

Article

Maple Encoding of Optimal Systems for Solvable Lie Algebras

Nimra Sher Asghar ^{1,2,*} , Hassan Azad ³ and Tooba Feroze ^{1,*} ¹ School of Natural Sciences, National University of Sciences and Technology, H-12, Islamabad 44000, Pakistan² School of Mathematical Sciences, University of Nottingham, University Park, Nottingham NG7 2RD, UK³ Abdus Salam School of Mathematical Sciences, Government College University, Lahore 54600, Pakistan; hassan.azad@sms.edu.pk

* Correspondence: nimra.phdmaths22sns@student.nust.edu.pk or nimra.sherasghar@nottingham.ac.uk (N.S.A.); tferoze@sns.nust.edu.pk (T.F.)

Abstract

This paper presents a Maple code for the systematic computation of optimal systems of solvable Lie algebras, built on an ideal based algorithmic framework. The main contribution is a fully automated Maple implementation that computes ideals, possible representative forms, adjoint factorizations, and conjugacy classes in a single reproducible workflow. The code is validated on a 9-dimensional solvable Lie algebra, where all ideals were determined in approximately five minutes, while the subsequent forms, adjoint factorizations, and conjugacy classes were each obtained in under a minute, a task that would take considerably longer manually. One-dimensional representatives are further extended to two-dimensional subalgebras through normalizers and quotient algebras. The code is then applied to the Lax equation, where its optimal system is used to derive invariant solutions, which are analyzed graphically to illustrate their behavior in detail.

Keywords: solvable lie algebra; ideals; conjugacy classes; invariant solutions**MSC:** 17B80

1. Introduction

An optimal system ($\mathcal{OS}$) is a collection of subalgebras chosen from a Lie algebra that aids in finding functionally independent invariant solutions of differential equations ($\mathcal{DE}$ s). In other words, the construction of an $\mathcal{OS}$ involves classifying subalgebras into inequivalence classes in such a way that every subalgebra is equivalent to one representative of these classes. Selecting one representative from each class ensures that all possible invariant solutions are considered without redundancy. This is one of the main reasons why optimal systems are important in symmetry analysis of differential equations.

The formation of one-dimensional $\mathcal{OS}$ s for Lie algebras was first introduced by Ovsiannikov. He employed the global matrix approach for the adjoint transformations and also discussed the construction of $\mathcal{OS}$ s of higher dimensions, as given in refs. [1,2].

Mathematical physics has witnessed substantial advancements in this technique. In particular, Patera et al. extended Ovsiannikov's work, as discussed in ref. [3]. Regarding $\mathcal{OS}$ s of higher-dimensional Lie algebras, Galas further advanced Ovsiannikov's approach by considering the problem of non-solvable algebras and by eliminating equivalent subalgebras, as discussed in ref. [4]. Patera and Winternitz provided conjugacy classes of all real Lie algebras of dimensions less than and equal to four. Their results were obtained by using the group of inner automorphisms, i.e., the Lie group obtained by exponentiating the adjoint representation of the considered Lie algebra, as discussed in ref. [5].



Academic Editor: Baccouch Mahboub

Received: 22 May 2026

Revised: 8 July 2026

Accepted: 10 July 2026

Published: 14 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Another way of determining $\mathcal{OS}$ s was introduced by Ibragimov in refs. [6–8]. In this approach, a generator is subjected to suitable adjoint transformations and is simplified as much as possible. This method is useful and has been widely applied in the construction of invariant solutions of $\mathcal{DE}$ s. However, it usually involves complex algebraic calculations and lacks an algorithmic approach. This situation leads to the problem of developing computational methods for determining conjugacy classes of subalgebras.

Addressing this challenge, Amata and Oliveri developed an algorithmic procedure, described in ref. [9], without detailed proofs. Later they introduced the *SymboLie* package for the automatic determination of $\mathcal{OS}$ s. Recently, Amata, Oliveri and Sgroi presented another computational approach for determining optimal systems, as given in ref. [10]. These works provide valuable symbolic tools for Lie subalgebra classification; however, they may leave out some Lie subalgebras in certain cases.

The present work follows a different approach which converts the ideal-based construction of Ali et al. [11] into a practical Maple procedure for solvable Lie algebras. Although optimal systems can be obtained manually for small algebras, the manual process requires repeated computation of ideals, possible representatives, adjoint transformations, normalizers, quotient algebras and removal of conjugate cases. For higher-dimensional algebras, this becomes time-consuming, and some ideals or conjugacy classes may be missed. The proposed code performs these repeated steps systematically and produces intermediate outputs, so the final representatives can be checked and reproduced with less manual effort. Compared with Ibragimov's direct adjoint simplification [6–8], the implementation first uses ideals to reduce the possible forms and then applies adjoint transformations to those reduced cases. Compared with *SymboLie* and related computational approaches [9,10], the present code emphasizes a transparent ideal-based workflow and also extends one-dimensional representatives to higher-dimensional subalgebras through normalizers and quotient algebras.

The reason for using this method is that ideals naturally describe the structure of solvable Lie algebras. If $\mathcal{L}$ is a solvable Lie algebra and $\mathcal{X} \in \mathcal{L}$, then $\mathcal{X}$ is contained in a smallest ideal of $\mathcal{L}$. This smallest ideal determines the possible shape of the one-dimensional subalgebra generated by $\mathcal{X}$. Therefore, by determining all ideals, one obtains a systematic starting point for finding conjugacy classes. The adjoint group is then factorized according to these ideals, and conjugate cases are removed. This method ensures that every possible conjugacy class is addressed without overlap because ideals naturally partition the Lie algebra into disjoint substructures. This approach not only simplifies the analysis but also ensures that no conjugacy class is left out.

Researchers and practitioners across a range of domains can find the methodology more accessible and user-friendly as it offers a streamlined and efficient method of handling Lie symmetry generators and associated $\mathcal{OS}$ s. The organization of this article is as follows: In Section 2, we apply the algorithms given in ref. [11] to develop the theoretical framework for determining all ideals, possible forms, adjoint factorizations, and conjugacy classes of one-dimensional subalgebras within solvable Lie algebras; this section illustrates the overall computational workflow for constructing ideals and conjugacy classes of subalgebras of solvable Lie algebras using a flowchart. In Section 3, this framework is implemented in Maple and validated through a 9-dimensional (9- $\mathcal{D}$) solvable Lie algebra, ensuring the accuracy of the determined ideals and conjugacy classes. In Section 4, two-dimensional subalgebras are computed by extending the one-dimensional representatives via the normalizer construction. In Section 5, the application of this program in the domain of $\mathcal{DE}$ s is carried out: the Lie symmetries of the provided $\mathcal{DE}$ are first identified, and once the corresponding Lie algebra is obtained, the program is used to determine the $\mathcal{OS}$ s. These systems are then instrumental in determining the invariant solutions of the partial differen-

tial equations ($\mathcal{PDEs}$), providing deeper insight into their structure and potential solutions; the invariant solutions obtained are thoroughly analyzed, with a focus on their physical behavior and characteristics. Lastly, in Section 6, we present our conclusions.

2. Computational Framework for Constructing Optimal Systems of Solvable Lie Algebras

The theoretical basis of the present implementation is the ideal-based construction of conjugacy classes of subalgebras of finite-dimensional solvable Lie algebras developed in [11]. Let $\mathcal{L}$ be a finite-dimensional solvable Lie algebra. A subspace $\mathcal{I} \subseteq \mathcal{L}$ is an ideal if:

$$[\mathcal{X}, \mathcal{Y}] \in \mathcal{I}, \quad \forall \mathcal{X} \in \mathcal{L}, \quad \forall \mathcal{Y} \in \mathcal{I}.$$

Ideals are important here because they are invariant under inner automorphisms and therefore provide natural containers for conjugacy classes. If $\mathcal{Y} \in \mathcal{L}$, then the smallest ideal containing $\mathcal{Y}$ determines the shape from which the corresponding one-dimensional subalgebra is classified. Once these forms are known, the possible forms of one-dimensional subalgebras can be determined. The adjoint action is then applied only to these forms in order to remove conjugate cases. The implemented procedure is summarized step by step using the flowchart in Figure 1.

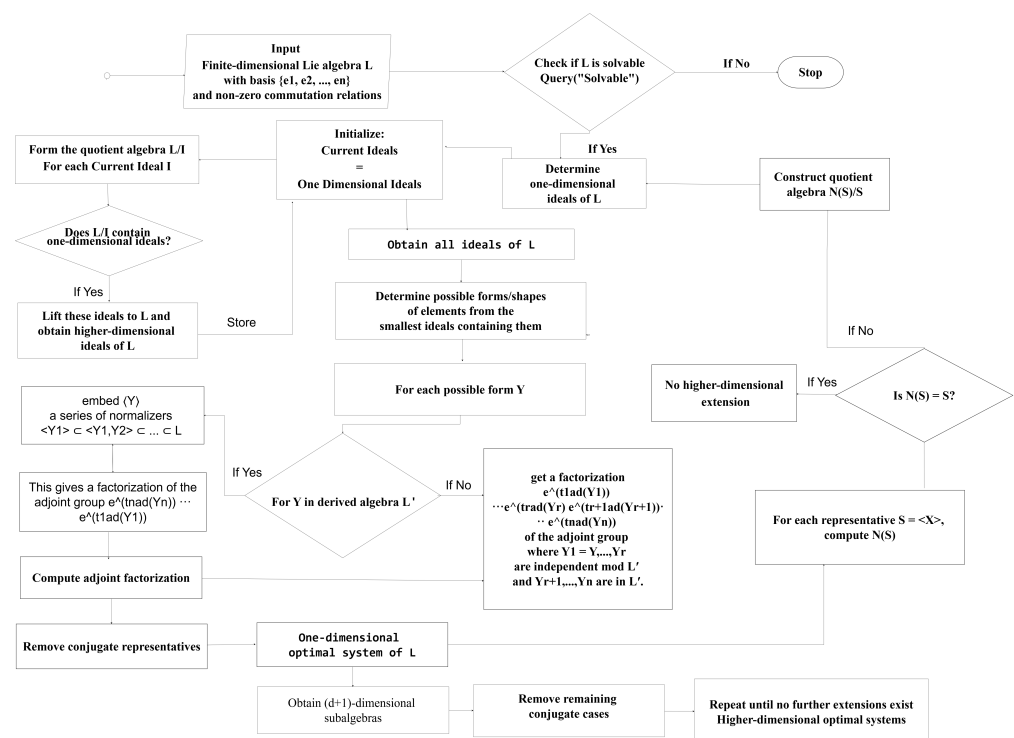


Figure 1. Computational workflow for constructing optimal systems of finite-dimensional solvable Lie algebras. **Step 1:** input the basis and commutation relations of $\mathcal{L}$, and verify solvability through the derived series. **Step 2:** compute one-dimensional ideals and then construct higher-dimensional ideals recursively by using quotient algebras. **Step 3:** determine possible representative forms from the smallest ideals containing the elements. **Step 4:** compute adjoint factorizations associated with these forms and remove conjugate representatives. **Step 5:** for higher-dimensional systems, extend each representative $S = \langle \mathcal{X} \rangle$ by computing its normalizer $N(S) = \{ \mathcal{Y} \in \mathcal{L} : [\mathcal{Y}, S] \subseteq S \}$ and classify one-dimensional subalgebras of the quotient algebra $N(S)/S$; the quotient representatives are lifted back to $\mathcal{L}$, and the procedure is repeated until no further extensions exist.

3. Maple Implementation and Validation on a Nine-Dimensional Solvable Lie Algebra

This section implements the framework of Section 2 in Maple and validates it on a nine-dimensional solvable Lie algebra. The presentation follows the main computational stages: construction of ideals, determination of possible representative forms, adjoint factorization, and extraction of one-dimensional conjugacy classes.

Consider the nine-dimensional Lie algebra with basis $\langle e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9 \rangle$. The non-zero commutation relations are:

$$\begin{aligned} [e_1, e_4] &= e_2, & [e_1, e_5] &= e_3, & [e_1, e_6] &= e_1, & [e_1, e_8] &= e_1, \\ [e_2, e_6] &= e_2, & [e_2, e_7] &= -e_3, & [e_3, e_6] &= e_3, & [e_3, e_7] &= e_2, \\ [e_4, e_7] &= -e_5, & [e_4, e_8] &= -e_4, & [e_6, e_7] &= e_4, & [e_5, e_8] &= -e_5. \end{aligned} \quad (1)$$

The 9- $\mathcal{D}$ Lie algebra is entered into Maple by specifying its basis and non-zero commutation relations, as shown in Listing 1.

Listing 1: Maple input for the nine-dimensional Lie algebra.

```
VectStrEq := ['Input The Lie Algebra'], ['Input The Basis'];
g := LieAlgebraData(VectStrEq, L1);
DGsetup(g, VectStrEq[2], [alpha]);
Basis := DGinfo(L1, "FrameBaseVectors");
```

The solvability of the algebra is verified by the command in Listing 2:

Listing 2: Maple command for verifying solvability.

```
Query("Solvable");
```

Which examines the derived series of the algebra. For the Lie algebra $\mathcal{L}$ defined by (1), Maple returns `true`. Hence, the code is applicable. The input and solvability verification are shown in Figure 2. The other functions used in the code are defined in Appendix A.

```
Lie_Algebra := op(VectStrEq[1]);
[e1, e4] = e2, [e1, e5] = e3, [e1, e6] = e1, [e1, e8] = e1, [e2, e6] = e2, [e2, e7] = -e3, [e3, e6] = e3,
[e3, e7] = e2, [e4, e7] = -e5, [e4, e8] = -e4, [e5, e7] = e4, [e5, e8] = -e5
Dimension_of_Lie_Algebra := print(nops(VectStrEq[2]), Dimensional Lie Algebra);
                               9, Dimensional Lie Algebra
Lie_Algebra_is_Solvable := Query("Solvable");
                               true
```

Figure 2. Input of the 9- $\mathcal{D}$ Lie algebra and verification of solvability.

The implementation proceeds through the following steps.

Step 1: How to Find All the Ideals of the Lie Algebra

The first step determines all ideals of the solvable Lie algebra $\mathcal{L}$. The computation begins with one-dimensional ideals, which are then used as initial data for the recursive construction of higher-dimensional ideals through quotient algebras. The Maple execution of this step is given below.

