

PAPER • OPEN ACCESS

## The $R^*$ -operation and combinatorial challenges at five loops

To cite this article: Ben Ruijl *et al* 2018 *J. Phys.: Conf. Ser.* **1085** 052006

View the [article online](#) for updates and enhancements.

You may also like

- [Error compensation in Brillouin optical correlation-domain reflectometry by combining bidirectionally measured frequency shift distributions](#)  
Guangtao Zhu, Kohei Noda, Heeyoung Lee et al.
- [Multi-robot Task Allocation for Search and Rescue Missions](#)  
Ahmed Hussein, Mohamed Adel, Mohamed Bakr et al.
- [Influence of the vibro-acoustic sensor position on cavitation detection in a Kaplan turbine](#)  
H Schmidt, O Kirschner, S Riedelbauch et al.



**ECS**  
The  
Electrochemical  
Society  
Advancing solid state &  
electrochemical science & technology

**DISCOVER**  
how sustainability  
intersects with  
electrochemistry & solid  
state science research

# The $R^*$ -operation and combinatorial challenges at five loops

Ben Ruijl<sup>1</sup>, Franz Herzog<sup>2</sup>, Takahiro Ueda<sup>2</sup>, J.A.M. Vermaseren<sup>2</sup>,  
Andreas Vogt<sup>3</sup>

<sup>1</sup> ETH Zurich, Wolfgang-Pauli-Str. 27, 8093 Zürich, Switzerland

<sup>2</sup> Nikhef Theory Group, Science Park 105, 1098 XG Amsterdam, NL

<sup>3</sup> Department of Mathematical Sciences, University of Liverpool, Liverpool L69 3BX, UK

E-mail: [benrl@nikhef.nl](mailto:benrl@nikhef.nl), [fherzog@nikhef.nl](mailto:fherzog@nikhef.nl), [tueda@nikhef.nl](mailto:tueda@nikhef.nl), [t68@nikhef.nl](mailto:t68@nikhef.nl),  
[Andreas.Vogt@liv.ac.uk](mailto:Andreas.Vogt@liv.ac.uk)

**Abstract.** We sketch how the  $R^*$ -operation can be used to compute the poles of Feynman diagrams. Next, we identify computational difficulties when performing five-loop calculations. We provide four solutions that drastically reduce the number of generated terms.

## 1. Introduction

Many physically interesting quantities can be computed from just the pole parts of Feynman diagrams. For example, anomalous dimensions (such as the beta function), splitting functions, and decay rates. Often, it is much easier to compute the pole parts of diagrams than the finite part.

In this work we sketch the  $R^*$ -operation [1, 2, 3], which is able to compute the pole parts of  $L$ -loop diagrams by a computation of at most  $L - 1$ -loop diagrams. Using a high-performance implementation of the  $R^*$ -operation combined with the FORCER program [4, 5, 6], we are able to compute the poles of five-loop massless propagator diagrams. Our code is all written in FORM [7, 8, 9].

Performing efficient computations of five-loop diagrams is hard, since the Feynman rules create many terms. To alleviate these issues, we describe four methods to reduce the number of terms: (1) removal of propagator insertions, (2) delaying of Feynman rule substitution, (3) canonicalization of Feynman diagrams, and (4) efficient tensor reduction. These optimizations are not specific to the  $R^*$ -method.

Using these optimizations, we have computed the five-loop beta function for Yang-Mills theory with fermions in six days on one 32-core machine [10]. We have also computed the  $R$ -ratio, the Higgs decay to quarks, and to gluons [11].

The outline of this paper is as follows. In section 2 we briefly describe the  $R^*$ -operation. In section 3 we describe the four optimizations. Finally, we present the conclusion in section 4.

## 2. The $R^*$ -operation

The  $R^*$ -operation can be used to compute the poles of Feynman diagrams [12, 13]. Recently, it has been extended to Feynman diagrams with arbitrary numerator structure [3]. In this section we briefly sketch how the  $R^*$ -operation works, focusing on UV-counterterms.



The basic object of the  $R^*$ -operation is the UV counterterm operation  $\Delta$  acting on a graph  $G$ , which is defined as the poles of  $G$  in the limit of all loop momenta going to infinity with all contributions from subdivergences subtracted.

