

# Unified computational approach to nilpotent algebra classification problems

*Shirali Kadyrov, Farukh Mashurov*

**Abstract.** In this article, we provide an algorithm with Wolfram Mathematica code that gives a unified computational power in classification of finite dimensional nilpotent algebras using Skjelbred-Sund method. To illustrate the code, we obtain new finite dimensional Moufang algebras.

## 1 Introduction

Let  $\mathfrak{A}$  be a finite dimensional algebra over the field of complex numbers  $\mathbb{C}$  equipped with a bilinear product denoted by  $x \cdot y$ . Given a variety defined over certain polynomial identities, one of the classical research problems is to classify, up to isomorphism, all algebras within the variety with fixed dimension. Varieties of associative, alternative, Lie, Novikov, Jordan, assosymmetric, Leibniz, Zinbiel, and Tortkara algebras are some of the most studied varieties, see e.g. [1],[2],[6],[8],[15],[17] and references therein. As described in the next section, classification involves several steps each requiring to symbolically solve systems of polynomial equations. Computational effort depends on many variables including the dimension of the variety, bilinear product, and number of identities and it may be required to solve system of more than  $n^3$  polynomial equations with  $n^2$  unknowns, where  $n$  is the fixed dimension of the variety. Obviously, the computations could become cumbersome even when the dimension is as little as 2, see e.g. [9]. As such, in part many authors rely on computer assisted classifications.

Our goal in this article is to provide a unified algorithm with a code written in Wolfram Mathematica to ease the computation aspects of the classification problem for nilpotent algebras and illustrate it with a new classification result. It is unified

---

2010 MSC: 17A30, 68W30

*Key words:* Algebra, Skjelbred-Sund classification, finite dimensional nilpotent algebra, Wolfram Mathematica, symbolic solver, algorithm

*Affiliation:*

Sh. Kadyrov – Suleyman Demirel University, Kaskelen, Kazakhstan

*E-mail:* shirali.kadyrov@sdu.edu.kz

F. Mashurov – Institute of Mathematics and Mathematical Modeling, Almaty and Suleyman Demirel University, Kaskelen, Kazakhstan

*E-mail:* f.mashurov@gmail.com

in the sense that a minimal effort is required to run the code, namely define the bilinear product and polynomial identities for the kind of algebraic variety being studied. Moreover, most of the computational steps are taken care of by the code as opposed to the previous works. Another significance of the current work is that it focuses on explaining the algorithms and the details of the full source code while the previous works were concentrated on algebraic classification with little attention to coding. The code is successfully tested on several previously obtained classification results. Moreover, it was used to classify assosymmetric algebras of dimension 4 in [13] and one generated assosymmetric algebras of dimension 5 and 6 in [19]. We illustrate our code and obtain new 4-dimensional Moufang algebra.

In the next section, we provide the background for steps needed to apply well-known Skjelbred-Sund classification method together with the algorithms that we follow in writing the code. In the follow up section § 3 we provide new results to illustrate our unified symbolic computational approach. The original code by the authors is provided open access through [14]. Finally in § 4 we conclude with possible future research directions.

## 2 Skjelbred-Sund classification method

Skjelbred-Sund classification method is one of the classical methods to classify finite dimensional nilpotent algebras which goes back to [21], where used central extensions of less than 6-dimensional Lie algebras to describe nilpotent 6-dimensional Lie algebras. More recently, [12] used Skjelbred-Sund in classification of all non-Lie central extensions of all 4-dimensional Malcev algebras and [10] in classification of five-dimensional nilpotent Jordan algebras. For various results where the same method was used for algebraic classification of finite dimensional nilpotent algebras from different varieties we refer to [7], [11], [16], and references therein.

For an excellent exposition of the method we refer to [10] and [12]. Here, we simply review some important notions used in the classification.

We let  $\mathfrak{A}$  denote an  $n$  dimensional algebra in certain variety  $\mathfrak{L}$  defined by set of polynomial identities and  $V$  be a vector space over  $\mathbb{C}$ . We define  $Z_{\mathfrak{L}}^2(\mathfrak{A}, V)$  as the set of bilinear forms  $\theta : \mathfrak{A} \times \mathfrak{A} \rightarrow V$  satisfying all identities in the variety  $\mathfrak{L}$ . That is,  $Z_{\mathfrak{L}}^2(\mathfrak{A}, V)$  is the set of closed bilinear forms, also known as *cocycles*. For a linear map  $f$  from  $\mathfrak{A}$  to  $V$ , we define a map  $\delta f : \mathfrak{A} \times \mathfrak{A} \rightarrow V$  given by  $\delta f(x, y) = f(xy)$ . It is easy to see that  $\delta f \in Z_{\mathfrak{L}}^2(\mathfrak{A}, V)$ . Therefore,

$$B_{\mathfrak{L}}^2(\mathfrak{A}, V) := \{\theta = \delta f \mid f \in \text{Hom}(\mathfrak{A}, V)\}$$

is a subspace of  $Z_{\mathfrak{L}}^2(\mathfrak{A}, V)$ . The elements of  $B_{\mathfrak{L}}^2(\mathfrak{A}, V)$  are called *coboundaries*. The second cohomolgy space is define as

$$H_{\mathfrak{L}}^2(\mathfrak{A}, V) := Z_{\mathfrak{L}}^2(\mathfrak{A}, V) / B_{\mathfrak{L}}^2(\mathfrak{A}, V).$$

We let  $\text{Aut}(\mathfrak{A})$  denote the automorphism group of  $\mathfrak{A}$  and take  $\phi \in \text{Aut}(\mathfrak{A})$ . If we define an action of  $\text{Aut}(\mathfrak{A})$  on  $Z_{\mathfrak{L}}^2(\mathfrak{A}, V)$  via  $\phi\theta(x, y) = \theta(\phi(x), \phi(y))$  for every  $\theta \in Z_{\mathfrak{L}}^2(\mathfrak{A}, V)$ , then necessarily  $\phi\theta \in Z_{\mathfrak{L}}^2(\mathfrak{A}, V)$ . That is,  $\text{Aut}(\mathfrak{A})$  acts on  $Z_{\mathfrak{L}}^2(\mathfrak{A}, V)$ . Furthermore,  $\phi\theta \in B_{\mathfrak{L}}^2(\mathfrak{A}, V)$  if and only if  $\theta \in B_{\mathfrak{L}}^2(\mathfrak{A}, V)$ , so that  $\text{Aut}(\mathfrak{A})$  also acts on  $H_{\mathfrak{L}}^2(\mathfrak{A}, V)$ .

