

On the algorithm to find S-related Lie algebras

Carlos Inostroza¹, Igor Kondrashuk², Nelson Merino³
and Felip Nadal⁴

¹ Departamento de Física, Universidad de Concepción, Casilla 160-C, Concepción, Chile

² Grupo de Matemática Aplicada, Departamento de Ciencias Básicas, Universidad del Bío-Bío, Campus Fernando May, Casilla 447, Chillán, Chile

³ APC, Université Paris Diderot, 10, rue Alice Domon et Leonie Duquet, 75205 Paris Cedex 13, France

⁴ Instituto de Física Corpuscular (IFIC), Edificio Institutos de Investigación, c/ Catedrático José Beltrán, 2, E-46980 Paterna, España

E-mail: nemerino@gmail.com

Abstract. In this article we describe the Java library that we have recently constructed to automatize the S-expansion method, a powerful mathematical technique allowing to relate different Lie algebras. An important input in this procedure is the use of abelian semigroups and thus, we start with a brief review about the classification of non-isomorphic semigroups made in the literature during the last decades, and explain how the lists of non-isomorphic semigroups up to order 6 can be used as inputs in many of the methods of our library. After describing the main features of the classes that compose our library we present a new method called *fillTemplate* which turns out to be very useful to answer whether two given algebras can be S-related.

1. Introduction

The S-expansion [1] is a powerful mathematical technique allowing to relate different Lie algebras, which contains as a particular case the Inönü-Wigner contraction [2, 3] and its generalizations [4, 5, 6]. This method also turns out to be a valuable tool to study the relations between different physical theories. Since we already gave a technical description of the S-expansion in the related proceeding contribution [7], as well as a review about its different physical applications, here we begin by summarizing the main ingredients needed to understand how the Java library [8] works. We have constructed this library to automatize the S-expansion procedure.

Basically, the S-expansion method combines the structure constants of a Lie algebra with the inner multiplication law of an abelian semigroup in such a way that leads to the definition of other Lie algebras, called *S-expanded* algebras. The inputs are:

- a Lie algebra \mathcal{G} with generators $\{X_i\}$ and Lie product $[X_i, X_j] = C_{ij}^k X_k$ where C_{ij}^k are the structure constants (convention sum on repeated indices is assumed).
- an abelian finite semigroup $S = \{\lambda_\alpha, \alpha = 1, \dots, n\}$, whose inner operation can be represented either by its multiplication table $A = (a_{\alpha\beta}) \equiv (\lambda_\alpha \cdot \lambda_\beta)$, with entries in λ_α , or by the *selectors* $K_{\alpha\beta}^\rho$ which are defined through the relation $\lambda_\alpha \cdot \lambda_\beta = \lambda_{\gamma(\alpha,\beta)} = K_{\alpha\beta}^\rho \lambda_\rho$, where $K_{\alpha\beta}^\rho = 1$ if $\rho = \gamma(\alpha, \beta)$ and $K_{\alpha\beta}^\rho = 0$ if $\rho \neq \gamma(\alpha, \beta)$. Thus, the components of a selector can only take the values 0 or 1.

As is well-known, the structure constants and the selectors provide respectively a matrix representation for \mathcal{G} and S . This allows to show that the Kronecker product of those representations $\mathcal{G}_S = S \otimes \mathcal{G}$ is also a Lie algebra with generators $X_{(i,\alpha)} \equiv \lambda_\alpha \otimes X_i$, Lie product defined by $[X_{(i,\alpha)}, X_{(j,\beta)}] \equiv \lambda_\alpha \cdot \lambda_\beta \otimes [X_i, X_j] = C_{(i,\alpha)(j,\beta)}^{(k,\gamma)} X_{(k,\gamma)}$ and structure constants given by $C_{(i,\alpha)(j,\beta)}^{(k,\gamma)} = K_{\alpha\beta}^\gamma C_{ij}^k$. The explicit proof can be found in Ref. [1].

