PH 354 Computational Physics Course Project Extremal Effective Field Theories

Aryaman Bhutani[1]

[1] Indian Institute of Science, Physics(UG), S.R. No.: 18721
*e-mail: aryamanb@iisc.ac.in

ABSTRACT

Effective Field Theory parametrise the long-distance effects of short-distance dynamics whose details may not be known. At short distances, if the physics is compatible with causality and unitarity, the low-energy parameters of the EFT obey certain inequalities. The paper explores the subset of EFT parameters (g_k) captured by $2 \to 2$ scattering. The parameters g_k along with a set of null constraints, create an optimisation problem. Solving the optimisation problems provides tight numerical bounds on the values of the g_k set of parameters. The numerical aspect of the problem reduces to solving a constrained optimisation problem, utilising linear or semidefinite programming. Using constrained optimisation, the optimal bound on parameters and the shape of the allowed space of parameters can be obtained. The new insight from the numerical analysis is the universal existence of two-sided bounds. The observations suggest the seemingly universal existence of two-sided bounds. Furthermore, it also allows the interpretation of the shape of the allowed space of parameters.

Key points: Effective Field Theories (EFT), Constrained optimisation, Linear Programming (LP), Semidefinite programming (SDP) **Advisor:** Prof. Aninda Sinha, Centre for High Energy Physics, Indian Institute of Science **Primary Reference:** Caron-Huot, S., Van Duong, V. Extremal effective field theories. J. High Energ. Phys. 2021, 280 (2021). https://doi.org/10.1007/JHEP05(2021)280
Link to GitHub Repository https://github.com/Ary276/Extremal_Effective_Field_Theories_PH354_Project

Part I

Preface

1 Theoretical Foundations

1.1 Effective Field Theories

Our understanding of physical phenomena spans a wide range of energies, length and time scales. The separation between the magnitudes of energies in several phenomena allows us to use strategies like perturbation theory in quantum mechanics. Similarly, in High Energy Physics, the separation of energy scales or any dimensional quantity is utilised to build a precise model approximating physical reality. An essential ingredient of EFT theories is symmetries. The symmetries help constrain the dynamics of the system. In fact, in the problem worked in this project, the critical ingredient required was null constraints which were generated from crossing symmetries. The main advantage EFT's offer is that they simplify complicated calculations by focusing on relevant degrees of freedom and using systematic models to take care of unknown physics.

2 Model: 2→2 Scattering Process

The work focuses on the simplest example: a single non-gravitating real scalar field. Typically the mass cut-off M is much larger than the mass of the particles. In particular, the 2 → 2 scattering of massless identical real scalars in a Poincare invariant theory has been considered. The lowest-order approximation in Perturbation theory here is the tree-level approximation. Using this, an effective Lagrangian for the process is obtained along with the amplitude. The effective Lagrangian in the tree-level approximation is given by

$$\begin{aligned}\mathcal{L}_{low} = &-\frac{1}{2}(\partial_\mu\phi)^2 - \frac{g}{3!}\phi^3 - \frac{\lambda}{4!}\phi^4 + \frac{g_2}{2}[(\partial_\mu\phi)^2]^2 \\ &+ \frac{g_3}{3}(\partial_\mu\partial_\nu\phi)^2(\partial_\sigma\phi)^2 + 4g_4[(\partial_{mu}\partial_{nu}\phi)^2]^2 + \dots\end{aligned}$$

In particular, the amplitude can be written as a series in terms of the Mandelstam variables (s,t,u) based on the above Lagrangian:

$$\begin{aligned}\mathcal{M}_{low}(s,t) =& g^2\left[\frac{1}{s}+\frac{1}{t}+\frac{1}{u}\right] - \lambda + g_2(s^2+t^2+u^2) \\ &+ g_3(stu) + g_4(s^2+t^2+u^2)^2 \\ &+ g_5(s^2+t^2+u^2)(stu) + \dots\end{aligned}$$