We let  $V$  be a vector space of dimension  $m$  and  $\theta \in Z_{\mathfrak{L}}^2(\mathfrak{A}, V)$ . With a multiplication “ $\cdot_{\mathfrak{A}_\theta}$ ” on  $\mathfrak{A}_\theta := \mathfrak{A} \oplus V$  given by

$$(x + v_1) \cdot_{\mathfrak{A}_\theta} (y + v_2) = x \cdot y + \theta(x, y) \text{ for all } x, y \in \mathfrak{A} \text{ and } v_1, v_2 \in V,$$

$\mathfrak{A}_\theta$  becomes an algebra in the variety  $\mathfrak{L}$ , called *m-dimensional central extension* of  $\mathfrak{A}$  by vector space  $V$ .

Let  $\{e_1, e_2, \dots, e_m\}$  be a basis for  $V$ . Then, any  $\theta \in Z_{\mathfrak{L}}^2(\mathfrak{A}, V)$  can be uniquely written as  $\theta(x, y) = \sum_{i=1}^m \theta_i(x, y)e_i$ , for some  $\theta_i \in Z_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$ . The set  $Ann(\theta) := \{x \in \mathfrak{A} : \theta(x, \mathfrak{A}) + \theta(\mathfrak{A}, x) = 0\}$  is called the *annihilator* of  $\theta$ . The *annihilator* of an algebra  $\mathfrak{A}$  is defined as the ideal  $Ann(\mathfrak{A}) = \{x \in \mathfrak{A} : x\mathfrak{A} + \mathfrak{A}x = 0\}$ .

We know that every finite-dimensional nilpotent algebra is a central extension of some nilpotent algebra of lower dimension, see e.g. [12]. Therefore, to classify all nilpotent algebras of a fixed dimension in variety  $\mathfrak{L}$ , all we need is to classify cocycles of nilpotent algebras  $\mathfrak{A}$  of lower dimension with condition  $Ann(\mathfrak{A}) \cap Ann(\theta) = 0$  and central extensions that appear from them (see Lemma 5 in [12]).

Let  $G_m(H_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C}))$  be the set of all  $m$ -dimensional subspaces of  $H_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$ . We define

$$T_m(\mathfrak{A}) = \{W := \langle [\theta_1], [\theta_2], \dots, [\theta_m] \rangle \in G_m(H_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})) : \cap_{i=1}^m Ann(\theta_i) \cap Ann(\mathfrak{A}) = 0\}.$$

We define action of  $Aut(\mathfrak{A})$  on  $T_m(\mathfrak{A})$  via  $\phi W = \langle [\phi\theta_1], [\phi\theta_2], \dots, [\phi\theta_m] \rangle$  for  $\phi \in Aut(\mathfrak{A})$  and  $W \in T_m(\mathfrak{A})$ . The set  $Orb(W)$  is the orbit of  $W \in T_m(\mathfrak{A})$  under automorphism group of  $\mathfrak{A}$ .

**Definition 1.** Let  $\mathbf{A}$  be an algebra and  $I$  be a subspace of  $Ann(\mathbf{A})$ . If  $\mathbf{A} = \mathbf{A}_0 \oplus I$  then  $I$  is called an *annihilator component* of  $\mathbf{A}$ .

Assume that a vector space  $V$  has dimension  $m$ . For a given algebra  $\mathfrak{A}$  in variety  $\mathfrak{L}$ , we let  $E(\mathfrak{A}, V)$  be the set of all non-split central extensions of  $\mathfrak{A}$  by  $V$ , where a non-split central extension is central extension of  $\mathfrak{A}$  without annihilator components.

**Theorem 1 (see [12]).** *There is a one-to-one correspondence between the set of  $Aut(\mathfrak{A})$ -orbits on  $T_m(\mathfrak{A})$  and the set of isomorphism classes of  $E(\mathfrak{A}, V)$ .*

Finally, we have the following steps to construct from the algebra  $\mathfrak{A}$  of dimension  $n - m$  all non-split algebras in  $\mathfrak{L}$  of dimension  $n$  with  $m$ -dimensional annihilator. We need to determine the following sets for given algebra  $\mathfrak{A}$  :

1. Compute base for  $Z_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$ ;
2. Compute base for  $B_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$  and  $H_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$ ;
3. Compute  $Aut(\mathfrak{A})$ ;
4. Compute base for  $Ann(\mathfrak{A})$  and  $Ann(\mathfrak{A}) \cap Ann(\theta)$ ;
5. Compute  $Aut(\mathfrak{A})$ -orbits on  $T_m(\mathfrak{A})$ ;

6. Construct the algebra in the variety  $\mathfrak{L}$  associated with a representative of each orbit.

We now describe algorithms to handle steps from 1 to 4. The remaining two steps are worked out by hand.

