

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL - UFRGS
INSTITUTO DE FÍSICA
MESTRADO EM FÍSICA TEÓRICA**

Rafael Eberhardt Sarate

**A PYTHON LIBRARY FOR GROUP THEORY CALCULATIONS IN PARTICLE
PHYSICS**

PORTO ALEGRE

2021

Rafael Eberhardt Sarate

**A PYTHON LIBRARY FOR GROUP THEORY CALCULATIONS IN PARTICLE
PHYSICS**

Trabalho de dissertação de mestrado apresentado como requisito parcial para a obtenção do título de Mestre em Física Teórica, pelo curso de Pós Graduação em Física - Física Teórica, da Universidade Federal do Rio Grande do Sul - UFRGS.

Orientador: Prof. Dr. Daniel Gammerman

Porto Alegre

2021

Summary

1	Introduction	1
2	Theoretical Background	2
2.1	Angular Momentum Operators	3
2.2	Lie Groups	6
2.3	Generalization for any Lie Algebra	7
2.3.1	The Cartan-Weyl Base	8
2.3.2	Graphical Representation of Root Vectors	10
2.3.3	Dynkin Diagrams	12
2.3.4	Cartan Matrices	14
3	Presentation of the Library	14
3.1	Generating Dynkin Diagrams	14
3.2	Building Irreps	16
3.2.1	An Algorithmic Approach	17
3.2.2	Multiplicity of Simple Roots Inside Irreps	18
3.2.2.1	Freudenthal's formula	18
3.2.2.2	The Weyl Orthogonal Group	19
3.3	Dimension of an Irrep	19
3.3.1	The Nomenclature Problem	19
3.4	Multiplication of Irreps	21
3.5	Symmetry Breaking	22
3.5.1	Decomposing Algebras into Non-Semisimple Subalgebras	23
3.5.2	Decomposing Algebras into Semisimple Lie Algebras	23
3.5.3	Breaking Symmetry of Irreps	24
3.6	Multiplet Diagrams in Quantum Number Space	27
4	Applying to Real Life Problems	28
5	Conclusions	33
	BIBLIOGRAPHY	34
	Appendices	37
1	Library's Commands List	39
1.1	Module LieAlgebras	39
1.1.1	The Group type	39
1.1.2	The Weight type	41
1.1.3	The Irrep type	41
1.2	Drawing Diagrams	43
2	Comparisson With Popular Similar Libraries	43

2.1	Calculations for the group E_6	44
2.1.1	Irreps with dimension 27	44
2.1.2	Irreps with dimension 78	45
2.2	Calculations for the group E_8	47
2.2.1	Irreps with dimension 248	47

Acknowledgements

First I ought to acknowledge CAPES for funding my research.

I would like to specially acknowledge the guidance and infinite patience of my supervisor Professor dr. Daniel Gamermann, who turned my almost non-existing knowledge of Group Theory into something appreciable, if I may. Also for believing in me even when I would not. PS. Do not try to code next to him, or you **are** going to feel like a child.

Special thanks should be said to Professor dr. Dimiter Hadjimichef, for being an inspiring figure as he has always been.

I would also want to acknowledge the whole Physics Department from Federal University of Rio Grande do Sul, for promptly helping me trough any issue I could have found along the way.

I thank my girlfriend, Nathalia Rammé, a fellow in the scientific community, for making me company through most of my master's program and building with me a whole new life style together along the way.

I would like also to thank my brother Keith Eberhardt Sarate, for indicating me for a role in the company he works for and providing me the chance to get a job as soon as I finish my master's program.

I want to thank as well my little sister Maria Luiza Eberhardt, for giving me pride as she grows up smart and making life more meaningful. And because she would be jealous of Keith otherwise.

I thank my other little sister Carolina Sarate, for listening so patiently to my explanations about nature and trying to replicate it even though she is just learning how to talk.

I would like to thank my colleague Gustavo Dartora, for being my training partner and because he asked for.

I could not forget to thank my colleagues Erison Rocha, Guilherme Peccinni and Guilherme Volkmer for being examples in the Physics community as well as islands of reason throughout this pandemic times.

1 Introduction

Symmetries are essential in Theoretical Physics, playing important role in Classical Physics in Lagrangian and Hamiltonian formulation, but even more fundamental in modern Physics.

Group Theory is the area of Mathematics whose subject is the study of symmetries, therefore, being present in most of the work of theoretical physicists going back to the formulation of Quantum Mechanics itself. More precisely, the *theory of representations* of groups by linear transformations was essential to the adequate description of the quantum phenomena, for, aside the principal quantum number, the other quantum numbers we know are associated to *representations* of *Lie Groups*. When it comes to gauge theories, one associates fields to representations of such groups and their algebras; interaction terms in the Lagrangian are built upon tensor products of Lie algebra's representations; mechanisms of spontaneous symmetry breaking are developed via decomposition of those into their subalgebras (THOLE, 2019), (SLANSKY,1981). Therefore, being able to perform calculations of such nature is essential for modelling new phenomena or even for simply understanding already existing theories.

Said calculations are rather tedious, though, also not always simple task and, so far, there are not many computational options to take care of matters. Some options found in the literature are presented here (FEGER; KEPHART, 2012), (FEGER; KEPHART; SASKOWSKI, 2019), (LEEUVEN; COHEN; LISSER, 1992), (WYBOURNE, 2002), (NUTMA, 2009). So far as this date, however, there is no similar software built on *Python* language.

Here I present a user friendly Python2 library which is able to extract most of the important information from Lie Groups and Lie Algebras and supply it in a clean and straightforward manner.

Even though the cited libraries are much similar to the one developed in the present work, Python is a programming language widely more popular than the ones utilized in their works, and arguably much more pleasant to use - for it reads much more like the way humans think. Even when compared to other vastly applied programming languages by the academic population, such as C or FORTRAN, Python tends to grow in preference year by year.

Python is a high level programming language and, as such, it is known to be slow, rendering itself not very useful when performance is crucial. For the purpose of dealing with such issue, most of the methods provided by the library are built on C language and translated to Python by means of extensions, more precisely, Python C-API, *Application Programming Interface*. Performance-wise, I write a whole section in the appendices where I run some calculations on our library in comparison with similar calculations run in one the libraries above cited.

Furthermore, the fact that C is also a popular language brings another advantage to

the present work, for then the code can be easily understood or modified by the reader, which is a crucial point: this application aims to be a modern open source alternative, potentially being subjected to peer development, which, from my perspective is what lies in the most essential nature of the scientific endeavor.

In terms of utility, the library is able to deal with classical Lie algebras corresponding to the groups $SU(N)$, $SO(N)$, $Sp(2N)$ and $SO(2l)$ as well as with the exceptional Lie algebras G_2 , F_4 , E_6 , E_7 and E_8 . It has the capability to build *irreducible representations*, or *irreps*, of those algebras; calculate tensor products between irreps; draw pictorial depiction of the algebras, with Dynkin diagrams, and of their irreps, with Hasse diagrams; as well as depict the consequences symmetry breaking over those objects, between others.

The possibility to create diagrams by means of the library is considered a strong didactic characteristic it presents, for it is, arguably, much easier to learn about Group Theory applied to Particle Physics when there exists an easy way of depicting the subject the way I mentioned above.

When it comes to model building, it is not simple to construct a model that is mathematically consistent. In many occasions, Group Theory is a good starting point. The theory provides a framework for constructing analogies or models from abstractions, and for the manipulation of those abstractions to design new systems, make new predictions and propose new hypotheses (SLANSKY, 1981).

The rest of the work is presented in three sections, according to the following structure:

- Theoretical Background: starting from familiar objects like the $SU(2)$ group and its algebra, I develop the generalization of concepts easily understood in Quantum Mechanics and introduce the Group Theory language necessary to understand how the library works, as well as its applicability;
- Presentation of the Library: after introducing the fundamental knowledge regarding Lie algebras, I cover the main functionalities of the library, like building irreducible representations; Dynkin diagrams; diagrams depicting quantum numbers of irreps and symmetry breaking, all by means of examples.
- Applying to Real Life Problems: I show that from the group $SO(10)$ we are able to retrieve the particles of the Standard Model of Particle Physics by means of symmetry breaking and, consequently, that it can be considered as option when it comes to building extensions to the SM.

2 Theoretical Background

Before we get to the core of what is needed for understanding the capabilities of the library and its applications, meaning the study of Lie Groups - or more precisely, Lie Algebras - let us get back a little and make a connection of what is to come with very

basic notions from Classical and Quantum Mechanics. Thus, with the purpose of building intuition, I find it particularly interesting to begin with the discussion of rotations and, consequently, angular *momentum* - which is conserved under central potentials, therefore it can be used to classify quantum states. Then, we are able to define the commutation relations for the angular *momentum* operators and understand they present a typical algebraic structure, making a connection with the symmetry *group of rotations* in Euclidean Space, $SO(3)$. After discussing how to calculate the *spectrum* of the \vec{L}_i operators (get the quantum numbers), we extend such notions to more general Lie algebras.

2.1 Angular *Momentum* Operators

Shall we remember that classically, the orbital angular *momentum* is defined as the vector $\vec{L} = \vec{r} \times \vec{p}$. In Quantum Mechanics, as much as in Classical Physics, there are very important problems of systems under the influence of central fields, e.g. the Hydrogen atom, and, as early as in the first days of the theory as we know today, the counterpart of \vec{L} in QM has been introduced (BORN; JORDAN, 1925), (BORN; HEISENBERG; JORDAN, 1926).

By means of the *correspondence principle* then, the orbital angular *momentum* is defined as a vector *operator* whose components are, in analogy to \vec{L} , given by:

$$\hat{L}_x = \hat{y}\hat{P}_z - \hat{z}\hat{P}_y; \quad (1)$$

$$\hat{L}_y = \hat{z}\hat{P}_x - \hat{x}\hat{P}_z; \quad (2)$$

$$\hat{L}_z = \hat{x}\hat{P}_y - \hat{y}\hat{P}_x. \quad (3)$$

The algebra of such operators is defined by their commutation relations, e.g.:

$$[\hat{L}_x, \hat{L}_y] = i\hbar\hat{L}_z. \quad (4)$$

Applying the equivalent of (4) between the other components of $\hat{\vec{L}}$, which can be easily done by hand, we get the well known algebra of the angular *momentum* operator:

$$[\hat{L}_i, \hat{L}_j] = i\hbar\epsilon_{ijk}\hat{L}_k. \quad (5)$$

It is also easy to check by hand that (5) respects the Jacobi identity:

$$[L_i, [L_j, L_k]] + [L_j, [L_k, L_i]] + [L_k, [L_i, L_j]] = 0, \quad (6)$$

and that information is going to be important later.

Even before the publication of the papers above mentioned, experiments like the one known today as Stern-Gerlach experiment (GERLACH; STERN, 1922) suggested

the existence of another kind of angular *momentum*, an intrinsic one with no classical counterpart. Not much time later, Wolfgang Pauli described the "proper moment" of the electron, what we call *spin* today, as an operator respecting the same algebra as \hat{L} , Eq. (5), (PAULI, 1927).

So let us postulate that Eq. (5) is respected by the components any kind of angular *momentum* operator, and we refer to it as \hat{J} . It is valid, then:

$$[\hat{J}_i, \hat{J}_j] = i\hbar\epsilon_{ijk}\hat{J}_k. \quad (7)$$

Having that settled, we can build an operator that commutes with all components of the angular *momentum*, with the consequence that there are simultaneous eigenvectors between them:

$$\hat{J}^2 = \hat{J}_x^2 + \hat{J}_y^2 + \hat{J}_z^2, \quad (8)$$

and

$$[\hat{J}^2, \hat{J}_i] = 0. \quad (9)$$

We have that \hat{J}^2 and any component of the angular *momentum* are compatible observables.

With the purpose of finding the spectrum of \hat{J}^2 and one of the \hat{J}_i 's, we might still want to define two more new (non-hermitian) operators, \hat{J}_+ and \hat{J}_- as:

$$\hat{J}_\pm = \hat{J}_x \pm i\hat{J}_y. \quad (10)$$

Then we can consider the following eigenvalue problems

$$\hat{J}^2|a, b\rangle = a\hbar^2|a, b\rangle, \quad (11)$$

and

$$\hat{J}_z|a, b\rangle = b\hbar|a, b\rangle. \quad (12)$$

Note that any component of \hat{J}_i could have been used in (12), and we chose \hat{J}_z arbitrarily. Applying (10) on (11) from the right, we get:

$$\hat{J}^2(\hat{J}_\pm|a, b\rangle) = \hat{J}_\pm(\hat{J}^2|a, b\rangle) = a\hbar^2(\hat{J}_\pm|a, b\rangle), \quad (13)$$

for \hat{J}_\pm commutes with \hat{J}^2 , which is promptly seen since both components of \hat{J}_\pm commute with \hat{J}^2 .

Eq. (13) simply means, and that's an important conclusion, that the action of \hat{J}_\pm does not change the eigenvalue of the square of the angular *momentum*.

Now, in order to analyse the action of \hat{J}_\pm on \hat{J}_z , first we need to calculate the commutation relation between them:

$$[\hat{J}_z, \hat{J}_\pm] = [\hat{J}_z, \hat{J}_x \pm i\hat{J}_y] = \pm\hbar\hat{J}_\pm. \quad (14)$$

Going on, we have then:

$$\hat{J}_z \hat{J}_\pm |a, b\rangle = ([\hat{J}_z, \hat{J}_\pm] + \hat{J}_\pm \hat{J}_z) |a, b\rangle = (\pm \hbar \hat{J}_\pm + \hat{J}_\pm \hat{J}_z) |a, b\rangle = (b \pm 1) \hbar \hat{J}_\pm |a, b\rangle, \quad (15)$$

where Eqs. (12) and (14) have been used.

We see, then, that the \hat{J}_+ and \hat{J}_- operators change (*raise* and *lower*, respectively), the value of the eigenvalue b of \hat{J}_z by one unit and, for that reason, are called *step operators*. With *step by step* operations of the step operators, we can find the spectrum of \hat{J}^2 and \hat{J}_z .

From eq. (15) one sees that, $\hat{J}_\pm |a, b\rangle$ is an eigenvector of \hat{J}_z just as much as $|a, b\rangle$, but the fact that the eigenvalue is $b \pm 1$ indicates that $\hat{J}_\pm |a, b\rangle$ is proportional to $|a, b \pm 1\rangle$:

$$\hat{J}_\pm |a, b\rangle = c |a, b \pm 1\rangle. \quad (16)$$

In order to calculate the modulus of c , we multiply (16) from the left with its conjugate:

$$\langle a, b | \hat{J}_\pm^\dagger \hat{J}_\pm |a, b\rangle = c^2. \quad (17)$$

It is easy to see that $\hat{J}_\pm^\dagger = \hat{J}_\mp$, then we have

$$c^2 = \langle a, b | \hat{J}_\mp \hat{J}_\pm |a, b\rangle. \quad (18)$$

So we need to calculate $\hat{J}_\mp \hat{J}_\pm$, which is easily seen to be hermitian:

$$\hat{J}_\mp \hat{J}_\pm = (\hat{J}_x \mp i \hat{J}_y)(\hat{J}_x \pm i \hat{J}_y) = \hat{J}_x^2 + \hat{J}_y^2 \pm i[\hat{J}_x, \hat{J}_y] = \hat{J}^2 - \hat{J}_z^2 \mp \hbar \hat{J}_z. \quad (19)$$

We can now use (19) to calculate the following:

$$\hat{J}_\mp \hat{J}_\pm |a, b\rangle = (\hat{J}^2 - \hat{J}_z^2 \mp \hbar \hat{J}_z) |a, b\rangle = \hbar^2 (a - b(b \pm 1)) |a, b\rangle, \quad (20)$$

so, finally, we have

$$c = \hbar \sqrt{a - b(b \pm 1)}, \quad (21)$$

(except for a phase), and

$$\hat{J}_\pm |a, b\rangle = \hbar \sqrt{a - b(b \pm 1)} |a, b \pm 1\rangle. \quad (22)$$

From Eq. (22) we extract the fact that, given successive applications of \hat{J}_+ , there must be a maximum value for $b(b+1)$, otherwise the eigenvalue would not be real. We may say $b_{max} = l$, then $a = l(l+1)$.