The coefficients obtained here, $g_2, g_3, g_4, \dots$, will be the main focus of the work and describe the EFT couplings. The task is to show that these parameters are bounded. In particular, the bounding of these parameters will show that *in any causal and unitary theory, dimensional analysis scaling is a theorem*. The particular observation here is the *universal existence of a two-sided bound.*

3 The Equations:

To obtain the equations for the parameters, a family of "Sum Rules" is utilised. These Sum Rules express the EFT coefficients as averages over high-energy probabilities. An intuitive understanding of the same will be presented later. In the low-energy regime, we use the EFT description, while in the high-energy regime, scattering theory and partial wave decomposition takes importance. In particular, we have the S-matrix, given by $S = 1 + i\mathcal{M}$, where $\mathcal{M}$ is the scattering amplitude. In the high-energy regime, we have imposed the conditions of Unitarity and Causality to constrain the unknown physics. This gives us $SS^\dagger = 1$. This puts bounds on the *spectral density* $\rho_J(s)$: $0 \le \rho_J(s) \le 2$, where ρ_J indicates complete absorption and $\rho_J = 2$ an elastic phase shift of π.

The two properties of the amplitude, which are obtained as a consequence of causality and unitarity, are given below

- For fixed $t < 0$ and $|s|$ is sufficiently small, $\mathcal{M}(s,t)$ is analytic in s away from the real axis.
- For fixed $t < 0$, $\lim_{|s|\to\infty}\left|\frac{\mathcal{M}(s,t)}{s^2}\right| = 0$ along any line of constant phase

These two properties are used in the following section.

To obtain the coefficients in the EFT as high-energy averages, we need to connect the high-energy regime to the low-energy regime. This is done analytically using dispersion relations. The cut-off between the low and high scales is in terms of Mass. Analytic results yield high-energy averages that can be represented as a summation in terms of the parameters g_k. One such expression is shown below:

$$B_2 : 2g_2 - g_3 t + 8g_4 t^2 + \dots = \left\langle \frac{(2m^2+t)\mathcal{P}_J(1+\frac{2t}{m^2})}{m^2(m^2+t)^3} \right\rangle$$

This is a sum rule from the $B_k(t)$ family of sum rules. By expanding the RHS of the above and matching it with the LHS term by term in terms of the Mandelstam variable t, we attain the expressions for the g_k parameters. The expressions for those studied in this project are given below.

$$g_2 = \left\langle \frac{1}{m^4} \right\rangle, \quad g_3 = \left\langle \frac{3 - \frac{4}{d-2}\mathcal{J}^2}{m^6} \right\rangle, \quad g_4 = \left\langle \frac{1}{2m^8} \right\rangle \qquad (1)$$

Further, dimensionless ratios of these constraints are defined, which are important in the optimisation process.

$$\tilde{g}3 = g_3\frac{M^2}{g_2} \qquad \tilde{g}4 = g_4\frac{M^4}{g_2} \qquad (2)$$

Also important here are the null constraints which are obtained from crossing symmetry. These form an infinite set and are orthogonal to a crossing-symmetric $\rho_J(s)$. Their expressions are also obtained from an expansion of the B_k sum rules. These constraints are critical in the project as they prevent large spins from contributing too much. As a result, it becomes possible to lower bound the parameters. Since they also average to 0 over an EFT loop, they offer a convenient way of setting up the numerical problem. The equation for the n_4 null constraint is given below.

$$\langle n_4(m^2,J)\rangle = \left\langle \frac{\mathcal{J}^2(2\mathcal{J}^2-5d+4)}{m^8} \right\rangle \tag{3}$$

Note that in the above equations, $\mathcal{J}^2$ is the Casimir operator $\mathcal{J}^2 = J(J+d-3)$ and d is the spacetime dimension (usually taken as 4).

