TTP19-011
P3H-19-010

Double Higgs production in the high- and low-energy limits

Joshua Davies

Institut für Theoretische Teilchenphysik, Karlsruhe Institute of Technology (KIT), Wolfgang-Gaede Str. 1, 76128 Karlsruhe, Germany

E-mail: joshua.davies@kit.edu

Abstract. In this talk we discuss some of the computational aspects of some recent computations of double Higgs production in gluon fusion. We consider the challenges encountered in computing the high-energy limit of the NLO virtual corrections and the large top quark mass limit of the NNLO virtual corrections.

1. Introduction

A direct measurement of the triple–Higgs boson coupling will be a key task in the coming years, particularly once the High Luminosity LHC is running. To that end, there has been much activity in the theory community to prepare predictions for relevant cross sections, the most important of which is double Higgs boson production in gluon fusion. Leading order (LO) results, exact in the top quark mass, have been known for many years [1, 2]. Since then, results have been computed numerically at next-to-leading order (NLO) [3, 4, 5], and in various limits at both NLO [6, 7, 8, 9, 10, 11] and NNLO [7, 12, 13, 14]. These limits can be used to form various approximations of the cross section over the whole kinematic range, see for e.g. [15, 16].

In these proceedings we summarize a talk presented at the 19th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2019), based on Refs. [10, 11, 17]. We consider an NLO computation of the high-energy limit of the virtual contribution to double Higgs boson production in gluon fusion, that is for $s, |t| \gg m_t^2 > m_H^2$. This limit was previously unknown in the literature and provides information about the tails of kinematic distributions. We also consider the large-m_t limit at NNLO. The first three terms of this expansion were already known; we discuss here the challenges involved in computing a further two expansion terms.

We now briefly outline the notation used in this paper, but refer the reader to Refs. [10, 11] for further details. The amplitude for the process $g(q_1)g(q_2) \to H(q_3)H(q_4)$ is given by

$$\mathcal{M}^{ab} = \varepsilon_{1,\mu}\varepsilon_{2,\nu}\mathcal{M}^{\mu\nu,ab} = \varepsilon_{1,\mu}\varepsilon_{2,\nu}\delta^{ab}\left(\mathcal{M}_1 A_1^{\mu\nu} + \mathcal{M}_2 A_2^{\mu\nu}\right), \tag{1}$$

where a, b are adjoint colour indices. The Lorentz-scalar coefficients $\mathcal{M}_1, \mathcal{M}_2$ of the structures $A_1^{\mu\nu}, A_2^{\mu\nu}$ may be projected from $\mathcal{M}^{\mu\nu}$ by using projectors $P_{1,\mu\nu}, P_{2,\mu\nu}$ which we do not typeset

here. We write them as follows,

$$\mathcal{M}_1 = X_0\, s\, \left(\frac{3m_H^2}{s-m_H^2} F_{\rm tri} + F_{\rm box1}\right), \quad \mathcal{M}_2 = X_0\, s\, F_{\rm box2}\,, \quad \text{where } X_0 = \frac{G_F\, \alpha_s(\mu)}{\sqrt{2}\ 4\pi}. \tag{2}$$

We call the objects F_X, $X \in \{\text{tri,box1,box2}\}$, "form factors"; it is the computation of these form factors which we discuss in these proceedings. We define the coefficients of their expansion in the strong coupling $\alpha_s(\mu)$ as $F_X = \sum_{i\geq 0} (\alpha_s(\mu)/\pi)^i\, F_X^{(i)}$. As usual we define the Mandelstam variables $s = (q_1+q_2)^2$, $t = (q_1+q_3)^2$ and $u = (q_2+q_3)^2$, and m_H denotes the mass of the Higgs boson.

2. High-energy limit of the NLO virtual corrections

In this section we summarize the methodology of Refs. [10, 11]. We assume the hierarchy $s, |t| \gg m_t > m_H$ (where m_t is the top quark mass) and perform a Taylor expansion for $m_H \to 0$ and an asymptotic expansion for $m_t \to 0$ in order to describe the amplitude in this kinematic region. The procedure is as follows,

(i) Use `qgraf` [18] to generate 8 one-loop and 118 two-loop Feynman diagrams. `q2e/exp` [19] is used to map the diagrams onto integral topologies and produce, using `FORM 4.2` [20], an amplitude for non-zero values of m_H and m_t; it has the form of a linear combination of Lorentz-scalar Feynman integrals, with coefficients which are functions of s, t, m_t, m_H.