Analogously, for successive applications of \hat{J}_- , there must be a b_{min} such that $a = l(l+1) = b_{min}(b_{min}+1)$. In other words, $b_{min} = -l$.

In conclusion, we have $-l \leq b \leq l$, and b is allowed to vary 1 unit up or down on each application of \hat{J}_\pm . Also, the whole interval $[-l, l]$ is only completely covered if l is an integer or semi-integer number.

We may now, after everything, write:

$$\hat{J}^2|l, m\rangle = \hbar^2 l(l+1)|l, m\rangle; \quad (23)$$

$$\hat{J}_z|l, m\rangle = \hbar m|l, m\rangle; \quad (24)$$

$$\hat{J}_{\pm}|l, m\rangle = \hbar\sqrt{l(l+1) - m(m \pm 1)}|l, m \pm 1\rangle, \quad (25)$$

where $l = 0, \frac{1}{2}, 1, \frac{3}{2}, 2, \dots$ and $m = -l, -l+1, -l+2, \dots, 0, \dots, l-2, l-1, l$.

We are done with the eigenvalue problem, and it is important for our purposes to emphasize the fact that all the results so far are just algebraic consequences of Eq. (7), showing the power of being able to manipulate such a structure. In fact, it would show to be very useful if we could generalize the procedure above to more complex systems, and that is exactly the point of this whole work. We are going to see that any system provided with symmetry is subject to be treated by means of similar tools. For some systems, though, the calculations might be extensive and boring, then our library can do the job for you.

2.2 Lie Groups

Everything we have seen so far comes naturally through the development of Quantum Mechanics, but we need to understand it as the outcome of more abstract concepts such as *groups* and *algebras*, more specifically, *Lie Groups* and *Lie Algebras*. Let us start, then, by defining what consists a group.

A group, \mathcal{G} , is a set, g , provided with an operation called *product*, $*$, respecting the following conditions:

1. **Closure:** the set g is closed under the product: $\forall a, b \in g, a * b \in g$.
2. **Associativity:** $\forall a, b, c \in g, (a * b) * c = a * (b * c)$;
3. **Identity:** There exist some $I_{\mathcal{G}}$ in \mathcal{G} , also called *neutral element*, such that $\forall a \in g, a * I_{\mathcal{G}} = I_{\mathcal{G}} * a = a$;
4. **Inverse:** For all element $a \in g$, there must be an inverse, $a^{-1} \in g$, such that $a * a^{-1} = a^{-1} * a = I_{\mathcal{G}}$.

In order to build intuition, and make an important connection between symmetries and groups, we might consider the example of rotations that preserve the symmetry of systems which indeed possess rotational symmetry. If we represent such rotations by rotation matrices, it is easy to show that they respect the above conditions with the matrix multiplication as the product.

Back to Quantum Mechanics, the rotation operator, \hat{D} , is a matrix whose dimension depends on the ket space it acts on. For infinitesimal rotations around $\hat{\mathbf{n}}$, of an angle $\delta\phi$, it assumes the form

$$\hat{D}(\hat{\mathbf{n}}, \delta\phi) = \mathbb{1} - \frac{1}{\hbar} \hat{\mathbf{n}} \cdot \hat{\mathbf{J}} \delta\phi. \quad (26)$$

In the case of finite rotations, it takes the form

$$\hat{D}(\hat{\mathbf{n}}, \phi) = \lim_{N \rightarrow \infty} \left(\mathbb{1} - \frac{\hat{\mathbf{n}} \cdot \hat{\mathbf{J}} \delta\phi}{\hbar N} \right)^N = \exp \left\{ -\frac{i\phi \hat{\mathbf{n}} \cdot \hat{\mathbf{J}}}{\hbar} \right\}, \quad (27)$$

where the angular *momentum*, $\hat{\mathbf{J}}$, is said to be the *generator* of the rotation.

It is important for us to know that the elements of any group associated to *continuous* transformations just like the above, denominated a Lie Group ¹, have the same form as (27). Also, in order to get information about continuous groups, suffices to know the relationship between its generators instead of working with the group elements themselves - anytime we want, we can take the exponential of the former to retrieve the latter. Those relationships are in fact the algebra of the generators and, in the case of a Lie Group, they are called Lie Algebras.

Equations (5) and (7) are examples of Lie Algebras, both associated to *rotation* groups, but of slightly different kinds. The first is the group of all 3×3 *orthogonal* matrices whose determinant is +1, called $SO(3)$. The second is the group formed by all 2×2 *unitary* matrices of determinant +1, denominated $SU(2)$. The fact that $SO(3)$ and $SU(2)$ have identical algebras is called *isomorphism*.

We have seen that utilizing algebraic tools for dealing with the angular *momentum* operator has proven to be very useful in the problem of determining its spectrum, therefore it would be really convenient if we were able to find similar structures in other algebras, for then we could just copy the same procedures we are already familiar with.

2.3 Generalization for any Lie Algebra

Much of what we are going to encounter next is the work of a Russian mathematician called Eugene Dynkin, who, in 1947, published a paper in which he presented complex Lie Algebras in the structure we are about to study, as well as invented what we call *root system*, fundamental for our present work, and the *Dynkin diagrams*, that enable us to visualize algebras in a very intuitive way (DYNKIN, 1947). Just like we have done for the case of $SU(2)$, it is by means of the root system of a *semisimple Lie Algebra* that we are able to describe the effect of the lowering and raising operators of the group's algebra on the eigenvalues (quantum numbers) of the diagonal generators, and provide geometrical interpretation of the commutation relations.

¹ After the Norwegian mathematician Sophus Lie.

At this point, we are almost ready to develop the knowledge necessary to comprehend our library in its full glory. Before going ahead, though, we should clarify the concept of semisimple Lie Algebra just mentioned. We say a Lie Algebra is *semisimple* when it is the direct sum of *simple* Lie Algebras, meaning the algebras of simple Lie Groups. The latter are, in turn, groups that do not possess a *continuous invariant subgroup*. Finally, an *invariant subgroup*, a , is a subgroup of \mathcal{G} such that the action of the elements of \mathcal{G} on its elements does not take them out of a .

2.3.1 The Cartan-Weyl Base

After finding the commutation relation between the generators of a group, like Eqs. (5) and (7), which is the *standard* form of the algebra associated to it, it is always possible to make a base transition by means of a linear combination of the generators.

What is important is the fact that, for any basis, the commutation relation between the generators may be written in the following form ((DYNKIN, 1947), (DYNKIN, 1946), (THOLE, 2019)):

$$[\hat{J}_i, \hat{J}_j] = \sum_{k=1}^n C_{ij}^k \hat{J}_k, \quad (28)$$

where n is the dimension of \mathcal{G} and C_{ij}^k are the so called structure constants which, locally, determine \mathcal{G} completely.

It is always possible, then, to write the algebra in a basis such that it contains the *largest possible number of generators commuting with each other*, or, in other words, all the *simultaneously diagonalizable generators* - or, yet, the maximal *abelian subalgebra*. We call that the *Cartan subalgebra* of the \mathcal{G} . The cardinality of the Cartan subalgebra, l , corresponds to the *rank* of the group - which is analog to the notion of dimension of a vector space - and that also means the maximum amount of structure constants equal to 0 for that algebra. The rest of the non-zero structure constants are going to play a very crucial role in the development of our work, receiving the special name of *roots* of the algebra.

Said basis is called Cartan-Weyl basis and, in order to construct it, we start with the Cartan subalgebra:

$$[\hat{H}_i, \hat{H}_j] = 0; \quad i, j = 1, 2, 3, \dots, l, \quad (29)$$

where \hat{H}_i have act like *weight operators*.

Than we demand the other generators, to which we will refer as \hat{E}_α to respect the following commutation relation:

$$[\hat{H}_i, \hat{E}_\alpha] = \alpha_i \hat{E}_\alpha; \quad i, j = 1, 2, 3, \dots, l, \quad (30)$$

where the *weight of the operators*, the α_i are the contravariant components of a vector, $\vec{\alpha}$, in an l -dimensional space, called *root vector*. It is important to point out that the E_α act as the step operators we defined for $SU(2)$ (GEORGI, 1999).

For the algebra to be complete, we need two more commutation relations, that can be obtained from the first two.

By taking the hermitian conjugate of (30), we have:

$$[\hat{H}_i, \hat{E}_\alpha]^\dagger = -\alpha_i \hat{E}_\alpha^\dagger, \quad (31)$$

meaning $E_\alpha^\dagger = E_{-\alpha}$. And we interpret E_α as a *raising* and $E_{-\alpha}$ as a *lowering* operator.

Now, the generators of the algebra written in this basis still respect the Jacobi identity (BRIAN, 2015). So we have that

$$[\hat{H}_i, [\hat{E}_\alpha, \hat{E}_{-\alpha}]] = -[\hat{E}_\alpha, [\hat{E}_{-\alpha}, \hat{H}_i]] - [\hat{E}_{-\alpha}, [\hat{H}_i, \hat{E}_\alpha]] = 0, \quad (32)$$

meaning $[E_\alpha, E_{-\alpha}]$ is a generator of the Cartan subalgebra, then it might be written as a linear combination of the \hat{H}_i :

$$[\hat{E}_\alpha, \hat{E}_{-\alpha}] = \sum_{i=1}^l \alpha_i \hat{H}_i = \vec{\alpha} \cdot \vec{H}. \quad (33)$$

By means of the Jacobi identity again, we have also that

$$[\hat{H}_i, [\hat{E}_\alpha, \hat{E}_\beta]] = -[\hat{E}_\alpha, [\hat{E}_\beta, \hat{H}_i]] - [\hat{E}_\beta, [\hat{H}_i, \hat{E}_\alpha]] = (\alpha_i + \beta_i)[\hat{E}_\alpha, \hat{E}_\beta] = N_{\alpha\beta} \hat{E}_{\alpha+\beta}, \quad (34)$$

where $\alpha + \beta \neq 0$ and $N_{\alpha\beta}$ if $\alpha + \beta$ is not a root.

Finally, we conclude that an algebra written in the Cartan-Weyl basis has the general form ²:

$$[\hat{E}_\alpha, \hat{E}_\beta] = N_{\alpha\beta} \hat{E}_{\alpha+\beta}, \quad \alpha \neq \beta; \quad (35)$$

$$[\hat{E}_\alpha, \hat{E}_{-\alpha}] = \alpha^i H_i; \quad (36)$$

$$[\hat{H}_i, \hat{E}_\alpha] = \alpha_i \hat{E}_\alpha; \quad (37)$$

$$[\hat{H}_i, \hat{H}_j] = 0. \quad (38)$$

It should be clear that this way of writing the algebra is completely analogous to the algebras of $SO(3)$ and $SU(2)$ we have seen above, just written in a different base. In this language, operators like \hat{J}^2 receive a new denomination. One characteristic that renders it useful in $SU(2)$ is the fact that it commutes with all components of \hat{J} , or, in other

² The reader may find a rich derivation of all that algebra in (JACOBSON, 1979) and (THOLE, 2019) or, in more straightforward manner, in (GREINER; MÜLLER, 1994).

words, with all generators of the group. Operators with that property are called *Casimir operators*, and their number in a group corresponds to its rank.

Being able to write the algebra in the above fashion is advantageous because then the Casimir operators together with the weight operators form a commuting set of hermitian operators and, as such, there are states that can be defined with respect to them and used to *define quantum numbers*. Also, the manner in which we defined and utilized the step operators in $SU(2)$ can now be reproduced for more complex algebras.

2.3.2 Graphical Representation of Root Vectors

Since the roots of the algebras written in the Cartan-Weyl basis are vectors, we are able to define the scalar product between them:

$$(\alpha, \beta) = \alpha^i \beta_i. \quad (39)$$

Then, follow three important consequences ((DYNKIN, 1947), (DYNKIN, 1946)), that I present without proof³:

1. If α and β are roots, then $2(\alpha, \beta)/(\alpha, \alpha)$ is an integer and $\beta - 2(\alpha, \beta)/(\alpha, \alpha)$ is also a root.
2. If α is a root vector, then $\alpha, 0$ and $-\alpha$ are the only integer multiples of α that are also root vectors.
3. A root series based on α which contains another root β consists of no more than four roots, satisfying $2(\alpha, \beta)/(\alpha, \alpha) = 0, \pm 1, \pm 2, \pm 3$.

From what is stated above, if we draw vectors corresponding to each root, an l -dimensional diagram is built and it is unique for each algebra, therefore an useful way of visualizing quantum numbers.

Still, as a consequence of what has been stated, it can be shown that the angle between those vectors might only be $\phi = 0^\circ, 30^\circ, 45^\circ, 60^\circ$ and 90° (CARTER, 2005).

It is also easy to show the connection between those angles and the ratios of the roots α and β ⁴, $(\alpha, \alpha)/(\beta, \beta)$ ((CARTER, 2005), (GEORGI, 1999)):

$$\begin{aligned} \phi = 30^\circ, \quad \frac{(\alpha, \alpha)}{(\beta, \beta)} = 3; \quad \phi = 45^\circ, \quad \frac{(\alpha, \alpha)}{(\beta, \beta)} = 2; \\ \phi = 60^\circ, \quad \frac{(\alpha, \alpha)}{(\beta, \beta)} = 1; \quad \phi = 90^\circ, \quad \frac{(\alpha, \alpha)}{(\beta, \beta)} : \text{not enough information to know.} \end{aligned}$$

Some graphics generated by our library can be seen in Figs. 1, 2 and 3. Those figures are important because the irreps they represent are going to appear again in other examples along the text.

³ The reader may find the whole development of those properties in (CARTER, 2005), or (GREINER; MÜLLER, 1994)

⁴ Here, care must be taken to the fact that the shorter root is chosen to be β .

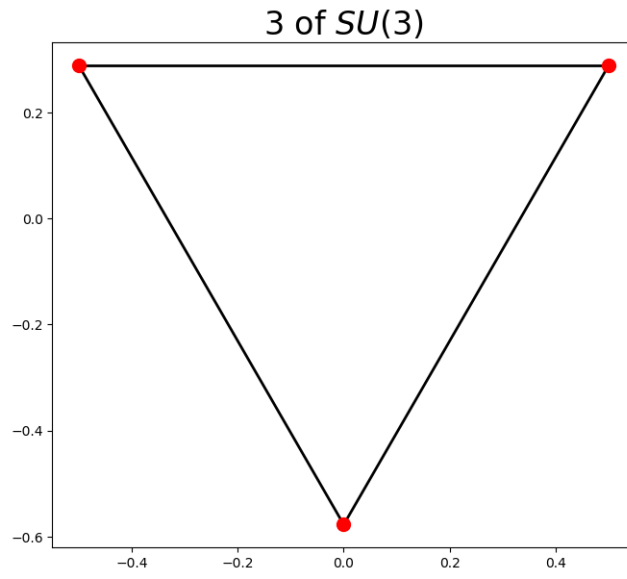


Figure 1 – Diagram for 3 of $SU(3)$ in quantum number space.