4 Importance of the Results

An important and frequently used argument in Effective Field Theories is that of *dimensional analysis scaling*. Specifically, the coefficient of a $(k+d)$-dimensional operator in the Low-Effective Lagrangian to scale like $1/M^k$. This comes as a consequence of the action having dimension 1 in mass terms in the natural units. Hence any term of the form $g_k(\mathcal{O})^{(k+d)}$ (where $\mathcal{O}$ is an operator) will have dimension 1 when integrated over. Hence, there is a question of whether this argument can be justified as a by *rigorous numerical bounds*. Essentially that can "accidentally large" EFT coefficients be ruled out. The work presented in the primary reference proves the existence of two-sided numerical bounds for the couplings. Further it shows that the ratios $\tilde{g_3}, \tilde{g_4}, ...$ are actually $\mathcal{O}(1)$ numbers. Since the dimensionless ratio of these numbers has been taken, the fact that they are $\mathcal{O}(1)$ rigorously proves the dimensional scaling as a consequence of causality and unitarity. Hence, the main finding can be summarised as: *In any causal and unitary theory, dimensional analysis scaling is theorem.*

Part II
Numerical Methods

5 Three Sum Rule Warm-up Problem

As a warm-up to a general formulation of the problem, consider the problem of bounding $\tilde{g_3}$ with a single constraint. First, we consider the coefficients and cast them into a simpler form as follows:

$$h_2 = \left\langle \frac{1}{m^4}, \right\rangle \qquad h_3 = \left\langle \frac{a-\mathcal{J}^2}{m^6} \right\rangle, \tag{4}$$

$$0 = \left\langle \frac{\mathcal{J}^4 - b\mathcal{J}^2}{m^8} \right\rangle \equiv \left\langle n(m^2,J) \right\rangle \tag{5}$$

These are identical when $a = \frac{3(d-2)}{4}$, $b = \frac{5d-4}{2}$. Since we wish to lower bound g_3, the warm-up problem becomes to lower bound h_3. One way to do this is by applying the Cauchy-Schwarz inequality, which tells that the ratios of EFT couplings are $\mathcal{O}(1)$ numbers in units of cutoff scale M; i.e., the rations $\tilde{g}_k$ are $\mathcal{O}(1)$ numbers. However, the bounds computed by this are not optimal.

Consider now a functions $h_i(m^2,J)$ whose averages yield h_i. To cast them into a semi-definite programming framework, a positive-definite combination of these three averages can be constructed.

$$\begin{aligned} F(m^2,J) \equiv& \alpha h_2(m^2,J) + h_3(m^2,J) + \beta n(m^2,J) && (6)\\ =& \frac{\alpha}{m^4} + \frac{a-\mathcal{J}^2}{m^6} + \beta\frac{\mathcal{J}^4 - b\mathcal{J}^2}{m^8} && (7) \end{aligned}$$

Where we must find α, β such that $F(m^2,J) \geq 0, \quad \forall J = 0,2,4,..., \quad m \geq M$. The average of any such F proves that $h_3 \geq -\alpha h_2$ as the average over the EFT loop of the null constraint n_4 is 0.

This warm-up problem is set with only one null constraint and allows the possibility to be computed analytically. However, for a higher number of null constraints, the minimum values converge to a tighter bound, and the general framework is presented in the next section.

6 The Problem Setup:

The general problem set-up takes the form of the "dual" of an optimisation problem, which allows carving out the space of EFT coefficients $\tilde{g_3}, \tilde{g_4}, ...$ allowed by unitarity and positivity. We require two objects:

1. The representative averages $g_k(m^2,J)$ which measure each desired g_k as $g_k = \left\langle g_k(m^2,J) \right\rangle$
2. The null functions whose heavy averages vanish: $0 = \left\langle n_4(m^2,J) \right\rangle = \left\langle n_5(m^2,J) \right\rangle = \left\langle n_6(m^2,J) \right\rangle, ...$

First, consider the problem of lower bounding g_3 given g_2 by being agnostic about the other g_k's. We define a vector which combined g_2, g_3 along with the null constraints as

$$v(m^2,J) \equiv (g_2(m^2,J), M^2 g_3(m^2,J), n_4(m^2,J), n_5(m^2,J), ...) \tag{8}$$