Listing 3: Maple code for determining all ideals of the Lie algebra.

```

All_Ideals := [ ];
onedimensionalideals := FindONEDIMIdeals();
current_ideal_list := onedimensionalideals;
if current_ideal_list = [ ] then
    d_alggebra := DerivedAlgebra();
    d_alggebra := sort(d_alggebra);
    if d_alggebra <> [ ] then
        d_al:=evalDG([LinearAlgebra:-DotProduct([seq(1,i=1..nops(d_alggebra))],
        [op(d_alggebra)], conjugate = false)]);
        dall := MyFunction(op(d_al));
        g_prime := dall;
        subsets := generateSubsets(g_prime);
        subsets := 'minus'({op(subsets)}, {[ ]});
        ide := [ ];
        for ele in subsets do
            if Query(ele, "Subalgebra") then
                if Query(ele, "Ideal") then
                    ide := [op(ide), ele]:
                end if:
            end if:
        end do;
        current_ideal_list := ide:
    end if:
end if;
All_Ideals := [op(All_Ideals), op(current_ideal_list)];
Current_ideal := [ ];
for elements in current_ideal_list do
    idealss_id:=evalDG([LinearAlgebra:-DotProduct([seq(1,i=1..nops(elements))],
    elements,conjugate = false)]);
    co_effs:=DGinfo(evalDG(op(idealss_id)),"CoefficientSet",{op(VectStrEq[2])});
    if nops(co_effs) = 1 then
        Current_ideal := [op(Current_ideal), elements]:
    else
        co_effs := 'minus'(co_effs, {1});
        eqs := map(proc (c) options operator, arrow; c = 0 end proc,co_effs):
        elements := subs(eqs, elements):
        Current_ideal := [op(Current_ideal), elements]:
    end if:
end do;
current_ideal_list := Current_ideal;
current_ideal_list := map(sortList, current_ideal_list);
current_ideal_list := convert(current_ideal_list, set);
for elem to nops(VectStrEq[2]) do
    ideal_list := find_other_ideal();
    ideal_list := convert(ideal_list, set);
    ideal_list := [op(ideal_list)];
    idealss := ideal_list;
    current_ideal_list := idealss;
    if current_ideal_list = [ ] then
        d_alggebra := DerivedAlgebra();
        d_alggebra := sort(d_alggebra);
        if d_alggebra <> [ ] then
            d_al:=evalDG([LinearAlgebra:-DotProduct([seq(1,i=1..nops(d_alggebra))],
            [op(d_alggebra)],conjugate = false)]);
            dall := MyFunction(op(d_al));
            g_prime := dall;
            subsets := generateSubsets(g_prime);
            subsets := 'minus'({op(subsets)}, {[ ]});
            ide := [ ];
            for ele in subsets do
                if Query(ele, "Subalgebra") = true then

```

```

        if Query(ele, "Ideal") = true then
            ide := [op(ide), ele]:
        end if:
    end if:
end do;
current_ideal_list := ide:
end if:
end if;
All_Ideals := [op(All_Ideals), op(current_ideal_list)];
Current_ideal := [ ];
for elements in current_ideal_list do
    idealss_id:=evalDG([LinearAlgebra:-DotProduct([seq(1,i=1..nops(elements)),
    elements, conjugate = false)]);
    co_effs := DGinfo(evalDG(op(idealss_id)), "CoefficientSet",{op(VectStrEq[2])})
    ;
    if nops(co_effs) = 1 then
        Current_ideal := [op(Current_ideal), elements];
    else
        co_effs := `minus` (co_effs, {1});
        eqs := map(proc (c) options operator, arrow; c = 0 endproc, co_effs);
        elements := subs(eqs, elements);
        Current_ideal := [op(Current_ideal), elements];
    end if:
end do:
current_ideal_list := Current_ideal:
current_ideal_list := map(sortList, current_ideal_list):
current_ideal_list := convert(current_ideal_list, set):
end do:
iideals := [ ];
for elements in All_Ideals do
    elements := sort(elements);
    iideals := [op(iideals), elements];
end do:
All_Ideals_algebraa := convert(iideals, set);
All_Idealss_algebra:=`minus`({op(All_Ideals_algebraa)},{All_Ideals_algebraa[-1]});
reem_el := `minus`({op(VectStrEq[2])}, {op(All_Ideals_algebraa[-2])});
result := Query([op(reem_el), op(All_Ideals_algebraa[-2])],
"Ideal");
if result = true then
All_Ideals_algebraa:=[op(All_Idealss_algebra),[op(reem_el),op(All_Ideals_algebraa
[-2])]]:
end if:
All_Ideals_algebra := All_Ideals_algebraa;
originalList := All_Ideals_algebra;
for elements in originalList do
    for i to nops(originalList) do
        or_result := has(elements, {a[i]});
        if or_result = true then
            elo := `minus`({op(elements)}, {elements[1]});
            originalList := remove(has, originalList, {sort([op(elements[1])[1],
            op(elo)])}):
        end if:
    end do:
end do:
All_Ideals_of_liealgebra := originalList;

```

Applying this code to the nine-dimensional Lie algebra generates the complete list of ideals of $\mathcal{L}$. The output is shown in Figure 3.

<i>Ideals</i>	<i>All_Ideals_of_lie_algebra</i> $[[e9], [e2, e3], [e1, e2, e3], [e2, e3, e9], [e6 + a_2 e9, e1, e2, e3], [e1, e2, e3, e9], [e2, e3, e4, e5], [e7 + a_2 e9, e2, e3, e4, e5], [e1, e2, e3, e4, e5], [e1, e2, e3, e6, e9], [e2, e3, e4, e5, e9], [e8 + a_4 e9, e1, e2, e3, e4, e5], [e7 + a_3 e8 + a_4 e9, e1, e2, e3, e4, e5], [e6 + a_2 e7 + a_3 e8 + a_4 e9, e1, e2, e3, e4, e5], [e1, e2, e3, e4, e5, e9], [e2, e3, e4, e5, e7, e9], [e7 + a_3 e8, e1, e2, e3, e4, e5, e9], [e7 + a_3 e9, e1, e2, e3, e4, e5, e8], [e8 + a_3 e9, e1, e2, e3, e4, e5, e6], [e8 + a_3 e9, e1, e2, e3, e4, e5, e7], [e6 + a_2 e7 + a_3 e8, e1, e2, e3, e4, e5, e9], [e6 + a_2 e7 + a_3 e9, e1, e2, e3, e4, e5, e8], [e6 + a_2 e8 + a_3 e9, e1, e2, e3, e4, e5, e7], [e7 + a_2 e8 + a_3 e9, e1, e2, e3, e4, e5, e6], [e1, e2, e3, e4, e5, e8, e9], [e6 + a_2 e7, e1, e2, e3, e4, e5, e8, e9], [e6 + a_2 e8, e1, e2, e3, e4, e5, e7, e9], [e6 + a_2 e9, e1, e2, e3, e4, e5, e7, e8], [e7 + a_2 e8, e1, e2, e3, e4, e5, e6, e9], [e7 + a_2 e9, e1, e2, e3, e4, e5, e6, e8], [e8 + a_2 e9, e1, e2, e3, e4, e5, e6, e7], [e1, e2, e3, e4, e5, e7, e8, e9], [e6, e1, e2, e3, e4, e5, e7, e8, e9]]$
---------------	---

Figure 3. All Ideals of the defined 9- $\mathcal{D}$ Lie algebra

Step 2: Determining the Possible Shapes

Starting from the ideals of smallest dimension, the code determines the possible representative forms of elements of $\mathcal{L}$. The execution of this step in Maple is as follows:

Listing 4: Maple code for determining the possible representative forms.

```

constantt := {seq(a[i], i = 1 .. nops(VectStrEq[2]))};
updateid := [];
Idealss := 'minus'({op(All_Ideals_of_liealgebra)}, {All_Ideals_of_liealgebra[-1]});
for elements in Idealss do
    has_element := has(elements, constantt);
    if has_element = true then
        result_id := evalDG([LinearAlgebra:-DotProduct([seq(1, i=1..nops(elements))],
            elements, conjugate = false)]);
        result_id := MyFunction(op(result_id));
        check_r := member(result_id, Idealss);
        if 'or'(check_r = true, nops(result_id) = nops(VectStrEq[2])) then
            updateid := [op(updateid), elements];
        end if;
    end if;
end do;
UpdateID := updateid;
ideals_algebra := 'minus'(Idealss, {op(UpdateID)});
if Query("Abelian") then
    Form := FindONEDIMIdeals();
else
    idealss := [];
    for element in ideals_algebra do
        idealssid := evalDG([LinearAlgebra:-DotProduct([seq(1, i=1..nops(element))],
            element, conjugate = false)]);
        idealssid := MyFunction(op(idealssid));
        idealss := [op(idealss), idealssid];
    end do;
    idealss := convert(idealss, set);
    idealss := [op(idealss), All_Ideals_of_liealgebra[-1]];
    ceff := cat(c, 1 .. 20);
    newestlist := [];
    for i to nops(idealss) do

```

```

constant_list := [1, seq(ceff[i][k], k = 1 .. nops(idealss[i])-1)];
newestlist := [op(newestlist), constant_list];
end do;
forms := CalculateDotProducts(newestlist, idealss);
Form := forms:
end if;
Form := Form;

```

The output of this step gives the possible forms of the elements in the 9- $\mathcal{D}$ Lie algebra. These forms are shown in Figure 4.

Shapes	Form
	$ \begin{aligned} & [e^9], [e^2 + c_2 e^3], [e^1 + c_3 e^2 + c_3 e^3], [e^2 + c_4 e^3 + c_4 e^9], [e^1 + c_5 e^2 \\ & + c_5 e^3 + c_5 e^9], [e^2 + c_6 e^3 + c_6 e^4 + c_6 e^5], [e^1 + c_7 e^2 + c_7 e^3 \\ & + c_7 e^4 + c_7 e^5], [e^1 + c_8 e^2 + c_8 e^3 + c_8 e^6 + c_8 e^9], [e^2 + c_9 e^3 \\ & + c_9 e^4 + c_9 e^5 + c_9 e^9], [e^1 + c_{10} e^2 + c_{10} e^3 + c_{10} e^4 + c_{10} e^5 \\ & + c_{10} e^9], [e^2 + c_{11} e^3 + c_{11} e^4 + c_{11} e^5 + c_{11} e^7 + c_{11} e^9], [e^1 \\ & + c_{12} e^2 + c_{12} e^3 + c_{12} e^4 + c_{12} e^5 + c_{12} e^8 + c_{12} e^9], [e^1 + c_{13} e^2 \\ & + c_{13} e^3 + c_{13} e^4 + c_{13} e^5 + c_{13} e^6 + c_{13} e^8 + c_{13} e^9], [e^1 + c_{14} e^2 \\ & + c_{14} e^3 + c_{14} e^4 + c_{14} e^5 + c_{14} e^7 + c_{14} e^8 + c_{14} e^9], [e^6 + c_{15} e^1 \\ & + c_{15} e^2 + c_{15} e^3 + c_{15} e^4 + c_{15} e^5 + c_{15} e^7 + c_{15} e^8 + c_{15} e^9] \end{aligned} $

Figure 4. Possible shapes of the defined 9- $\mathcal{D}$ Lie algebra

Step 3: Determining the Factorization of the Adjoint Group

If the form is an ideal, it cannot be conjugate to any other element of $\mathcal{L}$. If it is not an ideal, the code checks whether it lies in the derived algebra $\mathcal{L}'$. In that case, the element $\mathcal{Y}$ is inserted from the $\mathcal{L}'$ into a normalizer series $\langle \mathcal{Y} = \mathcal{Y}_1 \rangle$, $\langle \mathcal{Y}_1, \mathcal{Y}_2 \rangle$, and so on until the series reaches the full Lie algebra, or until no further enlargement is possible. When the full algebra is reached, this series gives a factorization of the adjoint group into exponential factors $e^{\epsilon_k ad(\mathcal{Y}_k)} \dots e^{\epsilon_1 ad(\mathcal{Y}_1)}$, where ad is the adjoint representation is given by:

$$ad(e^{\epsilon \mathcal{Y}_i} \mathcal{Y}_j) = \mathcal{Y}_j + \epsilon [\mathcal{Y}_i, \mathcal{Y}_j] + \frac{\epsilon^2}{2!} [\mathcal{Y}_i, [\mathcal{Y}_i, \mathcal{Y}_j]] + \frac{\epsilon^3}{3!} [\mathcal{Y}_i, [\mathcal{Y}_i, [\mathcal{Y}_i, \mathcal{Y}_j]]] + \dots \quad (2)$$

If $\mathcal{Y} \notin \mathcal{L}'$, then the adjoint group will be decomposed into $e^{\epsilon_1 ad(\mathcal{Y}_1)} \dots e^{\epsilon_m ad(\mathcal{Y}_m)} e^{\epsilon_{m+1} ad(\mathcal{Y}_{m+1})} \dots e^{\epsilon_k ad(\mathcal{Y}_k)}$, where $\mathcal{Y}_{m+1} \dots \mathcal{Y}_k \in \mathcal{L}'$ and these $\mathcal{Y}_1 = \mathcal{Y}, \dots, \mathcal{Y}_m$ elements not in $\mathcal{L}'$. The Maple execution of this step is as follows:

Listing 5: Maple code for determining the adjoint factorizations.

```

d_algebra := DerivedAlgebra();
d_algebra := sort(d_algebra);
if d_algebra <> [] then
  d_al:=evalDG(LinearAlgebra:-DotProduct([seq(1,i=1..
    nops(d_algebra))],[op(d_algebra)], conjugate = false));
  d_algebra := MyFunction(d_al):
end if;
g_prime := d_algebra:

List_of_Adjoint_Factorization_with_ID_subspaces := [ ]:
for element in Form do
  result := Query(element, "Ideal");
  if result = true then

```

```

List_of_Adjoint_Factorization_with_ID_subspaces:={op(List
_of_Adjoint_Factorization_with_ID_subspaces), [[ ],
element]]:
else
  dcall := MyFunction(op(element)):
  ooo := {op(dcall)}:
  res := check_elements(g_prime, ooo);
  if res = true then
    l_list := [];
    Sub_Nor := element;
    l_list := [op(l_list), op(Sub_Nor)];
    Sub_Nor := SubalgebraNormalizer(Sub_Nor);
    l_list := [l_list, Sub_Nor];
    Sub_Nor := DGinfo(evalDG(LinearAlgebra:-DotProduct([seq(l, i = 1 ..
nops(Sub_Nor)]), [op(Sub_Nor)], conjugate = false)), "ObjectGenerators");
    aab1 := [op(Sub_Nor)];
    dcall1 := [];
    for eles in aab1 do
      c_all1 := [_DG(["vector", L1, []], [[eles], 1]]);
      dcall1 := [op(dcall1), op(c_all1)];
    end do;
    Sub_Nor := [op(dcall1)];
    Sub_Nor := SubalgebraNormalizer(Sub_Nor);
    l_list := [op(l_list), Sub_Nor];
    if nops(Sub_Nor) <> nops(VectStrEq[2]) then
      Sub_Nor := SubalgebraNormalizer(Sub_Nor);
      l_list := [l_list, Sub_Nor];
    end if;
    Normalizer_Series := l_list;
    K := [ooo[1]];
    Fac := 'minus'({op(Normalizer_Series[-1])}, {op(K)});
    for ee in Fac do
      ABE := Query([ee, op(element)], "Abelian");
      if ABE = true then
        Fac := 'minus'(Fac, {ee}):
      end if;
    end do;
    adj_fac := [op(Fac)];
    List_of_Adjoint_Factorization_with_ID_subspaces := {op(
List_of_Adjoint_Factorization_with_ID_subspaces),
[adj_fac, element]}:
  else
    bas := DGinfo(L1, "FrameBaseVectors");
    c_R := 'minus'({op(bas)}, {op(g_prime)});
    fac_R := {op(c_R), op(g_prime)};
    c_R := 'minus'({op(bas)}, {op(element[1])[1]});
    c_R := [op(c_R)];
    for ee in c_R do
      ABE := Query([ee, op(element)], "Abelian");
      if ABE = true then
        c_R := 'minus'({op(c_R)}, {ee}):
      end if;
    end do;
    c_R_adj_fac := [op(c_R)];
    List_of_Adjoint_Factorization_with_ID_subspaces := {op
(List_of_Adjoint_Factorization_with_ID_subspaces),
[c_R_adj_fac, element]}:
  end if;
end if:
end do:
List_of_Adjoint_Factorization_with_ID_subspaces;
List_of_Adjoint_Factorization_with_forms := List_of_Adjoint_
Factorization_with_ID_subspaces;

```

The program generates an adjoint factorization for each possible form of the 9- $\mathcal{D}$ Lie algebra. The output is shown in Figure 5, where each form is displayed with its corresponding adjoint factors so that the classification can be verified from the intermediate data.