(ii) Taylor expand the amplitude for $m_H \to 0$, to $\mathcal{O}(m_H^2)$. The Feynman integrals are expanded by `LiteRed` [21, 22] to the same order.

(iii) The (now m_H-independent) Feynman integrals can be reduced to master integrals using `FIRE 5.2` [23]. We obtain 10 master integrals at LO and 161 master integrals at NLO, 30 of which are non-planar.

(iv) Compute the master integrals as an asymptotic expansion for $m_t \to 0$. We compute a system of differential equations for the master integrals w.r.t. m_t^2. By assuming an ansatz for the master integrals of the form (ϵ is the usual dimensional regularization parameter)

$$J = \sum_i \sum_j \sum_k C_{ijk}(s,t)\, \epsilon^i\, (m_t^2)^j\, \log^k(m_t^2), \tag{3}$$

the system of differential equations reduces to a system of linear equations for the coefficients $C_{ijk}(s,t)$. By providing the leading terms of this expansion as boundary conditions (see Ref. [24] for details) the system of equations can be solved to an arbitrarily high expansion depth in m_t^2.

After these steps, the expansions of the master integrals can be substituted into the amplitude to produce results for the form factors. We compute the master integrals to a sufficient depth in m_t^2 to provide the amplitude to $\mathcal{O}(m_t^{32})$. We refer the reader to [10, 11] for analysis of the results.

The most difficult step here, computationally, is step (iii). Despite having removed a scale (m_H^2) from the problem (by Taylor expansion), the reduction of two-loop $2 \to 2$ integrals depending on four variables ($d = 4-2\epsilon, s, t, m_t^2$) is not trivial. The leading term of the Taylor expansion produces approximately 26,000 Feynman integrals to be reduced. Around 120,000 additional integrals are produced, when we consider the $\mathcal{O}(m_H^2)$ terms of the Taylor expansion, as well as the differential equations of step (iv). The reduction of the most difficult topologies takes about 2 weeks on powerful machines with hundreds of gigabytes of memory. Based on studies at LO, we determined the $\mathcal{O}(m_H^4)$ corrections to not be worth the computational investment.

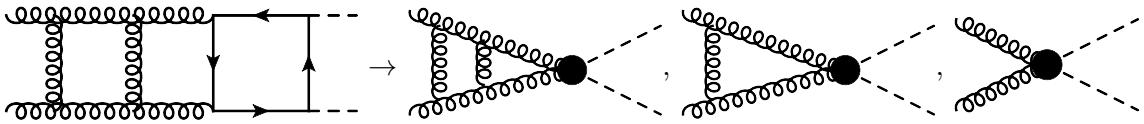


Figure 1. Asymptotic expansion for $m_t \to \infty$ produces massless graphs multiplied by large sums of massive tensor vacuum integrals, represented by the ● vertices.

Another aspect to consider here is how to efficiently apply the results of the IBP reduction to the (rather large) amplitude and differential equations. We convert the `FIRE` tables into a `FORM` `TableBase` so that we can apply the relations in `FORM`. Extensive use of `FORM`'s `PolyRatFun` functionality is necessary to simplify the resulting expressions.

3. Large-m_t limit of the NNLO virtual corrections

In this section we move on to the discussion of the low-energy limit at NNLO. We produce an expansion for the limit $m_t \to \infty$, that is, in powers of $(1/m_t^2)$. The leading order of this expansion ($(1/m_t^2)^0$) was computed in Ref. [12]. In Ref. [14] the first three orders ($(1/m_t^2)^{0,1,2}$) were computed, but presented for the differential cross section rather than the individual form factors.

Here we discuss a computation of the leading five expansion terms ($(1/m_t^2)^{0,\ldots,4}$) of the form factors. The procedure here is in principle rather straightforward. We first generate the 5,703 three-loop diagrams with `qgraf`. `q2e/exp` is then used match the diagrams to integral families and perform an asymptotic expansion for $m_t \gg q_1, q_2, q_3$. As a result, the diagrams become linear combinations of massless graphs (of at most two loops) multiplied by massive vacuum tensor graphs (of at most three loops), as depicted in Fig. 1.