Hence, the dual optimisation problem can be posed as :

$$\begin{cases} \text{maximise :} & A \\ \text{subject to:} & 0 \leq (-A, 1, c_4, c_5, ...) \cdot v(m^2,J) \\ & \forall m \geq M, \quad \forall J = 0,2,4,.. \end{cases} \tag{9}$$

After finding the optimal vector $(-A, 1, c_4, c_5, ...)$, consider the average of it's dot product with v, which yields $\langle -Ag_2 + M^2 g_3 + c_4 n_4 + c_5 n_5 + ...\rangle = -Ag_2 + M^2 g_3 \geq 0$. Hence this finally leads to the desired *optimal bound* as

$$A \leq \tilde{g_3} \quad \left(= g_3 \frac{M^2}{g_2}\right) \tag{10}$$

Similarly, we can find a maximum bound by:

$$\begin{cases} \text{minimise :} & B \\ \text{subject to:} & 0 \leq (B, -1, c_4, c_5, ...) \cdot v(m^2,J) \\ & \forall m \geq M, \quad \forall J = 0,2,4,.. \end{cases} \tag{11}$$

With this, it follows as above that $\tilde{g}_3 \leq B$. This yields a convex region defined by $A \leq \tilde{g}_3 \leq B$.

To obtain the optimal bounds for $\tilde{g}_4$ given optimal g_2 and g_3, we can modify the framework a little and cast into a similar problem. Consider now the vector as $v \to (g_2, M^2 g_3, M^4 g_4, n^4, n^5, ...)$ for which we impose the positivity of $(-A, -B, 1, c_4, c_5, ...) \cdot v(m^2, J)$ for all m and J. This yields the lower bound as $\tilde{g}_4 \geq A + B\tilde{g}_3{}^{(0)}$. This carves out a space in the $(\tilde{g}_3, \tilde{g}_4)$ plane defined by $\tilde{g}_4 - B\tilde{g}_3 \geq 0$. By repeating the calculation for distinct $\tilde{g}_3{}^{(0)}$, the convex shape of allowed values is carved out.

7 Implementation:

In order to implement the dual problem using a solver, two important points need to be noted:

- Since constrained optimisation problems are based on *polynomial* constraints, with semi-definite solvers accepting polynomials with $x \geq 0$, we set $m \to M^2(1+x)$ and remove a common denominator.
- Since the solver can only accept a finite list of polynomial constraints, we tabulate a finite list of spins, $J = 0, 2, 4, 6, .., J_{max}$, with J_{max} typically being 40, and adding a single function of x corresponding to $J \to \infty$

With these substitutions, our vector v gets modified to $v \equiv M^4(1+x)^4 \times (g_2, M^2 g_3, M^4 n_4)$. This leads to a purely polynomial set of constraints. It is important to note that multiplying with the common denominator doesn't affect the previously laid framework as $m^2 = M^2(1+x)^2 \geq 0 \ \forall x \geq 0$. The vector at infinite spin is written as $v(x, \infty) = (0, 0, 2)$. This can be found in the limiting scenario. Hence, in general, the vector v can be written as $v \to M^n(1+x)^n(g_2, M^2 g_3, M^4 n_4, ..., M^n n_n)$ where n is the highest Mandelstam order. Now with this in hand, we can use implementations of linear or semidefinite optimisation to find optimal parameters.

8 Numerical Methods: Constrained Optimisation

Optimisation problems that require finding an optimal solution given a set of constraints come under the scope of constrained optimisation. The optimisation problem can be further classified by the nature of the constraint equations. If the constrained equations satisfy only equalities, they can be converted to unconstrained optimisation through Lagrange multipliers. However, for inequality constraints, further depends on whether the constraints are linear, quadratic or nonlinear. Fortunately, here the constraints are linear and semidefinite, for which algorithms exist to solve the problem in polynomial time.