There are two cases where one can extract from \mathcal{G}_S smaller algebras, with interesting properties. The first one happens when the original algebra has a subspace decomposition, e.g.¹, of the type $\mathcal{G} = V_0 \oplus V_1$ satisfying $[V_0, V_0] \subset V_0$, $[V_0, V_1] \subset V_1$, $[V_1, V_1] \subset V_0$. If the semigroup has a decomposition in subsets $S = S_0 \cup S_1$ satisfying the so called *resonant decomposition*:

$$S_0 \cdot S_0 \in S_0, \quad S_0 \cdot S_1 \in S_1, \quad S_1 \cdot S_1 \in S_0 \quad (1)$$

then, one can show that $\mathcal{G}_{S,R} \equiv (S_0 \otimes V_0) \oplus (S_1 \otimes V_1)$ is a subalgebra of \mathcal{G}_S , which is called the *resonant subalgebra*. The other case occurs when the semigroup have a zero element 0_S satisfying $\lambda_\alpha \cdot 0_S = 0_S$ for any element $\lambda_\alpha \in S$. In that case the whole sector $0_S \otimes \mathcal{G}$ can be erased from \mathcal{G}_S in such a way that what remains is still a Lie algebra, which is called the *0_S -reduced algebra* $\mathcal{G}_{S,\text{red}}$ (which is not necessarily a subalgebra of \mathcal{G}_S).

In summary, depending on the features of the abelian semigroup S that one uses in the S-expansion, the following types of algebra can be generated

- (a) the expanded algebra $\mathcal{G}_S = S \otimes \mathcal{G}$,
- (b) the resonant subalgebra $\mathcal{G}_{S,R}$, if S has at least one resonant decomposition of the type (1),
- (c) the 0_S -reduced algebra $\mathcal{G}_{S,\text{red}}$, if S has a zero element 0_S ,
- (d) the 0_S -reduction of the resonant subalgebra $\mathcal{G}_{S,R,\text{red}}$, if S has both a zero element and a resonant decomposition.

Having introduced the main ingredients used in the S-expansion method, in the next sections we give a description of the Java library [8] that automatizes this procedure. In Section 2 we review some aspects about the classification of non-isomorphic semigroups, while in Section 3 we briefly describe the library. Then, in Section 4 we explain how the lists of all non-isomorphic abelian semigroups can be used to classify all possible S-expanded algebras (a-d). Finally, in Section 5, we will briefly describe a new method that extends our library and that is useful to answer if two given Lie algebras can be S-related.

2. Some aspects about the semigroup classification

As shown in the following table,

order	$Q = \#$ semigroups	
2	4	
3	18	
4	126	[Forsythe '54]
5	1,160	[Motzkin, Selfridge '55]
6	15,973	[Plemmons '66]
7	836,021	[Jurgensen, Wick '76]
8	1,843,120,128	[Satoh, Yama, Tokizawa '94]
9	52,989,400,714,478	[Distler, Kelsey, Mitchell '09]
10	12,418,001,077,381,302,684	[Distler, Jefferson, Kelsey, Kotthoff '16]

the problem of enumerating the all non-isomorphic finite semigroups of a certain order is a non-trivial problem, as the number Q of semigroups increases very quickly with the order

¹ As formulated in the original work [1], one can also deal with more general decompositions. However, on the first version of our library [8] we only focus in this case $\mathcal{G} = V_0 \oplus V_1$.

of the semigroup. This classification has been made by many different authors (see e.g. [9, 10, 11, 12, 13, 14, 15, 16, 17] and references therein). In lower orders, a list of the multiplication tables representing those semigroups can be explicitly constructed. For example, in Ref. [18] it has been claimed that its program *gen.f* allows to generate, in lexicographical ordering, the lists *sem.n* of these tables for the orders $n = 2, \dots, 7$. After running that program, we got them only up to order 6 (for some reason the program stops when reaches the 835,927th of the 836,021 semigroups of order 7). In any case, we remark that similar lists are not explicitly known for orders higher than 8 and that the number of non-isomorphic semigroups in those orders has been reached only by using indirect techniques (see e.g., [19, 20]).