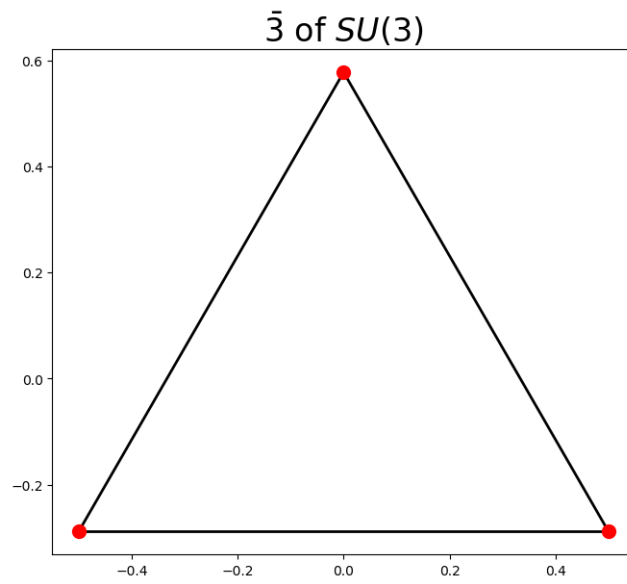


Figure 2 – Diagram for $\bar{3}$ of $SU(3)$ in quantum number space.

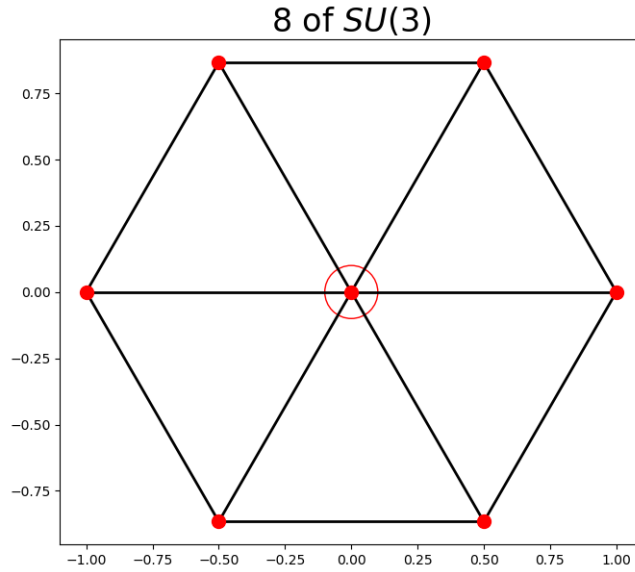


Figure 3 – Diagram for 8 of $SU(3)$ in quantum number space.

2.3.3 Dynkin Diagrams

Dynkin realized that all the information about an algebra can be retrieved from a small subset of its root vectors, what he called the *simple roots*. And it is possible to build a diagram representing them, which we call today a *Dynkin diagram*, from which is easy to deduce the angles between the positive roots, as well as the ratio between their lengths.

In order to define a simple root then (which is a definition depending on the basis choice), we need first to define what is a *positive root*, which is a root whose first non-negative coordinate, in that basis, is positive. Now, we say that a root is *simple* if it is not the sum of two positive roots.

There are also three conditions regarding simple roots, that, again, I present without proof ((CARTER, 2005),(GREINER; MÜLLER, 1994), (THOLE, 2019)):

1. If α and β are simple roots, $\alpha - \beta$ is not a simple root.
2. If α and β are simple roots, $2(\alpha, \beta)/(\alpha, \alpha) = -p$, where $p \in \mathbb{Z}^+$.
3. The angles between two simple roots might only be 90° , 120° , 135° or 150°

If $(\beta, \beta) \leq (\alpha, \alpha)$, it follows:

$$\phi = 120^\circ, \quad \frac{(\alpha, \alpha)}{(\beta, \beta)} = 1; \quad \phi = 135^\circ, \quad \frac{(\alpha, \alpha)}{(\beta, \beta)} = 2;$$

$$\phi = 150^\circ, \quad \frac{(\alpha, \alpha)}{(\beta, \beta)} = 3; \quad \phi = 90^\circ, \quad \frac{(\alpha, \alpha)}{(\beta, \beta)} : \text{not enough information to know.}$$

Dynkin prescribed that in the diagram, the simple roots are represented by circles connected by one, two or three lines in case the angles between them are 120° , 135°

or 150° respectively, while orthogonal roots are disconnected. The shortest root vectors are represented by filled circles, as opposed to the unfilled ones regarding the largest vectors. Consequently, each algebra is uniquely represented as we are able to see in Table 1⁵:

Cartan's Notation	Group	Dynkin Diagram
A_l	$SU(l+1)$	
B_l	$SO(2l+1)$	
C_l	$Sp(2l)$	
D_l	$SO(2l)$	
G_2	G_2	
F_4	F_4	
E_6	E_6	
E_7	E_7	
E_8	E_8	

Table 1 – Dynkin diagram for classical and exceptional Lie Algebras.

⁵ Other connected diagrams are not possible.

2.3.4 Cartan Matrices

Now it has come the time to talk about the most fundamental object to our library. To each algebra it is possible to relate a matrix, called *Cartan Matrix*, containing pretty much the same information as in the Dynkin diagrams (CARTER, 2005). Therefore, upon having one, the other may be obtained.

The elements of the Cartan matrix are given by:

$$A_{ij} = 2 \frac{(\alpha_i, \alpha_j)}{(\alpha_i, \alpha_i)}, \quad (40)$$

where α_i are simple roots.

The usage of our library will typically start with the construction of a group based on the respective Cartan Matrix, from which all information about the algebra is generated.

I reckon we have already established enough theoretical background to move on and present the library already. I am going to treat topics like the construction of *irreducible representations*, or *irreps*, along with their dimensions and their classification accordingly; products of irreps; symmetry breaking and the creation of diagrams containing information about the algebras. Besides, I am going to illustrate each point with examples of how the library works. The subject is going to be developed in a way to prepare the reader to follow along a practical application of the library and its tools, which is going to involve symmetry breaking of groups larger than $SU(3) \otimes SU(2)$ to try and fit the Standard Model inside a possible candidate for extensions of the theory.

3 Presentation of the Library

Here I illustrate the underlying reasoning by which some of the main capabilities possessed by the library have been developed. I do not have the pretension of writing a manual, since I do not want to show how to use the library, but how it actually produces the results it is able to generate.

That being said, I mean to cover only part of the vast endeavor that comprehends the library, for the rest of its functionalities are all built upon what I am about to expose. In the appendices, the reader find all the methods the library possesses and the comparison of its performance with a similar library found in the literature.

3.1 Generating Dynkin Diagrams

We have already seen that there are conditions respected by the simple roots of a semisimple Lie Algebra ((CARTER, 2005), (THOLE, 2019)), and that taking them in consideration, Dynkin proposed the construction of the diagrams now known by his name (DYNKIN, 1946).

I am going to use the same conditions to work out an example of how the library builds the referred diagram *from* a given Cartan Matrix. So far I have used $SU(n)$ as example whenever one was given, but the diagrams corresponding to those algebras are rather too simple (see Table 1). Therefore, it might be interesting to work on the example of F_4 's algebra, whose Cartan Matrix is the following:

$$A_{ij} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -2 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \quad (41)$$

Since $A_{02} = A_{03} = 0$ in (41), we gather $(\alpha_0, \alpha_2) = (\alpha_0, \alpha_3) = 0$, meaning those roots are orthogonal to each other, so they are not connected in the diagram.

Furthermore, since $A_{01} = A_{10} = -1$, it means that

$$\frac{A_{01}}{A_{10}} = \frac{(\alpha_1, \alpha_1)}{(\alpha_0, \alpha_0)} = 1, \quad (42)$$

where Eq. (40) was used.

Therefore, the angle between α_0 and α_1 is 120° and, by the rules proposed by Dynkin, the related circles should be connected by one line only. Equation (42) also implies that α_0 and α_1 have the same length.

Similarly, from (41) also we have that

$$\frac{A_{12}}{A_{21}} = \frac{(\alpha_2, \alpha_2)}{(\alpha_1, \alpha_1)} = 2, \quad (43)$$

meaning the angle between α_1 and α_2 is 135° , and the corresponding circles should be connected by two lines. Also, the last result points out α_1 has the shortest length compared with α_2 , and the corresponding circle must be filled, while the other remains unfilled. Consequently, the circle bound to α_0 is also filled.

By the similar reasoning, α_2 and α_3 have the same length and the circle corresponding to α_3 must be unfilled as well. The resulting diagram can be seen in Figure 4.

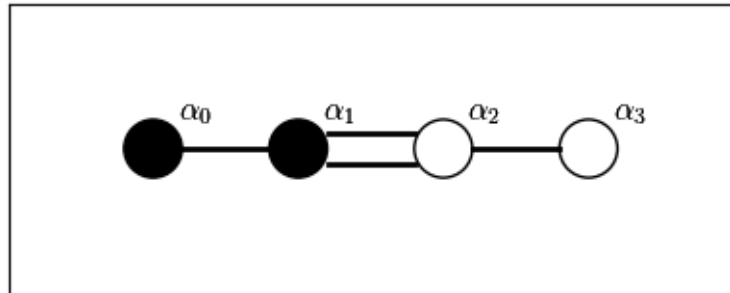


Figure 4 – Dynkin diagram for the algebra of F_4 .

The library is also capable of producing Dynkin diagrams for groups whose symmetry has been broken, and we come back the issue later, when we will see that it makes the library really useful to work with GUTS.

3.2 Building Irreps

In Quantum Mechanics, we are used to the concept of *multiplet*, consisting of states defined by the eigenvalues of some observables appropriate to define quantum number. Also, by means of the step operators, one is able to “navigate” between the states belonging to the multiplet, but no state outside the multiplet can be accessed by their action. The notion of multiplet can be extended for more complex algebras, and henceforth, we are going to regard it as an *irreducible representation* of an algebra, or *irrep*.

The basis vectors $|u\rangle$ of an irrep are classified by their eigenvalues with respect to the operators H_i of the Cartan subalgebra (SLANSKY, 1981):

$$H_i|u\rangle = \lambda_i(u)|u\rangle. \quad (44)$$

We call the *weight* of a state the vector

$$\Lambda(u) = (\lambda_1(u), \dots, \lambda_l(u)), \quad (45)$$

where l is the rank of the Lie algebra.

We can express the weights in terms of the simple roots of the algebra ((BOER, 2021/22), (SLANSKY, 1981)):

$$\lambda = \sum_{i=1}^l \mu_i \alpha_i; \quad \mu_i \in \mathcal{Q}. \quad (46)$$

We can also define an ordering relation between weight vectors: we say λ is greater than λ' if the first non-zero $\lambda_i - \lambda'_i, i = 1, 2, \dots, N$ is positive. As such we uniquely determine a *maximal weight*, Λ_{max} for each irrep, with the properties:

1. Λ_{max} is non-degenerate;
2. There is a one-to-one correspondence between Λ_{max} and the corresponding irrep.

It is important to point out that in (45), the λ 's correspond to the quantum numbers of the irrep. Now, in order to make them useful for our purposes, we must write them in the so called Dynkin basis, where, for each Λ , its components in this form are ((GREINER; MÜLLER, 1994), (THOLE, 2019)) the following:

$$a_i = \frac{2(\lambda, \alpha_i)}{(\alpha_i, \alpha_i)}. \quad (47)$$

Now, it is easy to see that, from the definition above, the a_i 's are all integers, turning the calculations easier. As an example, a triplet of $SU(3)$, in the Dynkin basis, is $(1, 0)$ or $(0, 1)$.

Now I am able to illustrate, by means of examples, how the library builds irreps based on the facts above mentioned.

3.2.1 An Algorithmic Approach

What I am going to present is an algorithmic procedure, for that is how the library works. We start by choosing the group with which we want to work and the maximal weight of the irrep we want to construct. I say it might be interesting to build the irrep for $\Lambda_{max} = (1, 0, 1)$, in $SU(4)$, given its algebra is not as simple as the $SU(2)$'s and $SU(3)$'s but still not so complex to the point of turning the following calculations too absorbing.

To begin with, we need the Cartan matrix of $SU(4)$, which can be easily deduced from the corresponding Dynkin diagram (Table 1). However, since our library is able to provide the Cartan matrix for any Lie group, I am going to skip ahead and start from the point where said matrix is given:

$$A_{ij} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \quad (48)$$

Now, the lines of the Cartan matrix coincide with the simple roots of the algebra written in the Dynkin basis, and that is a powerful consequence of what we just discussed above. Therefore, we get, as simple roots of $SU(4)$, in said basis, the following:

$$\alpha_0 = (2, -1, 0); \quad \alpha_1 = (-1, 2, -1); \quad \alpha_2 = (0, -1, 2). \quad (49)$$

Now, from $\Lambda_{max} = (1, 0, 1)$ we can “go down” subtracting α_0 , α_1 or α_2 in the correct fashion:

- since the first element of Λ_{max} is ≥ 0 and equal to 1, we subtract α_0 from Λ_{max} *once* and the resulting vector is a root;
- since the second element is 0, we cannot subtract α_1 from Λ_{max} ;
- since the third element is again ≥ 0 and equal to 1, we subtract α_2 from Λ_{max} *once* and the resulting vector is a root.

We repeat the process for each root arising from the last step till the point where no positive root is produced ⁶. The result of the whole process is shown in the Figure 5, which is a Hasse diagram generated by our very library.

We should now attend to two facts:

⁶ If any element of Λ_{max} were greater than 1, one should subtract the convenient simple root (α_0 , α_1 or α_2) an amount of times equal to the integer element even if the corresponding element in an resulting root along the way was ≤ 0 .

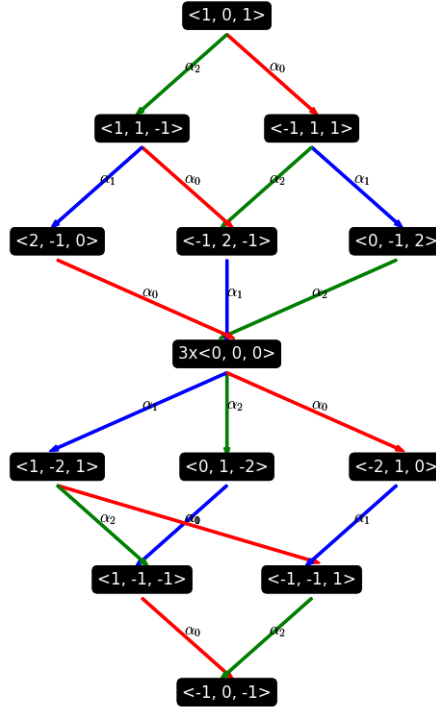


Figure 5 – Irreducible representation of dimension 15 in $SU(4)$.

1. The root with weight $(-1, 2, -1)$ has been produced twice in the second step (once for each root obtained in the first round of subtractions), as well as $(1, -1, -1)$, $(-1, -1, 1)$ and $(-1, 0, -1)$ later on, but it does not necessarily mean the states in question are degenerate, although they might be. Similarly, $(0, 0, 0)$ has been obtained three times, but has not necessarily multiplicity 3. We will turn back to this fact later.
2. Since every element of the resulting root in the third step is 0, we could only go on because there were elements equal to 2 in the roots⁷ from the step before, so we get the right to evolve one more step despite.

Remains the problem of determining the multiplicity of each obtained root, and, for that matter, one might use both the methods furnished next.

