

Numerical Evaluation of The $\mathcal{M}_{gg \rightarrow gH}$ Amplitude : Higgs + Jet Production through Gluon Fusion

Chaitanya Paranjape - IIT(ISM) Dhanbad, India

Abstract

This is the project report on the numerical evaluation of the amplitude for the $gg \rightarrow gH$ process at 1-Loop LO. We will successfully demonstrate the use of *sum_package* with pySecDec [2],[3] to numerically evaluate the 1-Loop amplitude for $gg \rightarrow gH$, by using the linear combination of master integrals. This example can be used as a template example to extend on towards multi-Loop calculations or some other amplitude at 1-loop Level.

I am humbled and very grateful to have had Prof. Dr. Gudrun Heinrich's Invaluable guidance during this project. I had the incredible opportunity to attend the Collider Phenomenology Lectures by Prof. Dr. Gudrun Heinrich, which were of great help with this project and my future studies. I would also like to thank the German Academic Exchange Service (DAAD), who offered the DAAD WISE Scholarship, which made this project possible.

- 1 Introduction**
- 2 The Master Integrals**
- 3 The Form Factors**
- 4 Analytic Expressions**
- 5 Implementation in pySecDec**
- 6 References**
- 7 Appendix : A**
- 8 Appendix : B**

Introduction

The aim of this project is to numerically evaluate the LO (1-Loop) squared matrix Element for the process : $gg \rightarrow gH$, using the newly developed *sum_package* of **pySecDec** [2],[3]. More about **pySecDec** can be found out from **pySecDec** (<https://secdec.readthedocs.io/en/latest/>). The *sum_package* is designed to numerically evaluate the weighted superposition of a set of feynman integrals. Therefore, our purpose would be to express the required amplitude in terms of the so called master integrals ¹. The derivation for our amplitude can be found out from [1]. Most of the equations we will be using on-wards have been borrowed from the same reference only.

In the framework of Conventional Dimensional Regularization(CDR), spin and color averaged absolute square of our amplitude is given by :

$$|\overline{\mathcal{M}}_{gg \rightarrow gH}|^2 = \frac{1}{(d-2)^2} \cdot \frac{1}{(N_c^2 - 1)^2} \cdot |\mathcal{M}_{gg \rightarrow gH}|^2 \quad (\text{II.46})$$

Where we have divided by the number of polarization $(d-2)$ & the number of colors $(N_c^2 - 1)$ for each of the incoming gluon. The absolute square of the matrix element is expressed in terms of the form factors ².

$$\begin{aligned} |\mathcal{M}_{gg \rightarrow gH}|^2 = \frac{1}{4} \cdot \left[\frac{(d-2)|F_{212}|^2 s_{12}^3 s_{23}}{s_{13}} + F_{212} F_{332}^* s_{12} s_{23}^2 + F_{212} F_{311}^* s_{12}^2 s_{13} + (d-2) F_{212} F_{312}^* s_{12}^2 s_{23} \right. \\ + F_{332} F_{212}^* s_{12} s_{23}^2 + \frac{(d-2)|F_{332}|^2 s_{13} s_{23}^3}{s_{12}} + F_{332} F_{311}^* s_{13}^2 s_{23} + (d-2) F_{332} F_{312}^* s_{13} s_{23}^2 \\ + F_{311} F_{212}^* s_{12}^2 s_{13} + F_{311} F_{332}^* s_{13}^2 s_{23} + \frac{(d-2)|F_{311}|^2 s_{12} s_{13}^3}{s_{23}} + (d-2) F_{311} F_{312}^* s_{12} s_{13}^2 \\ + (d-2) F_{312} F_{212}^* s_{12}^2 s_{23} + (d-2) F_{312} F_{332}^* s_{13} s_{23}^2 + (d-2) F_{312} F_{311}^* s_{12} s_{13}^2 \\ \left. + (3d-8)|F_{312}|^2 s_{12} s_{13} s_{23} \right] \quad (\text{II.48}) \end{aligned}$$

The form factor definitions can be obtained from [1]. Note here that only 4 of the form factors contribute to the physical amplitude. Our job would be to express these form factors in terms of the master integrals, such that we can provide them to the **pySecDec** for the numerical evaluation using the *sum_package*. We can then evaluate the required matrix element squared and verify our results with the analytic expressions in the Heavy Top quark Limit (HTL).

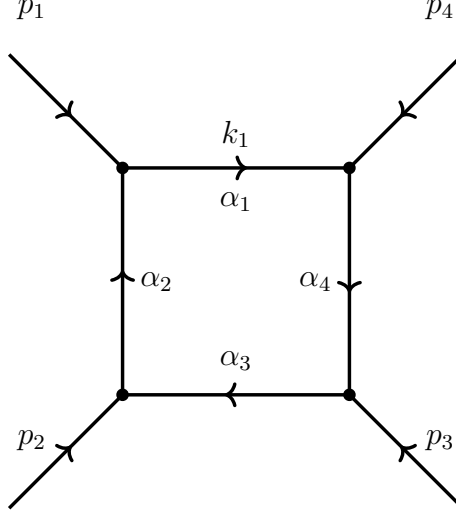
The project report has been divided into two parts, first dealing with the calculations of integrals, form factors and setting up all the equations we will be using. Second part will be focusing on the implementation of the same with **pySecDec** and verifying our results with the previously set up analytic expressions. We will now start with the first part by identifying the master integrals in the 1-Loop-Box topology of our $gg \rightarrow gH$ process.

¹The analytic calculation of the amplitude involves evaluating a large number of scalar feynman integrals. Instead of evaluating all of the integrals, we can find out relations between these integrals with the help of the Integration By Parts(IBP) identities. Using this method of IBP Reduction, we can express all the integrals in terms of a finite set of integrals called as the master integrals.

²The form factors are the Lorentz invariant coefficients defined in the tensorial Decomposition of the amplitude, the definitions for which can be found out from [1].