Adjoint Factorizations	List_of_Adjoint_Factorization_with_forms
	$\{[[], [e_9]], [[e_6, e_7], [e_2 + c_{21} e_3]], [[e_6, e_7], [e_2 + c_{41} e_3 + c_{42} e_9]], [[e_6, e_7, e_8], [e_2 + c_{61} e_3 + c_{62} e_4 + c_{63} e_5]], [[e_1, e_6, e_7, e_8], [e_2 + c_{91} e_3 + c_{92} e_4 + c_{93} e_5 + c_{94} e_9]], [[e_4, e_5, e_6, e_7, e_8], [e_1 + c_{31} e_2 + c_{32} e_3]], [[e_4, e_5, e_6, e_7, e_8], [e_1 + c_{51} e_2 + c_{52} e_3 + c_{53} e_9]], [[e_4, e_5, e_6, e_7, e_8], [e_1 + c_{71} e_2 + c_{72} e_3 + c_{73} e_4 + c_{74} e_5]], [[e_4, e_5, e_6, e_7, e_8], [e_1 + c_{101} e_2 + c_{102} e_3 + c_{103} e_4 + c_{104} e_5 + c_{105} e_9]], [[e_4, e_5, e_6, e_7, e_8], [e_1 + c_{121} e_2 + c_{122} e_3 + c_{123} e_4 + c_{124} e_5 + c_{125} e_8 + c_{126} e_9]], [[e_1, e_2, e_3, e_4, e_5, e_7, e_8], [e_6 + c_{151} e_1 + c_{152} e_2 + c_{153} e_3 + c_{154} e_4 + c_{155} e_5 + c_{156} e_7 + c_{157} e_8 + c_{158} e_9]], [[e_1, e_3, e_4, e_5, e_6, e_7, e_8], [e_2 + c_{111} e_3 + c_{112} e_4 + c_{113} e_5 + c_{114} e_7 + c_{115} e_9]], [[e_2, e_3, e_4, e_5, e_6, e_7, e_8], [e_1 + c_{81} e_2 + c_{82} e_3 + c_{83} e_6 + c_{84} e_9]], [[e_2, e_3, e_4, e_5, e_6, e_7, e_8], [e_1 + c_{131} e_2 + c_{132} e_3 + c_{133} e_4 + c_{134} e_5 + c_{135} e_6 + c_{136} e_8 + c_{137} e_9]], [[e_2, e_3, e_4, e_5, e_6, e_7, e_8], [e_1 + c_{141} e_2 + c_{142} e_3 + c_{143} e_4 + c_{144} e_5 + c_{145} e_7 + c_{146} e_8 + c_{147} e_9]]\}$

Figure 5. Adjoint factorization of the defined 9- $\mathcal{D}$ Lie algebra

Step 4: Determining Conjugacy Classes of One-Dimensional Subalgebras

Finally, the adjoint factorizations obtained in Step 3 are applied to the possible forms obtained in Step 2. This gives the conjugacy classes of one-dimensional subalgebras of $\mathcal{L}$. The Maple function `adj()` implements the adjoint action given in (2). The Maple execution is as follows:

Listing 6: Maple procedure for computing one-dimensional conjugacy classes.

```

adj := proc (a, b, t)
    local term, series, m;
    series := b;
    term := b;
    m := 1;
    if a = [ ] then
        series := b;
    else
        while m <= 4 do
            term := LieBracket(a, term);
            series := series + t^m * term / factorial(m);
            m := m + 1;
        end do;
    end if;
    return series
end proc;

#In the adj function, if the coefficients of the result is a
series, the function returns the first five terms of the
exponential series.
conjugacy_class := [ ];
res_ult := evalDG(adj('Input the Adjoint_Factorization', 'Input
the ID_subspaces ', ' Input any variable e.g r,s,t etc.));
yes_conjugacy_class := evalDG(res_ult);
conjugacy_class := [op(conjugacy_class), yes_conjugacy_class];
OneDimensional_Conjugacy_Classes := conjugacy_class;

```

By inserting the forms and their corresponding adjoint factorizations into the ‘adj’ function, we obtain the conjugacy classes of the one-dimensional subalgebra of the 9- $\mathcal{D}$ solvable Lie algebra. The output is shown in Figure 6.

Conjugacy Classes	<i>One_Dimensional_Conjugacy_Classes</i> where, $\epsilon = \pm 1$ $[e_9, e_2 + c_2 e_3, e_2 + c_4 e_3 + \epsilon e_9, e_2 + c_6 e_3 + \epsilon e_4 + \epsilon e_5, e_4 + c_9 e_5$ $+ \epsilon e_9, e_1, e_1 + c_5 e_9, e_1 + \epsilon e_4 + \epsilon e_5, e_1 + \epsilon e_4 + \epsilon e_5 + \epsilon e_9, e_1 + \epsilon e_8$ $+ \epsilon e_9, A e_1 + e_6 + c_{15_6} e_7 + c_{15_7} e_8 + c_{15_8} e_9, e_7 + c_{11_5} e_9, e_1 + \epsilon e_6$ $+ \epsilon e_9, e_1 + \epsilon e_6 + \epsilon e_8 + \epsilon e_9, e_1 + \epsilon e_7 + \epsilon e_8 + \epsilon e_9]$ where, $\epsilon = \pm 1$
-------------------	---

Figure 6. Conjugacy classes of the defined nine-dimensional solvable Lie algebra.

The one-dimensional subalgebras listed in Figure 6 are non-conjugate under the adjoint action. Therefore, they form a one-dimensional optimal system of 9- $\mathcal{D}$ Lie algebra (1) as follows:

$$\begin{aligned}
 \mathcal{OS}_1(\mathcal{L}) = & \langle e_9 \rangle, \quad \langle e_1 \rangle, \quad \langle e_2 + c_2 e_3 \rangle, \quad \langle e_2 + c_4 e_3 + \epsilon e_9 \rangle, \\
 & \langle e_2 + c_6 e_3 + \epsilon e_4 + \epsilon e_5 \rangle, \quad \langle e_1 + c_5 e_9 \rangle, \quad \langle e_1 + \epsilon e_4 + \epsilon e_5 \rangle, \\
 & \langle e_1 + \epsilon e_4 + \epsilon e_5 + \epsilon e_9 \rangle, \quad \langle e_1 + \epsilon e_8 + \epsilon e_9 \rangle, \quad \langle e_7 + c_{11_5} e_9 \rangle, \\
 & \langle A e_1 + e_6 + c_{15_6} e_7 + c_{15_7} e_8 + c_{15_8} e_9 \rangle, \quad \langle e_1 + \epsilon e_6 + \epsilon e_9 \rangle, \\
 & \langle e_1 + \epsilon e_6 + \epsilon e_8 + \epsilon e_9 \rangle, \langle e_1 + \epsilon e_7 + \epsilon e_8 + \epsilon e_9 \rangle, \\
 & \langle e_4 + c_9 e_5 + \epsilon e_9 \rangle, \quad \epsilon = \pm 1.
 \end{aligned} \tag{3}$$

This validation confirms that the Maple implementation follows the ideal-based procedure and produces the required representatives for the considered nine-dimensional solvable Lie algebra. These results confirm the accuracy and efficiency of the code in handling high-dimensional solvable Lie algebras and provide a robust foundation for further applications in symmetry analysis and invariant solutions of $\mathcal{PDE}$ s.

4. Two-Dimensional Subalgebras

In this section, we use the one-dimensional optimal system obtained in Section 3 to construct two-dimensional subalgebras of the nine-dimensional solvable Lie algebra. The detailed extension procedure has already been described in Section 2; therefore, we apply it here directly to the representatives obtained from the Maple computation.

Let $S_i = \langle X_i \rangle$ be a representative of a one-dimensional conjugacy class. For each such S_i , we compute its normalizer $N(S_i)$ in $\mathcal{L}$. If $N(S_i) = S_i$, then no two-dimensional extension arises from this representative. If $N(S_i)$ properly contains S_i , then the quotient algebra $N(S_i)/S_i$ is formed. The one-dimensional representatives of this quotient algebra determine the possible extension directions. Lifting these representatives back to $\mathcal{L}$ gives two-dimensional subalgebras of the form $\langle X_i, Y \rangle$. After the lifted subalgebras are obtained, conjugate cases are removed under the adjoint action of $\mathcal{L}$.

The following subsections list the results obtained by applying this procedure to the one-dimensional representatives of the nine-dimensional solvable Lie algebra. For each case, we give the normalizer, the quotient algebra, the one-dimensional representatives in the quotient, and the corresponding two-dimensional subalgebras.

4.1. Extension of $\langle e_9 \rangle$

For the one-dimensional subalgebra $\langle e_9 \rangle$, the normalizer is the following full Lie algebra:

$$N(\langle e_9 \rangle) = \langle e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9 \rangle.$$

Therefore,

$$N(\langle e_9 \rangle) / \langle e_9 \rangle = \langle \bar{e}_1, \bar{e}_2, \bar{e}_3, \bar{e}_4, \bar{e}_5, \bar{e}_6, \bar{e}_7, \bar{e}_8 \rangle.$$

The quotient algebra is determined by the following non-zero commutation relations:

$$\begin{aligned} [\bar{e}_1, \bar{e}_4] &= \bar{e}_2, & [\bar{e}_1, \bar{e}_5] &= \bar{e}_3, & [\bar{e}_1, \bar{e}_6] &= \bar{e}_1, & [\bar{e}_1, \bar{e}_8] &= \bar{e}_1, & [\bar{e}_2, \bar{e}_6] &= \bar{e}_2, \\ [\bar{e}_2, \bar{e}_7] &= -\bar{e}_3, & [\bar{e}_3, \bar{e}_6] &= \bar{e}_3, & [\bar{e}_3, \bar{e}_7] &= \bar{e}_2, & [\bar{e}_4, \bar{e}_7] &= -\bar{e}_5, & [\bar{e}_4, \bar{e}_8] &= -\bar{e}_4, \\ [\bar{e}_6, \bar{e}_7] &= \bar{e}_4, & [\bar{e}_5, \bar{e}_8] &= -\bar{e}_5. \end{aligned} \quad (4)$$

Applying the Maple code to this 8-dimensional solvable quotient algebra gives the following one-dimensional representatives:

$$\begin{aligned} &\langle \bar{e}_2 + \alpha_{11}\bar{e}_3 \rangle, \quad \langle \bar{e}_1 \rangle, \quad \langle \bar{e}_2 + \alpha_{12}\bar{e}_3 \pm \bar{e}_4 \pm \bar{e}_5 \rangle, \\ &\langle \bar{e}_1 \pm \bar{e}_4 \pm \bar{e}_5 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_8 \rangle, \quad \langle \bar{e}_6 + \alpha_{13}\bar{e}_7 + \alpha_{14}\bar{e}_8 \rangle, \\ &\langle \bar{e}_7 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_6 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_7 \pm \bar{e}_8 \rangle. \end{aligned} \quad (5)$$

Lifting these representatives to $\mathcal{L}$ gives the following two-dimensional subalgebras:

$$\begin{aligned} &\langle e_2 + \alpha_{11}e_3, e_9 \rangle, \quad \langle e_1, e_9 \rangle, \quad \langle e_2 + \alpha_{12}e_3 \pm e_4 \pm e_5, e_9 \rangle, \quad \langle e_1 \pm e_4 \pm e_5, e_9 \rangle, \\ &\langle e_1 \pm e_8, e_9 \rangle, \quad \langle e_6 + \alpha_{13}e_7 + \alpha_{14}e_8, e_9 \rangle, \quad \langle e_7, e_9 \rangle, \quad \langle e_1 \pm e_6, e_9 \rangle, \quad \langle e_1 \pm e_7 \pm e_8, e_9 \rangle. \end{aligned} \quad (6)$$

4.2. Extension of $\langle e_1 \rangle$

For the representative $\langle e_1 \rangle$, its normalizer is:

$$N(\langle e_1 \rangle) = \langle e_1, e_2, e_3, e_6, e_7, e_8, e_9 \rangle. \quad (7)$$

Thus,

$$N(\langle e_1 \rangle) / \langle e_1 \rangle = \langle \bar{e}_2, \bar{e}_3, \bar{e}_6, \bar{e}_7, \bar{e}_8, \bar{e}_9 \rangle. \quad (8)$$

The induced 6-dimensional solvable quotient algebra has the following non-zero commutators:

$$[\bar{e}_2, \bar{e}_6] = \bar{e}_2, \quad [\bar{e}_2, \bar{e}_7] = -\bar{e}_3, \quad [\bar{e}_3, \bar{e}_6] = \bar{e}_3, \quad [\bar{e}_3, \bar{e}_7] = \bar{e}_2. \quad (9)$$