As we will see, the lists *sem.n* up to order $n = 6$ can be used as inputs in our library (although its methods are not restricted to that order²). In each list, a semigroup $S_{(n)}^a$ of order n is uniquely identified by the number $a = 1, \dots, Q$ and the semigroup elements λ_α are represented by α with $\alpha = 1, \dots, n$. The program *com.f* of [18] selects from those lists only the abelian semigroups (which are the ones of interest in the context of the S-expansion). For example, for $n = 2$ the semigroup elements are $\{1, 2\}$ and the program *com.f* gives:

$$\begin{array}{c|cc} S_{(2)}^1 & 1 & 2 \\ \hline 1 & 1 & 1 \\ 2 & 1 & 1 \end{array}, \begin{array}{c|cc} S_{(2)}^2 & 1 & 2 \\ \hline 1 & 1 & 1 \\ 2 & 1 & 2 \end{array}, \begin{array}{c|cc} S_{(2)}^4 & 1 & 2 \\ \hline 1 & 1 & 2 \\ 2 & 2 & 1 \end{array}. \quad (3)$$

However, to avoid renaming the abelian semigroups in a new list, in our library we use directly the lists *sem.n* and select the abelian ones with a simple method when is needed³.

3. Description of the Java library to perform S-expansions

To use the Java library it is necessary to download the linear algebra package *jama.jar*⁴ from [21] and the following files from [8]:

File	Brief description
<i>data.zip</i>	Contains the the files <i>sem.n</i>
<i>sexpansion.jar</i>	Is the library itself
<i>ReadMe.pdf</i>	File with detailed installation instructions
<i>examples.zip</i>	Example programs explained in Section 5 of the Ref. [22]
<i>Output_examples.zip</i>	Output samples of the example programs

(4)

Our library is composed of 11 classes⁵, each of them containing different methods allowing us to automatize the S-expansion procedure. As the full documentation of the library is available in [23], here we only describe the main features of these classes.

First, the class *SetS* allows to represent a set of integers, where none of them is repeated. It contains methods allowing to represent permutations, which are useful to determine isomorphisms between different semigroup tables. It also contains methods allowing to generate all possible subsets S_0 and S_1 of a semigroup S , with respectively $n_0 \leq n$ and $n_1 \leq n$ elements, such that $S = S_0 \cup S_1$ (notice that some elements can be simultaneously in S_0 and S_1). Thus, the methods of this class also provide the basics elements to the study of resonances.

² The methods of our library also allow us to perform semigroup calculations (like checking associativity, commutativity, finding zero element, isomorphisms and resonances) when the order n is higher than 6. The only issue is that we do not have the full list of non-isomorphic tables for those higher orders.

³ It is also worth to mention that the lists *sem.n* are exhaustive, i.e., if one finds a semigroup S of order $n \leq 6$ whose multiplication table is not contained in the list *sem.n*, then S must be isomorphic to one and only one semigroup in that list. This can be easily checked with the methods implemented in our library and thus, we are able to work with arbitrary semigroups, i.e., not necessarily having a lexicographical ordering.

⁴ In particular, we use the methods belonging to the class *Matrix* of the library *jama.jar*.

⁵ To see the source code one only have to unzip the file *sexpansion.jar*.

The class *Semigroup* is used to represent discrete semigroups by means of the multiplication table $A = (a_{\alpha\beta}) \equiv (\alpha \cdot \beta)$ and contains methods allowing us to perform basic semigroup operations like checking associativity, commutativity, finding the zero element and loading the lists *sem.n* when is needed. It also contains methods that, when combined with those of the class *SetS*, allows to calculate isomorphisms and study all possible resonant decompositions of any given semigroup.