The Master Integrals

All of the master integrals in our case correspond to the different sub-topology of the 1-Loop-Box topology. Let us set up the 1-Loop-Box topology for the $gg \rightarrow gH$ process.



The kinematic constraints imposed will be $p_1^2 = p_2^2 = p_3^2 = 0$ & $p_4^2 = m_H^2$ along with the momentum conservation : $p_1 + p_2 + p_3 + p_4 = 0 \Rightarrow m_H^2 = s_{12} + s_{13} + s_{23}$.

Where we have defined the kinematic in-variants as : $s_{ij} = (p_i + p_j)^2$.

Conveniently, the number of independent scalar products is same as the number of propagators. As a result, we can immediately define the general scalar feynman integral as follows :

$$I_{\alpha_1, \alpha_2, \alpha_3, \alpha_4} = \int \frac{d^d k_1}{i\pi^{d/2}} \cdot \frac{1}{D_1^{\alpha_1} D_2^{\alpha_2} D_3^{\alpha_3} D_4^{\alpha_4}}$$

with the propagators given by :

$$\begin{aligned} D_1 &= k_1^2 - m_t^2 \\ D_2 &= (k_1 - p_1)^2 - m_t^2 \\ D_3 &= (k_1 - p_1 - p_2)^2 - m_t^2 \\ D_4 &= (k_1 - p_1 - p_2 - p_3)^2 - m_t^2 \end{aligned}$$

Topology of the integral depends on the propagator powers, therefore, we can obtain the different sub-topology by shrinking down the propagators, which would be setting the corresponding propagator power to zero. Key thing to note here is that, shrinking the propagators does not violate the momentum conservation at any point and the integral still belongs to the same topology family. As a result, they can still be related by the IBP identities. Which is how we reduce the set of integrals down to the set of master integrals. ³

The master integrals for our example have already been well defined and can be found out from [1]. The 9 master integrals are as follows : (* have crossed Integrals)

$$\begin{aligned} g_1(m_t^2) &= I_{2,0,0,0} & * \quad g_2(s_{12}, m_t^2) &= I_{2,0,1,0} & g_3(s_{23}, m_t^2) &= I_{0,2,0,1} \\ g_4(m_H^2, m_t^2) &= I_{2,0,0,1} & * \quad g_5(s_{12}, m_t^2) &= I_{1,1,1,0} & g_6(s_{23}, m_t^2) &= I_{0,1,1,1} \\ * \quad g_7(s_{12}, m_H^2, m_t^2) &= I_{1,0,1,1} & g_8(s_{23}, m_H^2, m_t^2) &= I_{1,1,0,1} & * \quad g_9(s_{12}, s_{23}, m_H^2, m_t^2) &= I_{1,1,1,1} \end{aligned}$$

³For more information about master Integrals and IBP reduction, kindly refer to [4].

Along with these 9 master integrals, some crossed-integrals also appear in the expressions of form-factors and amplitude. There are 5 extra crossed-integrals which we will also need in our expressions. The required 5 crossed integrals are obtained as :

- $g_2(s_{13}, m_t^2)$: obtained from $g_2(s_{12}, m_t^2)$ by Exchanging p_2 and p_3 .
- $g_5(s_{13}, m_t^2)$: obtained from $g_5(s_{12}, m_t^2)$ by Exchanging p_2 and p_3 .
- $g_7(s_{13}, m_H^2, m_t^2)$: obtained from $g_7(s_{12}, m_H^2, m_t^2)$ by Exchanging p_2 and p_3 .
- $g_9(s_{12}, s_{13}, m_H^2, m_t^2)$: obtained from $g_9(s_{12}, s_{23}, m_H^2, m_t^2)$ by Exchanging p_1 and p_2 .
- $g_9(s_{23}, s_{13}, m_H^2, m_t^2)$: obtained from $g_9(s_{12}, s_{23}, m_H^2, m_t^2)$ by Exchanging p_2 and p_3 .

Now, we have defined all the integrals we will need. For convenience, we will set a fixed order for these integrals, and will stick to this order for all the labelling in the calculations and pySecDec implementaion. The ordered 14 integrals are :

1. $g_1(m_t^2) = I_{2,0,0,0}$
2. $g_2(s_{12}, m_t^2) = I_{2,0,1,0}$
3. $g_2(s_{13}, m_t^2) = I_{2,0,1,0} : (p_2 \leftrightarrow p_3)$
4. $g_3(s_{23}, m_t^2) = I_{0,2,0,1}$
5. $g_4(m_H^2, m_t^2) = I_{2,0,0,1}$
6. $g_5(s_{12}, m_t^2) = I_{1,1,1,0}$
7. $g_5(s_{13}, m_t^2) = I_{1,1,1,0} : (p_2 \leftrightarrow p_3)$
8. $g_6(s_{23}, m_t^2) = I_{0,1,1,1}$
9. $g_7(s_{12}, m_H^2, m_t^2) = I_{1,0,1,1}$
10. $g_7(s_{13}, m_H^2, m_t^2) = I_{1,0,1,1} : (p_2 \leftrightarrow p_3)$
11. $g_8(s_{23}, m_H^2, m_t^2) = I_{1,1,0,1}$
12. $g_9(s_{12}, s_{23}, m_H^2, m_t^2) = I_{1,1,1,1}$
13. $g_9(s_{12}, s_{13}, m_H^2, m_t^2) = I_{1,1,1,1} : (p_1 \leftrightarrow p_2)$
14. $g_9(s_{23}, s_{13}, m_H^2, m_t^2) = I_{1,1,1,1} : (p_2 \leftrightarrow p_3)$

Now we can turn our attention to the 4 form factors and express them as superposition of the above integrals.

