



You have downloaded a document from
RE-BUŚ
repository of the University of Silesia in Katowice

Title: Reductions and Contractions of 1-loop Tensor Feynman Integrals

Author: Ievgen Dubovyk, Janusz Gluza, Andrea A. Almasy, Tord Riemann

Citation style: Dubovyk Ievgen, Gluza Janusz, Almasy Andrea A., Riemann Tord. (2013). Reductions and Contractions of 1-loop Tensor Feynman Integrals. "Acta Physica Polonica B" (Vol. 44, no. 11 (2013), s. 2241-2247), doi 10.5506/APhysPolB.44.2241



Uznanie autorstwa - Licencja ta pozwala na kopiowanie, zmienianie, rozprowadzanie, przedstawianie i wykonywanie utworu jedynie pod warunkiem oznaczenia autorstwa.



UNIWERSYTET ŚLĄSKI
W KATOWICACH



Biblioteka
Uniwersytetu Śląskiego



Ministerstwo Nauki
i Szkolnictwa Wyższego

REDUCTIONS AND CONTRACTIONS
OF 1-LOOP TENSOR FEYNMAN INTEGRALS*

IEVGEN DUBOVYK, JANUSZ GLUZA

Institute of Physics, University of Silesia
Uniwersytecka 4, 40-007 Katowice, Poland

ANDREA A. ALMASY, TORD RIEMANN

Deutsches Elektronen-Synchrotron, DESY
Platanenallee 6, 15738 Zeuthen, Germany*(Received November 11, 2013)*

*Dedicated to the memory of our late Colleague and Friend
Jochem Fleischer (1937–2013). The present work is based on
a decade of common fruitful research, which would have been
impossible without his impetus, imagination and dedication*

We report on the progress in constructing contracted one-loop tensors. Analytic results for rank $R = 4$ tensors, cross-checked numerically, are presented for the first time.

DOI:10.5506/APhysPolB.44.2241

PACS numbers: 11.80.Cr, 12.38.Bx

1. Introduction

There are a few source-open programs for 5- and 6-point reductions: LoopTools/FF ($n \leq 5$, rank ≤ 4) — T. Hahn [1, 2], Golem95 — T. Binoth *et al.* [3], PJFry ($n \leq 5$, rank ≤ 5) — V. Yundin *et al.* [4, 5]. Some of these packages need, in addition, a library of scalar functions: 't Hooft, Veltman [6], LoopTools/FF, QCDloop/FF — K. Ellis and G. Zanderighi [2, 7] or OneLOop with complex masses — van Hameren [8]. In most cases, these packages suffice to calculate one-loop processes, however, there are at least two reasons for improvements. First, which is always desirable, speed

* Presented by I. Dubovyk and T. Riemann at the XXXVII International Conference of Theoretical Physics “Matter to the Deepest” Ustroń, Poland, September 1–6, 2013.

improvements. This is important when calculations are included into precise measurements using Monte Carlo methods. As already discussed in [9], available methods are at the edge of applications at low energy calculations. Second, not all of them are able to fulfil high demands concerning accuracy at the very specific kinematic points; an example has been shown in [10].

We are working on independent calculations for tensor contractions as an alternative to the PJFry reductions [11], based on work by Davydychev–Tarasov–Fleischer–Jegerlehner–Riemann–Yundin (DTFJRY) [12–16]. First numerical studies have been discussed in [17]. In this material, we report on analytic results for rank 4 and comment on improvements concerning the OLEC package [18].