3.2.2 Multiplicity of Simple Roots Inside Irreps

3.2.2.1 Freudenthal's formula

Let R contain the roots of the algebra in question, then it is valid the following recurrence formula, called *Freudenthal's formula* (BERTRAM, 1959):

$$((\Lambda_{max} + \delta, \Lambda_{max} + \delta) - (\Lambda' + \delta, \Lambda' + \delta))n_{\Lambda'} = 2 \sum_{\alpha \in R; \alpha \geq 0} \sum_{k=1}^{\infty} n_{\Lambda'+k\alpha}(\Lambda' + k\alpha, \alpha), \quad (50)$$

⁷ Here I interchange the words *weight* and *root* when talking about irreps, for we do not actually need to distinguish both concepts in our scope.

where $\delta = \sum_{\alpha \in R; \alpha \geq 0} \alpha/2$ and $n_\Lambda = 0$ if Λ is not a weight of the representation. In addition, for any $\Lambda' \neq \Lambda_{max}$, $n_{\Lambda'} \neq 0$. Note also, that $n_{\Lambda_{max}} = 1$, always.

There is still another, much easier way of determining the multiplicity of the roots: observing the symmetry associated to the roots and their reflections, and we see more details below.

3.2.2.2 The Weyl Orthogonal Group

From Figure 5 we gather the diagram is provided with symmetry and, based only on that observation, we can determine if the number of ways each root was generated while building the irrep corresponds or not to its multiplicity.

Before going on, we should remember that, as exposed earlier, in a root system R for each $\alpha \in R$, $\beta - 2\alpha \frac{(\alpha, \beta)}{(\alpha, \alpha)}$ is also a root ($\beta \in R$), which now we recognize as a reflection of S_α about the hyperplane perpendicular to it. We call the *Weyl group*, W , the subgroup formed by all the reflections S_α in R (JACOBSON, 1979).

Now, given the symmetry of α and S_α , we know that they should have the same multiplicity. Then, if we look to the roots $\alpha = (-1, 2, -1)$ and $S_\alpha = (1, -2, 1)$ we see that the latter has multiplicity 1, thus the former should also be non-degenerate. As another example, if we compare the roots $(1, 0, 1)$ and $(-1, 0, -1)$, we see that the latter should also be non-degenerate, for the former has multiplicity 1 as well.

After all we should conclude that the only root that can have multiplicity greater than 1 is $(0, 0, 0)$ and, as already pointed out in Fig. 5, it is three times degenerate.

Finally, counting up all the roots in the irrep according to their multiplicity, we conclude that its dimension is 15.

Before going on, it is important to say that our library is able to calculate the multiplicity of the roots in a irrep by means of both methods discussed above.

3.3 Dimension of an Irrep

It is not necessary to build the irrep to know its dimension given the highest weight, $\vec{\Lambda}_{max}$, by means of the *Weyl dimension* formula:

$$\dim(\vec{\Lambda}_{max}) = \prod_{\alpha^+} \frac{(\alpha^+, \vec{\Lambda}_{max} + \vec{\delta})}{(\alpha^+, \vec{\delta})}, \quad (51)$$

where I use α^+ to refer to a positive root and $\vec{\delta} = (1, 1, \dots, 1)$ (with the same number of components as the given weight).

3.3.1 The Nomenclature Problem

It is possible to have more than one irrep with the same dimension. One example is the 3 and $\bar{3}$ of $SU(3)$, where $\bar{3}$ is the conjugate of 3. But that is still not the only possibility

of confusion the library is going to find along the way, for there are irreps with the same dimension in an algebra which are not the conjugate of each other.

For solving that issue, we define an index to the irrep, given as follows:

$$Ind = \frac{dim(\vec{\Lambda}_{max})}{ord(\mathcal{G})} \vec{\Lambda}_{max} \cdot (\vec{\Lambda}_{max} + 2\vec{\delta}), \quad (52)$$

where $ord(\mathcal{G})$ is the *order* of the group, meaning the number of elements present in it.

If both the dimension and index are the same for two different Λ_{max} , the irreps differ by *congruency number*, c , which is actually a single number for all algebras we have here encountered except for D_l , being a two component vector in this case. Then, given an irrep α , its "congruency number" is given by ((MCKAY, 1981), (LEMIRE; PATERA, 1980)):

$$A_l : c = \sum_{k=1}^l k\alpha_k \pmod{l+1}. \quad (53)$$

$$B_l : c = \alpha_l \pmod{2}. \quad (54)$$

$$C_l : c = \sum_{k=0}^{\frac{l-1}{2}} \alpha_{2k+1} \pmod{2}. \quad (55)$$

$$D_l : c = \left(\begin{array}{c} \alpha_{n-1} + \alpha_n \pmod{2} \\ \sum_{k=0}^{\frac{l-3}{2}} \alpha_{2k+1} + (l-2)\alpha_{l-1} + l\alpha_l \pmod{4} \end{array} \right). \quad (56)$$

$$E_6 : c = \alpha_1 - \alpha_2 + \alpha_4 - \alpha_5 \pmod{3}. \quad (57)$$

$$E_7 : c = \alpha_4 - \alpha_6 + \alpha_7 \pmod{2}. \quad (58)$$

$$E_8, F_4, G_2 : c = 0. \quad (59)$$

As an example, all the irreps whose Λ_{max} are following have dimension 20: $(0, 2, 0)$, $(1, 1, 0)$, $(0, 1, 1)$, $(3, 0, 0)$, $(0, 0, 3)$.

From which fololows:

- The irrep associated with $(0, 2, 0)$ has index 16.000000 and congruence class 0.
- The irrep associated with $(1, 1, 0)$ has index 13.000000 and congruence class 3.
- The irrep associated with $(0, 1, 1)$ has index 13.000000 and congruence class 1.
- The irrep associated with $(3, 0, 0)$ has index 21.000000 and congruence class 3.
- The irrep associated with $(0, 0, 3)$ has index 21.000000 and congruence class 1.



Figure 6 – Hasse diagram for 3 in $SU(3)$.

Now, the weights $(1, 1, 0)$ and $(0, 1, 1)$ have the lowest indices among all irreps above, therefore they are not primed in their names. Next is the self conjugated irrep associated with the label $(0, 2, 0)$ it receives, therefore, a single prime. Finally, irreps with the third highest index receive two primes. And the prime counts would keep increasing were there more irreps of dimension 20. They are then expressed, respectively, as $20, \bar{20}, 20', 20'', \bar{20}''$.

3.4 Multiplication of Irreps

The evident majority of systems in nature are made of multiple particles. Then, in the case of coupling between the particles, the individual quantum numbers might not be conserved, and we are not able to work with the states corresponding to the individual parts. In that case, we might want to work with the space generated from the product between irreps, which is reducible.

Again I am going to present an algorithmic approach in order to illustrate how the library works out the tensor product between irreps. Let us then go about the product $3 \otimes 3$ in $SU(3)$ ⁸.

We start by building the irreps we want to multiply, the way we have already learned. In our case, both are the same, so we can focus on the algorithm. The Hasse diagram for the 3 in $SU(3)$ can be seen in Figure 6.

Next, we sum term by term each root in the first irrep to each root in the second. In sequence, we search for the highest positive weight in the obtained list and build the

⁸ In general, the user don't need to know the dimensions of the irreps being multiplied, only the maximum weights.

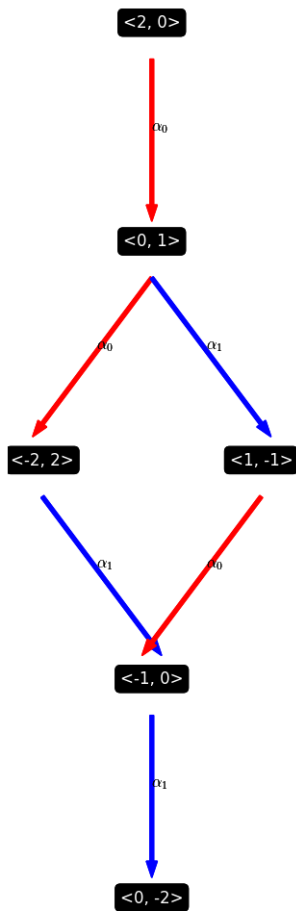


Figure 7 – Hasse diagram for $\bar{6}$ in $SU(3)$.

corresponding irrep. After eliminating from the list the weights belonging to that irrep, we repeat the process until the whole list is spanned.

In our example, the first step gives $(2, 0)$, $(0, 1)$, $(1, -1)$, $(0, 1)$, $(-2, 2)$, $(-1, 0)$, $(1, -1)$, $(-1, 0)$ and $(0, -2)$. The highest weight in the list is $(2, 0)$, whose corresponding irrep can be seen in Figure 7, and has dimension 6.

Among the weights left, the only positive is $(0, 1)$, which is clearly a $\bar{3}$ (mind that the maximum weight of $\bar{3}$ is $(1, 0)$), and obviously the two weights left belong to it.

Finally, we should conclude that⁹, in $SU(3)$, $3 \otimes 3 = \bar{6} \oplus \bar{3}$.

3.5 Symmetry Breaking

It is important to point out that the Standard Model of Particle Physics, SM, is built in terms of Lie Algebras (PESKIN; SCHROEDER, 1995), meaning all fundamental forces except gravity are underlined by means of symmetry groups, more specifically, $SU(3)$, $SU(2)$ and $U(1)$. These correspond, respectively, to the strong, weak and electromagnetic forces, where each existing particle corresponds to a representation of those groups.

⁹ In the majority of the literature, this result is presented as $3 \otimes 3 = 6 \oplus \bar{3}$, because of a historical convention (see (ZEE, 1982)) that we do not obey in the library.

The SM is the most successful theory for the description of elementary particles and their interactions mankind has developed so far, but there are limitations. The model presents 19 free parameters (25 if we take in consideration the masses of neutrinos and the parameters associated to cross-generational mixing between them) not derived from principles, but which have been introduced so the model agrees with observation (THOMSON, 2013).

Among other reasons, that is enough to motivate the search for groups with larger symmetries than the Standard Model from which $SU(3) \otimes SU(2) \otimes U(1)$ emerges under *symmetry breaking*. Some of the most explored candidates as symmetry group for the so called GUT's - Grand Unified Theories - are the classical Lie Groups $SU(5)$ (GEORGI; GLASHOW, 1974) and the $SO(10)$ (HEINEMEYER; MONDRAGON; ZOUPANOS, 2012), (FRITZSCH; MINKOWSKI, 1975) and the exceptional Lie Group E_6 (GURSEY P. RAMOND, 1976).

Even inside the Standard Model, mechanisms of spontaneous symmetry breaking plays important role in giving the masses of the particles (HIGGS, 1964). Because of the above reasons, one very useful feature of our library its set of tools developed to deal with symmetry breaking. It is able to generate symmetry breaking following two different paths: by eliminating one node from do Dynkin diagram for the algebra we are working with or by eliminating a node from the corresponding *extended* Dynkin diagram. The first breaks the original algebra into *non-semisimple subalgebras*, while the second breaks it into semisimple subalgebras.

We have already seen that every semisimple Lie Algebra decomposes as a direct sum of simple Lie algebras (BRIAN, 2015). Now, a non-semisimple Lie algebra is a semisimple Lie algebra times $U(1)$.

3.5.1 Decomposing Algebras into Non-Semisimple Subalgebras

In this case, we start from the Dynkin diagram for the algebra whose symmetry we want to break. In order to promote symmetry breaking, we need to chose one of the nodes in the diagram to be eliminated. The result is two or more disconnected Dynkin diagrams representing semisimple Lie algebras, while the eliminated dot represents $U(1)$.

An interesting example is breaking the symmetry of $SU(5)$ by eliminating the dot in the Dynkin diagram corresponding to α_2 (see Table 1 for the A_4 's diagram). The diagram depicting the consequences of the breaking can be seen in 8, from which we gather $SU(5)$ breaks into $SU(3) \otimes SU(2) \otimes U(1)$.

3.5.2 Decomposing Algebras into Semisimple Lie Algebras

In this case, before the breaking, we introduce another node in the Dynkin diagram, which depicts, then, the so-called *extended Lie algebra*. This is generating by adding the

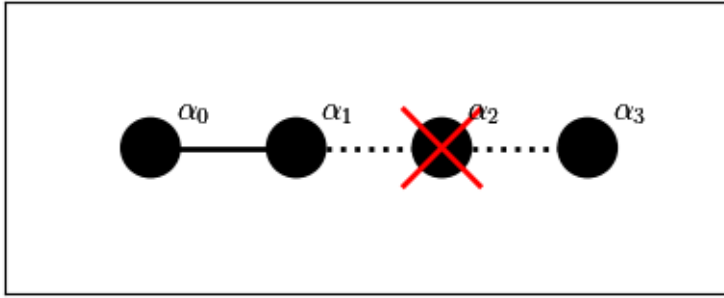


Figure 8 – Dynkin diagram for a non-semisimple Lie algebra emerging from breaking the symmetry of $SU(5)$.

negative of the *highest root*, which is the weight corresponding to the *adjoint* representation of the same algebra, to the set of the simple roots of the original algebra.

An adjoint representation is a map from the group on the *general linear group*, or the the group of invertible matrices, or yet a representation with dimension of the group, given by linear transformations on the Lie algebra. The reader may find the highest roots for some of the algebras we are dealing with in Table 2.

Algebra	Highest Root (γ)
A_4	(1, 0, 0 1)
B_4	(0, 1, 0, 0)
C_4	(2, 0, 0, 0)
D_4	(0, 1, 0, 0)
E_6	(0, 0, 0, 0, 0, 1)
E_7	(1, 0, 0, 0, 0, 0, 0)
E_8	(0, 0, 0, 0, 0, 0, 1, 0)
F_4	(1, 0, 0, 0)
G_2	(1, 0)

Table 2 – Highest root for some representative algebras.

All the extended Dynkin diagrams for the algebras of Table 1 can be found in Table 3.

Note that breaking the symmetry of the extended $SU(n)$ algebra gives back the same algebra. So, as an example, breaking the symmetry of the extended algebra for E_6 by means of eliminating α_3 , generates the diagram in Figure 9, where we see two disconnected Dynkin diagrams corresponding to $SU(6)$ and $SU(2)$, so E_6 breaks into $SU(6) \otimes SU(2)$.

3.5.3 Breaking Symmetry of Irreps

With the purpose of studying the consequence of symmetry breaking on the irreps of an algebra, I am going to use the $SU(4)$ example once more, for we are already familiar with it. Now, say we chose to eliminate the node corresponding to α_2 (see Table 1 for A_3), then the corresponding Dynkin diagram can be seen in Figure 10.

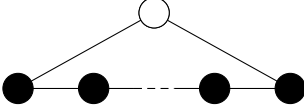
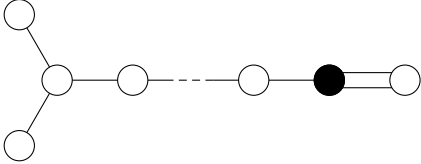
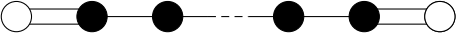
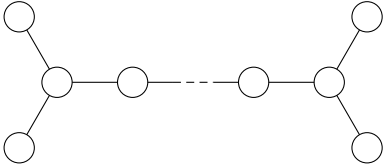


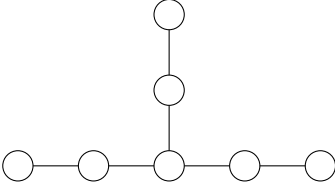
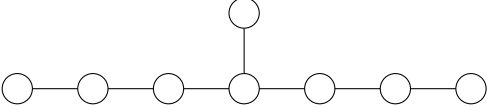
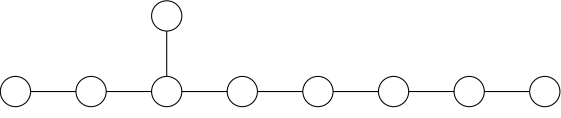
Original Algebra	Extended Dynkin Diagram
A_l	
B_l	
C_l	
D_l	
G_2	
F_4	
E_6	
E_7	
E_8	

Table 3 – Dynkin diagram for extended Lie Algebras.