The Form Factors

There are only 4 form factors which appear in the physical amplitude, namely $F_{212}, F_{311}, F_{332}$ & F_{312} . The expressions and derivations for which can be obtained from [1]. Our task would be to simplify the provided definitions and express them as superposition of the 14 Integrals.

Out of the 4 factors : $F_{212}, F_{311}, F_{332}$ are related through the permutation in-variance of the amplitude. $F_{212}(s_{12}, s_{13}, s_{23}) = F_{311}(s_{13}, s_{23}, s_{12}) = F_{332}(s_{23}, s_{12}, s_{13})$. So, once we have F_{212} ready, pySecDec can calculate the other two easily by just changing the arguments.

Therefore, F_{212} and F_{312} are the two independent form factors we need to deal with.

Let's first focus on F_{212} :

We define temporary set of coefficients T_1, T_2, \dots, T_{13} ⁴ such that the F_{212} can be expressed as follows : (defined at Pg 97 of [1])

$$\begin{aligned}
 F_{212}(s_{12}, s_{13}, s_{23}) = & f^{abc} C_\epsilon \cdot 2m_t e g_{ht} g_s^3 \cdot \left[T_1 \cdot A(m_H^2, m_t^2) \right. \\
 & + T_2 \cdot B(s_{12}, m_H^2, m_t^2) \\
 & + T_3 \cdot B(s_{13}, m_H^2, m_t^2) \\
 & + T_4 \cdot B(s_{23}, m_H^2, m_t^2) \\
 & + T_5 \cdot C(s_{12}, s_{13}, s_{23}, m_t^2) \\
 & + T_6 \cdot C(s_{13}, s_{23}, s_{12}, m_t^2) \\
 & + T_7 \cdot C(s_{23}, s_{12}, s_{13}, m_t^2) \\
 & + T_8 \cdot g_7(s_{12}, m_H^2, m_t^2) \\
 & + T_9 \cdot g_7(s_{13}, m_H^2, m_t^2) \\
 & + T_{10} \cdot g_8(s_{23}, m_H^2, m_t^2) \\
 & + T_{11} \cdot g_9(s_{12}, s_{23}, m_H^2, m_t^2) \\
 & + T_{12} \cdot g_9(s_{12}, s_{13}, m_H^2, m_t^2) \\
 & \left. + T_{13} \cdot g_9(s_{23}, s_{13}, m_H^2, m_t^2) \right] \\
 \Rightarrow F_{212}(s_{12}, s_{13}, s_{23}) = & f^{abc} C_\epsilon \cdot 2m_t e g_{ht} g_s^3 \cdot f_{212}(s_{12}, s_{13}, s_{23})
 \end{aligned}$$

The definitions for $A(), B()$ and $C()$ functions [1] are as follows :

$$\begin{aligned}
 A(m_H^2, m_t^2) &= g_1(m_t^2) + (4m_t^2 - m_H^2)g_4(m_H^2, m_t^2) \\
 B(x, m_H^2, m_t^2) &= (4m_t^2 - x)g_{2,3}(x, m_t^2) - (4m_t^2 - m_H^2)g_4(m_H^2, m_t^2) \\
 C(x, y, z, m_t^2) &= (d-2)g_{5,6}(x, m_t^2) - (d-2)\frac{y+z}{x}g_{7,8}(x, m_H^2, m_t^2)
 \end{aligned}$$

Where $g_{i,j}$ can be either g_i or g_j depending on the argument x .

We define the normalized form factor as f_{212} . This is what we will be providing as input to pySecDec. Substituting the definitions for $A(), B()$ and $C()$ functions, we can express our normalized form factor f_{212} as linear combination of the 14 Integrals with 14 coefficients. We call these coefficients as P coefficients.

⁴Refer to the appendix: A, for the exact expressions of the T and N coefficients

The 14 P coefficients P_1, P_2, \dots, P_{14} , are defined as follows :

$$\begin{aligned}
f_{212}(s_{12}, s_{13}, s_{23}) = & \left[P_1 \cdot g_1(m_t^2) + P_2 \cdot g_2(s_{12}, m_t^2) \right. \\
& + P_3 \cdot g_2(s_{13}, m_t^2) + P_4 \cdot g_3(s_{23}, m_t^2) + P_5 \cdot g_4(m_H^2, m_t^2) \\
& + P_6 \cdot g_5(s_{12}, m_t^2) + P_7 \cdot g_5(s_{13}, m_t^2) + P_8 \cdot g_6(s_{23}, m_t^2) \\
& + P_9 \cdot g_7(s_{12}, m_H^2, m_t^2) + P_{10} \cdot g_7(s_{13}, m_H^2, m_t^2) + P_{11} \cdot g_8(s_{23}, m_H^2, m_t^2) \\
& + P_{12} \cdot g_9(s_{12}, s_{23}, m_H^2, m_t^2) + P_{13} \cdot g_9(s_{12}, s_{13}, m_H^2, m_t^2) \\
& \left. + P_{14} \cdot g_9(s_{23}, s_{13}, m_H^2, m_t^2) \right]
\end{aligned}$$

The P Coefficients are obtained in terms of the T Coefficients as following :

$$\begin{aligned}
P_1 &= T_1 \\
P_2 &= T_2 \cdot (4m_t^2 - s_{12}) \\
P_3 &= T_3 \cdot (4m_t^2 - s_{13}) \\
P_4 &= T_4 \cdot (4m_t^2 - s_{23}) \\
P_5 &= (T_1 - T_2 - T_3 - T_4) \cdot (4m_t^2 - s_{12} - s_{13} - s_{23}) \quad \because (m_H^2 = s_{12} + s_{13} + s_{23}) \\
P_6 &= T_5 \cdot (d - 2) \\
P_7 &= T_6 \cdot (d - 2) \\
P_8 &= T_7 \cdot (d - 2) \\
P_9 &= T_8 - T_5 \cdot (d - 2) \cdot \frac{(s_{13} + s_{23})}{s_{12}} \\
P_{10} &= T_9 - T_6 \cdot (d - 2) \cdot \frac{(s_{23} + s_{12})}{s_{13}} \\
P_{11} &= T_{10} - T_7 \cdot (d - 2) \cdot \frac{(s_{13} + s_{12})}{s_{23}} \\
P_{12} &= T_{11} \\
P_{13} &= T_{12} \\
P_{14} &= T_{13}
\end{aligned}$$

