists an element $a \in S$ with $Tr_U(a) = 1$, where $Tr_U(a)$ stands for the ordinary trace of the action of a on U.

Below we show that semi linearity and other useful proprieties hold for the trace functions T_j on certain ideals.

Lemma 3.4 Let $L \leq \mathsf{M}_n(k)$ be a matrix algebra over the field k of characteristic p. Assume that $L \neq Rad(L)$. Let $(0) = U_0 < U_1 < \ldots < U_r = U$ be a composition series of the L – module $U = k^n$. Let $I_1, I_2, \ldots, I_t$ be the minimal elements of the set of ideals of L properly containing $Rad(L)$. For every index $i \in \{1, 2, \ldots, t\}$ fix a simple L – module V_i that belongs to the ideal I_i and denote the multiplicity of V_i in the composition series by m_i. $Put \quad l = \lfloor log_p n \rfloor$ and define the ideals $R'_0, R'_1, \ldots, R'_{l+1}$ as

$$R_j = Rad(L) + \sum_{p^j / m_i} I_i;$$

(*)

Then:

i) $R'_{l+1} = Rad(L)$;

ii) For every $j \in \{0, \ldots, l\}$ the function T_j is p^j - semilinear on R'_j (in the sense of Theorem 3.2)

iii) $T_j(ab) = T_j(ba)$ for every $j \in \{0, \ldots, l\}$, $b \in L$ and $a \in R'_j$.

iv) T_j is identically zero on $R'_{j+l} \quad (j = 0, \ldots, l)$.

v) T_j is not identically zero on ideals J_i such that the multiplicity m_i is divisible by p^j but not by $p^{j+l} \quad (j = 0, 1, \ldots, l)$.

The last lemma provides a tool to inductively verify that the subsets R_j coincide with the ideal R'_j defined in that lemma.

Lemma 3.5 Keeping the notation of Lemma 3.4 for each $j \in \{0,\ldots,l\}$ we have
$R'_{j+1} = \{a \in R'_j \mid T_j(ab) = 0 \text{ for every } b \in \{I\} \bigcup L\}$

Remark 3.6 The approach presented here is a simplified and specialized version of a result from [4] where arbitrary fields of characteristic p are allowed. In that general case the characterization of the ideals R_j is slightly more complicated then formula (*).

4. ABOUT COMPUATIONS IN LIE ALGEBRAS

Just like associative algebras, Lie algebras can be conveniently described by structure constants. If G is a Lie algebra over a field *k* and $e_1, e_2, \ldots, e_n$ is a basis of G, then the bracket is described if we have the products e_ie_j as linear combinations of the basis elements:

$$\lfloor e_i, e_j \rfloor = \gamma_{ij_1} e_1 + \ldots + \gamma_{ij_n} e_n$$

The coefficients $\gamma_{ij_k} \in k$ are called structure constants.

Now we outline algorithms for computing the nilpotent and the solvable radical of a Lie algebra. These problems can be reduced to associative radical computations. First we consider the nilradical. We need a theorem of Jacobson [10].

Let G be a finite dimensional Lie algebra over an arbitrary field *k*.

Theorem 4.1 (Jacobson's Theorem) Let L be the associative (matrix -) algebra generated by the linear transformations $ad(x), x \in L, i.e.,$ the image $ad(L)$ of the adjoint representation of L. Then an element $x \in L$ is in the nilradical N(L) if and only if $ad(x) \in Rad(L)$.

This result offers a reasonable way to computing $N(L)$ if the ground field k is a finite field or an algebraic number field. Indeed, we can compute first a basis of *L*, and then compute $Rad(L)$ with the algorithms of the previous section. We calculate the intersection of the *k* – subspaces $ad(L)$ and $Rad(L)$ by solving a system of linear equations. By Jacobson's Theorem the inverse image in *L* of the intersection $ad(L) \cap Rad(L)$ is $N(L)$. A formal description of our method reads as follows.

```
Nilradical (L): =
        L: = associative algebra generated by $ad(L)$
        return $ad^{-1}Rad(L)$.
```

Corollary 4.3 Let G be a finite dimensional Lie algebra over the field k, where k is either a finite field or on algebraic number field. Suppose that G is given as a collection of structure constants. Then the nilradical N(G) can be computed in time polynomial in the input size.

Next we address the problem of computing the solvable radical *R* (*G*).

For finite *k* the problem of computing *R* (*G*) can be reduced efficiently to the problem of computing $N(G)$.