The class *Selector* allows to represent semigroups by means of their selectors of $K_{\alpha\beta}^\rho$ which, as explained in the introduction, are very convenient to describe S-expansions. It also contains methods allowing to determine if the semisimplicity is preserved under the S-expansion. The classes *SelectorReduced*, *SelectorResonant*, *SelectorResonantReduced* are constructed as a child of the class *Selector*, i.e., they extend that class for the cases where the semigroup have a zero element, a resonance and both simultaneously.

The class *StructureConstantSet* allows to represent the original algebra \mathcal{G} by means of its structure constants C_{ij}^k . It also contains methods to perform basics calculations, like to determine the Killing-Cartan metric of \mathcal{G} .

The class *StructureConstantSetExpanded* allows to represent the Lie algebra resulting of performing the S-expansion of \mathcal{G} with S . Finally, the child classes *StructureConstantSetExpandedReduced*, *StructureConstantSetExpandedResonant* and *StructureConstantSetExpandedResonantReduced* allow to represent the expanded algebras $\mathcal{G}_{S,\text{red}}$, $\mathcal{G}_{S,R}$ and $\mathcal{G}_{S,R,\text{red}}$ when it corresponds.

Further details can be found in Ref. [22], which has been written as a handbook to use the library. Indeed, it explains in detail how the methods of our library has been constructed, how do they work and also explains the example programs that allow to check the results obtained in Refs. [24] and [25].

4. Classification of all possible S-expansions

The first motivation that lead us to construct our library was to provide the computational tool to study all possible S-expansions of the type (a-d) that one can perform with all the abelian non-isomorphic semigroups provided by the lists *sem.n*. On one side, this implies to identify all the abelian semigroups having a zero element, something that can be easily done not only by our library but also by the programs given in [18]. On the other side, a study of all possible resonant decompositions of these semigroups is also needed. To our knowledge, this was not done in the literature before our works [24] and [25] where we elaborated the first basic methods to study resonances, which we have improved recently and presented in the form of a Java library [8] in Ref. [22]. Thus, this is the first computational tool allowing to perform a full study of those resonances, as well as to classify and represent all possible S-expansions of the type (a-d) of a given Lie algebra. This result can be summarized in the following table,

	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$
# \mathcal{G}_S	3	12	58	325	2,143
#pss	2	5	16	51	201
# $\mathcal{G}_{S,\text{red}}$	2	8	39	226	1,538
#pss	1	3	9	34	135
# $\mathcal{G}_{S,R}$	1	8	48	299	2,059
#r	1	9	124	1,653	25,512
#pss	1	1	4	7	23
# $\mathcal{G}_{S,R,\text{red}}$	0	5	32	204	1,465
#r	0	6	92	1,295	20,680
#pss	0	1	1	6	12

(5)

It gives for each order the number of S-expansions of the type (a-d) that can be performed, as well as the number #pss of S-expansions preserving semisimplicity. Notice, a given semigroup

can have more than one resonance, and with them the same semigroup may lead to different resonant subalgebras. Thus, in the table we count not only the number of abelian semigroups having at least one resonance, but also the total number of different resonances $\#r$.

5. A method to find S-related algebras

Imagine that we have to find a semigroup with some multiplications fixed, for example

$$\begin{array}{c|cccc}
 & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \\
 \hline
 \lambda_1 & & \lambda_3 & & \lambda_4 \\
 \lambda_2 & \lambda_3 & & \lambda_4 & \lambda_4 \\
 \lambda_3 & & \lambda_4 & & \lambda_4 \\
 \lambda_4 & \lambda_4 & \lambda_4 & \lambda_4 & \lambda_4
 \end{array} . \tag{6}$$

As in Java an empty space cannot be left in an array, we choose to represent them by -1 in its place. For example, the template above will be represented by

$$\begin{pmatrix} -1 & 3 & -1 & 4 \\ 3 & -1 & 4 & 4 \\ -1 & 4 & -1 & 4 \\ 4 & 4 & 4 & 4 \end{pmatrix} . \tag{7}$$