The exact expressions for all the P and T coefficients can be viewed in the mathematica notebook for the F_{212} form factor at the Git-hub repository for this project : [PSD_gggH1L](#)

Now we have completely defined f_{212} form factor, ready for input to the pySecDec. We can deal with the f_{311} and f_{332} during the pySecDec implementation part. Let's focus on the F_{312} now. We again use the definition from [1]. Similarly as for F_{212} , we define temporary set of N Coefficients such that form factor F_{312} can be expressed as follows :

$$\begin{aligned}
F_{312}(s_{12}, s_{13}, s_{23}) &= f^{abc} C_\epsilon \cdot 2m_t e g_{ht} g_s^3 \cdot \left[N_1 \cdot A(m_H^2, m_t^2) \right. \\
&\quad + N_2 \cdot B(s_{12}, m_H^2, m_t^2) \\
&\quad + N_3 \cdot C(s_{12}, s_{13}, s_{23}, m_t^2) \\
&\quad + N_4 \cdot g_7(s_{12}, m_H^2, m_t^2) \\
&\quad + N_5 \cdot g_9(s_{12}, s_{23}, m_H^2, m_t^2) \\
&\quad + N_{11} \cdot A(m_H^2, m_t^2) \\
&\quad + N_{21} \cdot B(s_{13}, m_H^2, m_t^2) \\
&\quad + N_{31} \cdot C(s_{13}, s_{23}, s_{12}, m_t^2) \\
&\quad + N_{41} \cdot g_7(s_{13}, m_H^2, m_t^2) \\
&\quad + N_{51} \cdot g_9(s_{13}, s_{12}, m_H^2, m_t^2) \\
&\quad + N_{12} \cdot A(m_H^2, m_t^2) \\
&\quad + N_{22} \cdot B(s_{23}, m_H^2, m_t^2) \\
&\quad + N_{32} \cdot C(s_{23}, s_{12}, s_{13}, m_t^2) \\
&\quad + N_{42} \cdot g_7(s_{23}, m_H^2, m_t^2) \\
&\quad \left. + N_{52} \cdot g_9(s_{23}, s_{13}, m_H^2, m_t^2) \right] \\
\Rightarrow F_{312}(s_{12}, s_{13}, s_{23}) &= f^{abc} C_\epsilon \cdot 2m_t e g_{ht} g_s^3 \cdot f_{312}(s_{12}, s_{13}, s_{23})
\end{aligned}$$

Where we have defined the normalized form factor as f_{312} which is what we will be providing as input to pySecDec. Using the definitions of $A()$, $B()$ and $C()$, we can express our normalized form factor f_{312} as linear combination of the 14 integrals with 14 coefficients. We call these 14 coefficients as M coefficients. The 14 M coefficients M_1, M_2, \dots, M_{14} are defined as follows :

$$\begin{aligned}
f_{312}(s_{12}, s_{13}, s_{23}) &= \left[M_1 \cdot g_1(m_t^2) + M_2 \cdot g_2(s_{12}, m_t^2) \right. \\
&\quad + M_3 \cdot g_2(s_{13}, m_t^2) + M_4 \cdot g_3(s_{23}, m_t^2) + M_5 \cdot g_4(m_H^2, m_t^2) \\
&\quad + M_6 \cdot g_5(s_{12}, m_t^2) + M_7 \cdot g_5(s_{13}, m_t^2) + M_8 \cdot g_6(s_{23}, m_t^2) \\
&\quad + M_9 \cdot g_7(s_{12}, m_H^2, m_t^2) + M_{10} \cdot g_7(s_{13}, m_H^2, m_t^2) + M_{11} \cdot g_8(s_{23}, m_H^2, m_t^2) \\
&\quad + M_{12} \cdot g_9(s_{12}, s_{23}, m_H^2, m_t^2) + M_{13} \cdot g_9(s_{12}, s_{13}, m_H^2, m_t^2) \\
&\quad \left. + M_{14} \cdot g_9(s_{23}, s_{13}, m_H^2, m_t^2) \right]
\end{aligned}$$

The M coefficients in terms of N coefficients are expressed as follows :

$$\begin{aligned}
M_1 &= N_1 + N_{11} + N_{12} \\
M_2 &= (4m_t^2 - s_{12}) \cdot N_2 \\
M_3 &= (4m_t^2 - s_{13}) \cdot N_{21} \\
M_4 &= (4m_t^2 - s_{23}) \cdot N_{22} \\
M_5 &= (4m_t^2 - s_{12} - s_{13} - s_{23}) \cdot (N_1 + N_{11} + N_{12} - N_2 - N_{21} - N_{22}) \\
M_6 &= (d - 2) \cdot N_3 \\
M_7 &= (d - 2) \cdot N_{31}
\end{aligned}$$

$$\begin{aligned}
M_8 &= (d-2) \cdot N_{32} \\
M_9 &= N_4 - N_3 \cdot \frac{(d-2)(s_{13} + s_{23})}{s_{12}} \\
M_{10} &= N_{41} - N_{31} \cdot \frac{(d-2)(s_{12} + s_{23})}{s_{13}} \\
M_{11} &= N_{42} - N_{32} \cdot \frac{(d-2)(s_{12} + s_{13})}{s_{23}} \\
M_{12} &= N_5 \\
M_{13} &= N_{51} \\
M_{14} &= N_{52}
\end{aligned}$$

The exact expressions for all the M and N coefficients can be viewed in the mathematica notebook for the F_{312} form factor at the Git-hub repository for this project : [PSD_gggH1L](#)

We have completely defined both of the required form factors and are ready to provide them as input to pySecDec for numerical evaluation. But before that, we first set up the equation for matrix element squared with the normalized form factors and also investigate the analytic expressions in the Heavy Top quark Limit, which we will use for verification purpose.