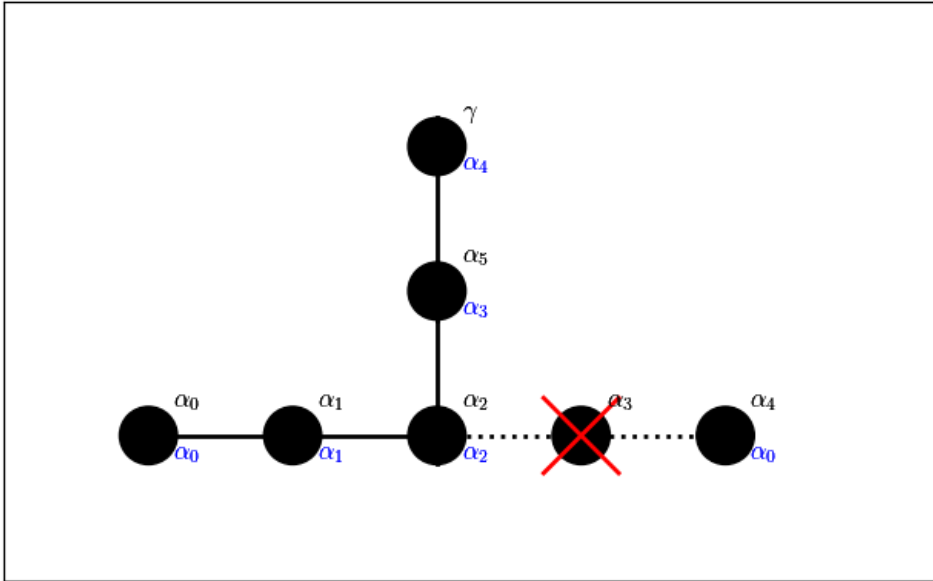


Figure 9 – $E_6 \rightarrow SU(6) \otimes SU(2)$.

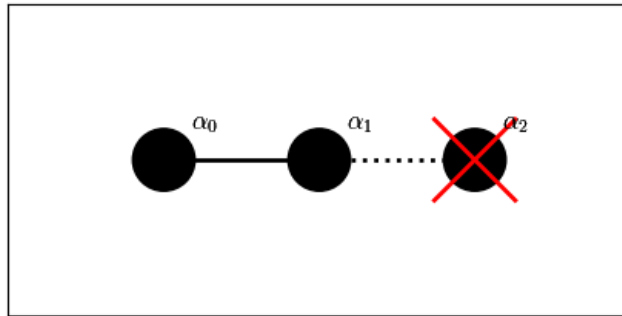


Figure 10 – $SU(4)$ group after symmetry breaking. Effectively, $SU(3) \otimes U(1)$.

As one might have expected, we end up with two connected nodes representing $SU(3)$ ¹⁰.

Considering the consequences suffered by the irreps of the original group, I take the one with maximal weight $(1, 0, 1)$ as example once more - a 15 in $SU(4)$. Obviously, we can anticipate that it breaks into a sum of irreps inside $SU(3)$, but we need to develop a way of determining the correct $SU(3)$ multiplets that will emerge from the process. We have already encountered examples of Hasse diagrams regarding irreps created by our library, and I have already covered the path taken in order to produce them (see Figure 5). Anyhow, there is still the possibility of producing the Hasse diagram depicting the weights from the irreps that get disconnected after breaking the underlying symmetry.

Since we chose to risk α_2 from the Dynkin diagram, we should also risk the corresponding element in each weight in the original irrep. Aiming to determine the resulting $SU(3)$ multiplets, every arrow departing from the roots in Figure 5 as consequence of the subtraction of α_2 must be eliminated. Then it will be created isolated subsets of roots,

¹⁰ The fact that $SU(3)$ is multiplied by $U(1)$ may be omitted, but the information is kept by the library so it might be used in order to break the symmetry in the irreps.

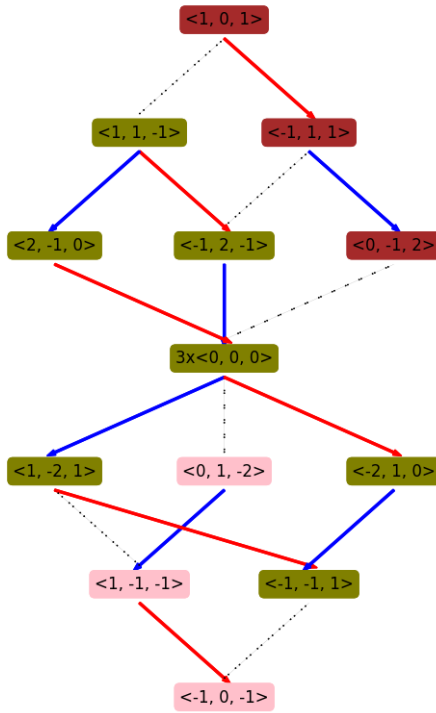


Figure 11 – Symmetry breaking in 15 of $SU(4)$.

each corresponding to an irrep in $SU(3)$, as one can see in Figure 11.

We should interpret Figure 10 in the following manner: in red we have the $SU(3)$ irrep whose maximal weight is $(1, 0)$, therefore a 3 ; in pink we have the $SU(3)$ irrep whose maximal weight is $(0, 1)$, therefore a $\bar{3}$; in green we have the $SU(3)$ irrep whose maximal weight is $(1, 1)$, therefore an 8 . But care must be taken, for the root $(0, 0)$ happens in the $SU(3)$'s 8 only twice, so the third $(0, 0)$ corresponds to a singlet, or an 1 of $SU(3)$. Consequently, the green section in the figure corresponds to $8 \oplus 1$. Finally, we conclude that the 15 in $SU(4)$ breaks into $3 \oplus 8 \oplus 1 \oplus \bar{3}$ in $SU(3)$.

Later we are going to encounter one more way the library permits the user to depict the same process of symmetry breaking in irreps. First, we need to talk about diagrams representing irreps in the quantum number space.

3.6 Multiplet Diagrams in Quantum Number Space

We have also seen already examples of diagrams regarding irreps in the quantum number space (Figs. 1, 2, 3). Now we are going to explore the path the library takes in order to construct them.

First, the user must provide the library with a list of vectors in the Euclidean space for some groups's root vectors. That process is expected to be automatized in future versions of the library. For now, we must keep in mind the Dynkin diagram for the group to which the irrep belongs.

As an example, I am going to cover the diagram for the 8 in $SU(3)$ (see Figure 3). From the Cartan matrix for this group (or from the corresponding Dynkin diagram) we are able to identify the ratio, and, consequently, the angle, between the chosen vectors. From Table 1 we see that, in this case, both have the same length and the angle between them is 120° . So we can choose the first as $\alpha = (1, 0)$, which makes the second $\beta = (-0.5, \sqrt{3}/2)$.

Given those vectors, then the library makes combinations between them: from the rules exposed in Section 2.3.2 we have that, besides the given vectors, $\alpha + \beta$ is a root, as well as $-\alpha$, $-\beta$ and $-(\alpha + \beta)$. The linear combination that generates the $(0, 0)$ must be the trivial one ($0\alpha + 0\beta$), or, otherwise, α and β would not be linear independent. Since the library is able to calculate the dimension of each root, that is used here to decide that the $(0, 0)$ happens twice, then completing the diagram.

It is easy to extrapolate the described method for plots in 3 dimensions. As an example, see Fig. 12, depicting the 15 in $SU(4)$.

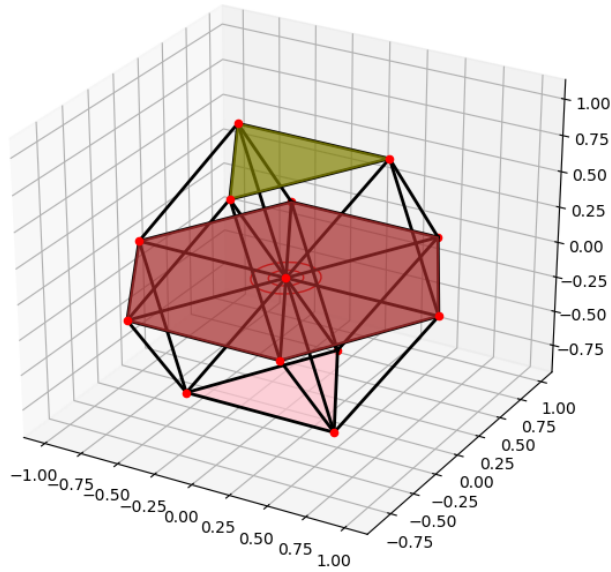


Figure 12 – Diagram for 15 of $SU(4)$ in quantum number space.

It is important to realize that the painted planes in the Figure 12 correspond to the 3, the $\bar{3}$ and the 8 in $SU(3)$, so it is one more way to understand how the smaller group fits inside the other and one more powerful pictorial way to understand the consequences of the symmetry breaking we discussed above.

4 Applying to Real Life Problems

A complete demonstration of the powers our library delivers could not be made without a proper demonstration of its functionalities in an practical problem.

I am going to show how we are able to retrieve all particles from the SM from the larger symmetry group $SO(10)$. When it comes to a Grand Unified Theory, the particles that do not couple in the subalgebras contained in the larger group may be part of an irrep in the latter, for they are subject to one unified force in intense energy regimes. So, for the considered GUT, all the 15 fermions of a generation are in the same 16-dimensional irrep of $SO(10)$.

In the SM, the generators of the groups $SU(3)$, $SU(2)$ and $U(1)$ are bosonic particles (the 8 gluons mediators of the strong interaction, the W^\pm and Z^0 , mediators of the weak interaction and the photon, carrier of the electromagnetic force), and, therefore, the way the fermions interact via those bosons depends on their representations. For example, the left-handed quarks up and down are triplets of the $SU(3)$ but doublets of the $SU(2)$, while right-handed electrons are singlets in both groups. Before going on, then, we should take a look at Table 4, containing the 15 fermions of the SM of the first generation.

Fermion	Representation	Hypercharge
Left-handed quarks	$3_{SU(3)} \otimes 2_{SU(2)}$	$1/3$
Left-handed leptons	$1_{SU(3)} \otimes 2_{SU(2)}$	-1
Right-handed up quark	$\bar{3}_{SU(3)} \otimes 1_{SU(2)}$	$4/3$
Right-handed down quark	$3_{SU(3)} \otimes 1_{SU(2)}$	$-2/3$
Right-handed electron	$1_{SU(3)} \otimes 1_{SU(2)}$	-2

Table 4 – The fermions of the Standard Model.

Now, the reader should also remember the $U(1)$ is always present when we eliminate a dot from a Dynkin diagram in order to promote the symmetry breaking. The physical meaning of such singlet varies from GUT to GUT and, sometimes, is interpreted as a right-handed neutrino ((CHU; SMIRNOV, 2016), (THOLE, 2019)).

In order to use the library for our purposes, first we create the $SO(10)$ group:

```
>>> from PyLA.LieAlgebras import *
>>> so10 = Group("SO10")
```

If we then eliminate the node corresponding to α_4 from the Dynkin diagram for $SO(10)$, we get $SU(5) \otimes U(1)$ (see Fig. 13):

```
>>> from PyLA.drawDiagrams import *
>>> so10.breakSym(4)
[< Group SU(5) or A4 >]
>>> fig = drawDynkinDiagramBrokenSym(so10)
>>> fig.savefig("so10cross.png")
```

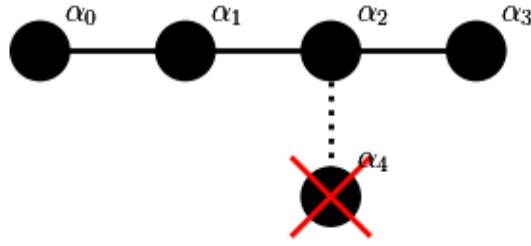


Figure 13 – $SO(10) \rightarrow SU(5)$.

We also want to investigate the symmetry breaking in the irrep with dimension 16. First, then, we use search for the possible labels corresponding to that dimension, then we generate the irrep and break the symmetry (see Fig. 14):

```
>>> labs = findLabels(16, so10)
[[0, 0, 0, 1, 0], [0, 0, 0, 0, 1]]
>>> irp = so10.irrep(labs[0])
>>> print irp
| 16 >
>>> irp.breakSym(4)
[ (| 5 bar >, 0.75), (| 10 >, -0.25), (| 1 (cc) >, -1.25) ]
>>> fig = drawIrrepBrokenDiagram(irp, frame =False)
>>> fig.savefig("16so10broken.png")
```

We get as result that the 16 of $SO(10)$ breaks into three irreps in $SU(5)$:

$$16 \rightarrow 10 \oplus \bar{5} \oplus 1.$$

Now we need to find the path by which we break $SU(5) \rightarrow SU(3) \otimes SU(2)$. Using the library, that can even be done by try and error, but it is not hard to grasp, from the observation of the Dynkin diagram for $SU(5)$ (see Table 1), that the elimination of the α_2 will break it into $SU(3) \otimes SU(2) \otimes U(1)$. Anyhow, this has already been done in 3.5.1 (see Figure 8).