The one-dimensional subalgebras of the quotient algebra given in Equation (9) are represented by:

$$\begin{aligned} &\langle \bar{e}_9 \rangle, \quad \langle \bar{e}_8 + \alpha_{21}\bar{e}_9 \rangle, \quad \langle \bar{e}_2 + \alpha_{22}\bar{e}_3 \rangle, \quad \langle \bar{e}_2 + \alpha_{23}\bar{e}_3 \pm \bar{e}_9 \rangle, \\ &\langle \bar{e}_2 + \alpha_{24}\bar{e}_3 \pm \bar{e}_8 \pm \bar{e}_9 \rangle, \quad \langle \bar{e}_6 + \alpha_{25}\bar{e}_7 + \alpha_{26}\bar{e}_8 + \alpha_{27}\bar{e}_9 \rangle, \\ &\langle \bar{e}_2 \pm \bar{e}_6 \pm \bar{e}_8 \pm \bar{e}_9 \rangle, \quad \langle \bar{e}_3 \pm \bar{e}_7 \pm \bar{e}_8 \pm \bar{e}_9 \rangle. \end{aligned} \quad (10)$$

Therefore, the corresponding two-dimensional subalgebras are:

$$\begin{aligned} &\langle e_9, e_1 \rangle, \quad \langle e_8 + \alpha_{21}e_9, e_1 \rangle, \quad \langle e_2 + \alpha_{22}e_3, e_1 \rangle, \quad \langle e_2 + \alpha_{23}e_3 \pm e_9, e_1 \rangle, \\ &\langle e_2 + \alpha_{24}e_3 \pm e_8 \pm e_9, e_1 \rangle, \quad \langle e_6 + \alpha_{25}e_7 + \alpha_{26}e_8 + \alpha_{27}e_9, e_1 \rangle, \\ &\langle e_2 \pm e_6 \pm e_8 \pm e_9, e_1 \rangle, \quad \langle e_3 \pm e_7 \pm e_8 \pm e_9, e_1 \rangle. \end{aligned} \quad (11)$$

4.3. Extension of $\langle e_2 + c_2 e_3 \rangle$

For the one-dimensional subalgebra $\langle e_2 + c_2 e_3 \rangle$, the normalizer is:

$$N(\langle e_2 + c_2 e_3 \rangle) = \langle e_1, e_2, e_3, e_4, e_5, e_6, e_8, e_9 \rangle. \quad (12)$$

The 7-dimensional solvable quotient algebra is:

$$N(\langle e_2 + c_2 e_3 \rangle) / \langle e_2 + c_2 e_3 \rangle = \langle \bar{e}_1, \bar{e}_3, \bar{e}_4, \bar{e}_5, \bar{e}_6, \bar{e}_8, \bar{e}_9 \rangle. \quad (13)$$

The non-zero commutators are:

$$\begin{aligned} [e_1, e_4] &= -c_2 e_3, & [e_1, e_5] &= e_3, & [e_1, e_6] &= e_1, & [e_1, e_8] &= e_1, \\ [e_3, e_6] &= e_3, & [e_4, e_8] &= -e_4, & [e_5, e_8] &= -e_5. \end{aligned} \quad (14)$$

The one-dimensional representatives in the quotient algebra are:

$$\begin{aligned} &\langle \bar{e}_3 \rangle, \quad \langle \bar{e}_9 \rangle, \quad \langle \bar{e}_5 \rangle, \quad \langle \bar{e}_3 \pm \bar{e}_9 \rangle, \quad \langle \bar{e}_4 + \alpha_3 \bar{e}_5 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_9 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_4 \pm \bar{e}_5 \rangle, \\ &\langle \bar{e}_1 \pm \bar{e}_5 \pm \bar{e}_9 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_4 \pm \bar{e}_5 \pm \bar{e}_9 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_5 \pm \bar{e}_8 \pm \bar{e}_9 \rangle, \quad \langle \bar{e}_6 + \alpha_3 \bar{e}_8 + \alpha_3 \bar{e}_9 \rangle, \\ &\langle \bar{e}_1 \pm \bar{e}_6 \pm \bar{e}_9 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_4 \pm \bar{e}_5 \pm \bar{e}_6 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_4 \pm \bar{e}_6 \pm \bar{e}_9 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_5 \pm \bar{e}_6 \pm \bar{e}_9 \rangle, \\ &\langle \bar{e}_1 \pm \bar{e}_4 \pm \bar{e}_5 \pm \bar{e}_6 \pm \bar{e}_9 \rangle. \end{aligned} \quad (15)$$

Hence, the two-dimensional subalgebras obtained by extending $\langle e_2 + c_2 e_3 \rangle$ are:

$$\begin{aligned} &\langle e_3, e_2 + c_2 e_3 \rangle, \quad \langle e_9, e_2 + c_2 e_3 \rangle, \quad \langle e_5, e_2 + c_2 e_3 \rangle, \quad \langle e_3 \pm e_9, e_2 + c_2 e_3 \rangle, \\ &\langle e_4 + \alpha_3 e_5, e_2 + c_2 e_3 \rangle, \quad \langle e_1 \pm e_9, e_2 + c_2 e_3 \rangle, \quad \langle e_1 \pm e_4 \pm e_5, e_2 + c_2 e_3 \rangle, \\ &\langle e_1 \pm e_5 \pm e_9, e_2 + c_2 e_3 \rangle, \quad \langle e_1 \pm e_4 \pm e_5 \pm e_9, e_2 + c_2 e_3 \rangle, \\ &\langle e_1 \pm e_5 \pm e_8 \pm e_9, e_2 + c_2 e_3 \rangle, \quad \langle e_6 + \alpha_3 e_8 + \alpha_3 e_9, e_2 + c_2 e_3 \rangle, \\ &\langle e_1 \pm e_6 \pm e_9, e_2 + c_2 e_3 \rangle, \quad \langle e_1 \pm e_4 \pm e_5 \pm e_6, e_2 + c_2 e_3 \rangle, \\ &\langle e_1 \pm e_4 \pm e_6 \pm e_9, e_2 + c_2 e_3 \rangle, \quad \langle e_1 \pm e_5 \pm e_6 \pm e_9, e_2 + c_2 e_3 \rangle, \\ &\langle e_1 \pm e_4 \pm e_5 \pm e_6 \pm e_9, e_2 + c_2 e_3 \rangle. \end{aligned} \quad (16)$$

4.4. Extension of $\langle e_2 + c_4 e_3 \pm e_9 \rangle$

For the one-dimensional subalgebra $\langle e_2 + c_4 e_3 \pm e_9 \rangle$, the normalizer and the following 6-dimensional solvable quotient algebra:

$$N(\langle e_2 + c_4 e_3 \pm e_9 \rangle) = \langle e_1, e_2, e_3, e_4, e_5, e_8, e_9 \rangle / \langle e_2 + c_4 e_3 \pm e_9 \rangle = \langle \bar{e}_1, \bar{e}_3, \bar{e}_4, \bar{e}_5, \bar{e}_8, \bar{e}_9 \rangle, \quad (17)$$

has the non-zero commutators:

$$[\bar{e}_1, \bar{e}_4] = -c_4 \bar{e}_3 - \bar{e}_9, \quad [\bar{e}_1, \bar{e}_5] = \bar{e}_3, \quad [\bar{e}_1, \bar{e}_8] = \bar{e}_1, \quad [\bar{e}_4, \bar{e}_8] = -\bar{e}_4, \quad [\bar{e}_5, \bar{e}_8] = -\bar{e}_5. \quad (18)$$

The one-dimensional representatives of 6-dimensional quotient algebra are:

$$\begin{aligned} &\langle \bar{e}_1 \rangle, \quad \langle \bar{e}_5 \rangle, \quad \langle \bar{e}_8 \rangle, \quad \langle \bar{e}_9 \rangle, \quad \langle \bar{e}_3 + \alpha_4 \bar{e}_9 \rangle, \quad \langle \bar{e}_5 \pm \bar{e}_9 \rangle, \\ &\langle \bar{e}_4 \pm \bar{e}_5 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_5 \rangle, \quad \langle \bar{e}_1 \pm \bar{e}_4 \pm \bar{e}_5 \rangle. \end{aligned} \quad (19)$$

Thus, the corresponding two-dimensional subalgebras obtained by extending $\langle e2 + c4_1e3 \pm e9 \rangle$ are:

$$\begin{aligned} &\langle e1, e2 + c4_1e3 \pm e9 \rangle, \quad \langle e5, e2 + c4_1e3 \pm e9 \rangle, \quad \langle e8, e2 + c4_1e3 \pm e9 \rangle, \\ &\langle e9, e2 + c4_1e3 \pm e9 \rangle, \quad \langle e3 + \alpha4_1e9, e2 + c4_1e3 \pm e9 \rangle, \\ &\langle e5 \pm e9, e2 + c4_1e3 \pm e9 \rangle, \quad \langle e4 \pm e5, e2 + c4_1e3 \pm e9 \rangle, \\ &\langle e1 \pm e5, e2 + c4_1e3 \pm e9 \rangle, \quad \langle e1 \pm e4 \pm e5, e2 + c4_1e3 \pm e9 \rangle. \end{aligned} \quad (20)$$

4.5. Extension of $\langle e4 + c9_3e5 \pm e9 \rangle$

For the one-dimensional subalgebra $\langle e4 + c9_3e5 \pm e9 \rangle$, the corresponding 5-dimensional solvable quotient algebra is:

$$N(\langle e4 + c9_3e5 \pm e9 \rangle) / \langle e4 + c9_3e5 \pm e9 \rangle = \langle \bar{e}2, \bar{e}3, \bar{e}5, \bar{e}6, \bar{e}9 \rangle, \quad (21)$$

and has the following non-zero commutators:

$$[\bar{e}2, \bar{e}6] = \bar{e}2, \quad [\bar{e}3, \bar{e}6] = \bar{e}3. \quad (22)$$

The representatives of one-dimensional subalgebras of 5-dimensional algebra (22) are:

$$\begin{aligned} &\langle \bar{e}3 \rangle, \quad \langle \bar{e}9 \rangle, \quad \langle \bar{e}2 + \alpha5_1\bar{e}3 \rangle, \quad \langle \bar{e}3 \pm \bar{e}9 \rangle, \quad \langle \bar{e}5 + \alpha5_2\bar{e}9 \rangle, \quad \langle \bar{e}2 + \alpha5_3\bar{e}3 \pm \bar{e}5 \rangle, \\ &\langle \bar{e}2 + \alpha5_4\bar{e}3 \pm \bar{e}9 \rangle, \quad \langle \bar{e}2 \pm \bar{e}5 \pm \bar{e}9 \rangle, \quad \langle \bar{e}3 \pm \bar{e}5 \pm \bar{e}9 \rangle, \quad \langle \bar{e}2 \pm \bar{e}6 \pm \bar{e}9 \rangle, \\ &\langle \bar{e}2 + \alpha5_5\bar{e}3 \pm \bar{e}5 \pm \bar{e}9 \rangle, \quad \langle \bar{e}5 + \alpha5_6\bar{e}6 + \alpha5_7\bar{e}9 \rangle. \end{aligned} \quad (23)$$

Hence, the two-dimensional subalgebras obtained by one-dimensional extension of $\langle e4 + c9_3e5 \pm e9 \rangle$ are:

$$\begin{aligned} &\langle e3, e4 + c9_3e5 \pm e9 \rangle, \quad \langle e9, e4 + c9_3e5 \pm e9 \rangle, \quad \langle e2 + \alpha5_1e3, e4 + c9_3e5 \pm e9 \rangle, \\ &\langle e3 \pm e9, e4 + c9_3e5 \pm e9 \rangle, \quad \langle e5 + \alpha5_2e9, e4 + c9_3e5 \pm e9 \rangle, \\ &\langle e2 + \alpha5_3e3 \pm e5, e4 + c9_3e5 \pm e9 \rangle, \quad \langle e2 + \alpha5_4e3 \pm e9, e4 + c9_3e5 \pm e9 \rangle, \\ &\langle e2 \pm e5 \pm e9, e4 + c9_3e5 \pm e9 \rangle, \quad \langle e3 \pm e5 \pm e9, e4 + c9_3e5 \pm e9 \rangle, \\ &\langle e2 \pm e6 \pm e9, e4 + c9_3e5 \pm e9 \rangle, \quad \langle e2 + \alpha5_5e3 \pm e5 \pm e9, e4 + c9_3e5 \pm e9 \rangle, \\ &\langle e5 + \alpha5_6e6 + \alpha5_7e9, e4 + c9_3e5 \pm e9 \rangle. \end{aligned} \quad (24)$$

4.6. Extension of $\langle e2 + c6_1e3 \pm e4 \pm e5 \rangle$

For the one-dimensional subalgebra $\langle e2 + c6_1e3 \pm e4 \pm e5 \rangle$, the following 5-dimensional solvable quotient algebra

$$N(\langle e2 + c6_1e3 \pm e4 \pm e5 \rangle) / \langle e2 + c6_1e3 \pm e4 \pm e5 \rangle = \langle \bar{e}3, \bar{e}4, \bar{e}5, -\bar{e}6 + \bar{e}8, \bar{e}9 \rangle, \quad (25)$$

has the following non-zero commutators:

$$[\bar{e}3, -\bar{e}6 + \bar{e}8] = -\bar{e}3, \quad [\bar{e}4, -\bar{e}6 + \bar{e}8] = -\bar{e}4, \quad [\bar{e}5, -\bar{e}6 + \bar{e}8] = -\bar{e}5. \quad (26)$$

Its one-dimensional subalgebras representatives are:

$$\begin{aligned} &\langle \bar{e}5 \rangle, \quad \langle \bar{e}9 \rangle, \quad \langle \bar{e}4 + \alpha6_1\bar{e}5 \rangle, \quad \langle \bar{e}5 \pm \bar{e}9 \rangle, \quad \langle \bar{e}3 + \alpha6_2\bar{e}4 + \alpha6_3\bar{e}5 \rangle, \\ &\langle \bar{e}4 + \alpha6_4\bar{e}5 \pm \bar{e}9 \rangle, \quad \langle \bar{e}3 + \alpha6_5\bar{e}4 + \alpha6_6\bar{e}5 \pm \bar{e}9 \rangle, \quad \langle -\bar{e}6 + \bar{e}8 + \alpha6_7\bar{e}9 \rangle. \end{aligned} \quad (27)$$