Analytic Expressions

Once we have form factors in terms of the master integrals, all left to do is to substitute them into equation (II.48) to calculate the matrix element squared. One thing to note here is that, because we will be working with normalized form factors in pySecDec, We have to modify the equation (II.48) for normalized form factors.

The normalization factor for all form factors is $K = f^{abc}C_\epsilon \cdot 2m_t e g_{ht} g_s^3$. Therefore, we will normalize the amplitude squared with $|K|^2$.

$$K = f^{abc}C_\epsilon \cdot 2m_t e g_{ht} g_s^3 = f^{abc} \frac{1}{(4\pi)^2} \cdot \frac{2m_t^2}{v} g_s^3$$

$$\Rightarrow |K|^2 = N_c(N_c^2 - 1) \cdot \frac{m_t^4 \alpha_s^3}{\pi v^2}$$

where we have used the relations :

$$C_\epsilon = \frac{1}{(4\pi)^2} + \mathcal{O}(\epsilon), \quad e \cdot g_{ht} = \frac{m_t}{v}, \quad g_s^2 = 4\pi\alpha_s,$$

For the color algebra, we know

$$Tr(F^a F^b) = 2 \cdot T_R \cdot N_c \delta^{ab}$$

set $a = b$ and sum over, also use $T_R = 1/2$

$$\Rightarrow |f^{abc}|^2 = N_c(N_c^2 - 1)$$

Therefore, we can now write Equation (II.48) with normalized form factors as follows :

$$\begin{aligned} \frac{|\mathcal{M}_{gg \rightarrow gH}|^2}{|K|^2} = \frac{1}{4} \cdot \left[\frac{(d-2)|f_{212}|^2 s_{12}^3 s_{23}}{s_{13}} + f_{212} f_{332}^* s_{12} s_{23}^2 + f_{212} f_{311}^* s_{12}^2 s_{13} + (d-2) f_{212} f_{312}^* s_{12}^2 s_{23} \right. \\ + f_{332} f_{212}^* s_{12} s_{23}^2 + \frac{(d-2)|f_{332}|^2 s_{13} s_{23}^3}{s_{12}} + f_{332} f_{311}^* s_{13}^2 s_{23} + (d-2) f_{332} f_{312}^* s_{13} s_{23}^2 \\ + f_{311} f_{212}^* s_{12}^2 s_{13} + f_{311} f_{332}^* s_{13}^2 s_{23} + \frac{(d-2)|f_{311}|^2 s_{12} s_{13}^3}{s_{23}} + (d-2) f_{311} f_{312}^* s_{12} s_{13}^2 \\ + (d-2) f_{312} f_{212}^* s_{12}^2 s_{23} + (d-2) f_{312} f_{332}^* s_{13} s_{23}^2 + (d-2) f_{312} f_{311}^* s_{12} s_{13}^2 \\ \left. + (3d-8)|f_{312}|^2 s_{12} s_{13} s_{23} \right] \end{aligned} \quad (\text{II.48a})$$

This is the general expression for amplitude squared. (axial gauge expression has been used for polarization sums. for more details : [1])

Once, we have numerically evaluated the form factors, we can substitute them in above expressions to get final results. To verify our numerical results, we will use the analytic expression for the amplitude squared, in the Heavy Top quark Limit. We will go over the details of Heavy Top quark Limit(HTL) during the implementation part. For now, we use the expression for amplitude squared in HTL, from [1].

$$|\mathcal{M}_{gg \rightarrow gH}|^2 = \frac{4}{9} \cdot \frac{N_c(N_c^2 - 1)\alpha_s^3}{\pi v^2} \cdot \frac{(m_H^8 + s_{12}^4 + s_{13}^4 + s_{23}^4)}{s_{12} s_{13} s_{23}} + \mathcal{O}(\epsilon)$$

We normalize it with $|K|^2$ to compare the results with (II.48a).

$$\frac{|\mathcal{M}_{gg \rightarrow gH}|^2}{|K|^2} = \frac{4}{9m_t^4} \cdot \frac{(m_H^8 + s_{12}^4 + s_{13}^4 + s_{23}^4)}{s_{12}s_{13}s_{23}} + \mathcal{O}(\epsilon)$$

This will be the final analytic Expression we will use for verification of our numerical results. We will call this the normalized amplitude squared. The individual form factors can also be verified in HTL. They will be expressed as follows :

$$f_{212} = \frac{4}{3m_t^2 s_{23}}, \quad f_{311} = \frac{4}{3m_t^2 s_{12}}, \quad f_{332} = \frac{4}{3m_t^2 s_{13}}, \quad f_{312} = \frac{4}{3m_t^2} \cdot \left(\frac{1}{s_{12}} + \frac{1}{s_{13}} + \frac{1}{s_{23}} \right)$$

Substituting the above form factors in (II.48a) lead us back to the normalized amplitude squared.