Let  $\mathfrak{A}$  be an algebra with basis  $\{e_i : i = 1, 2, \dots, n\}$ . We use the following notations:  $\Delta_{i,j}$  is the bilinear form  $\Delta_{i,j} : \mathfrak{A} \times \mathfrak{A} \rightarrow \mathbb{C}$  such that

$$\Delta_{i,j}(e_l, e_k) = \delta_{il}\delta_{jk}. \quad (1)$$

The set  $\{\Delta_{i,j} : 1 \leq i, j \leq n\}$  is the basis of  $Z_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$ . Every  $\theta \in Z_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$  can be uniquely written as  $\theta = \sum_{1 \leq i, j \leq n} \lambda_{i,j} \Delta_{i,j}$  where  $\lambda_{i,j} \in \mathbb{C}$ .

Now, we give algorithms to compute the above mentioned steps. The first algorithm shows how to compute  $Z_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$  given the dimension, the product rule, and the polynomial identities. It amounts to defining the symbolic equations and call the symbolic solver from the relevant programming language.

---

**Algorithm 1:** Algorithm to compute the basis for the  $Z_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$

---

**Input:** Dimension of your algebra:  $n$ , Bilinear product rule:  $\text{pr}(\cdot, \cdot)$ ,  
Identities:  $\{\text{Iden}_1, \dots, \text{Iden}_m\}$ .

Define a symbolic basis for algebra  $\mathfrak{A}$ :  $\{e_1, \dots, e_n\}$

Define symbolic cocycles, a bilinear function:  $\theta(\cdot, \cdot) = \sum_{i,j} \lambda_{i,j} \Delta_{i,j}(\cdot, \cdot)$  Let  
 $k$  be the number of variables used in the Identities.

Define the system of symbolic  $n^k m$  nonlinear equations:

**for**  $(i_1, \dots, i_k) \in \{1, \dots, n\}^k$  **do**

**for**  $j=1:m$  **do**

        Define an  $eq[i_1, \dots, i_k, j]$  that is obtained by applying  $\theta$  to  $\text{Iden}_j$   
        when  $e[i_1], \dots, e[i_k]$  are substituted.

Use built in ‘solve’ function to solve the system  $\{eq\}$ .

**Output:** Basis for  $Z_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$ : Z2A

---

The next algorithm uses outcomes of Algorithm 1 together with the same inputs. In this case we aim to compute bases for  $B_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$  and  $H_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$ . It does not require any tricks to obtain a basis for  $B_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$  but simply write them down manually from the given polynomial identities. In terms of coding this means to ask the programming language to read coefficients of polynomial expressions. As for the second part, we recall that  $B_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C}) \subset Z_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$  and  $H_{\mathfrak{L}}^2(\mathfrak{A}, V) = Z_{\mathfrak{L}}^2(\mathfrak{A}, V) / B_{\mathfrak{L}}^2(\mathfrak{A}, V)$ . Thus, the problem of finding a basis for  $H_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$  is equivalent to completing the basis of  $Z_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$  given the basis of  $B_{\mathfrak{L}}^2(\mathfrak{A}, \mathbb{C})$ .

---

**Algorithm 2:** Algorithm to compute the bases for  $B_{\mathbb{C}}^2(\mathfrak{A}, \mathbb{C})$  and  $H_{\mathbb{C}}^2(\mathfrak{A}, \mathbb{C})$

---

**Input:** Dimension of your algebra:  $n$ , Bilinear product rule:  $\text{pr}(\cdot, \cdot)$ ,  
 Identities:  $\{\text{Iden}_1, \dots, \text{Iden}_m\}$ , Basis for  $Z^2(\mathfrak{A}, \mathbb{C}) : \{z_1, \dots, z_k\}$   
 From  $\text{pr}(\cdot, \cdot)$  obtain a basis for  $B^2(\mathfrak{A}) : B2A = \{b_1, \dots, b_s\}$   
 Define empty set  $H2A := \{\}$ ;  
**for**  $i=1:k$  **do**  
     **if**  $z_i \notin \text{span}(B2A)$  **then**  
         Add  $z_i$  to  $H2A$ ;  
         Add  $z_i$  to  $B2A$ ;

**Output:** Basis for  $B_{\mathbb{C}}^2(\mathfrak{A}, \mathbb{C})$  and  $H_{\mathbb{C}}^2(\mathfrak{A}, \mathbb{C})$ : B2A, H2A

---

Computing the  $\text{Aut}(\mathfrak{A})$ , Algorithm 3, is one of the main steps in the above described method and the one with large computational cost. We may represent an automorphism with an  $n \times n$  invertible square matrix that respects the bilinear product rule. This requires to define symbolic matrix and define system of symbolic equations and finally call the solve function.

---

**Algorithm 3:** Algorithm to find the automorphism group

---

**Input:** Dimension of your algebra:  $n$ , Bilinear product rule:  $\text{pr}(\cdot, \cdot)$   
 Define matrix  $\text{GAut}_{n \times n}$ ; Define homomorphism function  
 $F[\{x, y\}] = \text{pr}[F[x], F[y]] - F[\text{pr}[x, y]]$ ;  
 Define mapping of basis by  $F[e_i] = \sum_{j=1}^n \lambda_{i,j} e_j$ ;  
**for**  $i, j; n$  **do**  
     Substitute basis to  $F[\{e_i, e_j\}]$  and define by Eq  
**end**  
 Use "solve" to find is set of solutions  $\lambda_{i,j}$  and define by Solution;  
 Obtained solutions substitute to matrix  $\text{GAut}$ , that is  
**for**  $i=1; \text{Length}[\text{Solutions}]$  **do**  
      $\text{Mat}[i] = \text{GAut} / .\text{Solution}[[i]]$   
**end**  
 Define set Automorphism= $\{\}$ ;  
**if**  $\text{Det}[\text{Mat}[i]] \neq 0$  **then**  
     Add to Automorphism  
**end**

**Output:** Matrix forms of Automorphism group : Automorphism

---

The next three algorithms are allocated for step 4 to compute annihilators. The Algorithm 4 computes the action of the automorphism group on  $H_{\mathbb{C}}^2(\mathfrak{A}, \mathbb{C})$  and uses outcomes of Algorithm 2 and Algorithm 3. Action of automorphism group defined by  $\phi^T * M * \phi$  where  $\phi \in \text{Aut}(\mathfrak{A})$  and  $M$  is matrix form of  $H_{\mathbb{C}}^2(\mathfrak{A}, \mathbb{C})$ .