Additionally, we define the pole operator  $K$  on a Laurent series in  $\varepsilon$  as

$$K \sum_{i=-\infty}^{\infty} c_i \varepsilon^i = \sum_{i=-\infty}^{-1} c_i \varepsilon^i. \tag{1}$$

Then the  $R^*$  operation for some simple examples yields:

$$K \text{ (circle with 1, 2) } = \Delta \left( \text{circle with 1, 2} \right) \tag{2}$$

$$K \text{ (circle with 1, 2, 3, 4) } = \Delta \left( \text{circle with 1, 2, 3, 4} \right) + \Delta \left( \text{circle with 2, 3} \right) \cdot \text{circle with 1, 4}. \tag{3}$$

In general, all sets of non-overlapping divergent subdiagrams have to be considered:

$$\begin{aligned} K \text{ (circle with 1, 2, 3, 4, 5, 6) } &= \Delta \left( \text{circle with 1, 2, 3, 4, 5, 6} \right) \\ &+ \Delta \left( \text{circle with 5, 6} \right) \cdot \text{circle with 1, 2, 3, 4} + \Delta \left( \text{circle with 2, 3} \right) \cdot \text{circle with 4, 5, 6, 1} \\ &- \Delta \left( \text{circle with 5, 6} \right) \Delta \left( \text{circle with 2, 3} \right) \cdot \text{circle with 1, 4}. \end{aligned} \tag{4}$$

In the latter case, the contribution from two counterterms gets a minus sign, to prevent double counting.

For logarithmically divergent diagrams,  $\Delta$  does not depend on external momenta or masses. Consequently, we can infrared rearrange (IRR) [14]:

$$\Delta \left( \text{circle with 1, 2, 3} \right) = \Delta \left( \text{circle with 1, 2} \right) = \Delta \left( \text{circle with 1, 3} \right). \tag{5}$$

Using IRR and the definition of  $\Delta$ , we can express the UV-counterterm of  $G$  in terms of simpler diagrams:

$$\Delta(G) \stackrel{\text{IRR}}{=} \underbrace{\Delta(G')}_{\text{Simpler than G}} = K(G') - \underbrace{\text{subdivergences}(G')}_{\text{Lower-loop diagrams}}$$

Using this setup, we can rewrite all  $L$ -loop input diagrams to  $L - 1$  scalar massless propagator integrals. For the applications at five loops, all those integrals can be computed using the Forcer program [4, 5].

### 3. Optimisations

Performing computations at five loops introduces at least four new bottlenecks compared to four loops: (1) the number of diagrams and their complexity grow exponentially, (2) the substitution of the Feynman rules is slow and creates millions of terms, (3) the number of counterterms grows exponentially, and (4) tensors of rank 10 have to be reduced, which involves solving large systems.

In this section we address these issues by presenting four optimisations, namely improved treatment of propagator insertions in section 3.1, delayed Feynman rule substitution in section 3.2, a canonical form algorithm for Feynman diagrams in section 3.3, and an efficient tensor reduction algorithm in section 3.4.

3.1. Treatment of propagator insertions

Many of the higher-loop corrections are self-energies of propagators in the diagram. Due to the local nature of the Feynman rules, these self-energies only depend on their external momentum (there are no contractions with other parts of the larger diagram), so they can be ‘factorised’ out:

$$\left( \text{diagram 1} + \text{diagram 2} + \text{diagram 3} \right) \times \text{diagram 4} = \Sigma_2^{1PR} \text{diagram 4} \tag{6}$$

where the  $L$ -loop self-energy is replaced by  $(p^2)^{-\varepsilon L}$  in the larger diagram (marked by  $L$  crosses). In a sense, the subdiagram is integrated out. The resulting simpler topology is multiplied by the one-particle-reducible  $L$ -loop self-energy. Since the  $L$ -loop subdiagram is of lower order, these quantities have already been computed and can easily be tabulated to prevent recomputations. For example, a five-loop diagram may contain the expensive 4-loop gluon propagator as a subdiagram.