A linear programming algorithm finds a point in the polytope (which is necessarily convex) where this function has the smallest (or largest) value if such a point exists. They can be expressed in the following form:

```
Find a vector      $x$
That Maximises     $c^T x$
Subject to         $Ax \leq b$
and                $x \geq 0$
```

Where we actually modify the problem to fit the "dual" of the optimisation problem where we take $Ax \geq b$. Semidefinite (SDP) is a special case where it is framed as a problem in terms of matrices by requiring that the matrix of variables is semidefinite. SDP offers more efficient and stable algorithms for guessing the solution. SDP has become particularly important in physics, where it is used to constrain conformal field theories with the conformal bootstrap. Although Semidefinite programming offers some advantages, implementation with linear programming is possible here and offers reasonably accurate solutions with feasible running time. The algorithms utilised by both are generally similar and the Interior Point Methods guarantee a solution in cubic or sub-cubic time.

8.1 Naive Implementation

A naive algorithm here would involve a brute-force search throughout the space of allowed values to find the optimal value. However, as the search space can be large, this method is inefficient. Hence, a brute force algorithm was developed that optimises solving the LP at hand. Certain key physical assumptions are extremely helpful:

- The g_k's are known to be $\mathcal{O}(1)$ numbers. Hence, the value of A can be sufficiently restricted between say $(-100, 100)$.
- Since the parameters must scale as M^2, every subsequent parameter is about 1-2 orders of magnitude smaller than the previous one. This can be glanced from the solution of the warm-up problem that yields: $A \approx 10$(order 2) and $B \approx 0.07$ (order 4).

We can now formulate a general algorithm, for which with certain modifications, large speed-ups can be obtained.

```
$A = (A_{min}, ..n.., A_{max})$
$B = (B_{min}, ..n.., B_{max})$
while ||New Soln - Old Soln|| > eps do
    Iterate over all  A, B, J, x
    if $(-A, 1, B) \leq 0$ for any J, x then
        Exit iteration and go to next (A,B) pair
    else
        Save the Value of (A, B) to a vector
    end if
    End Loop
    Find optimal value of A and B from saved values.
    Redefine $A_{min}, A_{max}, B_{min}, B_{max}$ to be tighter around the found optimal value
end while
```

Now note that the idea of *tighter* has been left ambiguous. This is because by modifying what we meant by tighter, we can achieve faster convergence. A naive strategy is to use equally spaced n values between the maximum and minimum values. However, this strategy can be slow as it requires a slower convergence of outer limits as well as a large value of n for convergence.

A better strategy is to capitalise on the nature of the optimal solution. As the optimal solution in the next iteration is *close* to the current optimal value, a random set of parameters based on a Gaussian centred on the previous optimal value can be adopted. This strategy inspired by Monte Carlo optimisation methods offers considerable speed-up. It also offers a way to control the tightening of the bounds by decreasing the standard deviation and works well even with small n. A scaling of 2^{-k} has been used where k is the iteration number.

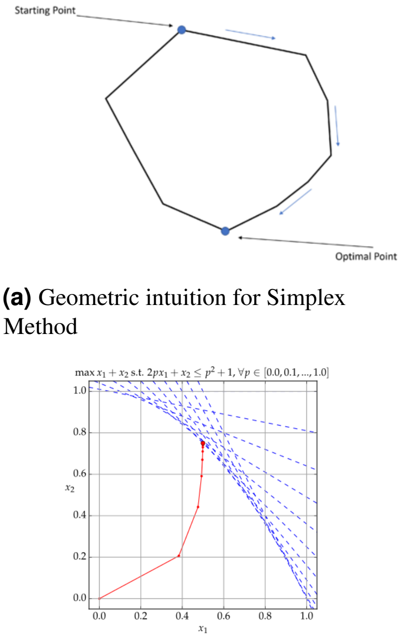
Even though the naive brute force methods were optimised and parallelised, they are much slower for algorithms which currently solve LP problems. The algorithm can be easily adapted to solve for g_4 and/or use a larger number of null constraints. However, any of these would add another parameter to be optimised and lead to an order of magnitude slower runtime.