---

**Algorithm 4:** Algorithm of action of the automorphism group on  $H_{\xi}^2(\mathfrak{A}, \mathbb{C})$

---

**Input:** Automorphism group of algebra, basis of  $H_{\xi}^2(\mathfrak{A}, \mathbb{C})$ .  
**for**  $i=1$ ;  $Length[Automorphism]$  **do**  
  |  $ActAut[i]=Transpose[Automorphism[[i]]].MatrixFormH2.Automorphism[[i]]$   
**end**  
**Output:** Action of automorphism group on  $H_{\xi}^2(\mathfrak{A}, \mathbb{C})$ :  $ActAut[i]$

---

Next algorithm uses Algorithm 4 to compute bases for  $Ann(\mathfrak{A})$ . Again one needs to define the system of polynomial equations and call the solver.

---

**Algorithm 5:** Finding basis of annihilator.

---

**Input:** Multiplication of basis elements  
Define linear combinations of basis elements of algebra by  
 $SumElemOFAnn$ ;  
Define empty set by  $ProdSumElemOFAnn$ ;  
**for**  $i=1$ ;  $dim$  **do**  
  |  $pr[SumElemOFAnn, e[i]] == 0$  add to  $ProdSumElemOFAnn$ ;  
  |  $pr[e[i], SumElemOFAnn] == 0$  add to  $ProdSumElemOFAnn$ ;  
**end**  
Solve  $ProdSumElemOFAnn$  after obtained solution put to  
 $SumElemOFAnn$ ;  
**Output:** Basis of Annihilator:  $SpanAnn$ .

---

Finally, the last algorithm below uses outcome of Algorithm 5 and gives conditions of  $Ann(\mathfrak{A}) \cap Ann(\theta) = 0$  :

---

**Algorithm 6:** Intersection condition of  $Ann(\mathfrak{A})$  and  $Ann(\theta)$  :

---

**Input:**  $\mathfrak{A}$  is  $n$  dimensional algebra, basis of  $Ann(\mathfrak{A})$  and basis of  $H_{\xi}^2(\mathfrak{A}, \mathbb{C})$   
Define the following :  
Linear combinations of basis elements of  $Ann(\mathfrak{A})$  by  $SpanAnn1$ .  
Define  $\Delta_{i,j}$  as in (1);  
 $OpenBraket[x_, y_] := x[y]$ ;  
 $ConditionOfAnnAndTheta = \{\}$ ;  
**for**  $j;n+1$  **do**  
  | **for**  $i;Length[H2]$  **do**  
    |  $Sum[\alpha_i OpenBraket[H2[[i]], \{SpanAnn1[[1]], e[j]\}] == 0$  add to  
       $ConditionOfAnnAndTheta$ ;  
    |  $Sum[\alpha_i OpenBraket[H2[[i]], \{e[j], SpanAnn1[[1]]\}] == 0$  add to  
       $ConditionOfAnnAndTheta$ ;  
  | **end**  
**end**  
**Output:** Condition of  $Ann(\mathfrak{A}) \cap Ann(\theta) = 0$  :  $ConditionOfAnnAndTheta$  .

---

### 3 Application: classification of four dimensional nilpotent Moufang algebras.

To illustrate the code we now obtain new 4-dimensional algebra for the variety of Moufang algebra. For the detailed previous study on Moufang algebras see in [18], [20] and here we adopt the definition given by [22].

An algebra  $\mathfrak{M}$  is called Moufang algebra if it satisfies the following polynomial identities:

$$\begin{aligned}
 (x, y, z) &= -(z, y, x), \\
 ((xy)z)t + ((zy)x)t &= x(y(zt)) + z(y(xt)), \\
 t(x(yz) + z(yx)) &= ((tx)y)z + ((tz)y)x, \\
 (xy)(tz) + (zy)(tx) &= (x(yt))z + (z(yt))x,
 \end{aligned}
 \tag{2}$$

where  $(x, y, z) = (xy)z - x(yz)$ .

Firstly, we need to give for the code some information about Moufang algebra in the following form:

We first input the number of identities denoted by  $n$ . In our case we set

$$n = 4;$$

The multiplication of basis elements defined by  $\text{pr}[* , *]$ . The identities defined such functions  $\text{Ident}[i][\{x_-, y_-, z_-, \dots\}]$ , where  $i \in \{1, \dots, n\}$  and first product should be written with capital ‘‘Pr’’. Given identities should be in homogeneous multi-linear form. Also, we denote the length of monomials in identities by  $\text{LengthOfMonomial}[i]$  where  $i \in \{1, \dots, n\}$ . Since, we consider Moufang algebra we input the identities (2) in the following form

$$\begin{aligned}
 \text{Ident}[1][\{x_-, y_-, z_-\}] &:= \\
 \text{Pr}[\text{pr}[x, y], z] - \text{Pr}[x, \text{pr}[y, z]] + \text{Pr}[\text{pr}[z, y], x] - \text{Pr}[z, \text{pr}[y, x]] \\
 \text{LengthOfMonomial}[1] &= 3;
 \end{aligned}$$