The two-dimensional subalgebras are:

$$\begin{aligned}
&\langle e5, e2 + c6_1 e3 \pm e4 \pm e5 \rangle, \quad \langle e9, e2 + c6_1 e3 \pm e4 \pm e5 \rangle, \\
&\langle e4 + \alpha 6_1 e5, e2 + c6_1 e3 \pm e4 \pm e5 \rangle, \quad \langle e5 \pm e9, e2 + c6_1 e3 \pm e4 \pm e5 \rangle, \\
&\langle e3 + \alpha 6_2 e4 + \alpha 6_3 e5, e2 + c6_1 e3 \pm e4 \pm e5 \rangle, \quad \langle e4 + \alpha 6_4 e5 \pm e9, e2 + c6_1 e3 \pm e4 \pm e5 \rangle, \\
&\langle e3 + \alpha 6_5 e4 + \alpha 6_6 e5 \pm e9, e2 + c6_1 e3 \pm e4 \pm e5 \rangle, \quad \langle -e6 + e8 + \alpha 6_7 e9, e2 + c6_1 e3 \pm e4 \pm e5 \rangle.
\end{aligned} \tag{28}$$

4.7. Extension of $\langle e1 + c5_3 e9 \rangle$

For the one-dimensional subalgebra $\langle e1 + c5_3 e9 \rangle$, the 5-dimensional solvable quotient algebra is:

$$N(\langle e1 + c5_3 e9 \rangle) / \langle e1 + c5_3 e9 \rangle = \langle \bar{e}2, \bar{e}3, \bar{e}7, -\bar{e}6 + \bar{e}8, \bar{e}9 \rangle, \tag{29}$$

and it has the following non-zero commutators:

$$[\bar{e}2, \bar{e}7] = -\bar{e}3, \quad [\bar{e}2, -\bar{e}6 + \bar{e}8] = -\bar{e}2, \quad [\bar{e}3, \bar{e}7] = \bar{e}2, \quad [\bar{e}3, -\bar{e}6 + \bar{e}8] = -\bar{e}3. \tag{30}$$

The one-dimensional subalgebras are:

$$\langle \bar{e}9 \rangle, \quad \langle \bar{e}2 + \alpha 7_1 \bar{e}3 \rangle, \quad \langle \bar{e}2 + \alpha 7_2 \bar{e}3 \pm \bar{e}9 \rangle, \quad \langle \bar{e}7 + \alpha 7_3 \bar{e}9 \rangle, \quad \langle -\bar{e}6 + \bar{e}8 + \alpha 7_4 \bar{e}9 \rangle. \tag{31}$$

Therefore, the two-dimensional subalgebras are:

$$\begin{aligned}
&\langle e9, e1 + c5_3 e9 \rangle, \quad \langle e2 + \alpha 7_1 e3, e1 + c5_3 e9 \rangle, \quad \langle e2 + \alpha 7_2 e3 \pm e9, e1 + c5_3 e9 \rangle, \\
&\langle e7 + \alpha 7_3 e9, e1 + c5_3 e9 \rangle, \quad \langle -e6 + e8 + \alpha 7_4 e9, e1 + c5_3 e9 \rangle.
\end{aligned} \tag{32}$$

4.8. Extension of $\langle e1 \pm e4 \pm e5 \rangle$

For the one-dimensional subalgebra $\langle e1 \pm e4 \pm e5 \rangle$, the following associated 4-dimensional solvable quotient algebra

$$\begin{aligned}
N(\langle e1 \pm e4 \pm e5 \rangle) / \langle e1 \pm e4 \pm e5 \rangle &= \langle e2, e3, -2e6 + e8, e9, e1 \pm e4 \pm e5 \rangle / \langle e1 \pm e4 \pm e5 \rangle \\
&= \langle \bar{e}2, \bar{e}3, -2\bar{e}6 + \bar{e}8, \bar{e}9 \rangle,
\end{aligned} \tag{33}$$

has the non-zero brackets are:

$$[\bar{e}2, -2\bar{e}6 + \bar{e}8] = -2\bar{e}2, \quad [\bar{e}3, -2\bar{e}6 + \bar{e}8] = -2\bar{e}3. \tag{34}$$

Its one-dimensional representatives are:

$$\langle \bar{e}3 \rangle, \quad \langle \bar{e}9 \rangle, \quad \langle \bar{e}2 + \alpha 8_1 \bar{e}3 \rangle, \quad \langle \bar{e}3 \pm \bar{e}9 \rangle, \quad \langle \bar{e}2 + \alpha 8_2 \bar{e}3 \pm \bar{e}9 \rangle, \quad \langle -2\bar{e}6 + \bar{e}8 + \alpha 8_3 \bar{e}9 \rangle. \tag{35}$$

Thus, the corresponding two-dimensional extensions of $\langle e1 \pm e4 \pm e5 \rangle$ are:

$$\begin{aligned}
&\langle e3, e1 \pm e4 \pm e5 \rangle, \quad \langle e9, e1 \pm e4 \pm e5 \rangle, \quad \langle e2 + \alpha 8_1 e3, e1 \pm e4 \pm e5 \rangle, \\
&\langle e3 \pm e9, e1 \pm e4 \pm e5 \rangle, \quad \langle e2 + \alpha 8_2 e3 \pm e9, e1 \pm e4 \pm e5 \rangle, \\
&\langle -2e6 + e8 + \alpha 8_3 e9, e1 \pm e4 \pm e5 \rangle.
\end{aligned} \tag{36}$$

4.9. Extension of $\langle e1 \pm e8 \pm e9 \rangle$

For the following one-dimensional subalgebra $\langle e1 \pm e8 \pm e9 \rangle$, the 5-dimensional solvable quotient algebra

$$N(\langle e1 \pm e8 \pm e9 \rangle) / \langle e1 \pm e8 \pm e9 \rangle = \langle \bar{e}2, \bar{e}3, \bar{e}1 + \bar{e}6, \bar{e}7, \bar{e}9 \rangle. \quad (37)$$

has the following non-zero commutators:

$$[\bar{e}2, \bar{e}1 + \bar{e}6] = \bar{e}2, \quad [\bar{e}2, \bar{e}7] = -\bar{e}3, \quad [\bar{e}3, \bar{e}1 + \bar{e}6] = \bar{e}3, \quad [\bar{e}3, \bar{e}7] = \bar{e}2. \quad (38)$$

Its one-dimensional representatives are:

$$\langle \bar{e}9 \rangle, \quad \langle \bar{e}2 \pm \bar{e}3 \rangle, \quad \langle \bar{e}2 + \alpha_9 \bar{e}3 \pm \bar{e}9 \rangle, \quad \langle \bar{e}3 \pm \bar{e}7 \pm \bar{e}9 \rangle, \quad \langle \bar{e}1 + \bar{e}6 + \alpha_9 \bar{e}7 + \alpha_9 \bar{e}9 \rangle. \quad (39)$$

Thus, the two-dimensional extensions are:

$$\begin{aligned} &\langle e9, e1 \pm e8 \pm e9 \rangle, \quad \langle e2 \pm e3, e1 \pm e8 \pm e9 \rangle, \quad \langle e2 + \alpha_9 e3 \pm e9, e1 \pm e8 \pm e9 \rangle, \\ &\langle e3 \pm e7 \pm e9, e1 \pm e8 \pm e9 \rangle, \quad \langle e1 + e6 + \alpha_9 e7 + \alpha_9 e9, e1 \pm e8 \pm e9 \rangle. \end{aligned} \quad (40)$$

4.10. Extension of $\langle e1 \pm e4 \pm e5 \pm e9 \rangle$

For the one-dimensional subalgebra $\langle e1 \pm e4 \pm e5 \pm e9 \rangle$, the quotient algebra is abelian, and its one-dimensional representatives are:

$$\langle \bar{e}9 \rangle, \quad \langle \bar{e}3 + \alpha_{10} \bar{e}9 \rangle, \quad \langle \bar{e}2 + \alpha_{10} \bar{e}3 + \alpha_{10} \bar{e}9 \rangle. \quad (41)$$

Hence, the corresponding two-dimensional extensions are:

$$\begin{aligned} &\langle e9, e1 \pm e4 \pm e5 \pm e9 \rangle, \quad \langle e3 + \alpha_{10} e9, e1 \pm e4 \pm e5 \pm e9 \rangle, \\ &\langle e2 + \alpha_{10} e3 + \alpha_{10} e9, e1 \pm e4 \pm e5 \pm e9 \rangle. \end{aligned} \quad (42)$$

4.11. Extension of $\langle e1 \pm e6 \pm e9 \rangle$

For the one-dimensional subalgebra $\langle e1 \pm e6 \pm e9 \rangle$, the following 5-dimensional solvable quotient algebra

$$N(\langle e1 \pm e6 \pm e9 \rangle) / \langle e1 \pm e6 \pm e9 \rangle = \langle \bar{e}1 + \bar{e}8, \bar{e}2 + \bar{e}4, \bar{e}3 + \bar{e}5, \bar{e}7, \bar{e}9 \rangle. \quad (43)$$

has the following two-dimensional extensions:

$$\begin{aligned} &\langle e9, e1 \pm e6 \pm e9 \rangle, \quad \langle e2 + e4 + \alpha_{11} (e3 + e5), e1 \pm e6 \pm e9 \rangle, \\ &\langle e2 + e4 + \alpha_{11} (e3 + e5) \pm e9, e1 \pm e6 \pm e9 \rangle, \\ &\langle e1 + e8 + \alpha_{11} e7 + \alpha_{11} e9, e1 \pm e6 \pm e9 \rangle. \end{aligned} \quad (44)$$

4.12. Extension of $\langle e1 \pm e6 \pm e8 \pm e9 \rangle$

For the one-dimensional subalgebra $\langle e1 \pm e6 \pm e8 \pm e9 \rangle$, the following quotient algebra

$$N(\langle e1 \pm e6 \pm e8 \pm e9 \rangle) / \langle e1 \pm e6 \pm e8 \pm e9 \rangle = \left\langle \frac{1}{2} \bar{e}1 + \bar{e}8, \bar{e}7, \bar{e}9 \right\rangle, \quad (45)$$

is abelian and hence, the corresponding two-dimensional extensions are:

$$\begin{aligned} &\langle e9, e1 \pm e6 \pm e8 \pm e9 \rangle, \quad \langle e7 + \alpha_{12_1} e9, e1 \pm e6 \pm e8 \pm e9 \rangle, \\ &\left\langle \frac{1}{2} e1 + e8 + \alpha_{12_2} e7 + \alpha_{12_3} e9, e1 \pm e6 \pm e8 \pm e9 \right\rangle. \end{aligned} \quad (46)$$

4.13. Extension of $\langle e1 \pm e7 \pm e8 \pm e9 \rangle$

For the following one-dimensional subalgebra $\langle e1 \pm e7 \pm e8 \pm e9 \rangle$, the 3-dimensional solvable quotient algebra

$$N(\langle e1 \pm e7 \pm e8 \pm e9 \rangle) / \langle e1 \pm e7 \pm e8 \pm e9 \rangle = \langle \bar{e}1 + \bar{e}6, \bar{e}1 + \bar{e}8, \bar{e}9 \rangle. \quad (47)$$

is abelian, and its two-dimensional extensions are:

$$\begin{aligned} &\langle e9, e1 \pm e7 \pm e8 \pm e9 \rangle, \quad \langle e1 + e8 + \alpha_{13_1} e9, e1 \pm e7 \pm e8 \pm e9 \rangle, \\ &\langle e1 + e6 + \alpha_{13_2} (e1 + e8) + \alpha_{13_1} e9, e1 \pm e7 \pm e8 \pm e9 \rangle. \end{aligned} \quad (48)$$

4.14. Extension of $\langle e6 + Ae1 + c_{15_6} e7 + c_{15_7} e8 + c_{15_8} e9 \rangle$

For the one-dimensional subalgebra $\langle e6 + Ae1 + c_{15_6} e7 + c_{15_7} e8 + c_{15_8} e9 \rangle$, assuming $c_{15_7} \neq -1$, the following 3-dimensional abelian quotient algebra

$$\begin{aligned} &N(\langle e6 + Ae1 + c_{15_6} e7 + c_{15_7} e8 + c_{15_8} e9 \rangle) / \langle e6 + Ae1 + c_{15_6} e7 + c_{15_7} e8 + c_{15_8} e9 \rangle \\ &= \left\langle \frac{A}{c_{15_7} + 1} \bar{e}1 + \bar{e}8, \bar{e}7, \bar{e}9 \right\rangle. \end{aligned} \quad (49)$$

has the following two-dimensional extensions:

$$\begin{aligned} &\langle e9, e6 + Ae1 + c_{15_6} e7 + c_{15_7} e8 + c_{15_8} e9 \rangle, \quad \langle e7 + \alpha_{14_1} e9, e6 + Ae1 + c_{15_6} e7 + c_{15_7} e8 + c_{15_8} e9 \rangle, \\ &\left\langle \frac{A}{c_{15_7} + 1} e1 + e8 + \alpha_{14_2} e7 + \alpha_{14_1} e9, e6 + Ae1 + c_{15_6} e7 + c_{15_7} e8 + c_{15_8} e9 \right\rangle. \end{aligned} \quad (50)$$

4.15. Extension of $\langle e7 + c_{11_5} e9 \rangle$

For the one-dimensional subalgebra $\langle e7 + c_{11_5} e9 \rangle$, the 4-dimensional solvable algebra is:

$$N(\langle e7 + c_{11_5} e9 \rangle) / \langle e7 + c_{11_5} e9 \rangle = \langle \bar{e}1, \bar{e}6, \bar{e}8, \bar{e}9 \rangle. \quad (51)$$

and it has the following non-zero commutators:

$$[\bar{e}1, \bar{e}6] = \bar{e}1, \quad [\bar{e}1, \bar{e}8] = \bar{e}1. \quad (52)$$

Thus, the corresponding two-dimensional extensions are:

$$\begin{aligned} &\langle e1, e7 + c_{11_5} e9 \rangle, \quad \langle e9, e7 + c_{11_5} e9 \rangle, \quad \langle e1 \pm e9, e7 + c_{11_5} e9 \rangle, \\ &\langle e1 \pm e8 \pm e9, e7 + c_{11_5} e9 \rangle, \quad \langle e6 + \alpha_{15_1} e8 + \alpha_{15_2} e9, e7 + c_{11_5} e9 \rangle. \end{aligned} \quad (53)$$