The massive three-loop tensor vacuum integrals can be treated by `MATAD` [25]. In practice, however, these become a large performance bottleneck for deeper expansion depths. For diagrams involving three-loop tensor vacuum integrals, we proceed here with an alternative method to avoid having to compute tensor integrals. The expansion produces terms which are combinations of the following ratios, $\{q_3\cdot q_3,\ q_1\cdot q_2,\ q_1\cdot q_3,\ q_2\cdot q_3\}/m_t^2$. We can thus avoid the computation of tensor vacuum integrals by projecting onto a basis

$$B = \sum_{L=0}^{L_{max}} \sum_{k+l+m+n}^{=L} C_{i,j,k,l}\ (q_3\cdot q_3)^k\,(q_1\cdot q_2)^l\,(q_1\cdot q_3)^m\,(q_2\cdot q_3)^n \tag{4}$$

using projectors $P_{k,l,m,n}$ such that $P_{k,l,m,n}B = C_{k,l,m,n}$. L_{max} depends on the expansion depth in $1/m_t^2$. The projectors are defined in terms of derivative operators w.r.t. the momenta q_1, q_2, q_3: $\Box_{a,b} = \frac{\partial}{\partial q_{a\,\mu}}\frac{\partial}{\partial q_b{}^{\mu}}$. For example,

$$C_{0,0,0,2} = P_{0,0,0,2}\,B = \left[\frac{1}{2d^2+2d-4}\Box_{2,3}\Box_{2,3} - \frac{1}{2d^3+2d^2-4d}\Box_{2,2}\Box_{3,3}\right] B. \tag{5}$$

It is understood that after taking the derivatives, all remaining momenta in the diagram should be set to zero. Applying these projectors to the diagrams yields only *scalar* massive vacuum integrals, which can be treated much more easily by `MATAD`. This is a generalization of the method presented in Ref. [26].

We organise the computation as follows,

(i) Sum all diagrams with the same colour factor. The 5,703 three-loop diagrams become 9 "super-diagrams". Although each super-diagram is large, many terms common to multiple diagrams merge which reduces unnecessary duplication of effort in the following steps.

(ii) For each super-diagram, multiply by one of the five Lorentz structures of the form factor projectors described above Eq. (2). We thus obtain 45 "projected super-diagrams".

(iii) Expand each of these 45 objects to in $(1/m_t^2)^4$, and store these intermediate results to disk in a compressed form (we use `gzip` for this).

Step (iii) comprises of 45 difficult tasks, computationally. We perform them on relatively large machines (with at least 12 cores, 96GB memory). The five most difficult of these tasks each required a wall time of around 10 days. That the intermediate results are compressed is important; they occupy 324GB and can not fit on the available network storage uncompressed. We can then proceed with the final step,

(iv) As separate tasks, we apply each of the derivatives of the projectors defined below Eq. (4). From the stored intermediate results, the tasks load *only* the part which will yield a non-zero result after taking the derivatives; the intermediate results are stored in a manner which facilitates this. The resulting scalar vacuum integrals are computed by `MATAD`.

The projectors required by the first five expansion orders in $(1/m_t^2)$ contain {15, 38, 88, 174, 324} derivatives which must be computed. Since after step (ii) we have 45 objects to compute, step (iv) comprises of {675, 1,710, 3,960, 7,830, 14,580} tasks. Fortunately each task is relatively simple and is run with just 4 cores and around 8GB of memory. Nonetheless, the combined wall time used by all tasks is about 1,600 days which corresponds to tasks running concurrently on our cluster for around 1 month.

3.1. Additional Optimizations

Despite having restructured the calculation as described above, some additional optimizations were necessary in order to compute the fourth and fifth terms of the expansion on the available resources. The first is the use of graph symmetries to reduce the number of terms in the intermediate expressions. Vacuum integrals are highly symmetric objects; one can often re-label the momenta of different graphs to merge them. By merging graphs into something like a "minimal set" the size of the expressions can be substantially reduced. Some examples of this re-labelling are shown in Tab. 1.