We can also evaluate the helicity amplitudes. Because each gluon can have two helicities, we have $2^3 = 8$ helicity amplitudes, although they are not all independent. They are all related by parity, such that only half of them are now independent. Furthermore, thanks to the permutation invariance of the amplitude, we finally have only two independent helicity amplitudes.

$$\mathcal{M}_{gg \rightarrow gH}^{h_1 h_2 h_3} = -\mathcal{M}_{gg \rightarrow gH}^{-h_1 -h_2 -h_3}$$

$$\mathcal{M}_{gg \rightarrow gH}^{++-}(s_{12}, s_{13}, s_{23}) = \mathcal{M}_{gg \rightarrow gH}^{+-+}(s_{13}, s_{23}, s_{12}) = \mathcal{M}_{gg \rightarrow gH}^{-++}(s_{23}, s_{12}, s_{13})$$

The two independent helicity amplitudes are represented in terms of the normalized form factors as follows : (Upto a phase factor)

$$\begin{aligned} \mathcal{M}_{gg \rightarrow gH}^{+++} &= \frac{\sqrt{s_{12}s_{13}s_{23}}}{\sqrt{2}} \cdot \left(\frac{s_{12}}{2s_{13}} f_{212} + \frac{s_{23}}{2s_{12}} f_{332} + \frac{s_{13}}{2s_{23}} f_{311} + f_{312} \right) \\ \mathcal{M}_{gg \rightarrow gH}^{++-} &= \sqrt{\frac{s_{12}s_{23}}{s_{13}}} \cdot \frac{s_{12}}{2\sqrt{2}} \cdot f_{212} \end{aligned}$$

We can now express the normalized amplitude squared as sum of helicity amplitudes squared.

$$\frac{|\mathcal{M}_{gg \rightarrow gH}|^2}{|K|^2} = 2 \cdot \left(|\mathcal{M}_{gg \rightarrow gH}^{+++}|^2 + |\mathcal{M}_{gg \rightarrow gH}^{++-}|^2 + |\mathcal{M}_{gg \rightarrow gH}^{+-+}|^2 + |\mathcal{M}_{gg \rightarrow gH}^{-++}|^2 \right) \quad (\text{II.54a})$$

We have now established all the theoretical framework we will need to feed our example to pySecDec for numerical evaluation. We can head on over to the implementation part.

Implementation in pySecDec

All the files required for implementing our example in pySecDec can be found in the gggH1L folder at the Git-hub repository for this project : [PSD_gggH1L](https://github.com/CpSquared/PSD-gggH1L) ⁵

We need to numerically evaluate the form factors with the *sum_package*. We have the form factors as linear combination of integrals with corresponding coefficients. Therefore, first we need to define all the 14 loop integrals in pySecDec. We can define them using any of the pySecDec functions, but using the propagators is convenient, because then we will just have to change the powers of propagators and pySecDec will take care of the rest. Extra care should be taken for defining the crossed integrals, depending on the legs interchanged. The integrals are ordered according to the previously established order only. IntegralsF212.py and IntegralsF312.py are both identical files with same list of integrals defined, which will be used for the form factors F_{212} and F_{312} .

CoefficientsF212.py and CoefficientsF312.py files have the corresponding coefficients defined in order of the integrals. The coefficients are calculated in the mathematica notebooks and copied using the raw input form. The numerator and denominator of coefficients are provided separately to the pySecDec. Key thing to remember here is that, coefficients should only be in terms of our real parameters $s_{12}, s_{13}, s_{23}, m_t^2, m_H^2$ and ϵ . Therefore, first substitute $d = 4 - 2\epsilon$ before providing them to pySecDec. Mind the order of the real parameters provided, same should be maintained for generation and integration. If we change it during the integration, we will get the different form factors, which we will get to later.

Once the integral coefficient files are ready, we can call the *sum_package* through the corresponding generate files for each form factor. It will provide the form factors as input to pySecDec, which we can then compile. After compilation, only job left to do is to integrate at a given set of conditions and match the results.

integrate_gggH1L.py file will perform the integration at given values of the real parameters and return the coefficient of ϵ^0 term, for all the form factors. Because our Amplitude is non-divergent, coefficients of ϵ^{-1} and ϵ^{-2} are just zero and we ignore them. Now, for the form factors f_{212} and f_{312} , we should maintain the correct order of real parameters, throughout all the files, to get correct results. However, because the form factors f_{311} and f_{332} are related to f_{212} through in-variance of amplitude under permutation, what we do is, we integrate the f_{212} two times again, at different order of real parameters, corresponding to the definitions f_{311} and f_{332} . This gives us the numerical results for f_{311}, f_{332} and thus, we have numerical results for all of the form factors. We use these values and substitute them in the equations (II.48a),(II.54a) with $d \rightarrow 4$ to get the numerical results for the normalized amplitude squared. We can also evaluate the analytic result for the same in HTL, and verify our numerical results.

Important thing while providing the values of real parameters for evaluation is that, they all must be in the physical scattering regions. The physical scattering region is given by :

$$s_{12} > 0, s_{13} < 0, s_{23} < 0 \text{ \& } m_H^2 = s_{12} + s_{13} + s_{23}$$

We verify our results in the Heavy Top quark Limit(HTL), where $m_t \rightarrow \infty$ such that, $m_t^2 \gg |s_{ij}| \text{ \& } m_t^2 \gg m_H^2$. In practice, to avoid large numerical uncertainties, we keep $m_t^2 = 1$ and choose the other real parameters to be very small ($\ll 1$) such that the ratio of scales is still small, but not smaller than 10^{-4} , otherwise pySecDec breaks down. Refer to the appendix: B, for evaluation of integrals when large scale differences are present.