Therefore, the representatives of the conjugacy classes of two-dimensional subalgebras of 9- $\mathcal{D}$ Lie algebra (1) are:

$$\begin{aligned}
\mathcal{OS}_2(\mathcal{L}) = \{ & \langle e2 + \alpha_{11}e3, e9 \rangle, \langle e1, e9 \rangle, \langle e2 + \alpha_{12}e3 \pm e4 \pm e5, e9 \rangle, \langle e1 \pm e4 \pm e5, e9 \rangle, \langle e1 \pm e8, e9 \rangle, \\
& \langle e6 + \alpha_{13}e7 + \alpha_{14}e8, e9 \rangle, \langle e7, e9 \rangle, \langle e1 \pm e6, e9 \rangle, \langle e1 \pm e7 \pm e8, e9 \rangle, \langle e8 + \alpha_{21}e9, e1 \rangle, \langle e2 + \alpha_{22}e3, e1 \rangle, \\
& \langle e2 + \alpha_{23}e3 \pm e9, e1 \rangle, \langle e2 + \alpha_{24}e3 \pm e8 \pm e9, e1 \rangle, \langle e6 + \alpha_{25}e7 + \alpha_{26}e8 + \alpha_{27}e9, e1 \rangle, \langle e2 \pm e6 \pm e8 \pm e9, e1 \rangle, \\
& \langle e3 \pm e7 \pm e8 \pm e9, e1 \rangle, \langle e3, e2 + c_{21}e3 \rangle, \langle e5, e2 + c_{21}e3 \rangle, \langle e3 \pm e9, e2 + c_{21}e3 \rangle, \langle e4 + \alpha_{31}e5, e2 + c_{21}e3 \rangle, \\
& \langle e1 \pm e9, e2 + c_{21}e3 \rangle, \langle e1 \pm e4 \pm e5, e2 + c_{21}e3 \rangle, \langle e1 \pm e5 \pm e9, e2 + c_{21}e3 \rangle, \langle e1 \pm e4 \pm e5 \pm e9, e2 + c_{21}e3 \rangle, \\
& \langle e1 \pm e5 \pm e8 \pm e9, e2 + c_{21}e3 \rangle, \langle e6 + \alpha_{32}e8 + \alpha_{33}e9, e2 + c_{21}e3 \rangle, \langle e1 \pm e6 \pm e9, e2 + c_{21}e3 \rangle, \\
& \langle e1 \pm e4 \pm e5 \pm e6, e2 + c_{21}e3 \rangle, \langle e1 \pm e4 \pm e6 \pm e9, e2 + c_{21}e3 \rangle, \langle e1 \pm e5 \pm e6 \pm e9, e2 + c_{21}e3 \rangle, \\
& \langle e1 \pm e4 \pm e5 \pm e6 \pm e9, e2 + c_{21}e3 \rangle, \langle e1, e2 + c_{41}e3 \pm e9 \rangle, \langle e5, e2 + c_{41}e3 \pm e9 \rangle, \langle e8, e2 + c_{41}e3 \pm e9 \rangle, \\
& \langle e3 + \alpha_{41}e9, e2 + c_{41}e3 \pm e9 \rangle, \langle e5 \pm e9, e2 + c_{41}e3 \pm e9 \rangle, \langle e4 \pm e5, e2 + c_{41}e3 \pm e9 \rangle, \langle e1 \pm e5, e2 + c_{41}e3 \pm e9 \rangle, \\
& \langle e1 \pm e4 \pm e5, e2 + c_{41}e3 \pm e9 \rangle, \langle e3, e4 + c_{93}e5 \pm e9 \rangle, \langle e9, e4 + c_{93}e5 \pm e9 \rangle, \langle e2 + \alpha_{51}e3, e4 + c_{93}e5 \pm e9 \rangle, \\
& \langle e3 \pm e9, e4 + c_{93}e5 \pm e9 \rangle, \langle e5 + \alpha_{52}e9, e4 + c_{93}e5 \pm e9 \rangle, \langle e2 + \alpha_{53}e3 \pm e5, e4 + c_{93}e5 \pm e9 \rangle, \\
& \langle e2 + \alpha_{54}e3 \pm e9, e4 + c_{93}e5 \pm e9 \rangle, \langle e2 \pm e5 \pm e9, e4 + c_{93}e5 \pm e9 \rangle, \langle e3 \pm e5 \pm e9, e4 + c_{93}e5 \pm e9 \rangle, \\
& \langle e2 \pm e6 \pm e9, e4 + c_{93}e5 \pm e9 \rangle, \langle e2 + \alpha_{55}e3 \pm e5 \pm e9, e4 + c_{93}e5 \pm e9 \rangle, \langle e5 + \alpha_{56}e6 + \alpha_{57}e9, e4 + c_{93}e5 \pm e9 \rangle, \\
& \langle e5, e2 + c_{61}e3 \pm e4 \pm e5 \rangle, \langle e4 + \alpha_{61}e5, e2 + c_{61}e3 \pm e4 \pm e5 \rangle, \langle e5 \pm e9, e2 + c_{61}e3 \pm e4 \pm e5 \rangle, \\
& \langle e3 + \alpha_{62}e4 + \alpha_{63}e5, e2 + c_{61}e3 \pm e4 \pm e5 \rangle, \langle e4 + \alpha_{64}e5 \pm e9, e2 + c_{61}e3 \pm e4 \pm e5 \rangle, \\
& \langle e3 + \alpha_{65}e4 + \alpha_{66}e5 \pm e9, e2 + c_{61}e3 \pm e4 \pm e5 \rangle, \langle -e6 + e8 + \alpha_{67}e9, e2 + c_{61}e3 \pm e4 \pm e5 \rangle, \\
& \langle e2 + \alpha_{71}e3, e1 + c_{53}e9 \rangle, \langle e2 + \alpha_{72}e3 \pm e9, e1 + c_{53}e9 \rangle, \langle e7 + \alpha_{73}e9, e1 + c_{53}e9 \rangle, \langle -e6 + e8 + \alpha_{74}e9, e1 + c_{53}e9 \rangle, \\
& \langle e3, e1 \pm e4 \pm e5 \rangle, \langle e2 + \alpha_{81}e3, e1 \pm e4 \pm e5 \rangle, \langle e3 \pm e9, e1 \pm e4 \pm e5 \rangle, \langle e2 + \alpha_{82}e3 \pm e9, e1 \pm e4 \pm e5 \rangle, \\
& \langle -2e6 + e8 + \alpha_{83}e9, e1 \pm e4 \pm e5 \rangle, \langle e2 \pm e3, e1 \pm e8 \pm e9 \rangle, \langle e2 + \alpha_{91}e3 \pm e9, e1 \pm e8 \pm e9 \rangle, \\
& \langle e3 \pm e7 \pm e9, e1 \pm e8 \pm e9 \rangle, \langle e1 + e6 + \alpha_{92}e7 + \alpha_{93}e9, e1 \pm e8 \pm e9 \rangle, \langle e3 + \alpha_{101}e9, e1 \pm e4 \pm e5 \pm e9 \rangle, \\
& \langle e2 + \alpha_{102}e3 + \alpha_{101}e9, e1 \pm e4 \pm e5 \pm e9 \rangle, \langle e2 + e4 + \alpha_{111}(e3 + e5), e1 \pm e6 \pm e9 \rangle, \\
& \langle e2 + e4 + \alpha_{112}(e3 + e5) \pm e9, e1 \pm e6 \pm e9 \rangle, \langle e1 + e8 + \alpha_{113}e7 + \alpha_{114}e9, e1 \pm e6 \pm e9 \rangle, \\
& \langle e9, e1 \pm e6 \pm e8 \pm e9 \rangle, \langle e7 + \alpha_{121}e9, e1 \pm e6 \pm e8 \pm e9 \rangle, \langle (1/2)e1 + e8 + \alpha_{122}e7 + \alpha_{123}e9, e1 \pm e6 \pm e8 \pm e9 \rangle, \\
& \langle e9, e6 + Ae1 + c_{156}e7 + c_{157}e8 + c_{158}e9 \rangle, \langle e7 + \alpha_{141}e9, e6 + Ae1 + c_{156}e7 + c_{157}e8 + c_{158}e9 \rangle, \\
& \langle (A/c_{157} + 1)e1 + e8 + \alpha_{142}e7 + \alpha_{141}e9, e6 + Ae1 + c_{156}e7 + c_{157}e8 + c_{158}e9 \rangle, \langle e1, e7 + c_{115}e9 \rangle, \\
& \langle e1 \pm e9, e7 + c_{115}e9 \rangle, \langle e1 \pm e8 \pm e9, e7 + c_{115}e9 \rangle, \langle e6 + \alpha_{151}e8 + \alpha_{152}e9, e7 + c_{115}e9 \rangle \}.
\end{aligned} \tag{54}$$

Here, all parameters are real constants, with the restriction $c_{15,7} \neq -1$. The repeated representatives have been omitted, since they generate the same two-dimensional subspace.

5. Application

This section demonstrates the use of the Maple implementation in the symmetry analysis of the Lax equation. The example is relevant for two reasons. First, the Lax equation is a classical nonlinear dispersive shallow-water wave equation. Second, its Lie point symmetry algebra is finite-dimensional and solvable, so it falls directly within the scope of the proposed code. After the symmetry algebra is obtained, the optimal system provides a minimal set of inequivalent reductions and prevents repeated analysis of conjugate subalgebras.

In the study of shallow water waves, Peter D. Lax derived the nonlinear dispersive wave equation, as described in ref. [12]:

$$\Delta := v_t + 10vv_{xxx} + 20v_xv_{xx} + 30v^2v_x + v_{xxxxx} = 0, \tag{55}$$

which is known as the Lax equation. Here, $v(x, t)$ denotes the fluid velocity at time t in the x direction.

5.1. Lie Symmetries and Optimal System

A general point-symmetry generator for a $\mathcal{PDE}$ in the variables (x, t, v) is written as:

$$V = \xi_x(x, t, v)\partial_x + \xi_t(x, t, v)\partial_t + \eta_v(x, t, v)\partial_v, \quad (56)$$

where $\xi_x(x, t, v)$, $\xi_t(x, t, v)$, and $\eta_v(x, t, v)$ are the coefficient functions. The generator (56) is a symmetry of Equation (55) if it satisfies the following invariance criterion:

$$\mathcal{E}V(\Delta)|_{\Delta=0} = 0, \quad (57)$$

where

$$\mathcal{E}V = V + \eta_v^x \partial_{v_x} + \eta_v^t \partial_{v_t} + \eta_v^{xx} \partial_{v_{xx}} + \eta_v^{xxx} \partial_{v_{xxx}} + \eta_v^{xxxx} \partial_{v_{xxxx}} + \eta_v^{xxxxx} \partial_{v_{xxxxx}}, \quad (58)$$

denotes the 5th prolongation of $\mathcal{V}$ for the Lax Equation (55). The coefficient η_J is given as:

$$\eta_J = D_J(\eta_v - \xi_x v_x - \xi_t v_t) + \xi_x v_{Jx} + \xi_t v_{Jt}. \quad (59)$$

Here, J is a multi-index and D_J denotes the corresponding total derivative operator. For explanation, one of them can be indicated as D_x and is defined as:

$$D_x = \partial_x + v_x \partial_v + v_{xx} \partial_{v_x} + v_{tx} \partial_{v_t} + v_{xxx} \partial_{v_{xx}} + \dots, \quad (60)$$

where $v_{tx} = D_x(v_t)$. Using the Lie symmetry analysis method, the determining equations of (55) are given as:

$$\begin{aligned} (\xi_t)_v &= 0, & (\xi_t)_x &= 0, & (\xi_t)_{tt} &= 0, & (\xi_x)_t &= 0, \\ (\xi_x)_v &= 0, & (\xi_x)_x &= \frac{1}{5}(\xi_t)_t, & \eta_v &= -\frac{2}{5}(\xi_t)_t v. \end{aligned} \quad (61)$$

Solving the determining system (61) gives the following infinitesimals:

$$\xi_x(x, t, v) = \frac{1}{5}a_3x + a_2, \quad \xi_t(x, t, v) = a_3t + a_1, \quad \text{and} \quad \eta_v(x, t, v) = -\frac{2}{5}a_3v. \quad (62)$$

Thus, the Lie algebra of symmetries of the Lax equation (55) is spanned by the following three-dimensional geometric vector fields:

$$V1 = \frac{\partial}{\partial t}, \quad V2 = \frac{\partial}{\partial x}, \quad V3 = \frac{1}{5}x \frac{\partial}{\partial x} + t \frac{\partial}{\partial t} - \frac{2}{5}v \frac{\partial}{\partial v}. \quad (63)$$

Table 1 gives the commutator structure of the basis (63).

Table 1. Commutator table.

$[,]$	V1	V2	V3
V1	0	0	V1
V2	0	0	$\frac{1}{5}V2$
V3	$-V1$	$-\frac{1}{5}V2$	0

The only two non-zero commutators are:

$$[V1, V3] = V1, \quad [V2, V3] = \frac{1}{5} V2. \quad (64)$$

This algebra is a special case of the example discussed in Lemma 4.2 of ref. [11]. We now determine its one-dimensional $\mathcal{OS}$ and use the resulting representatives to reduce the Lax equation. The adjoint representation Table of Lie algebra (64) is given by Table 2. Executing the Maple code for the Lie algebra (64) gives the one-dimensional $\mathcal{OS}$, such as $V1, V2, V1 \pm V2$, and $V3$. Maple output is shown in Figure 7.

Table 2. Adjoint representation.

Ad	V1	V2	V3
V1	V1	V2	$V3 - \epsilon V1$
V2	V1	V2	$V3 - \frac{\epsilon}{5} V2$
V3	$e^\epsilon V1$	$e^{\frac{\epsilon}{5}} V2$	V3

#OutPut <code>Lie_Algebra := op(VectStrEq[1]);</code> $[V1, V3] = V1, [V2, V3] = \frac{1}{5} V2$ <code>Dimension_of_Lie_Algebra := print(nops(VectStrEq[2]), Dimensional Lie Algebra);</code> $3, \text{Dimensional Lie Algebra}$ <code>Lie_Algebra_is_Solvable := Query("Solvable");</code> $true$	
Ideals	<i>All_Ideals_of_lie_algebra</i> $[[V1], [V2], [V1, V2], [V3, V1, V2]]$
Shapes	<i>Form</i> $[[V1], [V2], [V1 + c_3 V2], [V3 + c_4 V1 + c_4 V2]]$
Adjoint Factorizations	<i>List_of_Adjoint_Factorization_with_forms</i> $\{[[], [V1]], [[], [V2]], [[V3], [V1 + c_3 V2]], [[V1, V2], [V3 + c_4 V1 + c_4 V2]]\}$
Conjugacy Classes	<i>One_Dimensional_Conjugacy_Classes where, epsilon = ± 1</i> $[V1, V2, V1 + \epsilon V2, V3] \text{ where, } \epsilon = \pm 1$

Figure 7. Optimal system for the Lie algebra of the Lax equation.