In step (iii) we perform the large-m_t expansion. At this point in the calculation, the `FORM` representation of the terms is something like

```
+ Den(l1,mt) * Den(l1+q1,mt) * ... * Den(l2-q3,mt) * ( many terms )
```

where the `Den` functions denote the propagators which have to be large-m_t expanded; they have the form $1/(m_t^2 - (l_1 + q_1)^2)$, etc. The bracketed "`many terms`" are coefficients which do not depend on m_t. There can be many thousands of these coefficients, so it is important to keep them bracketed away during the multi-`FORM`-module expansion routine. This keeps the number of terms in the expressions as small as possible, and avoids expanding the same product of `Den` functions multiple times. In `FORM` this is typically achieved with a construction like

```
Bracket Den;
.sort
CFunction f;
Collect f;
```

which keeps the bracket contents inside the `CFunction f`. While this does reduce the number of terms, it does not (greatly) reduce the overall size of the expressions. Since the expressions are large enough to require disk-based sorting, there is still a large performance bottleneck due

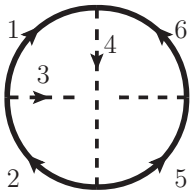
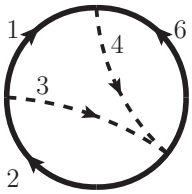
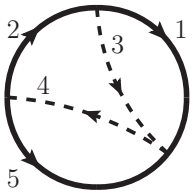
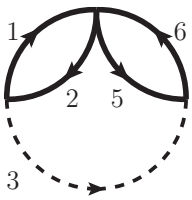
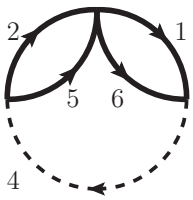
Top-level Topology	Graph 1	Graph 2	Re-labelling
			$p1 \to -p2$ $p2 \to -p1$ $p5 \to -p6$ $p6 \to -p5$ $p4 \to -p4$
			$p1 \to \ \ p5$ $p5 \to \ \ p1$ $p2 \to -p2$ $p6 \to -p6$ $p3 \to -p4$ $p4 \to -p3$

Table 1. The graphs in the "Graph 1" and "Graph 2" columns can be derived from the "Top-level Topology" by removing different lines. This yields equivalent graphs, but with the line momenta labelled differently. By changing the labelling, the equivalence is made manifest; we show example relabelling for the first and second row. The arrows on the lines denote the direction of momentum flow.

to repeated reading and writing of the expressions to and from the disk. We resolve this by additionally using the `FORM` statement `ArgToExtraSymbol f;`, which replaces the arguments of the `f`s with single symbols, the definitions of which are stored by `FORM`. More memory is required to store the symbol definitions, however the reduction in size of the expressions provides a large speed-up of the disk-based sorting in the expansion routine. After the expansion is finished, the original coefficients can be recovered from the symbols with the statement `FromPolynomial`. We additionally make use of this technique in certain expensive modules of `MATAD`'s routines for computing two-loop tensor vacuum integrals.

The results of these expansions are shown in Fig. 2. Below the threshold at $\sqrt{s} = 2m_t$, a reasonable convergence of the expansion can be observed. The reader is referred to Ref. [17] for an in-depth discussion.

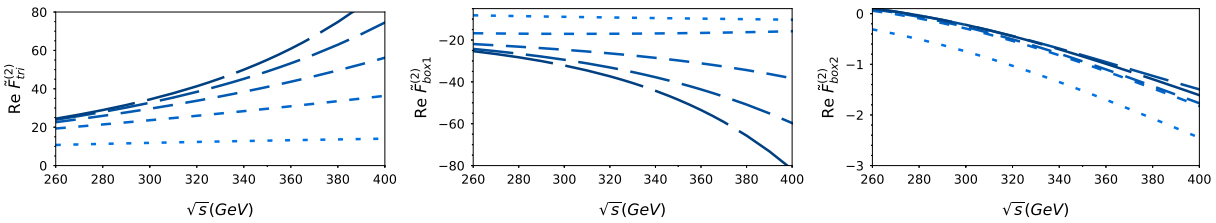


Figure 2. The real part of the three-loop form factors, plotted for $\mu^2 = -s$ and a fixed ratio $p_T^2/s = 0.1$, where $p_T^2 = (tu - m_H^4)/s$ is the transverse momentum of the Higgs bosons. The short- through long-dashed lines show the $(1/m_t^2)^0$ to $(1/m_t^2)^4$ expansions.

4. Conclusion

In these proceedings we have outlined the computational challenges encountered in an NLO computation of the high-energy limit of double Higgs boson production in gluon fusion, and a large-m_t expansion of the same amplitude at NNLO. We point out which steps, in particular, are challenging and how we overcome the bottlenecks to compute results which would otherwise be prohibitively expensive.