$$\begin{aligned}
 \text{Ident}[2][\{x_-, y_-, z_-, t_-\}] &:= \\
 \text{Pr}[\text{pr}[\text{pr}[x, y], z], t] - \text{Pr}[x, \text{pr}[y, \text{pr}[z, t]]] + \\
 \text{Pr}[\text{pr}[\text{pr}[z, y], x], t] - \text{Pr}[z, \text{pr}[y, \text{pr}[x, t]]] \\
 \text{LengthOfMonomial}[2] &= 4;
 \end{aligned}$$

$$\begin{aligned}
 \text{Ident}[3][\{x_-, y_-, z_-, t_-\}] &:= \\
 \text{Pr}[\text{pr}[\text{pr}[t, x], y], z] - \text{Pr}[t, \text{pr}[x, \text{pr}[y, z]]] + \\
 \text{Pr}[\text{pr}[\text{pr}[t, z], y], x] - \text{Pr}[t, \text{pr}[z, \text{pr}[y, x]]] \\
 \text{LengthOfMonomial}[3] &= 4;
 \end{aligned}$$

$$\begin{aligned}
 \text{Ident}[4][\{x_-, y_-, z_-, t_-\}] &:= \\
 \text{Pr}[\text{pr}[x, y], \text{pr}[t, z]] + \text{Pr}[\text{pr}[z, y], \text{pr}[t, x]] - \\
 \text{Pr}[\text{pr}[x, \text{pr}[y, t]], z] - \text{Pr}[\text{pr}[z, \text{pr}[y, t]], x] \\
 \text{LengthOfMonomial}[4] &= 4;
 \end{aligned}$$

#### 3.1 Choosing three dimensional Moufang algebras from the literature

As mentioned before, various three dimensional nilpotent algebra classifications were considered in [4] including Moufang algebras. For our purposes, we need to

extract the three dimensional algebras from this article.

To this end, we need to input dimension of algebra, in the code it is given as *dim*. So,

```
dim =3;
```

Next step, we input multiplication table of algebra in the form  $\text{pr}[e[i], e[j]] = e[k]$  where  $i, j \in \{1, \dots, n\}$ . If for some  $i, j$  the product  $\text{pr}[e[i], e[j]] = 0$  then drop it. For instance,

```
pr [ e [ 1 ] , e [ 1 ] ] := e [ 2 ] ;
pr [ e [ 1 ] , e [ 2 ] ] := e [ 3 ] ;
pr [ e [ 2 ] , e [ 1 ] ] := e [ 3 ] ;
```

Now the code has all the information that it needs. Next we select from [17] all two dimensional Moufang algebras and from [4] three dimensional nilpotent Moufang algebras by using the following code:

```
ChangeTopr [ x_ ] := x / . { Pr -> pr }
Unident = { };
Do [ Unident =
  Union [ Unident ,
    Expand [ ChangeTopr [ Map [ Ident [ i ] ,
      Tuples [ basis , LengthOfMonomial [ i ] ] ] ] ] , { i , 1 , n } ]
If [ Unident == { 0 } ,
  Print [ Grid [ { Text @
    Style [ # , " TableHeader " ] & / @ { " The identities hold true . " } } ,
    Frame -> All ] ] ,
  Print [ Grid [ { Text @
    Style [ # , " TableHeader " ] & / @ { " The identities do not hold . " } } ,
    Frame -> All ] ] ]
```

As result we have all 2-dimensional Moufang algebras:

$$\begin{array}{l|l}
 \mathfrak{M}_{01}^2 & A_3 & : & e_1 e_1 = e_2; \\
 \mathfrak{M}_{02}^2 & D_2(0, 0) & : & e_1 e_1 = e_1; \\
 \mathfrak{M}_{03}^2 & D_2(1, 1) & : & e_1 e_1 = e_2, \quad e_1 e_2 = e_2, \quad e_2 e_1 = e_2; \\
 \mathfrak{M}_{04}^2 & E_1(0, 0, 0, 0) & : & e_1 e_1 = e_1, \quad e_2 e_2 = e_2; \\
 \mathfrak{M}_{05}^2 & E_5(0) & : & e_1 e_1 = e_1, \quad e_1 e_2 = e_1, \quad e_2 e_1 = e_2, \quad e_2 e_2 = e_2; \\
 \mathfrak{M}_{06}^2 & E_5(1) & : & e_1 e_1 = e_1, \quad e_1 e_2 = e_2, \quad e_2 e_1 = e_1, \quad e_2 e_2 = e_2;
 \end{array}$$

**Remark.** It follows from Theorem 3.3 in [17] that any nontrivial 2-dimensional Moufang algebra is isomorphic to one of the  $\mathfrak{M}_{0i}^2$  algebras, where  $i \in \{1, \dots, 6\}$ . Moreover, from [17] we have: the variety of 2-dimensional Moufang algebras has the following irreducible components defined by the algebras:  $\mathfrak{M}_{04}^2$ ,  $\mathfrak{M}_{05}^2$  and  $\mathfrak{M}_{06}^2$ .

Also, we get 3-dimensional nilpotent Moufang algebras:



$$\begin{aligned}
 \mathfrak{M}_{01}^3 & : e_1e_1 = e_2, & e_1e_2 = e_3, & e_2e_1 = e_3; \\
 \mathfrak{M}_{02}^3 & : e_1e_1 = e_2; \\
 \mathfrak{M}_{03}^3 & : e_1e_2 = e_3, & e_2e_1 = e_3; \\
 \mathfrak{M}_{04}^3 & : e_1e_2 = e_3, & e_2e_1 = -e_3; \\
 \mathfrak{M}_{05}^3(\lambda) & : e_1e_1 = \lambda e_3, & e_2e_1 = e_3 & e_2e_2 = e_3;
 \end{aligned}$$

**3.2 Second cohomology of three dimensional nilpotent Moufang algebras.**

Here, we apply our code constructed by Algorithm 1 and Algorithm 2 to obtain  $Z_{\mathbb{C}}^2(\mathfrak{A}, \mathbb{C})$ ,  $B_{\mathbb{C}}^2(\mathfrak{A}, \mathbb{C})$ , and  $H_{\mathbb{C}}^2(\mathfrak{A}, \mathbb{C})$ .