After building the $SU(5)$ group with the library, then we should build the irreps with dimensions 5 and 10 and break their symmetry:

```
>>> su5 = Group("SU5")
>>> su5.breakSym(2) #Not necessary, but useful.
[< Group SU(3) or A2 >, < Group SU(2) or A1 >]
>>>
>>> labs1 = findLabels(5, su5)
>>> print labs1
[[1, 0, 0, 0], [0, 0, 0, 1]]
```

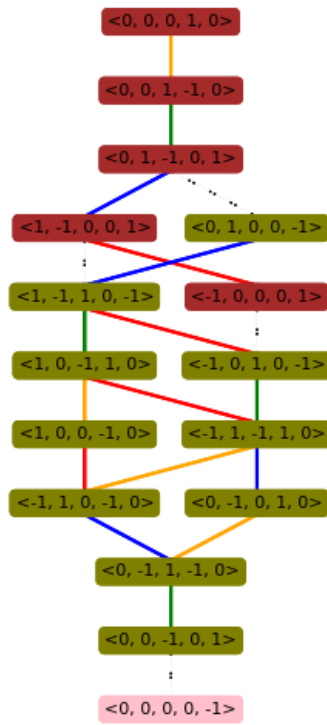


Figure 14 – $16 \rightarrow 10 \oplus \bar{5} \oplus 1$.

```

>>> print su5.irrep(labs1[1])
| 5 bar >
>>> irp1 = su5.irrep(labs1[1])
>>>
>>> labs2 = findLabels(10, su5)
>>> print labs2
[[0, 1, 0, 0], [0, 0, 1, 0]]
>>> print su5.irrep(labs2[0])
| 10 >
>>> irp2 = su5.irrep(labs2[0])
>>>
>>> irp1.breakSym(2)
[ (| 1 (cc) >, | 2 (cc) >, 0.6), (| 3 bar >, | 1 (cc) >, -0.4) ]
>>> irp2.breakSym(2)
[ (| 3 bar >, | 1 (cc) >, 0.8), (| 3 >, | 2 (cc) >, -0.2),
  (| 1 (cc) >, | 1 (cc) >, -1.2) ]
>>>
>>> fig1 = drawIrrepBrokenDiagram(irp1, frame=False)
>>> fig1.savefig("5barsu5broken.png")
>>> fig2 = drawIrrepBrokenDiagram(irp2, frame=False)
>>> fig2.savefig("10su5broken.png")

```

It is easy to interpret the results by means of the Figures 15 and 16. For example, a glance at the first permits us to see that, by eliminating the third weight from the highest root of the $\bar{5}$ in $SU(5)$ we end up with the product of irreps with maximum weights $(0, 0)$ and (1) plus the product of irreps with maximum weights $(0, 1)$ and (0) , which are easily recognizable as a singlet of $SU(3)$ times a doublet of $SU(2)$ plus a triplet of $SU(3)$ times a singlet of $SU(2)$:

$$\bar{5}_{SU(5)} \rightarrow 1_{SU(3)} \otimes 2_{SU(2)} \oplus \bar{3}_{SU(3)} \otimes 1_{SU(2)}. \quad (60)$$

Similarly, we also have:

$$10_{SU(5)} \rightarrow \bar{3}_{SU(3)} \otimes 1_{SU(2)} \oplus 3_{SU(3)} \otimes 2_{SU(2)} \oplus 1_{SU(3)} \otimes 1_{SU(2)}. \quad (61)$$

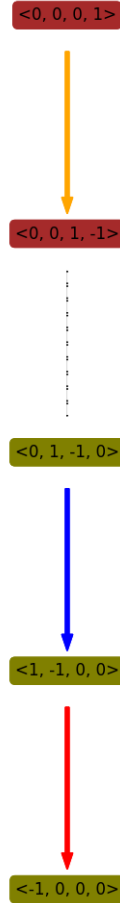


Figure 15 – $\bar{5}_{SU(5)} \rightarrow 1_{SU(3)} \otimes 2_{SU(2)} \oplus \bar{3}_{SU(3)} \otimes 1_{SU(2)}$.

The information about the hypercharge of each particle is also given by the library when the symmetry of the irrep is broken, but, because of the vectors I loaded to the group, its value is multiplied by 0.6. Having that in mind, from Table 4 and Equation (60) we see that the Left-handed anti-leptons and the right-handed down quark emerge as part of the same 5-dimensional irrep in the $SU(5)$. Similarly, by means of Equation (61) we gather

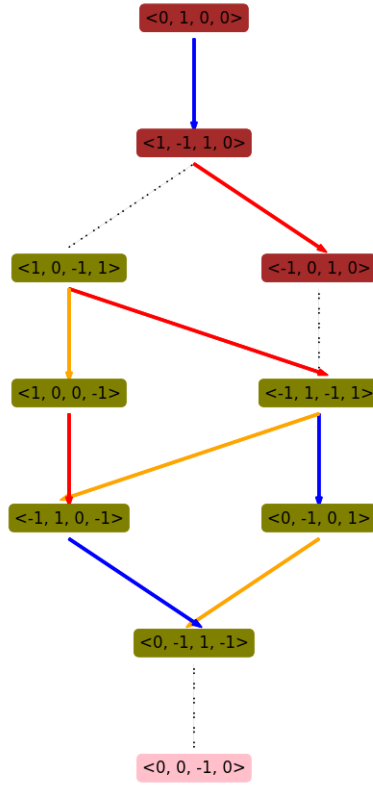


Figure 16 – $10_{SU(5)} \rightarrow \bar{3}_{SU(3)} \otimes 1_{SU(2)} \oplus 3_{SU(3)} \otimes 2_{SU(2)} \oplus 1_{SU(3)} \otimes 1_{SU(2)}$.

the 10 in $SU(5)$ breaks into the anti-right-handed electron plus the anti-right-handed up quark plus the left handed quarks.

5 Conclusions

I have presented an open source library application in Python2 language for calculating quantities associated to Lie Algebras and their representations for the use in Particle Physics. It is able to generate group objects from the related Cartan matrix, as well as Dynkin diagrams; diagrams depicting the multiplets both in 2D and 3D; irreducible representations, along with their classifications; Weyl reflections; tensor products of irreps; promoting symmetry breaking between other useful functions.

We have started this project with the idea of using the library to build effective Lagrangians for bound quarks states, but it has been shown to be specially useful for dealing with symmetry breaking, therefore, being a great tool for the study of GUTS. With that in mind, an example of the problem and how it would be treated by means of the library was illustrated.

For future versions, one expects to implement tools for working with Young tableaux and calculating Clebsch-Gordan coefficients, as well as translating the code to Python3 language.

Bibliography

- BERTRAM, K. A formula for the multiplicity of a weight. *Transactions of the American Mathematical Society*, American Physical Society, v. 93, p. 53–73, 1959. Cited in page 18.
- BOER, D. Lecture notes for the course of lie groups in physics. 2021/22. Cited in page 16.
- BORN, M.; HEISENBERG, W.; JORDAN, P. Zur quantenmechanik ii. *Zeitschrift f. Physik*, v. 35, p. 557–615, 1926. Cited in page 3.
- BORN, M.; JORDAN, P. Zur quantenmechanik. *Z. Phys.*, v. 34, p. 858–888, 1925. Cited in page 3.
- BRIAN, C. H. *Lie Groups, Lie Algebras, and Representations. An Elementary Introduction*. Switzerland: Springer International Publishing, 2015. Cited 2 times in pages 9 e 23.
- CARTER, R. The cartan matrix and the dynkin diagram. In: _____. *Lie Algebras of Finite and Affine Type*. [S.l.]: Cambridge University Press, 2005. (Cambridge Studies in Advanced Mathematics), p. 69–87. Cited 3 times in pages 10, 12 e 14.
- CHU, X.; SMIRNOV, A. Neutrino mixing and masses in so(10) guts with hidden sector and flavor symmetries. *J. High Energ. Phys.*, American Physical Society, Oct 2016. Cited in page 29.
- DYNKIN, E. B. Classification of the simple lie groups. *MSb*, v. 18, 3 (60), p. 347–353, 1946. Cited 3 times in pages 8, 10 e 14.
- DYNKIN, E. B. The structure of semi-simple lie algebras. *Uspekhi Matematicheskikh Nauk*, v. 2, 4, (20), p. 59–127, 1947. Cited 3 times in pages 7, 8 e 10.
- FEGER, R.; KEPHART, T. W. Lieart – a mathematica application for lie algebras and representation theory. *arXiv:1206.6379 [math-ph]*, 2012. Cited 2 times in pages 1 e 43.
- FEGER, R.; KEPHART, T. W.; SASKOWSKI, R. J. Lieart 2.0 – a mathematica application for lie algebras and representation theory. *arXiv:1912.10969v2 [hep-th]*, 2019. Cited 3 times in pages 1, 43 e 44.
- FRITZSCH, H.; MINKOWSKI, P. Unified interactions of leptons and hadrons. *Annals of Physics*, v. 93, n. 1, p. 193–266, 1975. ISSN 0003-4916. Disponível em: <https://www.sciencedirect.com/science/article/pii/0003491675902110>. Cited in page 23.
- GEORGI, H. *Lie Algebras in Particle Physics. from Isospin to Unified Theories*. [S.l.]: CRC Press, 1999. Cited 2 times in pages 9 e 10.
- GEORGI, H.; GLASHOW, S. L. Unity of all elementary-particle forces. *Phys. Rev. Lett.*, American Physical Society, v. 32, p. 438–441, Feb 1974. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevLett.32.438>. Cited in page 23.
- GERLACH, W.; STERN, O. Der experimentelle nachweis der richtungsquantelung im magnetfeld. *Zeitschrift für Physik*, v. 9(1), p. 349–352., 1922. Cited in page 3.

- GREINER, W.; MÜLLER, B. *Quantum Mechanics (Symmetries)*. [S.l.]: Springer, New York, 1994. Cited 4 times in pages 9, 10, 12 e 16.
- GURSEY P. RAMOND, P. S. H. A universal gauge theory model based on e_6 . *Phys.Lett.*, B60, p. 177, 01 1976. Cited in page 23.
- HEINEMEYER, S.; MONDRAGON, M.; ZOUPANOS, G. Finite unified theories and the higgs boson. 01 2012. Cited in page 23.
- HIGGS, P. W. Broken symmetries and the masses of gauge bosons. *Phys. Rev. Lett.*, American Physical Society, v. 13, p. 508–509, Oct 1964. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevLett.13.508>. Cited in page 23.
- JACOBSON, N. *Lie Algebras*. [S.l.]: Dover, New York, 1979. Cited 2 times in pages 9 e 19.
- LEEUWEN, M. van; COHEN, A.; LISSER, B. Lie, a package for lie group computations. *Computer Algebra Nederland, Amsterdam*, 1992. Disponível em: <http://young.sp2mi.univ-poitiers.fr/~marc/LiE/>. Cited 2 times in pages 1 e 43.
- LEMIRE, F.; PATERA, J. Congruence number, a generalization of $su(3)$ triality. *J.Math.Phys*, v. 21 (8), 1980. Cited in page 20.
- MCKAY, J. P. W. G. *Tables of dimensions, indices, and branching rules for representations of simple Lie algebras, Lecture Notes in Pure and Applied Mathematics*. Dekker and New York, NY: Springer International Publishing, 1981. Cited in page 20.
- NUTMA, T. Schur. 2009. Disponível em: <http://code.google.com/p/simplie/>. Cited 2 times in pages 1 e 43.
- PAULI, W. Zur quantenmechanik des magnetischen elektrons. *Zeitschrift f. Physik*, v. 43, p. 601–623, 1927. Cited in page 4.
- PESKIN, M. E.; SCHROEDER, D. V. *An Introduction to Quantum Field Theory*. [S.l.]: Addison-Wesley, 1995. Cited in page 22.
- SLANSKY, R. Group theory for unified model building. *Physics Reports*, v. 79, n. 1, p. 1–128, 1981. Cited 3 times in pages 1, 2 e 16.
- THOLE, J. *Family Symmetry Grand Unified Theories*. [S.l.]: University of Groningen, 2019. Cited 7 times in pages 1, 8, 9, 12, 14, 16 e 29.
- THOMSON, M. *Modern Particle Physics*. Cambridge: Cambridge University Press, 2013. Cited in page 23.
- WYBOURNE, B. G. Schur. 2002. Disponível em: <http://wwwold.fizyka.umk.pl/~bgw/schur.pdf>. Cited in page 1.
- ZEE, A. *Unity of Forces in the Universe*. [S.l.]: Singapore, World Scientific, 1982. Cited in page 22.

Appendices

1 Library's Commands List

1.1 Module LieAlgebras

The module also contains three objects:

- `Group` → Group object type. Contains the structure of a Group.
- `Weight` → Weight Object Type. Contains a Weight associated with a given group.
- `Irrep` → Irreducible representation object type. Contains the structure of an Irrep associated with a given group.

Follow the methods particular to the module:

- `hello` → Prints "Hello" in the screen.
- `cartanMatrix` → Returns the Cartan matrix for a given group.
- `combinator` → Returns all possible labels with a given sum for a given rank.
- `findLabels` → Returns all labels corresponding to irreps of a given dimension for a given group.
- `findLabelsBetween` → Returns all labels corresponding to irreps of dimension inside a given range for a given group.
- `groupStringReader` → Identifies the group in a string and prints on screen its cartan matrix.

1.1.1 The Group type

- `Group.roots` (method) → returns the list of roots as Weight types.
- `Group.positiveRoots` (method) → returns the list of positive roots as Weight types.
- `Group.weylVector` (method) → returns the weyl vector of the group as a Weight type.
- `Group.rank` (attribute) → the group's rank.
- `Group.npRoots` (attribute) → number of positive roots in the group.
- `Group.scalarP` (attribute) → Matrix with the resulting scalar product of the root vectors.
- `Group.show` (method) → prints on screen all computed information about the group.

- `Group.irrep` (method) → creates an irreducible representation of the group. Slower algorithm.
- `Group.irrep2` (method) → creates an irreducible representation of the group. Faster algorithm (uses weyl reflection.)
- `Group.labelsDimension` (method) → given a list of Dynking labels, evaluates the dimension of the corresponding irreducible representation.
- `Group.labelsDimensionDbl` (method) → same as `Group.labelsDimension`, but the result is a double, such that it can evaluate the dimension even if it is bigger than the maximum possible integer the compiler can deal with.
- `Group.breakSym` (method) → Breaks the symmetry by removing a node from the Dynkin diagram. Its input is the node's number to be removed (between zero and the group's rank minus one) and returns a list with the groups the symmetry breaks into (but ignores the $U(1)$ which is always a factor in such symmetry breakings).
- `Group.breakSymExt` (method) → Breaks the symmetry by removing a node from the Dynkin diagram of the extended algebra. Its input is the node's number to be removed (between zero and the group's rank minus one) and returns a list with the groups the symmetry breaks into.
- `Group.broken` (attribute) → If the group's symmetry has been previously broken, this attribute is an integer indicating the node removed from the corresponding Dynkin diagram. It is -1 otherwise.
- `Group.brokenExt` (attribute) → Is 0 or 1 indicating if, when the group symmetry was broken, it was by removing a node from its usual Dynking diagram or the extended one.
- `Group.brokenOrder` (method) → Return the way the roots are ordered when the symmetry was broken by removing a node from the extended algebra.
- `Group.gamma` (method) → If the symmetry was broken by removing a node from the extended algebra, this method returns a `Weight` object corresponding to the minimum weight that extends the algebra.
- `Group.loadVecs` (method) → This method receives a list where each element is a list with the vector components in Euclidian space for the group's root vectors. This information is used latter for computing the weights' quantum numbers for weights derived from the group.
- `Group.productDecomposition` (method) → Given two Irreps, evaluates the maximum weights resulting in the irrep product.

- `order` (attribute) → Group's order.
- `conjW` (method) → Returns the dynkin label associated to the irrep conjugate to the one represented by the labels given as argument in the method call.
- `indexAndCongru` (method) → Returns the index and congruency class for the dynkin label entered as argument in the method call.
- `latexNameFor` (method) → Returns in latex form the name of the irrep whose labels are given as argument in the method call.

1.1.2 The `Weight` type

- `Weight.ws` (attribute) → Tuple with the list of integers that define the weight.
- `Weight.deco` (attribute) → Tuple with the coefficients of the decomposition of the weight in terms of the group's simple roots.
- `Weight.rank` (attribute) → Rank of the group to which it belongs.
- `Weight.multiplicity` (attribute) → Multiplicity of the weight (is zero if the weight was not originated from an irreducible representation).
- `Weight.group` (attribute) → Group to which it belongs.
- `Weight.qnumbers` (method) → If the group associated to the weight has the root vectors in Euclidian space loaded to it, this method returns the quantum numbers (vector resulting from evaluating weight decomposition) associated to it.
- `Weight.show` (method) → Shows the weight information.

1.1.3 The `Irrep` type

- `Irrep.dimension` (attribute) → Dimension of the irreducible representation.
- `Irrep.nWs` (attribute) → Number of different weights in the irrep. It may be less than the dimension, since some weights might have multiplicity bigger than one.
- `Irrep.group` (attribute) → Group to which the irrep belongs.
- `Irrep.maxw` (attribute) → Tuple with the integers composing the representation maximum weight.
- `Irrep.broken` (attribute) → Same as `Group.broken`, but referring to the irrep.
- `Irrep.brokenExt` (attribute) → Same as `Irrep.broken`, but referring to the extended algebra.