2. Contractions for the 5-point functions with rank $R = 4$

We present here for the first time, results of contracted tensors of rank $R = 4$

$$I_5^{\mu\nu\lambda\rho} = I_5^{\mu\nu\lambda} \cdot Q_0^\rho - \sum_{s=1}^5 I_4^{\mu\nu\lambda,s} \cdot Q_s^\rho, \tag{1}$$

for lower rank results, see [15, 16]. After contraction with chords q (differences of external momenta), we get

$$q_{a,\mu}q_{b,\nu}q_{c,\lambda}q_{d,\rho} I_5^{\mu\nu\lambda\rho} = CE_{4,abcd} = -\frac{1}{2}CE_{3,abc}, Y_d + C_{5,abcd}. \tag{2}$$

Here,

$$Q_s^\rho = \sum_{i=1}^5 q_i^\rho \frac{\binom{s}{i}_5}{\binom{\rho}{5}}, \quad s = 0, \dots, 5. \tag{3}$$

The first term $q_{a,\mu}q_{b,\nu}q_{c,\lambda} I_5^{\mu\nu\lambda}$ is known [17], the second term has to be determined

$$C_{5,abcd} = - \sum_{s=1}^5 q_{a\mu}q_{b\nu}q_{c\lambda} I_4^{\mu\nu\lambda,s} \frac{1}{2} (\delta_{ds} - \delta_{5s}), \tag{4}$$

and it becomes

$$\begin{aligned} C_{5,abcd} = & \frac{1}{16} \left\{ G^5 + \delta_{ab}\delta_{ac}\delta_{ad}G^d - I_1^{5abc} - I_1^{5abd} - I_1^{5acd} - I_1^{5bcd} + I_1^{abcd} \right. \\ & - J_3^{a5} - J_3^{b5} - J_3^{c5} - J_3^{d5} + R^{5ab} + R^{5ac} + R^{5bc} + R^{5da} + R^{5db} + R^{5dc} \\ & \left. + \delta_{bc}\delta_{bd} \left(J_3^{ad} - J_3^{5d} \right) + \delta_{ac}\delta_{ad} \left(J_3^{bd} - J_3^{5d} \right) + \delta_{ab}\delta_{ad} \left(J_3^{cd} - J_3^{5d} \right) \right\} \end{aligned}$$

$$\begin{aligned}
 & + \delta_{ab}\delta_{ac} \left(J_3^{dc} - J_3^{5c} \right) + \delta_{ab}\delta_{cd}\tilde{J}_3^{db} + \delta_{ad}\delta_{bc}\tilde{J}_3^{dc} + \delta_{ac}\delta_{bd}\tilde{J}_3^{dc} \\
 & + \delta_{ab} \left(\tilde{J}_3^{5b} - R^{b5c} - R^{bd5} + R^{bcd} \right) + \delta_{ac} \left(\tilde{J}_3^{5c} - R^{c5b} - R^{cd5} + R^{cdb} \right) \\
 & + \delta_{ad} \left(\tilde{J}_3^{d5} - R^{d5b} - R^{d5c} + R^{dbc} \right) + \delta_{bc} \left(\tilde{J}_3^{5c} - R^{c5a} - R^{cd5} + R^{cda} \right) \\
 & + \delta_{bd} \left(\tilde{J}_3^{d5} - R^{d5a} - R^{d5c} + R^{dac} \right) + \delta_{cd} \left(\tilde{J}_3^{d5} - R^{d5a} - R^{d5b} + R^{dab} \right) \\
 & + \delta_{ab}\delta_{ad}J_4^d Y_c + \delta_{ac}\delta_{ad}J_4^d Y_b + \delta_{bc}\delta_{bd}J_4^d Y_a + \delta_{ab} \left(R^{bd} - R^{b5} \right) Y_c \\
 & + \delta_{ac} \left(R^{cd} - R^{c5} \right) Y_b + \delta_{ad} \left(R^{dc} - R^{d5} \right) Y_b + \delta_{ad} \left(R^{db} - R^{d5} \right) Y_c \\
 & + \delta_{bc} \left(R^{cd} - R^{c5} \right) Y_a + \delta_{bd} \left(R^{dc} - R^{d5} \right) Y_a + \delta_{bd} \left(R^{da} - R^{d5} \right) Y_c \\
 & + \delta_{cd} \left(R^{da} - R^{d5} \right) Y_b + \delta_{cd} \left(R^{db} - R^{d5} \right) Y_a + \left(I_4^d - I_4^5 \right) Y_a Y_b Y_c \\
 & + \left(I_3^{cd} - I_3^{5c} - I_3^{5d} + R^5 + \delta_{cd}R^d \right) Y_a Y_b \\
 & + \left(I_3^{bd} - I_3^{5b} - I_3^{5d} + R^5 + \delta_{bd}R^d \right) Y_a Y_c \\
 & + \left(I_3^{ad} - I_3^{5a} - I_3^{5d} + R^5 + \delta_{ad}R^d \right) Y_b Y_c \\
 & + \left(I_2^{bcd} - I_2^{5bc} - I_2^{5bd} - I_2^{5cd} - J_4^5 + R^{5b} + R^{5c} + R^{5d} \right) Y_a \\
 & + \left(I_2^{acd} - I_2^{5ac} - I_2^{5ad} - I_2^{5cd} - J_4^5 + R^{5a} + R^{5c} + R^{5d} \right) Y_b \\
 & + \left(I_2^{abd} - I_2^{5ab} - I_2^{5ad} - I_2^{5bd} - J_4^5 + R^{5a} + R^{5b} + R^{5d} \right) Y_c \}, \tag{5}
 \end{aligned}$$