$\mathfrak{M}$	$Z^2(\mathfrak{M}, \mathbb{C})$	$B^2(\mathfrak{M}, \mathbb{C})$	$H^2(\mathfrak{M}, \mathbb{C})$
$\mathfrak{M}_{01}^3$	$\langle \begin{matrix} \Delta_{1,1}, \Delta_{1,2} + \Delta_{2,1}, \\ \Delta_{1,3} + \Delta_{2,2} + \Delta_{3,1} \end{matrix} \rangle$	$\langle \Delta_{1,1}, \Delta_{1,2} + \Delta_{2,1} \rangle$	$\langle [\Delta_{1,3}] + [\Delta_{2,2}] + [\Delta_{3,1}] \rangle$
$\mathfrak{M}_{02}^3$	$\langle \begin{matrix} \Delta_{1,1}, \Delta_{1,3}, \\ \Delta_{1,2} + \Delta_{2,1}, \Delta_{3,1}, \\ \Delta_{2,3} + \Delta_{3,2}, \Delta_{3,3} \end{matrix} \rangle$	$\langle \Delta_{11} \rangle$	$\langle [\Delta_{1,3}], [\Delta_{1,2}] + [\Delta_{2,1}], \\ [\Delta_{3,1}], [\Delta_{2,3}] + [\Delta_{3,2}], [\Delta_{3,3}] \rangle$
$\mathfrak{M}_{03}^3$	$\langle \begin{matrix} \Delta_{1,1}, \Delta_{1,2}, \Delta_{2,1}, \\ \Delta_{2,2}, \Delta_{1,3} + \Delta_{3,1}, \\ \Delta_{2,3} + \Delta_{3,2} \end{matrix} \rangle$	$\langle \Delta_{1,2} + \Delta_{2,1} \rangle$	$\langle [\Delta_{1,1}], [\Delta_{2,1}], [\Delta_{2,2}], \\ [\Delta_{1,3}] + [\Delta_{3,1}], [\Delta_{2,3}] + [\Delta_{3,2}] \rangle$
$\mathfrak{M}_{04}^3$	$\langle \begin{matrix} \Delta_{1,1}, \Delta_{1,2}, \Delta_{2,1}, \\ \Delta_{2,2}, \Delta_{1,3} - \Delta_{3,1}, \\ \Delta_{2,3} - \Delta_{3,2} \end{matrix} \rangle$	$\langle \Delta_{1,2} - \Delta_{2,1} \rangle$	$\langle [\Delta_{1,1}], [\Delta_{1,2}], [\Delta_{2,2}], \\ [\Delta_{1,3}] - [\Delta_{3,1}], [\Delta_{2,3}] - [\Delta_{3,2}] \rangle$
$\mathfrak{M}_{05}^3(\alpha)$	$\langle \Delta_{1,1}, \Delta_{1,2}, \Delta_{2,1}, \Delta_{2,2} \rangle$	$\langle \lambda \Delta_{1,1} + \Delta_{2,1} + \Delta_{2,2} \rangle$	$\langle [\Delta_{1,1}], [\Delta_{1,2}], [\Delta_{2,2}] \rangle$

We see from the above table that the second cohomology spaces and automorphism groups of  $\mathfrak{M}_{01}^3, \mathfrak{M}_{02}^3, \mathfrak{M}_{03}^3, \mathfrak{M}_{04}^3$ , and  $\mathfrak{M}_{05}^3(\lambda)$  algebras coincide with the algebras  $\mathfrak{J}_{01}^3, \mathfrak{J}_{01}^{3*}, \mathfrak{J}_{02}^{3*}, \mathfrak{J}_{03}^{3*}$ , and  $\mathfrak{J}_{04}^{3*}$  in [5] respectively. As such, it is clear that these algebras have the same central extensions. In other words, our calculations are expected to produce the same four dimensional algebras, see Table 1. Indeed, that is what we will get next which in particular gives us an opportunity to verify our code.

**3.3 Computation of central extensions of  $\mathfrak{M}_{01}^3$ .**

The next part is computing  $\text{Aut}(\mathfrak{M}_{01}^3)$ . This part constructed by Algorithm 3 and we get the automorphism group  $\text{Aut}(\mathfrak{M}_{01}^3)$  denoted by the set ‘‘Automorphism’’ in the code. In our particular case, we get the following output:

$$\begin{pmatrix} \lambda_{1,1} & 0 & 0 \\ \lambda_{2,1} & \lambda_{1,1}^2 & 0 \\ \lambda_{3,1} & 2\lambda_{1,1}\lambda_{2,1} & \lambda_{1,1}^3 \end{pmatrix},$$

By Algorithm 4 the action of  $\phi \in \text{Aut}(\mathfrak{M}_{01}^3)$  on the subspace  $[\theta] = \alpha_1([\Delta_{1,3}] + [\Delta_{2,2}] + [\Delta_{3,1}])$  gives the following table:

$\mathfrak{M}$	$\text{Aut}(\mathfrak{M})$	Matrix form of $H^2(\mathfrak{M}, \mathbb{C})$	Action of automorphism group on $H^2(\mathfrak{M}, \mathbb{C})$	Where $\alpha_i^*$ 's are equal:
$\mathfrak{M}_{01}^3$	$\begin{pmatrix} \lambda_{1,1} & 0 & 0 \\ \lambda_{2,1} & \lambda_{1,1}^2 & 0 \\ \lambda_{3,1} & 2\lambda_{1,1}\lambda_{2,1} & \lambda_{1,1}^3 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & \alpha_1 \\ 0 & \alpha_1 & 0 \\ \alpha_1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} \beta_1 & \beta_2 & \alpha_1^* \\ \beta_2 & \alpha_1^* & 0 \\ \alpha_1^* & 0 & 0 \end{pmatrix}$	$\alpha_1^* = \alpha_1 \lambda_{1,1}^4$