8.2 Simplex Algorithm

There are two key points in LP programming: the feasible region forms a convex polytope and there is only one vertex (extreme point) of the feasible region that contains the optimal solution. Hence, this algorithm involves finding the vertices and hopping from vertex to vertex to find the optimal solution for the problem. This is done by a tabular method, hence in some contexts, this method is also called the Simplex Tabular method. This is based on the geometric assumption, and Figure 1a displays the geometric intuition behind the idea. While the method guarantees a solution and is polynomial time on average, the worst-case scenarios can complexity is not well known and is often exponential.

8.3 Interior Point Methods

Interior point methods are robust and are guaranteed to find a solution in polynomial time. They are used for the development of all high-performing LPs and SDPs. Further, they can also be applied to non-linear optimisations cases. The current algorithms for the problem rely on Cholesky decomposition along with Newton's methods and are heavily optimised for sparse matrices. An intuitive way of understanding this algorithm is that it an implementation of a gradient descent algorithm modified with the help of a barrier function. Hence, we start in the interior of the convex region and with each step find the optimal solution until the boundary is reached, where we find the optimal solution to the problem. The Figure 1b provides an intuitive understanding of the process that this algorithm follows.



(a) Geometric intuition for Simplex Method

(b) Intuitive understanding of the Interior Point Method

Figure 1. Geometric Ideas behind LP algorithms

Both of these algorithms have been implemented with the help of pre-existing libraries and programs. In particular, GLPK (GNU Linear Programming Kit) and MOSEK were used along with PuLP which is a Python package that acts as a wrapper for implementing the above-mentioned LP solvers.

Part III

Results

Here two key results will be presented. As the key finding of the paper was the existence of two-sided bounds, the computed two-sided bounds will be shown for $\tilde{g_3}$. In addition, the box bounds of $\tilde{g_4}$ are simple, but when a plot of $\tilde{g3} - \tilde{g_4}$ is plotted by computing the $\tilde{g_4}$ for a given value of $\tilde{g_3}^{(0)}$, a convex shape emerges. Further, this convex shape is shown to have a separation for spinless and spin particles.

9 Results from Three Sum Rule:

For the Three Sum Rule, the optimal vector computed alongside the optimal vector computed by authors has been shown side by side (Left side my results, Right side from paper):

$$0 \leq F(m^2,J) = \begin{pmatrix} 10.612492 \\ 1 \\ 0.00671869 \end{pmatrix} \cdot \begin{pmatrix} g_2(m^2,J) \\ M^2 g_3(m^2,J) \\ M^4 n_4(m^2,J) \end{pmatrix} \qquad = \begin{pmatrix} 10.61249 \\ 1 \\ 0.00671875 \end{pmatrix} \cdot \begin{pmatrix} g_2(m^2,J) \\ M^2 g_3(m^2,J) \\ M^4 n_4(m^2,J) \end{pmatrix} \tag{12}$$

As can be seen, the values are accurate up to a high degree of accuracy.

10 Results on Two-sided Bounds

The lower bound for $\tilde{g_3}$ was calculated as detailed in the numerical strategy above. The upper bound can be obtained from the $J = 0$ contributions. Similarly, for $\tilde{g_4}$, a purely rational upper bound is obtained from the spin-0 contribution as its expression is spin independent. The computed values of $\tilde{g_3}$ are listed below and compared against those found from the paper (all rounded upto 4th decimal):

n (Mandelstam order)	dim N (number of constraints)	min $\tilde{g_3}$ (Paper)	min $\tilde{g_3}$ (Brute Force)	min $\tilde{g_3}$ (LP)
4	1	-10.6125	-10.6125	-10.6125
5	2	-10.6125	-10.606	-10.6125
6	3	-10.3662	-10.3663	-10.3662
7	5	-10.3580	Too Long to Compute	-10.35799