⁵<https://github.com/CpSquared/PSD-gggH1L>

Now we have all the files ready to execute, we can try to verify our results in the HTL. We will choose a set of conditions as follows and integrate to obtain results :

$$[s_{12}, s_{13}, s_{23}, m_t^2, m_H^2] = [0.0009, -0.0003, -0.00059842873775, 1.0, 1.57126225000000002e - 06]$$

The explicit conditions we use for the integration are :

Integrator = Qmc verbose=True epsrel=1e-4 epsabs=1e-14

Integrating the form factors ⁶ we Obtain :

```
Numerical Results For All Normalized Form Factors upto a phase factor :
Normalized F212 : 2228.05719411905375 - 8.28197625818457052e-7*I
Normalized F311 : -1481.48161841203819 + 1.00900601941613428e-6*I
Normalized F332 : 4444.44484616036971 - 1.8636243843809954e-6*I
Normalized F312 : 5191.0537518748215 + 0.0000153041235741549214*I
Analytic results of all Normalized Form Factors with heavy top quark limit
Analytic Result Normalized F212 : -2228.0569919594127
Analytic Result Normalized F311 : 1481.4814814814813
Analytic Result Normalized F332 : -4444.4444444444444
Analytic Result Normalized F312 : -5191.019954922376
```

Performing the amplitude calculation, we get following results :

```
Numerical Result of The Normalized Amplitude Squared with eq (II.48a)
|M_(gg- > gH)|^2 : 0.00217977663809607881

Numerical Result of The Normalized Amplitude Squared with eq (II.54a)
|M_(gg- > gH)|^2 : 0.00217977663809607898

Analytic Result of The Normalized Amplitude Squared
|M_(gg- > gH)|^2 : 0.0021797762426380867
```

We have successfully demonstrated the use of *sum-package* with pySecDec to numerically evaluate the 1-Loop amplitude for $gg \rightarrow gH$, by using the linear combination of master integrals. This will build the ground for the multi-loop calculations of crucial processes which will play an important role in the collider phenomenology after the high luminosity upgrade at the LHC .

I would like to formally express my gratitude towards Prof. Dr. Gudrun Heinrich for spending some of her valuable time to guide me on this project despite the current pandemic situation. This project, along with the collider phenomenology lectures, motivated me to pursue research in particle phenomenology. I would also like to thank the German Academic Exchange Service (DAAD) for offering the DAAD WISE Scholarship and supporting undergraduates at the research frontier.

⁶The *I represents multiplication with imaginary number i

References :

- [1] The HiggsTools Handbook: Concepts and observables for deciphering the Nature of the Higgs Sector : <https://arxiv.org/abs/1711.09875>
- [2] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, J. Schlenk, T. Zirke, pySecDec: A toolbox for the numerical evaluation of multi-scale integrals, *Comput.Phys.Comm.* 222 (2018), doi:10.1016/j.cpc.2017.09.015, arXiv:1703.09692
- [3] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, J. Schlenk, A GPU compatible quasi-Monte Carlo integrator interfaced to pySecDec, *Comput.Phys.Commun.* 240 (2019), doi:10.1016/j.cpc.2019.02.015, arXiv:1811.11720
- [4] A. G. Grozin, Integration by parts: An Introduction, *Int. J. Mod. Phys. A*26 (2011) 2807 [1104.3993].
- [5] G. Heinrich, Collider Physics at the Precision Frontier, *Phys. Rept.* 922 (2021) 1 [2009.00516].

Appendix : A

- T Coefficient expressions :

$$\begin{aligned}
T_1 &= \frac{8(d-4)(s_{12}^2 - s_{13}s_{23})}{s_{12}s_{23}(d-3)(d-2)(s_{12} + s_{23})(s_{12} + s_{13})} \\
T_2 &= \frac{-4(d-4)}{s_{12}s_{23}(d-3)(d-2)} \\
T_3 &= \frac{-4s_{13}((d-4)s_{12}^2 + 2ds_{12}s_{23} + ds_{23}^2)}{s_{12}^2s_{23}(d-3)(d-2)(s_{12} + s_{23})^2} \\
T_4 &= \frac{-4((d-4)s_{12}^2 + 2ds_{12}s_{13} + ds_{13}^2)}{s_{12}^2(d-3)(d-2)(s_{12} + s_{13})^2} \\
T_5 &= \frac{-(d-4)(s_{13}^2 + s_{23}^2)}{s_{13}s_{23}^2(d-3)(d-2)} \\
T_6 &= \frac{-s_{13}^2((d-4)s_{12}^2 + ds_{23}^2)}{s_{12}^3s_{23}^2(d-3)(d-2)} \\
T_7 &= \frac{-s_{23}((d-4)s_{12}^2 + ds_{13}^2)}{s_{12}^3s_{13}(d-3)(d-2)} \\
T_8 &= \frac{2(s_{13} + s_{23})((d-2)s_{12} - 8m_t^2)}{s_{12}^2s_{23}(d-2)} \\
T_9 &= \frac{2((d-2)s_{12}(s_{12}^2 - s_{23}^2) - 8m_t^2(s_{12}^2 - 2s_{12}s_{23} - s_{23}^2))}{s_{12}^2s_{23}(s_{12} + s_{23})(d-2)} \\
T_{10} &= \frac{2((d-2)s_{12}(s_{12}^2 - s_{13}^2) - 8m_t^2(s_{12}^2 - 2s_{12}s_{13} - s_{13}^2))}{s_{12}^2s_{23}(s_{12} + s_{13})(d-2)} \\
T_{11} &= \frac{(d-4)s_{12}s_{23} + (d-3)s_{12}s_{13} - 4m_t^2s_{13}}{s_{12}s_{13}(d-3)} \\
T_{12} &= \frac{s_{13}((d-4)s_{12}s_{13} + (d-3)s_{12}s_{23} - 4m_t^2s_{23})}{s_{12}s_{23}^2(d-3)} \\
T_{13} &= \frac{s_{13}(-(d-3)s_{12}^2 + ds_{13}s_{23} + 12m_t^2s_{12})}{s_{12}^3(d-3)}
\end{aligned}$$