Also, one of the steps of the construction a non-split algebra is computing basis of annihilator and conditions required for  $\text{Ann}([\theta]) \cap \text{Ann}(\mathfrak{M}_{01}^3) = 0$ . By Algorithm 5

our code gives output `SpanAnn` which is the set of basis elements for  $Ann(\mathfrak{M}_{01}^3)$ . And by Algorithm 6 our code computes the conditions for  $Ann([\theta]) \cap Ann(\mathfrak{M}_{01}^3) = 0$ . That is, any element in  $u \in Ann(\mathfrak{M}_{01}^3)$  can be expressed by  $\lambda_3 e_3$  where  $\lambda_3 \in \mathbb{C}$  and it is defined in the code by `SpanAnn1`. Also, every  $[\theta] \in H^2(\mathfrak{M}\mathbb{C})$  is linear combinations of  $\alpha_1([\Delta_{1,3}] + [\Delta_{2,2}] + [\Delta_{3,1}])$ . Note that  $\Delta_{i,j}$  is function defined as (1). We just check if  $u = \lambda_3 e_3 \in \text{SpanAnn1}$ , when the following conditions  $\theta(e_j, u) = 0$  and  $\theta(u, e_j) = 0$  hold, where  $j \in \{1, \dots, n\}$ . So, we get the following output:

$Ann(\mathfrak{M}_{01}^3)$	Condition of $Ann([\theta]) \cap Ann(\mathfrak{M}_{01}^3)$
$\{e_3\}$	$\alpha_1 \lambda_3 = 0$

All of the above calculations are performed within seconds by using our code. At this stage, we need to obtain different orbits by hand. Since, we are interested in  $Ann([\theta]) \cap Ann(\mathfrak{M}_{01}^3) = 0$  we see that from above table that this holds whenever  $\alpha_1 \neq 0$ . If we take  $\lambda_{1,1} = \frac{1}{\sqrt[4]{\alpha_1}}$  we get  $\langle [\Delta_{1,3}] + [\Delta_{2,2}] + [\Delta_{3,1}] \rangle$ , which finally leads to new 4-dimensional nilpotent Moufang algebra from  $\mathfrak{M}_{01}^3$  :

$\mathfrak{M}_{01}^4$	$e_1 e_1 = e_2$	$e_1 e_2 = e_3$	$e_1 e_3 = e_4$	$e_2 e_1 = e_3$	$e_2 e_2 = e_4$	$e_3 e_1 = e_4$
-----------------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

### 4 Conclusion

In this article, we explained a unified approach to produce new nilpotent algebras which is one of the active research area in algebra. In part, we use the function “solve”, the built in symbolic solver of Wolfram Mathematica, in handling the system of polynomial equations. This is the main function that takes most of the compilation time. The codes written by the authors in other software including Matlab and Python gave worse results in terms of the running time and in some cases failing to provide any solutions. It is no doubt that solving system of symbolic nonlinear equations is not an easy task even for such advanced programming language. However, one may dig into the “solve” function and may improve to provide faster results for the kind of problems considered here.

### Acknowledgments.

First author acknowledges the support by a grant from the Ministry of Education and Science of the Republic of Kazakhstan within the framework of the Project “AP08051987”. Second author acknowledges the support by a grant from the Ministry of Education and Science of the Republic of Kazakhstan within the framework of the Project “AP09259551”. The authors are grateful to I. Kaygorodov for his essential comments.

### Appendix A.

Table 1: *The list of obtained 4-dimensional nilpotent Moufang algebras.*

$\mathfrak{M}$	Multiplication table			
$\mathfrak{M}_{01}^4$	$e_1 e_1 = e_2$	$e_1 e_2 = e_3$	$e_1 e_3 = e_4$	$e_2 e_1 = e_3$
Continued on next page				

**Table 1 – continued from previous page**

$\mathfrak{M}$	Multiplication table			
	$e_2e_2 = e_4$	$e_3e_1 = e_4$		
$\mathfrak{M}_{02}^4$ :	$e_1e_1 = e_2$	$e_2e_3 = e_4$	$e_3e_2 = e_4$	$e_3e_1 = e_4$
$\mathfrak{M}_{03}^4$ :	$e_1e_1 = e_2$	$e_1e_2 = e_4$	$e_2e_1 = e_4$	$e_3e_1 = e_4$
	$e_3e_3 = e_4$			
$\mathfrak{M}_{04}^4$ :	$e_1e_1 = e_2$	$e_1e_2 = e_4$	$e_2e_1 = e_4$	$e_3e_1 = e_4$
$\mathfrak{M}_{05}^4$ :	$e_1e_1 = e_2$	$e_2e_3 = e_4$	$e_3e_2 = e_4$	
$\mathfrak{M}_{06}^4$ :	$e_1e_1 = e_2$	$e_1e_2 = e_4$	$e_2e_1 = e_4$	$e_3e_3 = e_4$
$\mathfrak{M}_{07}^4$ :	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = e_3 + e_4$	$e_2e_3 = e_4$
	$e_3e_1 = e_4$	$e_3e_2 = e_4$		
$\mathfrak{M}_{08}^4$ :	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = e_3 + e_4$	$e_2e_2 = e_4$
	$e_3e_1 = e_4$			
$\mathfrak{M}_{09}^4$ :	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = e_3 + e_4$	$e_3e_1 = e_4$
$\mathfrak{M}_{10}^4$ :	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = e_3$	$e_2e_3 = e_4$
	$e_3e_1 = e_4$	$e_3e_2 = e_4$		
$\mathfrak{M}_{11}^4$ :	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = e_3$	$e_3e_1 = e_4$
$\mathfrak{M}_{12}^4$ :	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = e_3$	$e_2e_2 = e_4$
	$e_3e_1 = e_4$			
$\mathfrak{M}_{13}^4$ :	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = -e_3$	$e_3e_1 = -e_4$
$\mathfrak{M}_{14}^4$ :	$e_1e_1 = e_4$	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = -e_3$
	$e_3e_1 = -e_4$			
$\mathfrak{M}_{15}^4$ :	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = -e_3 + e_4$	$e_3e_1 = -e_4$
$\mathfrak{M}_{16}^4$ :	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = -e_3$	$e_2e_2 = e_4$
	$e_3e_1 = -e_4$			
$\mathfrak{M}_{17}^4$ :	$e_1e_1 = e_4$	$e_1e_2 = e_3$	$e_1e_3 = e_4$	$e_2e_1 = -e_3$
	$e_2e_2 = e_4$	$e_3e_1 = -e_4$		