For the  $R^*$ -operation, this representation has an issue: the non-integer power hides UV-divergent subdiagrams, which should be subtracted. However, since the exact contents of the  $(p^2)^{-\varepsilon L}$  is factorised out, we may replace it with *any*  $L$ -loop subdiagram. Therefore we choose the simplest configuration:  $L$  scalar one-loop bubbles side by side.

Thus, for the  $R^*$ -operation we can remove propagator insertions by using the following relation:

$$\text{diagram 1} = \frac{\text{diagram 2}}{L} \times \text{diagram 3} \tag{7}$$

3.2. Delayed Feynman rule substitution

Substituting the Feynman rules creates many terms. For example, the following fully gluonic five-loop graph creates 12 029 521 scalar integrals in the Feynman gauge:

$$\text{diagram 1} \tag{8}$$

The source of the blow-up is the Feynman rule for the triple gluon vertex, which can be written in the following way:

$$v_{3g}(p_1^{\mu,a}, p_2^{\nu,b}, p_3^{\rho,c}) = -if^{abc} [(p_1 - p_2)^\rho g_{\mu\nu} + (2p_2 + p_1)^\mu g_{\nu\rho} + (-2p_1 - p_2)^\nu g_{\mu\rho}] \tag{9}$$

Thus, for every vertex, six terms are created, of which some will evaluate to the same expression due to symmetries. For all these terms, expensive operations such as Taylor expansions and divergent subgraph recognitions have to be performed. However, these operations only depend on the momentum powers and are invariant under the way the momenta contract. So, we rewrite the triple gluon vertex in a way that exposes the momenta, but keeps all the contractions unsubstituted:

$$v_{3g}(p_1^{\mu,a}, p_2^{\nu,b}, p_3^{\rho,c}) = -if^{abc} p_1^\sigma t_3(\sigma, \nu, \rho, \mu) + if^{abc} p_2^\sigma t_3(\sigma, \mu, \rho, \nu), \quad (10)$$

where

$$t_3(\mu, \nu, \rho, \sigma) = g_{\mu\rho}g_{\nu\sigma} + g_{\mu\sigma}g_{\nu\rho} - 2g_{\mu\nu}g_{\rho\sigma}. \quad (11)$$

After rewriting  $v_{3g}$  in terms of  $t_3$ , there are only  $2^{10} = 1024$  terms for the Feynman diagram in eq. (8). We can keep our input in this compactified notation for as long as the actual contractions are not important, which is right until the tensor reduction.

We define the operation  $\circ$  that applies the remaining Feynman rules to all components of the  $R^*$ -operation. For example:

$$t_3(\mu, \nu, \rho, \sigma) \circ \Delta \left( \begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \right) = 2\Delta \left( \begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \right) - 2\Delta \left( \begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \right). \quad (12)$$

We stress that for this particular case contraction is *necessary*.

Similar rules can be devised for the other vertices and for the trace of gamma matrices. At five loops, the substitution of  $t_3$  and similar structures is an expensive part of the calculation, since the number of generated terms is high.

### 3.3. Canonical forms for Feynman diagrams

The  $R^*$ -operation applied to five-loop diagrams will create many counterterms. In order to reduce computation time, it is important to compute the counterterms of a specific graph only once. In turn, this requires an efficient way to detect if two graphs are equal. One straightforward option is to keep a list of all the graphs that have already been processed and test for isomorphisms on every element of the list until one is found. If no match is found, the current graph can be added to the list. The two downsides of this method are that (1) an isomorphism test can be rather slow at five loops and (2) that the list of topologies grows rapidly.

A better alternative is to construct a *canonical form* of a graph. A canonical form is an isomorphism of the graph that is designated as the smallest by some yet to be defined measure. To test for equality, one can simply compare the canonical forms. Since isomorphy is first and foremost a property of the vertices, we give each vertex a label from 1 to  $n$ . For simplicity, let us consider a graph that has no dot products and only has edges with power 1.

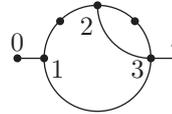
We convert our graph to an edge representation:

$$\begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} = e(0,1)e(1,2)e(2,3)e(2,3)e(1,3)e(3,4). \quad (13)$$

Here,  $e(n_1, n_2)$  is the edge function, in which we place the smallest vertex index as the first argument. The *edge list* is a lexicographically sorted list of edge functions, as is shown in eq. (13). Now we define the smallest isomorphism of a graph as the vertex labelling for which the edge list is lexicographically smallest.<sup>1</sup>

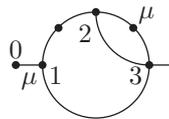
<sup>1</sup> In our program, we use the internal (deterministic) sorting order of FORM to determine the smallest isomorphism instead.