where we have introduced

$$J_3^{st} \equiv \frac{1}{\binom{st}{st}_5} \left\{ -\binom{s}{s}_5 I_3^{[d+],st} + \binom{ts}{0s}_5 R^{ts} - \sum_{u=1}^5 \binom{ts}{us}_5 R^{tsu} \right\}, \tag{6}$$

$$\tilde{J}_3^{st} \equiv \frac{1}{\binom{st}{st}_5} \left\{ \binom{s}{t}_5 I_3^{[d+],st} + \binom{st}{0t}_5 R^{ts} - \sum_{u=1}^5 \binom{st}{ut}_5 R^{tsu} \right\}, \tag{7}$$

$$G^s \equiv \frac{1}{\binom{s}{s}_5} \left\{ -2\binom{}{}_5 R^{[d+],s} + \binom{s}{0}_5 J_4^s - \sum_{t=1}^5 \binom{s}{t}_5 J_3^{ts} \right\}. \tag{8}$$

J_4^s and $R^{[d+],s}$ are given in Eqs. (2.24) and (2.44) of [15], respectively. For further abbreviations, see (2.24), (2.49), (2.9), (2.17), (2.34), (2.41) of [15]. There also R^s, R^{st}, R^{tsu} are defined, $Y_a = Y_{a5} - Y_{55}, Y_{ab} = -(q_a - q_b)^2 + m_a^2 + m_b^2$.

Some numerical results for a 5-point function with rank $R = 4$ are added in [18]. For scalar functions, we use the `OneLOop` package and compare the results with `LoopTools/FF`. For the considered kinematic points, full agreement has been obtained. The kinematics is that of the process $e^-e^+ \rightarrow \mu^-\mu^+\gamma$. Using an MC generator, we have checked thousands of points up to rank three. All of them agreed between `OLEC` and `LT/OneLOop`. For ranks 3 and the new results presented here, we made checks also against the public version of `Golem95`, available at <http://golem.hepforge.org/95/>. At the `OLEC` webpage [18], we added two sets of files with the output. In set I, the kinematics is chosen such that 3-point functions hit the IR singularities, while in set II, the 3-point functions are slightly off the IR singularities. The `Golem95` results were different from `LT`, `OneLOop` and `OLEC`, starting already from rank 3¹.

For rank four, we give below such an example:

```
p1s = 1.1163688400000000E-002  p2s = 2.6109999999999998E-007
p3s = 0.0000000000000000      p4s = 2.6109999999999998E-007
p5s = 1.1163688400000000E-002
s12 = -0.70858278190000001    s23 = -1.5343299000000002E-003
s34 = -0.12851860429999998    s45 = -0.61023937949999996
s15 = 0.92668942420000000    m1s = 1.1163688361676107E-002
m2s = 0.00000000000000000    m3s = 2.6112003932088364E-007
m4s = 2.6112003932088364E-007 m5s = 0.00000000000000000
```

The R=4 contractions, a,b,c,d=3,3,3,3

```
OLEC: ( -48094.1074 54542318      , -47802.08746 5035322 )
LoopTools: ( -48094.1074 65      , -47802.08746 05      )
```

The R=4 contractions, a,b,c,d=3,3,3,4

```
OLEC: ( -18463.1204 24842149      , -23446.4704 12257226 )
LoopTools: ( -18463.1204 31      , -23446.4704 09      )
```

The R=4 contractions, a,b,c,d=3,3,3,5

```
OLEC: ( 0.00000000000000000      , 0.00000000000000000 )
LoopTools ( 0.00000000000000000      , 0.00000000000000000 )
```

The last result with $d = 5$ is a virtue of the construction of chords, where $g_5 = 0$.