- `Irrep.weights` (method) → Returns a list of the irrep weights.
- `Irrep.show` (method) → Shows details about the irrep.
- `Irrep.links` (method) → returns the information on how the irrep was actually built by the algorithm. The returned list has tuples with three elements, the first is the weight from where it comes, the second to where it goes and the third through which simple root it went. This and the method below are important to draw diagrams, as will be clear latter.
- `Irrep.levels` (method) → returns a list indicating the level (how many steps from the maximum weight) each weight in the irrep appeared.
- `Irrep.slice` (method) → Receives an integer as input and returns a list with lists of weights, indicating groups of weights that get disconnected if a node is removed from the corresponding group's Dynking diagram (if symmetry is broken in such way).
- `Irrep.partition` (method) → Returns the partition of weights if symmetry was previously broken.
- `Irrep.breakSym` (method) → Receives an integer as input and returns a list with tuples of irreps of the grups into which the symmetry was broken by removing the input node.
- `Irrep.breakSymExt` (method) → Receives an integer as input and returns a list with tuples of irreps of the grups into which the symmetry was broken by removing the input node from the group's extended algebra.
- `Irrep.maxWbar` (method) → Returns a tuple with the integers corresponding to the maximum weight of the irrep's complex conjugate.
- `Irrep.OTimes` (method) → Multiplies the irrep to another irrep. This method is usually slower than using the `mul` (simple multiplication).
- `Irrep.productDecomposition` (method) → Returns the maximum weights in the product decomposition of multiplying the irrep by the irrep corresponding to the maximum weights given.
- `Irrep.pow` (method) → Takes the irrep to a given power (multiplies it by itself many times).
- `Irrep.isCC` → Checks if the irrep is its own complex conjugate.
- `Irrep.latexName` → Writes in latex form the dimension name of the irrep.

1.2 Drawing Diagrams

The package also has a module to draw some diagrams. This module depends on the matplotlib library for python.

The functions contained in this module are:

- `limitingWeights` → This function receives an Irrep object as input and returns a list of its weights that are in the “border” of the irrep (weights that if added to some of the group’s positive roots will result in weights outside the irrep).
- `drawDynkinDiagram` → Draws the dynking diagram for a Group object.
- `drawDynkinDiagramBrokenSym` → Draws the dynking diagram for a Group object with a node crossed (symmetry broken).
- `drawDynkinDiagramBrokenSymExt` → Draws the dynking diagram for a Group object extended algebra with a node crossed (symmetry broken).
- `drawIrrepDiagram` → Draws the diagram depicting the construction of an irrep.
- `drawIrrepBrokenDiagram` → Draws the diagram depicting the weights from a irrep that get disconnected when breaking the underlying symmetry.
- `drawMultiplet2D` → Draws a two dimensional slice of a multiplet (Irrep’s quantum numbers).
- `drawMultiplet3D` → Draws a three dimensional slice of a multiplet (Irrep’s quantum numbers).

2 Comparisson With Popular Similar Libraries

Although this library is the first to be build upon Python language, it is not the only existing library aiming to deal with Lie Algebras. In the literature, one may find similar applications like, for example, (LEEUVEN; COHEN; LISSER, 1992), (NUTMA, 2009), (FEGER; KEPHART, 2012) and (FEGER; KEPHART; SASKOWSKI, 2019) (the latter being a new version of the former).

Every one of the cited libraries, among others, has good reasons to be popular, and one could find that they have, in effect, been the tool for the writing of many published papers. But ours, being the first written in Python, is a good innovation, for the language grows in popularity and is really easy to use. Furthermore, although LIE is also open source, the others are not.

While SimpLie is free to use, although not open source, LieART, for being an aplication for Mathematica, is consequently not free to use. It should also be said that the first is

written in C language, and the second, in Java (therefore being also easy to use with a nice interface). For that reason, a reserve for the future the comparison of the performance of this library against LIE's, since the calculations in ours are also coded in C. This time, though, I have made the comparison with LieArt.

Bellow I compare the time taken to perform the same operations as in chapter 5 (Benchmarks) of (FEGER; KEPHART; SASKOWSKI, 2019). The results are in tables 5 - 7. In the last column of these tables I show the ratio between the reported time in the article and the time it took in my computer (Intel Core i5, with 8G RAM). The ratio fluctuates, but ours is at least more than 100 times faster for high dimensional irreps! Since I do not know exactly how they perform the operation in their software, I considered that maybe the construction time for the irrep is also included.

For generating the results, I first found all the possible Dynkin labels corresponding to a given dimension for an irrep inside a given algebra, then used the `irrep.pow` method and found the time taken to calculate the tensor product of the irrep with itself a given number of times. It is important to say that the `irrep.pow` method returns only the maximum weights in the product decomposition (it does not construct the irreps resulting). I report here also the time our algorithm took to construct each representation. I considered that maybe the construction time for the irrep is also included, so I also summed to the product time in our is the result in parenthesis in the tables.

2.1 Calculations for the group E_6

2.1.1 Irreps with dimension 27

In Table 5 we see the results for the irreps with dimension 27 of E_6 . The time taken for constructing irrep was 0.005627 s. Follow also the results of the calculations.

Power	Our Time (with Irrep constr.) [s]	LieArt Time [s]	LieArt/Our (with Irrep constr.)
2	0.017835 (0.023462)	0.136494	7.653196 (5.817648)
3	0.018531 (0.024158)	0.143171	7.725992 (5.926384)
4	0.061531 (0.067158)	0.436060	7.086854 (6.493048)
5	0.115320 (0.120947)	0.715819	6.207242 (5.918446)
6	0.198395 (0.204022)	1.533490	7.729479 (7.516291)
7	0.507153 (0.512780)	12.996200	25.625796 (25.344583)
8	1.507932 (1.513559)	93.807900	62.209638 (61.978353)

Table 5 – Powers of 27 in E6

$$27^2 = \overline{27} \oplus \overline{351} \oplus 351'$$

$$27^3 = 1 \oplus 2(78) \oplus 3(650) \oplus 2925 \oplus 3003 \oplus 2(5824)$$

$$27^4 = 6(27) \oplus 3(\overline{351}') \oplus 6(351) \oplus 8(1728) \oplus 6(\overline{7371}) \oplus 6(7722) \oplus 17550 \oplus 19305' \oplus 2(\overline{34398}) \oplus 3(51975) \oplus 3(54054)$$

$$27^5 = 15(\overline{27}) \oplus 26(\overline{351}) \oplus 20(351') \oplus 24(\overline{1728}) \oplus 30(7371) \oplus 15(\overline{7722}) \oplus 20(\overline{17550}) \oplus 20(19305) \oplus 34398 \oplus \overline{46332} \oplus 10(\overline{51975}) \oplus 10(61425) \oplus 100386 \oplus 20(\overline{112320}) \oplus 4(314496) \oplus 4(359424') \oplus 5(\overline{386100}) \oplus 5(\overline{412776}) \oplus 6(494208)$$

$$27^6 = 15(1) \oplus 65(78) \oplus 130(650) \oplus 45(2430) \oplus 110(2925) \oplus 50(3003) \oplus 15(\overline{3003}) \oplus 80(\overline{5824}) \oplus 136(5824) \oplus 144(34749) \oplus 43758 \oplus 90(70070) \oplus 45(\overline{78975}) \oplus 90(78975) \oplus 45(85293) \oplus 40(105600) \oplus 40(146432) \oplus 16(\overline{252252}) \oplus 80(252252) \oplus 15(371800) \oplus 442442 \oplus 30(\overline{600600}) \oplus 5(600600) \oplus 5(812175) \oplus 45(852930) \oplus 45(972972) \oplus 5(1337050) \oplus 5(1559376) \oplus 5(1896180) \oplus 9(2453814) \oplus 10(2977975) \oplus 9(3007368) \oplus 10(3309696) \oplus 16(4752384)$$

$$27^7 = 210(27) \oplus 225(\overline{351}') \oplus 385(351) \oplus 630(1728) \oplus 735(\overline{7371}) \oplus 595(7722) \oplus 525(17550) \oplus 105(19305') \oplus 300(\overline{19305}) \oplus 336(\overline{34398}) \oplus 315(46332) \oplus 735(51975) \oplus 441(54054) \oplus 105(\overline{61425}) \oplus 560(112320) \oplus 504(\overline{314496}) \oplus 504(359424) \oplus 210(386100) \oplus 71(393822) \oplus 21(412776) \oplus 21(\overline{459459}) \oplus 106(\overline{494208}) \oplus 210(579150) \oplus 105(638820) \oplus 6(741312) \oplus 70(853281) \oplus 420(967680) \oplus 210(\overline{1123200}) \oplus 140(\overline{1253070}) \oplus 210(1640925) \oplus 1706562 \oplus 21(1837836) \oplus 210(2088450) \oplus 90(4200768) \oplus 14(4582656) \oplus 15(5553900) \oplus 105(\overline{5776056}) \oplus 84(6110208) \oplus 14(\overline{6243237}) \oplus 105(\overline{6747300}) \oplus 15(7528950) \oplus 126(7601958) \oplus 6(8401536) \oplus 14(10378368) \oplus 14(\overline{14805504}) \oplus 14(16540524) \oplus 15(17453475) \oplus 21(17918901) \oplus 21(\overline{19297278}) \oplus 20(19768320) \oplus 35(30115800) \oplus 35(34906950')$$

$$27^8 = 820(\overline{27}) \oplus 1435(351') \oplus 1960(\overline{351}) \oplus 2800(\overline{1728}) \oplus 4165(7371) \oplus 2520(\overline{7722}) \oplus 3780(\overline{17550}) \oplus 105(\overline{19305}') \oplus 3220(19305) \oplus 1316(34398) \oplus 1736(\overline{46332}) \oplus 3675(\overline{51975}) \oplus 1092(\overline{54054}) \oplus 1960(61425) \oplus 196(100386) \oplus 5040(\overline{112320}) \oplus 4144(314496) \oplus 2400(\overline{359424}) \oplus 1120(359424') \oplus 2660(\overline{386100}) \oplus 1260(\overline{393822}) \oplus 1652(\overline{412776}) \oplus 1260(459459) \oplus 2856(494208) \oplus 421(\overline{579150}) \oplus 420(\overline{638820}) \oplus 112(\overline{741312}) \oplus 2240(\overline{967680}) \oplus 2240(1123200) \oplus 420(1253070) \oplus 1345(\overline{1640925}) \oplus 238(\overline{2088450}) \oplus 1344(2559843) \oplus 420(3281850) \oplus 210(3675672) \oplus 112(4088448) \oplus 2688(\overline{4200768}) \oplus 840(\overline{4582656}) \oplus 20(\overline{5501925}) \oplus 420(\overline{5553900}) \oplus 1260(5776056) \oplus 5895396 \oplus 455(6243237) \oplus 21(6675669) \oplus 140(6747300) \oplus 140(\overline{7528950}) \oplus 637(\overline{7601958}) \oplus 28(7757100') \oplus 1260(7757100) \oplus 630(\overline{9189180}) \oplus 560(\overline{10378368}) \oplus 448(12648636) \oplus 560(\overline{13478400}) \oplus 672(14017536) \oplus 140(\overline{17918901}) \oplus 28(19297278) \oplus 14(22007700) \oplus 140(\overline{23629320}) \oplus 28(\overline{26702676}) \oplus 272(\overline{30115800}) \oplus 140(30718116) \oplus 7(32424678) \oplus 35(36100350) \oplus 301(37459422) \oplus 14(\overline{41442192}) \oplus 252(\overline{46542600}) \oplus 280(48243195) \oplus 35(49017150) \oplus 64(54991872) \oplus 448(\overline{66830400}) \oplus 20(74826180) \oplus 64(\overline{75119616}) \oplus 21(77026950) \oplus 28(\overline{89791416}') \oplus 42(\overline{93459366}) \oplus 35(103169430) \oplus 56(\overline{123803316}) \oplus 56(136547775) \oplus 70(138881925) \oplus 70(\overline{184864680}) \oplus 64(192067200) \oplus 90(219490128)$$

2.1.2 Irreps with dimension 78

In Table 6 we see the results for the irreps with dimension 78 of E_6 . The time taken for constructing irrep was 0.005214 s. Follow also the results of the calculations.

$$78^2 = 1 \oplus 78 \oplus 650 \oplus 2430 \oplus 2925$$

$$78^3 = 1 \oplus 5(78) \oplus 4(650) \oplus 3(2430) \oplus 4(2925) \oplus 2(\overline{5824}) \oplus 2(5824) \oplus 3(34749) \oplus 43758 \oplus 70070 \oplus 2(105600)$$

Power	Our Time (with Irrep constr.) [s]	LieArt Time [s]	LieArt/Our (with Irrep constr.)
2	0.017835 (0.023462)	0.136494	7.653196 (5.817648)
3	0.018531 (0.024158)	0.143171	7.725992 (5.926384)
4	0.061531 (0.067158)	0.436060	7.086854 (6.493048)
5	0.115320 (0.120947)	0.715819	6.207242 (5.918446)
6	0.198395 (0.204022)	1.533490	7.729479 (7.516291)
7	0.507153 (0.512780)	12.996200	25.625796 (25.344583)
8	1.507932 (1.513559)	93.807900	62.209638 (61.978353)