5.2. Symmetry Reductions and Exact Solutions

We now use the representatives of the one-dimensional $\mathcal{OS}$ s to obtain inequivalent invariant reductions of the Lax equation (55). For a symmetry generator of the form (56), the characteristic equation is:

$$\frac{dx}{\xi_x(x, t, v)} = \frac{dt}{\xi_t(x, t, v)} = \frac{dv}{\eta_v(x, t, v)}. \quad (65)$$

5.2.1. Subalgebra $V1 = \frac{\partial}{\partial t}$

For $V1$, the characteristic equation is:

$$\frac{dx}{0} = \frac{dt}{1} = \frac{dv}{0}, \quad (66)$$

which yields the following invariants, $X = x$ and $v = \phi(X)$. Using these similarity variables, we determine the following reduced ordinary differential equation: ($\mathcal{ODE}$)

$$30\phi^2\phi' + 20\phi'\phi'' + 10\phi\phi''' + \phi'''' = 0. \quad (67)$$

The Lie point symmetries of (67) are spanned by the following two-dimensional geometric vector fields:

$$\begin{aligned} U1 &= \frac{\partial}{\partial x}, \\ U2 &= x \frac{\partial}{\partial x} - 2\phi \frac{\partial}{\partial \phi}. \end{aligned} \quad (68)$$

The commutator Table of (68) is given by Table 3, from which we can see that there is only one non-zero commutator:

$$[U1, U2] = U1. \quad (69)$$

This 2-dimensional Lie algebra is also discussed in Lemma 4.1 of ref. [11]. The adjoint representation Table of Lie algebra (69) is given by Table 4.

Table 3. Commutator table.

$[,]$	$U1$	$U2$
$U1$	0	$U1$
$U2$	$-U1$	0

Table 4. Adjoint representation.

Ad	$U1$	$U2$
$U1$	$U1$	$U2 - \epsilon U1$
$U2$	$e^\epsilon U1$	$U2$

Executing the code for the Lie algebra of the reduced Equation (67) gives the one-dimensional $\mathcal{OS}$, such as $U1$ and $U2$. The Maple output is shown in Figure 8.

#OutPut $Lie_Algebra := op(VectStrEq[1]);$ $Dimension_of_Lie_Algebra := print(nops(VectStrEq[2]), Dimensional Lie Algebra);$ $Lie_Algebra_is_Solvable := Query("Solvable");$	
	$[U1, U2] = U1$ $2, Dimensional Lie Algebra$ $true$
Ideals	$All_Ideals_of_lie_algebra$ $[[U1], [U2, U1]]$
Shapes	$Form$ $[[U1], [U2 + c_1 U1]]$
Adjoint Factorizations	$List_of_Adjoint_Factorization_with_forms$ $\{[[], [U1]], [[U1], [U2 + c_1 U1]]\}$
Conjugacy Classes	$One_Dimensional_Conjugacy_Classes where, \epsilon = \pm 1$ $[U1, U2] where, \epsilon = \pm 1$

Figure 8. Optimal system for the Lie algebra of equation (67).

Symmetry Reductions by Subalgebras $U1$ and $U2$

The $\mathcal{OS}$ of the reduced Equation (67) gives two inequivalent reductions. The corresponding characteristic equation is:

$$\frac{dx}{\xi_x(x, \phi)} = \frac{d\phi}{\eta_\phi(x, \phi)}, \quad (70)$$

Case I: $U1 = \frac{\partial}{\partial x}$

For $U1$, the characteristic equation is:

$$\frac{dx}{1} = \frac{d\phi}{0}, \quad (71)$$

which yields the invariant $\phi(x) = c$ and therefore gives the invariant solution $v(x, t) = c$ of the Lax equation (55), where c is an arbitrary constant.

Case II: $U2 = x\frac{\partial}{\partial x} - 2\phi\frac{\partial}{\partial \phi}$

For $U2$, the characteristic equation is:

$$\frac{dx}{x} = \frac{d\phi}{-2\phi}. \quad (72)$$

From (72), we get $\phi(x) = \rho x^{-2}$. Hence, the invariant solution of the Lax equation (55) is:

$$v(x, t) = \rho x^{-2}, \quad \text{where } \rho = 0, -2, \text{ and } -6. \quad (73)$$

Figure 9 illustrates the qualitative structure of the inverse-square family $v = \rho x^{-2}$. The parameter ρ determines the amplitude and sign of the stationary profile. The value $\rho = 0$ gives the trivial member of the family, whereas $\rho = -2$ and $\rho = -6$ give non-zero singular profiles. Since the solution is independent of t , the three-dimensional surface remains unchanged along the time direction. The singularity at $x = 0$ and the decay as $|x|^{-2}$ away from the origin show the steep local spatial structure of this reduced shallow-water wave state.

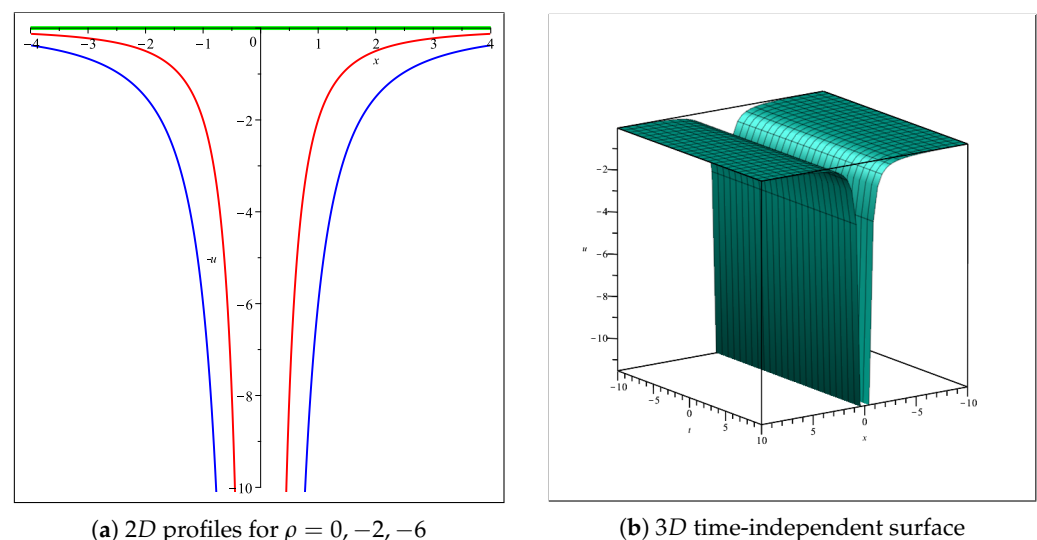


Figure 9. Graphical illustration of the stationary inverse-square solution $v = \rho x^{-2}$. Panel (a) shows the two-dimensional profiles for $\rho = \{0 \text{ (green)}, -2 \text{ (blue)}, -6 \text{ (red)}\}$, while panel (b) shows that the surface is independent of time.

5.2.2. Subalgebra $V2 = \frac{\partial}{\partial x}$

For $V2$, the Lax equation (55) is transformed into $f_T = 0$, with $T = t$ and $v = F(T)$. Therefore, the solution is given as $F(t) = c$. Hence, the invariant solution of the Lax equation (55) is $v(x, t) = c$, which corresponds to a spatially homogeneous equilibrium solution.

5.2.3. Subalgebra $V_3 = \frac{1}{5}x\frac{\partial}{\partial x} + t\frac{\partial}{\partial t} - \frac{2}{5}v\frac{\partial}{\partial v}$

The characteristic equation for the subalgebra V_3 :

$$\frac{dx}{x/5} = \frac{dt}{t} = \frac{dv}{-2v/5}, \quad (74)$$

yields the following invariants $p = xt^{-1/5}$ and $v = t^{-2/5}\psi(p)$. The Lax equation (55) is transformed into $\mathcal{ODE}$:

$$-\frac{1}{5}p\psi(p)' - \frac{2}{5}\psi(p) + 10\psi(p)\psi(p)''' + 20\psi(p)'\psi(p)'' + 30\psi(p)^2\psi(p)' + \psi(p)'''' = 0. \quad (75)$$

By using the power series solution, let

$$\psi(p) = \sum_{n=0}^{\infty} a_n p^n. \quad (76)$$

Substituting (76) in (75) and comparing coefficients, we obtain the following recursion formula:

$$\begin{aligned} a_{n+5} = & -\frac{1}{(n+1)(n+2)(n+3)(n+4)(n+5)} \left\{ -\frac{1}{5}na_n - \frac{2}{5}a_n + 10 \sum_{k=0}^n (n-k+3) \right. \\ & (n-k+2)(n-k+1)a_k a_{n-k+3} + 20 \sum_{k=0}^n (k+1)(n-k+2)(n-k+1)a_{k+1}a_{n-k+2} \\ & \left. + 30 \sum_{k=0}^n \sum_{l=0}^k (n-k+1)a_l a_{k-l}a_{n-k+1} \right\}, \quad \forall n = 0, 1, 2, \dots \end{aligned} \quad (77)$$

Thus, for arbitrarily chosen constants a_i ($i = 0, 1, \dots, 4$), we obtain:

$$a_5 = -\frac{1}{120} \left\{ -\frac{2}{5}a_0 + 60a_0a_3 + 40a_1a_2 + 30a_0^2a_1 \right\}. \quad (78)$$

Hence, the power series solution of (75) is:

$$\begin{aligned} \psi(p) = & a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5 + \sum_{n=1}^{\infty} a_{n+5}p^{n+5} \\ = & a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 - \frac{1}{120} \left\{ -\frac{2}{5}a_0 + 60a_0a_3 + 40a_1a_2 + 30a_0^2a_1 \right\} p^5 \\ & - \sum_{n=1}^{\infty} \left\{ \frac{1}{(n+1)(n+2)(n+3)(n+4)(n+5)} \left\{ -\frac{1}{5}na_n - \frac{2}{5}a_n + 10 \sum_{k=0}^n (n-k+3) \right. \right. \\ & (n-k+2)(n-k+1)a_k a_{n-k+3} + 20 \sum_{k=0}^n (k+1)(n-k+2)(n-k+1)a_{k+1}a_{n-k+2} \\ & \left. \left. + 30 \sum_{k=0}^n \sum_{l=0}^k (n-k+1)a_l a_{k-l}a_{n-k+1} \right\} \right\} p^{n+5}. \end{aligned} \quad (79)$$

Thus, the exact power series solution of the Lax equation (55) is:

$$\begin{aligned}
 v(x, t) = & a_0 t^{-\frac{2}{5}} + a_1 x t^{-\frac{3}{5}} + a_2 x^2 t^{-\frac{4}{5}} + a_3 x^3 t^{-1} + a_4 x^4 t^{-\frac{6}{5}} - \frac{1}{120} \left\{ -\frac{2}{5} a_0 + 60 a_0 a_3 \right. \\
 & + 40 a_1 a_2 + 30 a_0^2 a_1 \} x^5 t^{-\frac{7}{5}} - \sum_{n=1}^{\infty} \left\{ \frac{1}{(n+1)(n+2)(n+3)(n+4)(n+5)} \right. \\
 & \left\{ -\frac{1}{5} n a_n - \frac{2}{5} a_n + 10 \sum_{k=0}^n (n-k+3)(n-k+2)(n-k+1) a_k a_{n-k+3} \right. \\
 & + 20 \sum_{k=0}^n (k+1)(n-k+2)(n-k+1) a_{k+1} a_{n-k+2} + 30 \sum_{k=0}^n \sum_{l=0}^k (n-k \\
 & \left. \left. + 1) a_l a_{k-l} a_{n-k+1} \right\} \right\} x^{n+5} t^{-\frac{(n+7)}{5}}.
 \end{aligned} \quad (80)$$

Figure 10 illustrates the local profile of the similarity power-series solution. For the selected constants $a_i = 1, i = 0, 1, \dots, 4$, the solution evolves through the similarity variable $p = xt^{-1/5}$ and the amplitude factor $t^{-2/5}$. Therefore, the plotted profiles represent self-similar behaviour of the reduced Lax equation. As justified in the convergence argument below, the series is local near $p = 0$; consequently, the graphical representation should be interpreted as a local visualization of the similarity solution rather than as a global description for all x and t .

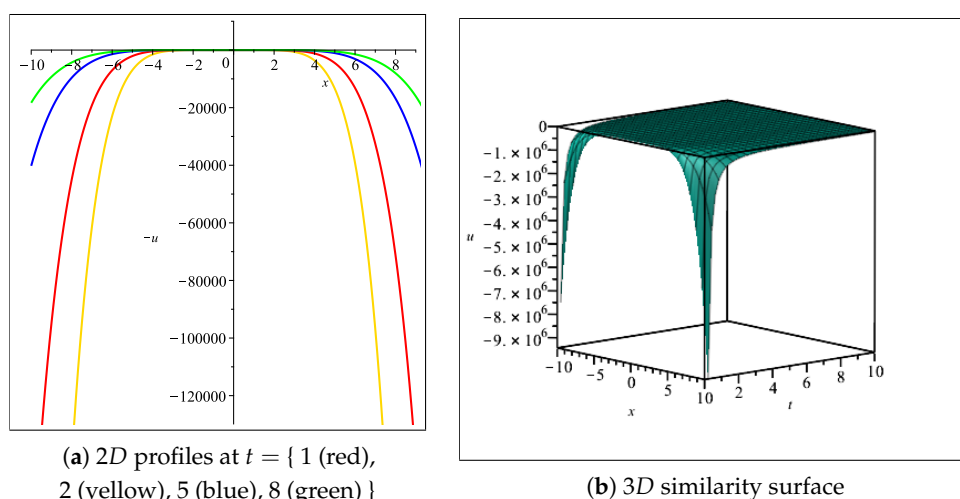


Figure 10. Graphical illustration of the similarity power-series solution (80) for $a_i = 1, i = 0, 1, \dots, 4$. Panel (a) shows two-dimensional sections at selected times, and panel (b) shows the corresponding three-dimensional surface.