Clearly, there are $4^4 = 256$ different ways to fill this template in a commutative way, where only some of them are associative so they are really semigroups. In general, for a template of order n where x independent component has been fixed, there are $n^{n(n+1)/2-x}$ different forms to fill it in a commutative way. This task is performed by the method *fillTemplate* of the class *Semigroup*, which reads the *template* and returns a *list* with all the tables that fill it. It is used as follow.

```
list = Semigroup.fillTemplate(template);
```

From all those tables one should select those which are associative and satisfy, if apply, a certain resonance. Applying the isomorphisms methods, we can finally identify the set of non-isomorphic semigroups that are solutions for the given template. As it will be explained in [26], this method turns out very useful to elaborate a general algorithm to answer if two given Lie algebras can be S-related.

6. Final Remarks

As our library has an open licence GNU, we expect that it can be improved and extended, for example to study other kind of decompositions like, e.g., $S = S_0 \cup S_1 \cup S_2$ where

$$\begin{aligned}
 S_0 \times S_0 &\subset S_0 , & S_0 \times S_1 &\subset S_1 , & S_0 \times S_2 &\subset S_2 \\
 S_1 \times S_1 &\subset S_0 \cap S_2 , & S_1 \times S_2 &\subset S_1 , & S_2 \times S_2 &\subset S_0 \cap S_2 .
 \end{aligned}$$

This could be useful for new physical applications because, as shown in [1], this type of decomposition can be used to study expansions of Lie super algebras (used in the context of supergravity and string theory), which have the subspace decomposition $\mathcal{G} = V_0 \oplus V_1 \oplus V_2$ with the following structure

$$\begin{aligned}
 [V_0, V_0] &\subset V_0 , & [V_0, V_1] &\subset V_1 , & [V_0, V_2] &\subset V_2 \\
 [V_1, V_1] &\subset V_0 \oplus V_2 , & [V_1, V_2] &\subset V_1 , & [V_2, V_2] &\subset V_0 \oplus V_2 .
 \end{aligned}$$

On the other hand, as mentioned in [27] it would be interesting to analyze if the S-expansion could help to fit to the classification of solvable Lie algebras of a fixed dimension using S-expansions of semisimple Lie algebras of the same dimension. Thus, our library might also be useful to analyze that problem.

Acknowledgments

We thank Andrei Kataev for his invitation to present this work at ACAT 2017. C.I. was supported by a Mecesup PhD grant and the Término de tesis grant from CONICYT (Chile). C.I. is very grateful to Local Organizing Committee and in particular to Gordon Watts for the financing his participation in ACAT. I.K. was supported by was supported by Fondecyt (Chile) grant 1050512 and by DIUBB (Chile) Grant Nos. 102609 and GI 153209/C. NM is supported by a Becas-Chile postdoctoral grant.