Table 6 – Powers of 27 in E6

$$78^4 = 5(1) \oplus 17(78) \oplus 24(650) \oplus 18(2430) \oplus 26(2925) \oplus 2(3003) \oplus 2(\overline{3003}) \oplus 16(\overline{5824}) \oplus 16(5824) \oplus 27(34749) \oplus 6(43758) \oplus 15(70070) \oplus 6(\overline{78975}) \oplus 6(78975) \oplus 3(85293) \oplus 16(105600) \oplus 8(\overline{252252}) \oplus 8(252252) \oplus 537966 \oplus \overline{600600} \oplus 600600 \oplus 6(812175) \oplus 6(852930) \oplus 2(1337050) \oplus 3(1911195) \oplus 3(2453814)$$

$$78^5 = 17(1) \oplus 90(78) \oplus 150(650) \oplus 110(2430) \oplus 175(2925) \oplus 24(3003) \oplus 24(\overline{3003}) \oplus 140(\overline{5824}) \oplus 140(5824) \oplus 255(34749) \oplus 50(43758) \oplus 170(70070) \oplus 90(\overline{78975}) \oplus 90(78975) \oplus 51(85293) \oplus 160(105600) \oplus 16(\overline{146432}) \oplus 16(146432) \oplus 120(\overline{252252}) \oplus 120(252252) \oplus 10(537966) \oplus 40(\overline{600600}) \oplus 40(600600) \oplus 95(812175) \oplus 120(852930) \oplus 24(972972) \oplus 24(\overline{972972}) \oplus 30(1337050) \oplus \overline{1559376} \oplus 1559376 \oplus 40(1911195) \oplus 75(2453814) \oplus 30(\overline{2977975}) \oplus 30(2977975) \oplus 15(3490695) \oplus 20(\overline{4752384}) \oplus 20(4752384) \oplus 4969107 \oplus 20(5054400) \oplus 20(\overline{5054400}) \oplus 10(11655930) \oplus 11(12514788) \oplus 4(\overline{19160064}) \oplus 4(19160064) \oplus 4(22843392) \oplus 20(23795200) \oplus 5(29422393) \oplus 5(34906950) \oplus 6(42134742)$$

$$78^6 = 90(1) \oplus 542(78) \oplus 1100(650) \oplus 840(2430) \oplus 1390(2925) \oplus 270(3003) \oplus 270(\overline{3003}) \oplus 1264(\overline{5824}) \oplus 1264(5824) \oplus 2496(34749) \oplus 465(43758) \oplus 1905(70070) \oplus 1170(\overline{78975}) \oplus 1170(78975) \oplus 720(85293) \oplus 1680(105600) \oplus 320(146432) \oplus 320(\overline{146432}) \oplus 1615(\overline{252252}) \oplus 1615(252252) \oplus 40(371800) \oplus 40(\overline{371800}) \oplus 115(537966) \oplus 745(\overline{600600}) \oplus 745(600600) \oplus 1370(812175) \oplus 1890(852930) \oplus 585(\overline{972972}) \oplus 585(972972) \oplus 495(1337050) \oplus 65(\overline{1559376}) \oplus 65(1559376) \oplus 585(1911195) \oplus 1350(2453814) \oplus 760(\overline{2977975}) \oplus 760(2977975) \oplus 45(\overline{3007368}) \oplus 45(3007368) \oplus 66(3162159) \oplus 66(\overline{3162159}) \oplus 90(3309696) \oplus 90(\overline{3309696}) \oplus 450(3490695) \oplus 15(4548180) \oplus 560(\overline{4752384}) \oplus 560(4752384) \oplus 15(4969107) \oplus 480(\overline{5054400}) \oplus 480(5054400) \oplus 80(\overline{7779200}) \oplus 80(7779200) \oplus 245(11655930) \oplus 515(12514788) \oplus 240(\overline{19160064}) \oplus 240(19160064) \oplus 80(22843392) \oplus 640(23795200) \oplus 115(29422393) \oplus 144(\overline{32752512}) \oplus 144(32752512) \oplus 210(34906950) \oplus 36685506 \oplus 225(42134742) \oplus 5(44767800) \oplus 5(\overline{44767800}) \oplus 41(47783736) \oplus 41(\overline{47783736}) \oplus 90(47849373) \oplus 90(\overline{47849373}) \oplus 90(53557504) \oplus 90(\overline{53557504}) \oplus 15(59073300) \oplus 15(\overline{59073300}) \oplus 46(64205141) \oplus 40(\overline{64414350}) \oplus 40(64414350) \oplus 45(66023100) \oplus 80(\overline{115287744}) \oplus 80(115287744) \oplus 15(119189070) \oplus 5(200449886) \oplus 5(203365305) \oplus 5(221077350) \oplus 30(226459233) \oplus 9(\overline{252808452}) \oplus 9(252808452) \oplus 10(303388800) \oplus 10(\overline{303388800}) \oplus 50(348985350) \oplus 45(350895402) \oplus 9(366235506) \oplus 10(476952476) \oplus 16(734557824)$$

$$78^7 = 542(1) \oplus 3962(78) \oplus 9156(650) \oplus 7413(2430) \oplus 12481(2925) \oplus 3024(3003) \oplus$$

$3024(\overline{3003}) \oplus 12474(\overline{5824}) \oplus 12474(5824) \oplus 26481(34749) \oplus 5055(43758) \oplus 22204(70070) \oplus$
 $14910(\overline{78975}) \oplus 14910(78975) \oplus 9786(85293) \oplus 19376(105600) \oplus 5236(146432) \oplus 5236(\overline{146432}) \oplus$
 $21420(\overline{252252}) \oplus 21420(252252) \oplus 1155(371800) \oplus 1155(\overline{371800}) \oplus 1505(537966) \oplus$
 $12145(\overline{600600}) \oplus 12145(600600) \oplus 19621(812175) \oplus 28161(852930) \oplus 10899(\overline{972972}) \oplus$
 $10899(972972) \oplus 8085(1337050) \oplus 1701(\overline{1559376}) \oplus 1701(1559376) \oplus$
 $175(1896180) \oplus 175(\overline{1896180}) \oplus 8910(1911195) \oplus 22365(2453814) \oplus 14770(\overline{2977975}) \oplus$
 $14770(2977975) \oplus 1575(\overline{3007368}) \oplus 1575(3007368) \oplus 2016(3162159) \oplus 2016(\overline{3162159}) \oplus$
 $2870(3309696) \oplus 2870(\overline{3309696}) \oplus 9729(3490695) \oplus 686(4548180) \oplus 11760(\overline{4752384}) \oplus$
 $11760(4752384) \oplus 231(4969107) \oplus 9450(\overline{5054400}) \oplus 9450(5054400) \oplus 2990(\overline{7779200}) \oplus$
 $2990(7779200) \oplus 5075(11655930) \oplus 13216(12514788) \oplus 420(14152320) \oplus 420(\overline{14152320}) \oplus$
 $6804(\overline{19160064}) \oplus 6804(19160064) \oplus 1624(22843392) \oplus 15114(23795200) \oplus 2730(29422393) \oplus$
 $5544(\overline{32752512}) \oplus 5544(32752512) \oplus 6111(34906950) \oplus 21(36685506) \oplus 5901(42134742) \oplus$
 $196(42398720) \oplus 196(\overline{42398720}) \oplus 455(44767800) \oplus 455(\overline{44767800}) \oplus 490(45741696) \oplus$
 $490(\overline{45741696}) \oplus 2156(47783736) \oplus 2156(\overline{47783736}) \oplus 3870(\overline{47849373}) \oplus 3870(47849373) \oplus$
 $3430(53557504) \oplus 3430(\overline{53557504}) \oplus 300(\overline{54991872'}) \oplus 300(54991872') \oplus 1260(\overline{59073300}) \oplus$
 $1260(59073300) \oplus 2541(64205141) \oplus 1400(\overline{64414350}) \oplus 1400(64414350) \oplus 2079(66023100) \oplus$
 $560(85974525) \oplus 560(\overline{85974525}) \oplus 315(89791416) \oplus 315(89791416) \oplus 3360(\overline{115287744}) \oplus$
 $3360(115287744) \oplus 525(119189070) \oplus 106(152423700) \oplus 175(200449886) \oplus 140(203365305) \oplus$
 $70(\overline{212838912}) \oplus 70(212838912) \oplus 386(221077350) \oplus 225961450 \oplus 1575(226459233) \oplus$
 $560(\overline{236487680}) \oplus 560(236487680) \oplus 819(\overline{252808452}) \oplus 819(252808452) \oplus 840(303388800) \oplus$
 $840(\overline{303388800}) \oplus 3465(348985350) \oplus 2100(350895402) \oplus 294(366235506) \oplus 210(\overline{392837445}) \oplus$
 $210(392837445) \oplus 505(466237200) \oplus 505(\overline{466237200}) \oplus 525(476952476) \oplus 14(\overline{532097280}) \oplus$
 $14(532097280) \oplus 504(\overline{537567030}) \oplus 504(537567030) \oplus 70(598998400) \oplus 70(598998400') \oplus$
 $210(625532544') \oplus 210(625532544) \oplus 15(649806300) \oplus 15(\overline{649806300}) \oplus 126(688740975) \oplus$
 $966(734557824) \oplus 105(797489550) \oplus 140(929510400) \oplus 140(\overline{929510400}) \oplus 21(944929700) \oplus$
 $420(\overline{1051315200}) \oplus 420(1051315200) \oplus 216(\overline{1177830720}) \oplus 216(1177830720) \oplus 6(1445558400) \oplus$
 $216(1478062080) \oplus 210(\overline{1525620096}) \oplus 210(1525620096) \oplus 84(1544524800) \oplus 84(\overline{1544524800}) \oplus$
 $14(1643241600''') \oplus 14(1643241600'') \oplus 14(3116305920) \oplus 20(\overline{3203785728}) \oplus 20(3203785728) \oplus$
 $14(3256917300) \oplus 84(3548188800) \oplus 119(3863940795) \oplus 15(4035297123) \oplus 105(4129204716) \oplus$
 $21(4790483775) \oplus 35(4942962024) \oplus 35(\overline{4942962024}) \oplus 21(4973434830) \oplus 141(4991693850) \oplus$
 $35(8713554850)$

2.2 Calculations for the group E_8

2.2.1 Irreps with dimension 248

In Table 7 we see the results for the irreps with dimension 248 of E_8 . The time taken for constructing irrep was 0.381720 s. Follow also the results of the calculations.

$$248^2 = 1 \oplus 248 \oplus 3875 \oplus 27000 \oplus 30380$$

$$248^3 = 1 \oplus 5(248) \oplus 3(3875) \oplus 3(27000) \oplus 4(30380) \oplus 2(147250) \oplus 3(779247) \oplus 1763125 \oplus$$

Power	Our Time (with Irrep constr.) [s]	LieArt Time [s]	LieArt/Our (with Irrep constr.)
2	0.000031 (0.381751)	0.850115	27428.005730 (2.226883)
3	0.000177 (0.381897)	0.883473	4994.008541 (2.313380)
4	0.000405 (0.382125)	31.985500	78962.278159 (83.704255)
5	0.001380 (0.383100)	49.929500	36181.669241 (130.330190)
6	0.009504 (0.391224)	90.112000	9481.639708 (230.333572)
7	0.228582 (0.610302)	7636.290000	33407.237497 (12512.314162)

Table 7 – Powers of 248 in E8

$2450240 \oplus 2(4096000)$

$248^4 = 5(1) \oplus 16(248) \oplus 17(3875) \oplus 18(27000) \oplus 23(30380) \oplus 13(147250) \oplus 21(779247) \oplus 6(1763125) \oplus 12(2450240) \oplus 16(4096000) \oplus 3(4881384) \oplus 6(6696000) \oplus 8(26411008) \oplus 6(70680000) \oplus 6(76271625) \oplus 79143000 \oplus 146325270 \oplus 2(203205000) \oplus 3(281545875) \oplus 3(344452500)$

$248^5 = 16(1) \oplus 79(248) \oplus 90(3875) \oplus 100(27000) \oplus 136(30380) \oplus 100(147250) \oplus 170(779247) \oplus 50(1763125) \oplus 109(2450240) \oplus 140(4096000) \oplus 36(4881384) \oplus 70(6696000) \oplus 100(26411008) \oplus 75(70680000) \oplus 90(76271625) \oplus 10(79143000) \oplus 36(146325270) \oplus 30(203205000) \oplus 40(281545875) \oplus 24(301694976) \oplus 60(344452500) \oplus 15(820260000) \oplus 30(1094951000) \oplus 20(2172667860) \oplus 20(2275896000) \oplus 2642777280 \oplus 10(3929713760) \oplus 10(4825673125) \oplus 6899079264 \oplus 20(8634368000) \oplus 4(12692520960) \oplus 5(17535336000) \oplus 4(20288765952) \oplus 5(21039669000) \oplus 6(23592339045)$

$248^6 = 79(1) \oplus 421(248) \oplus 575(3875) \oplus 675(27000) \oplus 924(30380) \oplus 775(147250) \oplus 1386(779247) \oplus 415(1763125) \oplus 1011(2450240) \oplus 1240(4096000) \oplus 405(4881384) \oplus 765(6696000) \oplus 1144(26411008) \oplus 895(70680000) \oplus 1125(76271625) \oplus 115(79143000) \oplus 554(146325270) \oplus 410(203205000) \oplus 510(281545875) \oplus 456(301694976) \oplus 855(344452500) \oplus 315(820260000) \oplus 605(1094951000) \oplus 405(2172667860) \oplus 470(2275896000) \oplus 15(2642777280) \oplus 15(2903770000) \oplus 195(3929713760) \oplus 45(4076399250) \oplus 325(4825673125) \oplus 125(6899079264) \oplus 80(8634368000) \oplus 480(8634368000) \oplus 80(12692520960) \oplus 115(17535336000) \oplus 216(20288765952) \oplus 165(21039669000) \oplus 180(23592339045) \oplus 144(45329752170) \oplus 45(63513702720) \oplus 45(66393847000) \oplus 69176971200 \oplus 90(83080364250) \oplus 90(85424220000) \oplus 40(110977024000) \oplus 40(124436480000) \oplus 15(152883490500) \oplus 15(220778105625) \oplus 80(234550030000) \oplus 267413986840 \oplus 30(355647996000) \oplus 5(417933862500) \oplus 5(492957660000) \oplus 45(508731738750) \oplus 45(574197082368) \oplus 5(627099023250) \oplus 9(841900509450) \oplus 5(919045960000) \oplus 10(1041872676000) \oplus 9(1283242632840) \oplus 10(1349926375875) \oplus 16(1813461073920)$

$248^7 = 421(1) \oplus 2674(248) \oplus 4081(3875) \oplus 5061(27000) \oplus 7007(30380) \oplus 6580(147250) \oplus 12306(779247) \oplus 3850(1763125) \oplus 9779(2450240) \oplus 11830(4096000) \oplus 4452(4881384) \oplus 8226(6696000) \oplus 12830(26411008) \oplus 10465(70680000) \oplus 13566(76271625) \oplus 1330(79143000) \oplus 7651(146325270) \oplus 5370(203205000) \oplus 6405(281545875) \oplus 7014(301694976) \oplus 11445(344452500) \oplus 5250(820260000) \oplus 9744(1094951000) \oplus 6714(2172667860) \oplus 8204(2275896000) \oplus 231(2642777280) \oplus 455(2903770000) \oplus 3290(3929713760) \oplus 1281(4076399250) \oplus 6475(4825673125) \oplus 3171(6899079264) \oplus$

8840(8634368000) \oplus 2296(8634368000') \oplus 1414(12692520960) \oplus 2240(17535336000) \oplus
 5124(20288765952) \oplus 3696(21039669000) \oplus 3717(23592339045) \oplus 4410(45329752170) \oplus
 1449(63513702720) \oplus 1674(66393847000) \oplus 21(69176971200) \oplus 2765(83080364250) \oplus
 3150(85424220000) \oplus 1190(110977024000) \oplus 1820(124436480000) \oplus 420(152883490500) \oplus
 1155(220778105625) \oplus 300(223850628000) \oplus 2835(234550030000) \oplus 546(267413986840) \oplus
 1155(355647996000) \oplus 105(417532087000) \oplus 140(417933862500) \oplus 175(492957660000) \oplus
 1575(508731738750) \oplus 315(560213725500) \oplus 2170(574197082368) \oplus 280(627099023250) \oplus
 294(841900509450) \oplus 875(919045960000) \oplus 420(1041872676000) \oplus 560(1198018560000) \oplus
 735(1283242632840) \oplus 756(1349926375875) \oplus 1473701482500 \oplus 756(1813461073920) \oplus
 105(3067797300750) \oplus 504(3191795712000) \oplus 504(3233052753920) \oplus 105(3431612952000) \oplus
 3754721200320 \oplus 210(3950782290000) \oplus 70(4007202600000) \oplus 210(4189713446646) \oplus
 21(4490627295000) \oplus 70(4779643627500) \oplus 5006235840320 \oplus 210(6458110083072) \oplus
 21(7723951192125) \oplus 420(8145764352000) \oplus 140(8715491428800) \oplus 6(10701806469120) \oplus
 210(12737135385000) \oplus 210(13532264250750) \oplus 84(19994148864000) \oplus 84(26125438976000) \oplus
 105(26461348084080) \oplus 14(28123973939490) \oplus 105(29369472656250) \oplus 14(30014459904000) \oplus
 15(33372802062000) \oplus 6(33699815424000) \oplus 126(33943999320000) \oplus 14(46678711926784) \oplus
 21(53540697687750) \oplus 14(56860936405000) \oplus 21(57306919524192) \oplus 20(57591234560000) \oplus
 15(58549130859375) \oplus 35(85471274280000) \oplus 35(107701303073000)