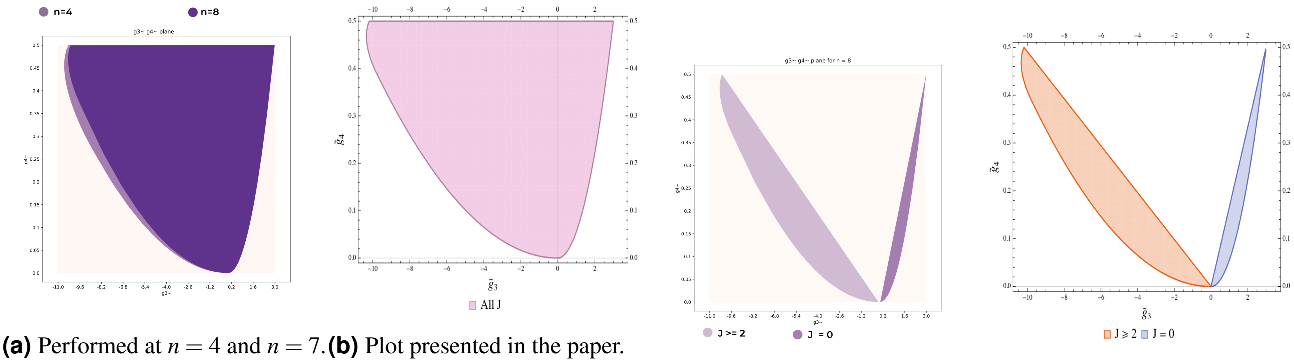
Note: Although the paper has generated optimal bounds for higher orders of constraints and also coefficients, the expressions for the same are not provided. To generate the expressions, the approach by the authors is not easy and requires the usage of crossing symmetry. Hence, only the provided expressions were used for computation.
The best optimal bounds computed in this project are:

$$-10.3580 \leq \tilde{g_3} \leq 3 \qquad\qquad 0 \leq \tilde{g_4} \leq \frac{1}{2} \tag{13}$$

The best optimal bounds given in the paper (at higher n) are $-10.346 \leq \tilde{g_3} \leq 3$ and $0 \leq \tilde{g_4} \leq \frac{1}{2}$

11 g3-g4 Plane

The mutual plot for $\tilde{g_3}$ and $\tilde{g_4}$ have been shown. The first of the two sets of plots are for the general case, and the second set of plots shows the separation between the spinless and non-zero spin states.



(a) Performed at $n = 4$ and $n = 7$. Plot made for project.

(b) Plot presented in the paper. Performed at $n = 10$

(c) Plot made for project

(d) Plot presented in the paper

Figure 2. (a) and (b): The $\tilde{g_3} - \tilde{g_4}$ allowed region. (c) and (d): The $\tilde{g_3} - \tilde{g_4}$) allowed region, segmented into theories without spinless particles ($J \geq 2$) and theories with only spinless particles ($J = 0$).

Acknowledgements (optional)

I want to thank my project mentor Prof. Aninda Sinha for guiding me through the project and ensuring its successful completion. His support and constant dialogue helped me troubleshoot and improve my code. I would also like to thank Prof Manish Jain for providing me with the opportunity to work on an interesting project during the semester. Further, the content covered by him in the course as well as the multiple discussions have contributed immensely to this project. I would also like to thank my friends, especially Saakshi Porwal and Nisarg Chadha who is also taking this course. Discussions regarding course content and projects helped foster new ideas and get exposed me to a wider variety of computational techniques. Finally, I would like to thank my roommate who had to endure my working through the night.

12 Additional References

- PuLP 2.7.0 Python Module: https://coin-or.github.io/pulp/index.html
- GLPK (GNU Linear Programming Kit): https://www.gnu.org/software/glpk/
- MOSEK: https://www.mosek.com/ (Free Academic License)
- Gurobi: https://www.gurobi.com/ (Free Academic License)