3. OLEC package, Fortran code

The idea of external contractions has been implemented for the first time in the C++ code `OLEC` for tensors up to rank 3 [17], and basic examples

¹ The problem in `Golem95` v.1.2.1 has been settled in the meantime for the set I with changelog 128 (11 Oct, 2013), <https://golem.hepforge.org/trac/changeset/128>

are given at [18]. In the meantime, a Fortran code has been written with the aim of further optimization. The calculation of contracted tensor integrals consists of two basic steps. The first and most time consuming step is a preparation of building blocks for a calculation — basic scalar integrals ($\sim 50\%$) and the so-called signed minors ($\sim 25\%$). We cannot make much about scalar integral libraries, unless new independent developments appear. For some recent efforts in this direction, see [19]. However, the calculation of signed minors can be improved, both the cache system and the computation algorithm. The second step is *performing* of the contractions ($\sim 25\%$), which also can be improved. During the initialization procedure, scalar integrals and signed minors are calculated and stored in the RAM memory. For an efficient usage of memory, appropriate data structures are constructed. *E.g.* for 5-point kinematics, we have, in total, only 30 different basic scalar integrals, while for signed minors situation it is more complicated because all minors with up to 4 scratched rows and columns are needed. The “natural” storage model for such objects in the form of multidimensional arrays is not an optimal solution. For example, for minors of rank 3 (3 rows and columns are excluded), one has an array with 46656 elements but according to the minors’ symmetry properties, only 210 of them are different and non-zero. For the purpose of minimal memory usage, a linear storage model is chosen. In general, minors of a given rank are stored in the memory as one-dimensional arrays and access to them can be done according to the following pattern: `minorrank[AddressTable[func(i, j, k...)]]`, as it was first done in [11]. For a given minor indices i, j, k, \dots , a function ‘func’ based on bitwise operations calculates an address of an element from the constant table ‘AddressTable’. This table is the same for all minors and contains positions of values of minors in array ‘minor_{rank}’. Finally, the cache contains a few hundred of double precision numbers for all building blocks like minors, scalar integrals and auxiliary functions.

In the new version of the library, the computation algorithm for signed minors has also been changed. In practical computations of one-loop cross-sections, usually at least all contractions up to rank 3 are needed (like in QED). It means that minors of all ranks are needed as well. The procedure is iterative, cache is filled first from minors of the highest rank 4 (with 8 indices) which are used further to calculate minors of rank 3. And so on. All loops connected with minor indices during this procedure are unrolled and the highest level of optimization is applied for compilation (these are algebraical manipulations for which “-O3” optimization does not spoil double accuracy). Unrolling and usage of optimization for algebraic part of the package allows to decrease extremely the matrix algebra computation time.

4. Summary and outlook

In recent years, the strategy described has been developed in a bunch of papers for explicit analytical and recursive treatment of heptagon, hexagon and pentagon tensor integrals of rank R in terms of pentagons and boxes of rank $R - 1$. A systematic derivation of expressions which are explicitly free of inverse Gram determinants $(\)_5$ until pentagons of rank $R = 5$ has been worked out. The numerical package OLEC for contracted tensor integrals in C++ and Fortran [18, 20] are under development and tests.

Our preliminary benchmarks show that the OLEC library in the present form is faster than 5-point tensor reductions implemented in LoopTools/FF by about an order of magnitude (tested up to rank 3). Work in progress includes a correct treatment of small Gram determinant cases for the reductions (expansion in small parameters [11] or using hypergeometric representations [21]), adding rank 4 (this paper) and rank 5 contractions for 5-point functions, programming contracted tensors for 6- and 7-point functions.