## References

- [1] H. Abdelwahab, A.J. Calderón, I. Kaygorodov: The algebraic and geometric classification of nilpotent binary Lie algebras. *International Journal of Algebra and Computation* 29 (6) (2019) 1113–1129.
- [2] L. Camacho, I. Kaygorodov, V. Lopatkin, M. Salim: The variety of dual Mock-Lie algebras. *Communications in Mathematics* 28 (2) (2020) 161–178.
- [3] S. Cicalò, W. De Graaf, C. Schneider: Six-dimensional nilpotent Lie algebras. *Linear Algebra and its Applications* 436 (1) (2012) 163–189.
- [4] A.J. Calderón, A.F. Ouaridi, I. Kaygorodov: On the classification of bilinear maps with radical of a fixed codimension. *Linear and Multilinear Algebra* (2020) 1–24. DOI: 10.1080/03081087.2020.1849001
- [5] D. Jumaniyozov, I. Kaygorodov, A. Khudoyberdiyev: The algebraic and geometric classification of nilpotent noncommutative Jordan algebras. *Journal of Algebra and its Applications* (2020). DOI: 10.1142/S0219498821502029
- [6] W. De Graaf: Classification of nilpotent associative algebras of small dimension. *International Journal of Algebra and Computation* 28 (1) (2018) 133–161.
- [7] I. Gorshkov, I. Kaygorodov, M. Khrypchenko: The algebraic classification of nilpotent Tortkara algebras. *Communications in Algebra* 48 (8) (2020) 3608–3623.

- [8] I. Gorshkov, I. Kaygorodov, A. Kytmanov, M. Salim: The variety of nilpotent Tortkara algebras. *Journal of Siberian Federal University. Mathematics & Physics* 12 (2) (2019) 173–184.
- [9] M. Goze, E. Remm: 2-dimensional algebras. *African Journal Of Mathematical Physics* 10 (2) (2011) 81–91.
- [10] A.S. Hegazi, H. Abdelwahab: Classification of five-dimensional nilpotent Jordan algebras. *Linear Algebra and its Applications* 494 (2016) 165–218.
- [11] A.S. Hegazi, H. Abdelwahab: The classification of  $n$ -dimensional non-associative Jordan algebras with  $(n - 3)$ -dimensional annihilator. *Communications in Algebra* 46 (2) (2018) 629–643.
- [12] A.S. Hegazi, H. Abdelwahab, A.J. Calderón Martin: The classification of  $n$ -dimensional non-Lie Malcev algebras with  $(n - 4)$ -dimensional annihilator. *Linear Algebra and its Applications* 505 (2016) 32–56.
- [13] N. Ismailov, I. Kaygorodov, F. Mashurov: The Algebraic and Geometric Classification of Nilpotent Assosymmetric Algebras. *Algebras and Representation Theory* 24 (2021) 135–148.
- [14] S. Kadyrov, F. Mashurov: Nilpotent algebra classifier code. <https://www.wolframcloud.com/obj/39e8169f-8fbf-4540-aa65-4b01f187f63f> (2020)
- [15] I. Karimjanov, I. Kaygorodov, A. Khudoyberdiyev: The algebraic and geometric classification of nilpotent Novikov algebras. *Journal of Geometry and Physics* 143 (2019) 11–21.
- [16] I. Kaygorodov, M.P. Páez-Guillán, V. Voronin: The algebraic and geometric classification of nilpotent bicommutative algebras. *Algebras and Representation Theory* 23 (2020) 2331–2347.
- [17] I. Kaygorodov, Yu. Volkov: The variety of 2-dimensional algebras over an algebraically closed field. *Canadian Journal of Mathematics* 71 (4) (2019) 819–842.
- [18] E. K. Loginov: Embedding of Strongly Simple Moufang Loops In Simple Alternative Algebras. *Mathematical Notes* 54 (6) (1993) 1230–1235.
- [19] F. Mashurov, I. Kaygorodov: One-generated nilpotent assosymmetric algebras. *Journal of Algebra and Its Applications* (2020). DOI: 10.1142/S0219498822500311
- [20] I. Shestakov, J.M. Pérez-Izquierdo: An envelope for Malcev algebras. *Journal of Algebra* 272 (1) (2004) 379–393.
- [21] T. Skjelbred, T. Sund: On the classification of nilpotent Lie algebras. *Preprint series. Pure Mathematics* (1977). <http://urn.nb.no/URN:NBN:no-48289>
- [22] G.W. Zinbiel: Encyclopedia of types of algebras 2010. *Proceedings of the International Conference in Nankai Series in Pure Applied Mathematics and Theoretical Physics* 9 (2012) 217–298.

Received: 11 March, 2020

Accepted for publication: 16 May, 2020

Communicated by: Ivan Kaygorodov