We can easily extend the graph notation to a graph where propagators can have different powers, by introducing a third argument to the edge function  $e$ :



$$= e(0, 1, 1)e(1, 2, 2)e(2, 3, 1)e(2, 3, 2)e(1, 3, 1)e(3, 4, 1), \quad (14)$$

where we again make sure that the first two arguments of  $e(n_1, n_2, \dots)$  are sorted. To add support for dot products and tensors, we extend the edge function even further:



$$= e(0, 1, 1, \mu)e(1, 2, 2)e(2, 3, 1)e(2, 3, 2, \mu)e(1, 3, 1)e(3, 4, 1). \quad (15)$$

We define the canonical signs of the momenta such that they always flow from the smallest vertex label to the highest. If a transformation changes the order, we flip the sign if the number of vectors in the momentum is odd:

$$e(2, 1, n, \mu_1, \dots, \mu_k) = (-1)^k e(1, 2, n, \mu_1, \dots, \mu_k). \quad (16)$$

Finally, the momentum label  $p_i$  of each edge is uniquely defined by the position  $i$  of the edge in the edge list.

Now that most properties of the Feynman integral are captured in the extended edge list and we have defined which edge list is smallest, we use McKay's canonicalisation algorithm [15] to efficiently rewrite the complete Feynman integral to canonical form. A simplified version of this algorithm is implemented in FORM code.

### 3.4. Efficient tensor reduction

It can be shown that the tensor reduction of ultraviolet and infrared subtraction terms, required for the  $R^*$ -operation, is equivalent to the tensor reduction of tensor vacuum bubble integrals. In general tensor vacuum integrals can be reduced to linear combinations of products of metric tensors  $g^{\mu\nu}$  whose coefficients are scalar vacuum integrals. Specifically a rank  $r$  tensor,  $T^{\mu_1 \dots \mu_r}$ , is written as a linear combination of  $n = r!/2^{(r/2)}/(r/2)!$  combinations of  $(r/2)$  metric tensors with coefficients  $c_\sigma$ , i.e.,

$$T^{\mu_1 \dots \mu_r} = \sum_{\sigma \in {}_2S_r} c_\sigma T_\sigma^{\mu_1 \dots \mu_r}, \quad T_\sigma^{\mu_1 \dots \mu_r} = g^{\mu_{\sigma(1)} \mu_{\sigma(2)}} \dots g^{\mu_{\sigma(r-1)} \mu_{\sigma(r)}}. \quad (17)$$

Here we define  ${}_2S_r$  as the group of permutations which do *not* leave the tensor  $T_\sigma^{\mu_1 \dots \mu_r}$  invariant. The coefficients  $c_\sigma$  can be obtained by acting onto the tensor  $T^{\mu_1 \dots \mu_r}$  with certain projectors  $P_\sigma^{\mu_1 \dots \mu_r}$ , such that

$$c_\sigma = P_\sigma^{\mu_1 \dots \mu_r} T_{\mu_1 \dots \mu_r}. \quad (18)$$

From this it follows that the orthogonality relation,

$$P_\sigma^{\mu_1 \dots \mu_r} T_{\tau, \mu_1 \dots \mu_r} = \delta_{\sigma\tau}, \quad (19)$$

must hold, where  $\delta$  is the Kronecker-delta. Since the projector  $P_\sigma^{\mu_1 \dots \mu_r}$  of each tensor can also be written in terms of a linear combination of products of metric tensors, inverting an  $n \times n$  matrix determines all the projectors. However, there are two issues. The first is that the size of the matrix grows rather rapidly as  $r$  increases. Instead of solving an  $n \times n$  linear system, the

symmetry group of the metric tensors can be utilised to reduce the size of the system. From eq. (19) it follows that the projector  $P_\sigma$  is in the same symmetry group (the group of permutations which leave it invariant) as  $T_\sigma$ . For example, given a permutation  $\sigma_1 = (123\dots(r-1)r)$ ,

$$T_{\sigma_1}^{\mu_1\dots\mu_r} = g^{\mu_1\mu_2} g^{\mu_3\mu_4} \dots g^{\mu_{r-1}\mu_r}. \quad (20)$$