We observe first that $\mathrm{N}\,(\mathrm{G}) \le \mathrm{R}\,(\mathrm{G})$ and if $\mathrm{N}\,(\mathrm{G}) = (0)$ then $\mathrm{R}\,(\mathrm{G}) = (0)$, because the next to last element of the derived series of $\mathrm{R}\,(\mathrm{G})$ is an abelian hence nilpotent ideal of *G*. With these we define the sequence G_i of Lie algebras as follows: let $G_0 = G$; if $\mathrm{N}\,(\mathrm{G}_\mathrm{i}) \ne (0)$ then let $G_{i+1} = G_i / \mathrm{N}\,(\mathrm{G}_\mathrm{i})$; if $\mathrm{N}\,(\mathrm{G}_\mathrm{i}) = (0)$ then G_{i+1} is not defined. This sequence of Lie algebras has no more than $\dim_k G + 1$ elements. From Corollary 4.3 we obtain that the algebras G_i can all be computed in polynomial time over finite *k*. Let G_j be the last algebra of the sequence. We then have $G_j \cong G / R(G)$. Moreover, we can construct a basis for $R(G)$ by keeping track of the preimages of the ideals we factored out during the computation of the sequence $G_0, G_1, \ldots, G_j$.

It is instructive to view this computation in terms of ideals of *G*. For i > 0 let J_i denote the kernel of the composition of the natural maps $G_0 \to G_1 \to \ldots \to G_i$. We then have $J_1 \subset J_2 \subset \ldots \subset J_j$, $\mathrm{N}\,(\mathrm{G/J_i}) = J_{i+1} / J_i$ for $0 < i < j$, and $J_j = R(G)$. From a basis of J_i a basis of J_{i+1} is obtained by a single call of the nilradical – algorithm with the algebra G / J_i as input. As a result, we obtain elements $h_1, h_2, \ldots, h_k \in G$ such that $h_1 + J_i, \ldots, h_k + J_i$ from a basis of J_{i+1} / J_i. Now the elements h_l together with a basis of J_i will constitute a basis of J_{i+1}. In *j* such rounds we obtain a basis of $R(G)$. Below we give a formal description of the algorithm:

```
Solvable Radical(G): =
        S: = (0):
        loop
           S: = Nilradical (G/S);
           $\phi$: = natural map $G \to G/S$;
           $S := \phi^{-1}(\overline{S})$;
        until $\overline{S} = (0)$;
        return S.
```

Corollary 4.4 Let G be a finite dimensional Lie algebra over k_q, given by structure constants. Then (a basis of) the solvable radical $R(G)$ can be computed in time polynomial in $dim_{k_q} G$ and $log\, q$.

Variants of the radical algorithms discussed here are implemented by Wilhelm de Graaf in a general library of Lie algebras algorithm called ELIAS (for Eindhoven LIe Algebra System), which is built into the computer algebra systems GAP4 and MAGMA.

BIBLIOGRAPHY

[1] E.H. Bareiss (1968): “Sylvester’s identify and multistep integer-preserving Gaussian elimination”, Mathematics of computation, 103, 565 – 578;

[2] R.E. Beck, B. Kolman, and I. N. Stewart (1977): “Computing the structure of a Lie algebra”, Computers in nonassociative rings and algebras, Academic Press, New York, 167 – 188.

[3] E.R. Berlenkamp (1970): “Factoring polynominals over large finite fileds”, Math. of Computation 24, 71375.

[4] A.M. Cohen, G. Ivanyos, and D.B. Wales (1997): “Finding the radical of an algebra of linear transformations“, J. of Pure and Applied Algebra 117&118, 177 – 193.

[5] G. E. Collins M. Mignotte, and F. Winkler (1983): “Arithmetic in basic algebraic domains”, in: Computer Algebra, Symbolic and Algebraic Computation, 2nd edn., Springer-Verlag, Berlin Heidelberg New York, 180 – 220.

[6] L.E. Dickson (1923): “Algebra and Their Arithmetic’s”, Univeristy of Chicago.

[7] Edmonds (1967): “System of distinct representatives and linear algebra”, Journal of Research of the National Bureau of Standards 718, 241 – 245.

[8] G. Ivanyos, L. Rónyai and Á. Szántó (1994): “Decomposition of algebras over $F_q(X_1,\ldots,X_m)$”, Applicable Algebra in Engineering, Communication and Computing 5; 71 – 90.

[9] J.E. Humphreys (1980): “Introduction to Lie Algebra and Representation Theory”, Graduate Texts in Mathematics 9, Springer-Verlag, Berlin Heidelberg New York.

[10] N. Jacobson (1962): “Lie Algebra”, John Wiley.

[11] D.E. Knuth (1981): “The art of computer programming”, Vol 2, Seminumerical algorithms, Addison-Wesley.

[12] R. Lidl and H. Niederreiter (1983): “Finite Fields”, Addison-Wesley.