Convergence of the power series. Equation (75) can be written in the normal analytic form:

$$\psi^{(5)} = \frac{1}{5} p \psi' + \frac{2}{5} \psi - 10 \psi \psi''' - 20 \psi' \psi'' - 30 \psi^2 \psi'.$$

The right-hand side is analytic in p, ψ, ψ', ψ'' and ψ''' . Hence, by the standard local existence theorem for analytic ordinary differential equations, for arbitrary initial values $a_0, a_1, \dots, a_4$ there exists a unique analytic solution in a neighbourhood of $p = 0$ [13]. The recursion Formula (77) gives exactly the Taylor coefficients of this local analytic solution. Therefore, the power series in (79) and the corresponding invariant solution (80) are locally convergent for sufficiently small $|p| = |xt^{-1/5}|$.

5.2.4. Subalgebra $V1 \pm V2 = \frac{\partial}{\partial t} + \frac{\partial}{\partial x}$

For $V1 \pm V2$, (55) is transformed into:

$$-\Phi' + 30\Phi^2\Phi' + 20\Phi'\Phi'' + 10\Phi\Phi''' + \Phi'''' = 0, \quad (81)$$

with $z = x \pm t$ and $v = \Phi(z)$. The Lie symmetry algebra of (81) is one-dimensional and is generated by $\mathcal{X}_1 = \frac{\partial}{\partial z}$. Applying the code to this algebra gives $\mathcal{OS} \mathcal{X}_1$ itself. The corresponding reduction gives $\Phi(z) = c$, and hence the invariant solution $v(x, t) = c$.

6. Conclusions

A Maple code for constructing optimal systems of solvable Lie algebras has been developed, based on the ideal-based framework of ref. [11]. The implementation computes ideals, possible representative forms, adjoint factorizations, and conjugacy classes in a single reproducible workflow, with checkable intermediate output at every stage. The method was validated on solvable Lie algebras up to nine dimensions, where the determination of all ideals took approximately five minutes, while the forms, adjoint factorizations, and conjugacy classes were each obtained in under a minute, so that the complete one-dimensional optimal system was found within a few minutes, a task that would take considerably longer if carried out manually. The underlying construction is not limited to this dimension, and the code can, in principle, be applied to solvable Lie algebras of higher orders as well. One dimensional representatives were further extended to two-dimensional subalgebras through normalizers and quotient algebras.

The application to the Lax equation shows how the computed optimal system supports symmetry reduction of a nonlinear dispersive shallow water model, yielding a constant solution, a nontrivial stationary inverse square solution, and a locally convergent similarity power series solution. These results connect the Lie algebraic classification obtained by the code with concrete invariant solutions of a nonlinear differential equation.

From a computational point of view, the main cost occurs in enumerating candidate forms from ideals and in repeated ideal/subalgebra and normalizer computations. Therefore, the runtime is expected to grow substantially for algebras with many one-dimensional ideals or many free structure parameters. The present implementation is restricted to solvable Lie algebras; extension to non-solvable algebras, for example through Levi decomposition into a semisimple part and a solvable radical, is a natural future direction.

Author Contributions: Conceptualization, H.A.; validation, N.S.A., H.A., and T.F.; Maple implementation, N.S.A.; writing—original draft preparation, N.S.A. writing—review and editing, N.S.A., H.A., and T.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Listing A1: Below are the functions used in the above-mentioned code:

```

> generateSubsets := proc(elements::list)
  local S, subset_list, sub_set;
  S := combinat:-subsets(elements);
  subset_list := [];
  while not S[finished] do
    sub_set := S[nextvalue]();
    subset_list := [op(subset_list), sub_set]
  end do;
  return subset_list
end proc;

> CreateBasisList := proc(inputList::list)
  local n, result;
  n := nops(inputList);
  result := [seq(inputList[i .. n], i = n .. 1, -1)];
  return result
end proc;

> CreateCeffList := proc(originalList)
  local resultList, i;
  resultList := [[]];
  for i to nops(originalList)-1 do
    resultList := [op(resultList), [1, op(originalList[-i ..
      -1])]];
  end do;
  return resultList;
end proc;

> DotProduct := proc(u::list, v::list)
  local i;
  return [add(u[i]*v[i], i = 1 .. min(nops(u), nops(v)))];
end proc;

> CalculateDotProducts:=proc(cefflist::(list(list)),basislist::
(list(list)))
  return zip(proc (a, b) options operator, arrow;
    DotProduct(a, b) end proc, cefflist, basislist);
end proc;

> MyFunction := proc(v_alg)
  local val_alg, vallg, element, v_all;
  val_alg := DGinfo(evalDG(v_alg), "ObjectGenerators");
  val_alg := [op(val_alg)];
  vallg := [];
  for element in val_alg do
    v_all := [_DG(["vector", L1, []], [[element], 1])];
    vallg := [op(vallg), op(v_all)];
  end do;
  return vallg;
end proc;

> sortList := proc(sublist)
  sort(sublist);
end proc;

> check_elements := proc(g_prime, elements_to_check)
  local i, element, results, result;
  results := [ ];
  for i to nops(elements_to_check) do

```

```

        element := elements_to_check[i];
        if member(element, g_prime) then
            results := [op(results), true]:
        else
            results := [op(results), false]:
        end if:
    end do:
    if has(results, false) then
        result := false:
    else
        result := true:
    end if:
    return result;
end proc:

> process_subsets := proc(subsets)
    local element, com, result, is_ideal, element_com, ele,
    new_subsets, Onedimensional_IDeals, basislist, cefflist;
    new_subsets := [];
    Onedimensional_IDeals := [];
    for element in subsets do
        com := LinearAlgebra:-DotProduct([seq(a[i], i = 1 ..
            nops(element))], [op(element)], conjugate = false);
        result := Query([com], "Subalgebra");
        if result then
            is_ideal := Query([com], "Ideal");
            if is_ideal then
                basislist := CreateBasisList(element);
                cefflist := CreateCeffList([seq(a[i], i = 1 ..
                    nops(element))]);
                element_com := CalculateDotProducts(cefflist,
                    basislist);
                for ele in element_com do
                    result := Query(ele, "Subalgebra");
                    if result then
                        is_ideal := Query(ele, "Ideal");
                        if is_ideal then
                            Onedimensional_IDeals := [op(Onedimensional_
                                IDeals), ele]:
                        end if:
                    end if:
                end do:
                new_subsets := generateSubsets(element);
                new_subsets := 'minus'({op(new_subsets)}, {[[]]});
                new_subsets := 'minus'({op(subsets)}, new_subsets);
                new_subsets := [op(new_subsets)];
                break
            end if
        end if
    end do;
    Onedimensional_IDeals := Onedimensional_IDeals;
    return [new_subsets, Onedimensional_IDeals];
end proc:

> FindONEDIMIdeals := proc()
    local centerr, c_alg, calg, d_al, dall, element, Z_g,
    d_algebra, g_prime, Onedimensional_IDeals_listt, result,
    is_ideal, Z_g_prime, subsets, i,
    gg_prime, resultt, iddeal;
    centerr := Center();
    centerr := sort(centerr);
    if centerr <> [ ] then
        c_alg := evalDG([LinearAlgebra:-DotProduct([seq(1, i=1..

```

```

      nops(centerr)), [op(centerr)], conjugate = false));
      calg := MyFunction(op(c_alg));
      if dall <> [ ] then
        centerr := calg;
      end if;
    end if;
    Z_g := centerr;
    d_algebra := DerivedAlgebra();
    d_algebra := sort(d_algebra);
    if d_algebra <> [ ] then
      d_al := evalDG([LinearAlgebra:-DotProduct([seq(1, i = 1..
        nops(d_algebra))
      ], [op(d_algebra)], conjugate = false)]);
      dall := MyFunction(op(d_al));
      if dall <> [ ] then
        d_algebra := dall;
      end if;
    end if;
    g_prime := d_algebra;
    gg_prime := d_algebra;
    Onedimensional_IDeals_listt := [ ];
    if Z_g <> [ ] then
      subsets := generateSubsets(Z_g);
      subsets := 'minus'({op(subsets)}, {[ ]});
      subsets := sort([op(subsets)], proc(a, b) options
        operator, arrow; nops(b) < nops(a) end proc);
      while subsets <> [ ] do
        resultt := process_subsets(subsets);
        subsets := resultt[1]; iddeal := resultt[2];
        Onedimensional_IDeals_listt := [op(Onedimensional_
          IDEals_listt), op(resultt[2])];
      end do;
    end if;
    Onedimensional_IDeals_listt:=Onedimensional_IDeals_listt;
    g_prime := 'minus'({op(g_prime)}, {op(Z_g)});
    g_prime := [op(g_prime)];
    if g_prime <> [ ] and g_prime <> Z_g then
      Z_g_prime := Centralizer(g_prime);
      if Z_g_prime <> [ ] then
        d_al := evalDG([LinearAlgebra:-DotProduct([seq(1,i=1..
          nops(Z_g_prime))
        ], [op(Z_g_prime)], conjugate = false)]);
        Z_g_prime := MyFunction(op(d_al));
      end if;
      Z_g_prime := sort(Z_g_prime);
      Z_g_prime:= 'minus'({op(Z_g_prime)}, {op(Z_g)});
      Z_g_prime := [op(Z_g_prime)];
      if Z_g_prime <> [ ] then
        subsets := generateSubsets(Z_g_prime);
        subsets := 'minus'({op(subsets)}, {[ ]});
        subsets := sort([op(subsets)], proc(a, b) options
          operator, arrow; nops(b) < nops(a) end proc);
        while subsets <> [ ] do
          resultt := process_subsets(subsets);
          subsets := resultt[1]; iddeal := resultt[2];
          Onedimensional_IDeals_listt:= [op(Onedimensional_
            IDEals_listt), op(resultt[2])];
        end do;
      end if;
    end if;
    Onedimensional_IDeals_listt:=Onedimensional_IDeals_listt;
    if Onedimensional_IDeals_listt = [ ] then
      if gg_prime <> [ ] then

```

```

subsets := generateSubsets(gg_prime);
subsets := 'minus'({op(subsets)}, {[ ]});
subsets := sort([op(subsets)], proc (a, b) options
operator, arrow; nops(b) < nops(a) end proc);
while subsets <> [ ] do
    resultt := process_subsets(subsets);
    subsets := resultt[1]; iddeal := resultt[2];
    Onedimensional_IDeals_listt := [op(Onedimensional_
    IDeals_listt), op(resultt[2])];
end do;
end if;
end if;
return Onedimensional_IDeals_listt;
end proc;

> processnext_subsets := proc(elements, subsets)
local element, com, result, is_ideal, element_com, ele,
new_subsets, dimensional_IDeals, next_ideal, basislist,
cefflist;
new_subsets := [];
next_ideal := [];
for element in subsets do
    com := LinearAlgebra:-DotProduct([seq(a[i] , i = 1 ..
nops(element))], [op(element)], conjugate = false);
    com := [op(elements), com];
    result := Query(com, "Subalgebra");
    if result then is_ideal := Query(com, "Ideal");
    if is_ideal then
        basislist := CreateBasisList(element);
        cefflist := CreateCeffList([seq(a[i], i = 1 .. nops
(element))]);
        element_com := CalculateDotProducts(cefflist,
basislist);
        for ele in element_com do
            result := Query([op(elements), op(ele)],
"Subalgebra");
            if result then
                is_ideal := Query([op(elements), op(ele)],
"Ideal");
                if is_ideal then
                    next_ideal := [op(next_ideal), [op(elements),
op(ele)]];
                end if;
            end if;
        end do;
        new_subsets := generateSubsets(element);
        new_subsets := 'minus'({op(new_subsets)}, {[ ]});
        new_subsets := 'minus'({op(subsets)}, new_subsets);
        new_subsets := [op(new_subsets)];
        break
    end if;
end if;
end do;
next_ideal := next_ideal;
return [new_subsets, next_ideal];
end proc;

> find_other_ideal := proc()
local IDeals_listt, elements, basiss, subsets, Basis,
resultt, iddeal;
IDeals_listt := [];
for elements in current_ideal_list do
    basiss := 'minus'({op(VectStrEq[2])}, {op(elements)});

```

```

Basis := [op(basiss)];
if Basis <> [] then
  subsets := generateSubsets(Basis);
  subsets := 'minus'({op(subsets)}, {[[]]});
  subsets := sort([op(subsets)], proc (a, b) options
operator, arrow; nops(b) < nops(a) end proc);
while subsets <> [] do
  resultt := processnext_subsets(elements, subsets);
  subsets := resultt[1];
  iddeal := resultt[2];
  IDEals_listt := [op(IDEals_listt), op(resultt[2])];
  IDEals_listt := sort(IDEals_listt);
end do;
end if;
end do;
return IDEals_listt;
end proc;

```

References

1. Ovsiannikov, L.V. *Group Properties of Differential Equations*; Siberian Section of the Academy of Science of USSR: Siberia, Russia, 1962.
2. Ovsiannikov, L.V. *Group Analysis of Differential Equations*; Academic Press: New York, NY, USA, 1982.
3. Patera, J.; Sharp, R.T.; Winternitz, P.; Zassenhaus, H. Invariants of real low dimension Lie algebras. *J. Math. Phys.* **1976**, *17*, 986–994.
4. Galas, F.; Richter, E.W. Exact similarity solutions of ideal MHD equations for plane motions. *Phys. D Nonlinear Phenom.* **1991**, *50*, 297–307.
5. Patera, J.; Winternitz, P. Subalgebras of real three-and four-dimensional Lie algebras. *J. Math. Phys.* **1997**, *18*, 1449–1455.
6. Ibragimov, N.H. *Selected Works. Vol. I*; ALGA publications: Canberra Australia, 2006.
7. Ibragimov, N.H. *Selected Works. Vol. II*; ALGA publications: Canberra Australia, 2006.
8. Ibragimov, N.H. *Selected Works. Vol. III*; ALGA publications: Canberra Australia, 2006.
9. Amata, L.; Oliveri, F. Automatic Determination of Optimal Systems of Lie Subalgebras: the Package SymboLie. *Commun. Contemp. Math.* **2023**, *789*, 1–16.
10. Amata, L.; Oliveri, F.; Sgroi, E. Optimal Systems of Lie Subalgebras: A Computational Approach. *J. Geom. Phys.* **2024**, *204*, 105290.
11. Ali, S.; Azad, H.; Biswas, I.; Mahomed, F.M. Ideals and conjugacy classes in solvable Lie algebras. *Int. J. Geom. Methods Mod. Phys.* **2023**, *20*, 2350202.
12. Lax, P.D. Integrals of nonlinear equations of evolution and solitary waves. *Commun. Pure Appl. Math.* **1968**, *21*, 467–490.
13. Coddington, E.A.; Levinson, N. *Theory of Ordinary Differential Equations*; McGraw–Hill: New York, NY, USA, 1955.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.