The corresponding projector  $P_{\sigma_1}^{\mu_1\dots\mu_r}$  must be symmetric under interchanges of indices such as  $\mu_1 \leftrightarrow \mu_2$ ,  $(\mu_1, \mu_2) \leftrightarrow (\mu_3, \mu_4)$  and so on. Grouping the metric tensors by the symmetry leads to the fact that  $P_\sigma$  is actually written in a linear combination of a small number of  $m$  tensors instead of  $n$  ( $m \leq n$ ),

$$P_\sigma^{\mu_1\dots\mu_r} = \sum_{k=1}^m b_k \sum_{\tau \in A_m^\sigma} T_\tau^{\mu_1\dots\mu_r}. \quad (21)$$

The set of groups  $\{A_k^\sigma | k = 1..m\}$  must therefore each be closed under the permutations which leaves  $T_\sigma$  invariant and at the same time their union must cover once the group  ${}_2S_n$ . Contracting  $P_\sigma$  with a representative in each group gives an  $m \times m$  matrix which can be inverted to yield the coefficients  $b_k$ . The number of unknowns  $m$  is  $m = 5$  for  $r = 8$  and  $m = 22$  for  $r = 16$ , whereas we have  $n = 105$  for  $r = 8$  and  $n = 2027025$  for  $r = 16$ . The comparison of these numbers illustrates that the exploitation of the symmetry of the projectors makes it possible to find the tensor reduction even for very large values of  $r$ , which could never have been obtained by solving the  $n \times n$  matrix.

The second issue with tensors of high rank is the large number of intermediate terms that are created. Even though the system for the projector can be solved efficiently,  $\mathcal{O}(n^2)$  terms will be created, of which some will merge due to symmetry. Let us consider rank 6, with 15 terms:

$$c_1 g^{\mu_1\mu_2} g^{\mu_3\mu_4} g^{\mu_5\mu_6} + c_2 g^{\mu_1\mu_3} g^{\mu_2\mu_4} g^{\mu_5\mu_6} + \dots \quad (22)$$

In most practical situations there is symmetry, both on the inside of the object that will be projected as on the outside. For example

$$A(p_1^{\mu_1} p_1^{\mu_2} p_1^{\mu_3} p_1^{\mu_4} p_2^{\mu_5} p_2^{\mu_6}) p_3^{\mu_1} p_3^{\mu_2} p_4^{\mu_3} p_4^{\mu_4} p_4^{\mu_5} p_4^{\mu_6} \quad (23)$$

is symmetric in exchanges of  $\mu_1, \dots, \mu_4$  and  $\mu_5, \mu_6$  inside  $A$ , and is symmetric in  $\mu_1, \mu_2$  and  $\mu_3, \dots, \mu_6$  outside  $A$ . The symmetry inside the object  $A$  will enforce that coefficient  $c_1$  and  $c_2$  (and others) will actually be the same. The symmetry on the outside will cause terms to merge. In the end, we could have used the symmetrised variant of eq. (22) instead:

$$c_1 \cdot (g^{\mu_1\mu_2} g^{\mu_3\mu_4} g^{\mu_5\mu_6} + 2g^{\mu_1\mu_3} g^{\mu_2\mu_4} g^{\mu_5\mu_6}) + c_3 (2g^{\mu_1\mu_2} g^{\mu_3\mu_5} g^{\mu_4\mu_6} + 10g^{\mu_1\mu_5} g^{\mu_2\mu_6} g^{\mu_3\mu_4}). \quad (24)$$

We see that only two coefficients have to be computed instead of 15 and that there are only 4 terms in the output instead of 15. The challenge is to prevent these terms from being created in the first place by exploiting symmetry, instead of starting from eq. (22). We make use of the optimised FORM command `dd_`, which creates the tensor structure  $T^{\mu_1, \dots, \mu_r}$  without generating duplicates. If we evaluate `dd_(p1,p1,p1,p1,p2,p2)` and strip the coefficient we get `p1.p1~2*p2.p2+p1.p1*p1.p2~2`. These two terms represent the structure outside of  $c_1$  and  $c_3$  in (24). For each of these two terms, we solve for the coefficient. Next, we recreate the metric structures that would give this specific contraction.