Work supported by the European Initial Training Network LHCPHENOnet PITN-GA-2010-264564.

REFERENCES

- [1] T. Hahn, M. Perez-Victoria, *Comput. Phys. Commun.* **118**, 153 (1999) [arXiv:hep-ph/9807565].
- [2] G. van Oldenborgh, *Comput. Phys. Commun.* **66**, 1 (1991); <http://www.xs4all.nl/~gjvo/>
- [3] T. Binoth *et al.*, *Comput. Phys. Commun.* **180**, 2317 (2009) [arXiv:0810.0992 [hep-ph]].
- [4] V. Yundin, “Massive loop corrections for collider physics”, Ph.D. Thesis, Humboldt-Universität zu Berlin, 2012, see <http://edoc.hu-berlin.de/docviews/abstract.php?id=39163> and <http://urn:nbn:de:kobv:11-100199626>
- [5] J. Fleischer, T. Riemann, V. Yundin, in: [22], preprint DESY 11-252, 2011, <http://www-library.desy.de/cgi-bin/showprep.pl?desy11-252>
- [6] G. 't Hooft, M. Veltman, *Nucl. Phys.* **B153**, 365 (1979).
- [7] R.K. Ellis, G. Zanderighi, *J. High Energy Phys.* **02**, 002 (2008) [arXiv:0712.1851 [hep-ph]].
- [8] A. van Hameren, *Comput. Phys. Commun.* **182**, 2427 (2011) [arXiv:1007.4716 [hep-ph]].
- [9] M. Gunia, see transparencies of talk at this conference, <http://indico.if.us.edu.pl/event/us2013>

- [10] J. Gluza, M. Gunia, T. Riemann, M. Worek, *PoS RADCOR2011*, 034 (2011) [[arXiv:1201.0968](#) [[hep-ph](#)]].
- [11] V. Yundin, C++ package PJFry, available at <https://github.com/Vayu/PJFry/>
- [12] A.I. Davydychev, *Phys. Lett.* **B263**, 107 (1991).
- [13] O. Tarasov, *Phys. Rev.* **D54**, 6479 (1996) [[arXiv:hep-th/9606018](#)].
- [14] J. Fleischer, F. Jegerlehner, O. Tarasov, *Nucl. Phys.* **B566**, 423 (2000) [[arXiv:hep-ph/9907327](#)].
- [15] J. Fleischer, T. Riemann, *Phys. Rev.* **D83**, 073004 (2011) [[arXiv:1009.4436](#) [[hep-ph](#)]].
- [16] J. Fleischer, T. Riemann, *Phys. Lett.* **B701**, 646 (2011) [[arXiv:1104.4067](#) [[hep-ph](#)]].
- [17] J. Fleischer *et al.*, *PoS LL2012*, 051 (2012) [[arXiv:1211.3921](#) [[hep-ph](#)]].
- [18] A.A. Almasy *et al.*, OLEC project; C++ and Mathematica packages OLEC v.0.9, available at the Silesian Univ., Katowice, webpage <http://prac.us.edu.pl/~gluza/olec/>
- [19] J.P. Guillet, E. Pilon, M. Rodgers, M. Zidi, [arXiv:1310.4397](#) [[hep-ph](#)].
- [20] A.A. Almasy, J. Fleischer, J. Gluza, T. Riemann, publication in preparation; A.A. Almasy, “Fortran package for rank 5 tensor integrals”, available on request.
- [21] J. Fleischer, F. Jegerlehner, O. Tarasov, *Nucl. Phys.* **B672**, 303 (2003) [[arXiv:hep-ph/0307113](#)].
- [22] J. Alcaraz Maestre *et al.*, Report of the SM and NLO Multileg and SM MC Working Groups for the Workshop “Physics at TeV Colliders”, Les Houches, France, 31 May–8 June, 2011, see [arXiv:1203.1488](#) [[hep-ph](#)] and [arXiv:1203.6803](#) [[hep-ph](#)].