References

- [1] Izaurieta F, Rodriguez E and Salgado P 2006 Expanding Lie (super)algebras through Abelian semigroups *J. Math. Phys.* **47** 123512 (*Preprint* hep-th/0606215)
- [2] Segal I E 1951 A class of operator algebras which are determined by groups *Duke Math. J.* **18** 221-65
- [3] İnönü E and Wigner E P 1953 On the contraction of groups and their representations *Proc. Nat. Acad. Sci. USA* **39** 510-24
- [4] Weimar-Woods E 2000 Contractions, generalized İnönü and Wigner contractions and deformations of finite-dimensional Lie algebras *Rev. Math. Phys.* **12** 1505-29
- [5] Hatsuda M and Sakaguchi M 2003 Wess-Zumino term for the AdS superstring and generalized Inonu-Wigner contraction *Prog. Theor. Phys.* **109** 853 (*Preprint* hep-th/0106114)
- [6] de Azcarraga J A, Izquierdo J M, Picon M and Varela O 2003 Generating Lie and gauge free differential (super)algebras by expanding Maurer-Cartan forms and Chern-Simons supergravity *Nucl. Phys. B* **662** 185 (*Preprint* hep-th/0212347)
- [7] Inostroza C, Kondrashuk I, Merino N and Nadal F 2017 On the algorithm to find S-related Lie algebras *Merged talk by C Inostroza at ACAT 2017*
- [8] <https://github.com/SemigroupExp/Sexpansion/releases/tag/v1.0.0>
- [9] Forsythe G E 1955 SWAC computes 126 distinct semigroups of order 4 *Proc. Amer. Math. Soc.* **6** 443-447
- [10] Motzkin T S and Selfridge J L 1956 Semigroups of order five *Bull. Amer. Math. Soc.* **62** 13-23
- [11] Plemmons R 1967 There are 15973 semigroups of order 6 *Math. Algorithms* **2** 2-17
- [12] Plemmons R 1970 Construction and analysis of non-equivalent finite semigroups *Computational Problems in Abstract Algebra: Proceedings of Conference (Oxford, 1967)* (Oxford: Pergamon) pp. 223-8
- [13] Jürgensen H and Wick P 1977 Die Halbgruppen der Ordnungen ≤ 7 *Semigroup Forum* **14** 69-79
- [14] Satoh S, Yama K, Tokizawa M 1994 Semigroups of order 8 *Semigroup Forum* **49** 7-29
- [15] Distler A and Kelsey T 2009 The monoids of orders eight, nine and ten *Ann. Math. Artif. Intell.* **56** 3-21
- [16] Distler A and T. Kelsey T 2008 The monoids of order eight and nine *9th Int. Conf. on Artificial Intelligence and Symbolic Computation AISC 2008: Proceedings of Conference (Birmingham, July 2008)* ed S Autexier, J Campbell, J Rubio, V Sorge, M Suzuki and F Wiedijk (*Lecture Notes in Computer Science vol 5144*) (Berlin: Springer) pp 61-76
- [17] Distler A and Mitchell J D 2011 Smallsemi - a GAP package, version 0.6.4 <http://tinyurl.com/jdmitchell/smallsemi/>
- [18] Hildebrandt J 2001 Handbook of Finite Semigroup Programs *LSU Mathematics Electronic Preprint Series Preprint 2001-24*
- [19] Distler A, Kelsey T and Mitchell J D <http://www-circa.mcs.st-and.ac.uk/>
- [20] Distler A, Jefferson C, Kelsey T and Kotthof L 2012 The Semigroups of Order 10 *18th Int. Conf. on Principles and Practice of Constraint Programming CP 2012: Proceedings of Conference (Québec City, October 2012)* ed M Milano (*Lecture Notes in Computer Science vol 7514*) (Berlin: Springer Heidelberg) pp 883-899
- [21] <http://math.nist.gov/javanumerics/jama/>
- [22] Inostroza C, Kondrashuk I, Merino N and Nadal F 2017 A Java library to perform S-expansions of Lie algebras *Preprint* arXiv:1703.04036 [cs.MS]
- [23] <https://github.com/SemigroupExp/Sexpansion/wiki>
- [24] Caroca R, Kondrashuk I, Merino N and Nadal F 2013 Bianchi spaces and their three-dimensional isometries as S-expansions of two-dimensional isometries *J. Phys. A* **46** 225201 (*Preprint* arXiv:1104.3541 [math-ph])
- [25] Andrianopoli L, Merino N, Nadal F and Trigiantè M 2013 General properties of the expansion methods of Lie algebras *J. Phys. A* **46** 365204 (*Preprint* arXiv:1308.4832 [gr-qc])
- [26] Inostroza C, Kondrashuk I, Merino N, Nadal F 2017 Algorithm to find S-related Lie algebras *in preparation*
- [27] Nesterenko M 2013 S-expansions of three-dimensional Lie algebras *6th International Workshop Group analysis of differential equations and integrable systems GADEIS-VI: Proceedings of Conference (Protaras, June 2012)* ed O O Vaneeva, C Sophocleous, R O Popovych, P G L. Leach, V M Boyko and P A Damianou (Nicosia: Department of Mathematics and Statistics, University of Cyprus) pp 147-154 (*Preprint* arXiv:1212.1820[math-ph])