A term generated by `dd_` consists of two different factors:  $(p \cdot p)^a$  and  $(p_1 \cdot p_2)^a$ . For  $(p \cdot p)^a$ , we collect all possible indices involved with  $p$ . For eq. (23), this would be  $\mu_1, \dots, \mu_4$ . Then we select all possible ways to get  $2a$  elements from that list with `distrib_`. Next, we use `dd_` on those indices. Thus, for  $p_1 \cdot p_1$  in the example we would get  $g^{\mu_1\mu_2} + g^{\mu_1\mu_3} + g^{\mu_2\mu_3}$ . For cases such

as  $(p_1 \cdot p_2)^a$ , we select  $a$  from the list of indices associated to  $p_1$  and  $a$  from the list of  $p_2$ . Then we permute over the list of  $p_2$ . Using this algorithm, one can generate all possible contractions from the result without generating duplicates. To apply the outside symmetry, one can easily fill in the outside momenta associated to the indices instead of the indices themselves. `distrib_` and `dd_` will take the symmetry into account automatically.

#### 4. Conclusion

We have sketched how the  $R^*$ -operation can be used to compute the poles of Feynman diagrams. Additionally, we have identified computational difficulties when performing five-loop calculations. We provide four solutions that reduce the number of generated terms.

Using these methods, we have computed the five-loop beta function for Yang-Mills theory with fermions in six days on one 32-core machine [10]. We have also computed the  $R$ -ratio, the Higgs decay to quarks, and to gluons [11].

#### Acknowledgements

This work is supported by the ERC Advanced Grant no. 320651, “HEPGAME”.

#### References

- [1] Chetyrkin K G and Tkachov F V 1982 *Phys. Lett.* **B114** 340–344
- [2] Chetyrkin K G and Smirnov V A 1984 *Phys. Lett.* **B144** 419–424
- [3] Herzog F and Ruijl B 2017 *JHEP* **2017** 37 ISSN 1029-8479 URL [http://dx.doi.org/10.1007/JHEP05\(2017\)037](http://dx.doi.org/10.1007/JHEP05(2017)037)
- [4] Ruijl B, Ueda T and Vermaseren J 2017 Forcer URL <https://doi.org/10.5281/zenodo.1043944>
- [5] Ruijl B, Ueda T and Vermaseren J A M 2017 (*Preprint* 1704.06650)
- [6] Ueda T, Ruijl B and Vermaseren J A M 2016 *PoS* **LL2016** 070 (*Preprint* 1607.07318)
- [7] Kuipers J, Ueda T, Vermaseren J A M and Vollinga J 2013 *Comput. Phys. Commun.* **184** 1453–1467 (*Preprint* 1203.6543)
- [8] Tentyukov M and Vermaseren J A M 2010 *Comput. Phys. Commun.* **181** 1419–1427 (*Preprint* hep-ph/0702279)
- [9] Ruijl B, Ueda T and Vermaseren J 2017 (*Preprint* 1707.06453)
- [10] Herzog F, Ruijl B, Ueda T, Vermaseren J A M and Vogt A 2017 *JHEP* **02** 090 (*Preprint* 1701.01404)
- [11] Herzog F, Ruijl B, Ueda T, Vermaseren J A M and Vogt A 2017 *JHEP* **08** 113 (*Preprint* 1707.01044)
- [12] Chetyrkin K G and Tkachov F V 1982 *Phys. Lett.* **B114** 340–344
- [13] Chetyrkin K G and Smirnov V A 1984 *Phys. Lett.* **B144** 419–424
- [14] Vladimirov A A 1980 *Theor. Math. Phys.* **43** 417 [Teor. Mat. Fiz.43,210(1980)]
- [15] McKay B D 1981 *Congressus Numerantium* **30** 45–87 URL <http://users.cecs.anu.edu.au/~bdm/nauty/pgi.pdf>