- N Coefficient expressions :

$$\begin{aligned}
N_1 &= \frac{8(d-4)}{(d-3)(d-2)(s_{12}s_{13} + s_{13}s_{23})} \\
N_2 &= \frac{4((d-2)s_{13}^2 + 2ds_{13}s_{23} + (d-2)s_{23}^2)}{s_{13}s_{23}(d-3)(d-2)(s_{13} + s_{23})^2} \\
N_3 &= \frac{s_{12}(s_{13}^2 + s_{23}^2)}{s_{13}^2s_{23}^2(d-3)} \\
N_4 &= \frac{4((d-2)(s_{13} + s_{23}) - 8m_t^2)}{s_{12}(d-2)(s_{13} + s_{23})} \\
N_5 &= \frac{((d-3)s_{13}^2 - (d-2)s_{12}s_{23} - 4m_t^2s_{13})}{s_{13}^2(d-3)}
\end{aligned}$$

$$N_{i1} = N_i(s_{12} \rightarrow s_{13}, s_{13} \rightarrow s_{23}, s_{23} \rightarrow s_{12}) \quad N_{i2} = N_i(s_{12} \rightarrow s_{23}, s_{13} \rightarrow s_{12}, s_{23} \rightarrow s_{13})$$

where $i = \{1, 2, 3, 4, 5\}$

Appendix : B

- Evaluation of master integrals with expansion by regions :

Expansion by regions is a systematic method to express the feynman integral as a series expansion in terms of a ‘smallness parameter’, which arises if there are large scale differences present between kinematic invariants or the masses of the integral. While evaluating the integrals regularly, pySecDec cannot handle large scale differences, which is why this particular method is very useful in such cases. Therefore, to evaluate the integrals in heavy top limit, we will employ this method and try to verify our results with the analytic expressions from [1]. For a brief review of this method, refer to the section 3.2.7 of [5].

Because we are working in the HTL, the integrals are expanded in inverse powers of m_t^2 . The analytic expressions for the master integrals ⁷ can be calculated to be as follows :

$$\begin{aligned}
g_2(s_{12}, m_t^2) &= \frac{-(6m_t^2 + s_{12})}{12m_t^4} + \epsilon \cdot \left(\frac{\gamma_E(6m_t^2 + s_{12})}{12m_t^4} + \frac{-(s_{12} + (6m_t^2 + s_{12}) \cdot \ln \frac{1}{m_t^2})}{12m_t^4} \right) \\
g_5(s_{12}, m_t^2) &= \frac{-(12m_t^2 + s_{12})}{24m_t^4} + \epsilon \cdot \left(\frac{\gamma_E(12m_t^2 + s_{12})}{24m_t^4} + \frac{-(s_{12} + (12m_t^2 + s_{12}) \cdot \ln \frac{1}{m_t^2})}{24m_t^4} \right) \\
g_7(s_{12}, m_H^2, m_t^2) &= \frac{-(m_H^2 + 12m_t^2 + s_{12})}{24m_t^4} + \epsilon \cdot \left(\frac{\gamma_E(m_H^2 + 12m_t^2 + s_{12})}{24m_t^4} + \right. \\
&\quad \left. \frac{-(m_H^2 + s_{12} + (m_H^2 + 12m_t^2 + s_{12}) \cdot \ln \frac{1}{m_t^2})}{24m_t^4} \right) \\
g_9(s_{12}, s_{23}, m_H^2, m_t^2) &= \frac{1}{6m_t^4} + \epsilon \cdot \left(\frac{1 + \ln \frac{1}{m_t^2} - \gamma_E}{6m_t^4} \right)
\end{aligned}$$

where we have dropped the terms $\sim \epsilon \cdot \mathcal{O}(\frac{1}{m_t^6})$, $\sim \mathcal{O}(\frac{1}{m_t^6})$ & $\sim \mathcal{O}(\epsilon^2)$

All the files required for the evaluation of these integrals using expansion by regions with pySecDec can be found in the ‘Expansion by regions’ folder at the Git-hub repository for this Project : [PSD_gggH1L](#)

We can define the loop integrals in the same way as we did earlier, but in order to introduce the ‘smallness parameter’- z , we multiply all the in-variants and masses (except m_t) with z . We will also add z to the list of real parameters, and provide our loop integral to the `loop_regions` method, which will expand the integral in different regions. We can then proceed to sum over all regions using the `sum_package`.

However, the $g_5(s_{12}, m_t^2)$ integral must be treated carefully, as it needs an additional regulator to evaluate it. The same will be notified by pySecDec, if the integral needs an additional regulator. We introduce the extra regulator n_1 in the powers of the propagators, by consequently dividing it with the prime numbers. Therefore, the modified conditions for $g_5(s_{12}, m_t^2)$ should look like following :
(remaining conditions same as other integrals)

```

powerlist = ["1+n1", "1+ n1/2", "1+n1/3", "0+n1/5"]
regulators=["n1","eps"] requested_orders = [0,1]

```

⁷Because the $g_1(m_t^2)$ integral has only a single scale dependence, we cannot expand it in small ratio of scales, thus we do not consider it.

Now we have all the integrals ready to evaluate, we can try to verify our results with the analytic expression mentioned above, in the HTL. We will choose a set of conditions as follows and integrate to obtain results :

$$[s_{12}, s_{13}, s_{23}, m_t^2, m_H^2] = [0.00383, -0.000966, -0.000966, 1.0, 0.000157126225]$$

The explicit conditions we use for the Integration are : (modify for $g_5(s_{12}, m_t^2)$ as mentioned earlier)

```
expansion_by_regions_order = 1 requested_orders = [1] z = 1
Integrator = Qmc transform="korobov3" fitfunction="polysingular"
```

Integrating $g_2(s_{12}, m_t^2)$ we Obtain :

```
Analytic Result :
eps^0 : -0.5003191666666666
eps^1 : 0.2884728937838142
Numerical Result :
eps^0 : -0.50032-0.00000*I +/- 3.40961e-17+1.93492e-17*I
eps^1 : 0.28847-0.00000*I +/- 1.96825e-17+1.11695e-17*I
```

Similarly, we can calculate for the other integrals and verify our procedure. The output summary for all the integrals is given in the `Output_Summary_exp_by_regions.txt` file in the ‘Expansion by regions’ folder mentioned earlier.

We have successfully demonstrated the evaluation of master integrals using the method of expansion by regions in the heavy top limit with `pySecDec`.