Acknowledgments

This research was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant 396021762 - TRR 257. The author thanks Go Mishima, Matthias Steinhauser and David Wellmann for collaboration.

References

[1] E. W. N. Glover and J. J. van der Bij, Nucl. Phys. B **309** (1988) 282.
[2] T. Plehn, M. Spira and P. M. Zerwas, Nucl. Phys. B **479** (1996) 46
Erratum: [Nucl. Phys. B **531** (1998) 655] [hep-ph/9603205].
[3] S. Borowka, N. Greiner, G. Heinrich, S. P. Jones, M. Kerner, J. Schlenk, U. Schubert and T. Zirke, Phys. Rev. Lett. **117** (2016) no.1, 012001
Erratum: [Phys. Rev. Lett. **117** (2016) no.7, 079901] [arXiv:1604.06447 [hep-ph]].
[4] S. Borowka, N. Greiner, G. Heinrich, S. P. Jones, M. Kerner, J. Schlenk and T. Zirke, JHEP **1610** (2016) 107 [arXiv:1608.04798 [hep-ph]].
[5] J. Baglio, F. Campanario, S. Glaus, M. Mühlleitner, M. Spira and J. Streicher, arXiv:1811.05692 [hep-ph].
[6] S. Dawson, S. Dittmaier and M. Spira, Phys. Rev. D **58** (1998) 115012 [hep-ph/9805244].
[7] J. Grigo, K. Melnikov and M. Steinhauser, Nucl. Phys. B **888** (2014) 17 [arXiv:1408.2422 [hep-ph]].
[8] G. Degrassi, P. P. Giardino and R. Gröber, Eur. Phys. J. C **76** (2016) no.7, 411 [arXiv:1603.00385 [hep-ph]].
[9] R. Bonciani, G. Degrassi, P. P. Giardino and R. Gröber, Phys. Rev. Lett. **121** (2018) no.16, 162003 [arXiv:1806.11564 [hep-ph]].
[10] J. Davies, G. Mishima, M. Steinhauser and D. Wellmann, JHEP **1803** (2018) 048 [arXiv:1801.09696 [hep-ph]].
[11] J. Davies, G. Mishima, M. Steinhauser and D. Wellmann, JHEP **1901** (2019) 176 [arXiv:1811.05489 [hep-ph]].
[12] D. de Florian and J. Mazzitelli, Phys. Lett. B **724** (2013) 306 [arXiv:1305.5206 [hep-ph]].
[13] D. de Florian and J. Mazzitelli, Phys. Rev. Lett. **111** (2013) 201801 [arXiv:1309.6594 [hep-ph]].
[14] J. Grigo, J. Hoff and M. Steinhauser, Nucl. Phys. B **900** (2015) 412 [arXiv:1508.00909 [hep-ph]].
[15] R. Gröber, A. Maier and T. Rauh, JHEP **1803** (2018) 020 [arXiv:1709.07799 [hep-ph]].
[16] M. Grazzini, G. Heinrich, S. Jones, S. Kallweit, M. Kerner, J. M. Lindert and J. Mazzitelli, JHEP **1805** (2018) 059 [arXiv:1803.02463 [hep-ph]].
[17] J. Davies and M. Steinhauser, in preparation.
[18] P. Nogueira, J. Comput. Phys. **105** (1993) 279.
[19] `http://sfb-tr9.ttp.kit.edu/software/html/q2eexp.html`.
[20] B. Ruijl, T. Ueda and J. Vermaseren, [arXiv:1707.06453 [hep-ph]].
[21] R. N. Lee, [arXiv:1212.2685 [hep-ph]].
[22] R. N. Lee, J. Phys. Conf. Ser. **523** (2014) 012059 [arXiv:1310.1145 [hep-ph]].
[23] A. V. Smirnov, Comput. Phys. Commun. **189** (2015) 182 [arXiv:1408.2372 [hep-ph]].
[24] G. Mishima, JHEP **1902** (2019) 080 [arXiv:1812.04373 [hep-ph]].
[25] M. Steinhauser, Comput. Phys. Commun. **134** (2001) 335 [hep-ph/0009029].
[26] J. Fleischer and O. V. Tarasov, Z. Phys. C bf 64 (1994) 413 [hep-ph/9403230].