DESY 11-252 **23 Dec 2011**

BI-TP 2011/51

PJFry – a C++ package for tensor reduction of one-loop Feynman integrals *

J. Fleischer[1], *T. Riemann*[2], *V. Yundin*[3]
[1]Fakultät für Physik, Universität Bielefeld, Universitätsstr. 25, 33615 Bielefeld, Germany
[2]Deutsches Elektronen-Synchrotron, DESY, Platanenallee 6, 15738 Zeuthen, Germany
[3]Niels Bohr International Academy and Discovery Center, Niels Bohr Institute, University of Copenhagen, Blegdamsvej 17, DK-2100, Copenhagen, Denmark

Abstract
The C++ package PJFry 1.0.0 [1, 2] – a one loop tensor integral library – is introduced. We use an algebraic approach to tensor reduction. As a result, the tensor integrals are presented in terms of scalar one- to four-point functions, which have to be provided by an external library, e.g. QCDLoop/FF or OneLOop or LoopTools/FF. The reduction is implemented until five-point functions of rank five. A numerical example is shown, including a special treatment for small or vanishing inverse four-point Gram determinants. Future modules of PJFry might cover the treatment of n-point functions with $n \geq 6$; the corresponding formulae are worked out. Further, an extremely efficient approach to tensor reduction relies on evaluations of complete contractions of the tensor integrals with external momenta. For this, we worked out an algorithm for the analytical evaluation of sums over products of signed minors with scalar products of chords, i.e. differences of external momenta. As a result, the usual multiple sums over tensor coefficients are replaced for the numerical evaluation by compact combinations of the basic scalar functions.

1. PJFry

The goal of the C++ package PJFry is a stable and fast open-source implementation of one-loop tensor reduction of Feynman integrals

$$I_n^{\mu_1\cdots\mu_R} \quad = \quad C(\epsilon) \int \frac{d^dk}{i\pi^{d/2}} \; \frac{\prod_{r=1}^{R} k^{\mu_r}}{\prod_{j=1}^{n}(k-q_j)^2 - m_j^2 + i\epsilon}, \tag{1}$$

suitable for any physically relevant kinematics.[1] The algorithm was invented in [2]. PJFry performs the reduction of 5-point 1-loop tensor integrals up to rank 5. The 4- and 3-point tensor integrals are obtained as a by-product. Main features are:

- Any combination of internal or external masses
- Automatic selection of optimal formula for each coefficient
- Leading $()_5$ are eliminated in the reduction
- Small $()_4$ are avoided using asymptotic expansions where appropriate
- Cache system for tensor coefficients and signed minors

*To appear in the proceedings of the 7^{th} "Workshop on Physics at TeV Colliders", 30 May to 17 June 2011, Les Houches, France

[1]An extended description of notations and of the formalism may be found in [2, 3, 4, 5]. The normalization of PJFry follows that chosen in the scalar library. For QCDLoop, $C(\epsilon) = \Gamma(1-2\epsilon)/[\Gamma(1+\epsilon)\Gamma^2(1-\epsilon)]$.

- Interfaces for C, C++, FORTRAN and Mathematica
- Uses QCDLoop [6, 7] or OneLOop [8] for 4-dim scalar integrals
- Available from the project webpage `https://github.com/Vayu/PJFry/` [1, 2]

The installation of PJFry may be performed following the instructions given at the project webpage.
The project subdirectories are
./src - the library source code
./mlink - the MathLink interface
./examples - the FORTRAN examples of library use, built with make check

A build on Unix/Linux and similar systems is done in a standard way by sequential performing ./configure, make, make install. See the INSTALL file for a detailed description of the ./configure options.

The functions for tensor coefficients for up to rank $R = 5$ pentagon integrals are declared in the Mathematica interface:

```
In:= Names["PJFry`*"]

Out= {A0v0, B0v0, B0v1, B0v2, C0v0, C0v1, C0v2, C0v3, \
D0v0, D0v1, D0v2, D0v3, D0v4, E0v0, E0v1, E0v2, \
E0v3, E0v4, E0v5, GetMu2, SetMu2}
```

The C++ and Fortran interface syntax is very close to that of e.g. LoopTools/FF:

```
E0v3[i,j,k,p1s,p2s,p3s,p4s,p5s,s12,s23,s34,s45,s15,m1s,m2s,m3s,m4s,m5s,ep=0]
```

where:[2]
`i,j,k` are indices of the tensor coefficient $(0 < i \leq j \leq k < n)$,
`p1s,p2s,...` are squared external masses p_i^2,
`s12,s23,...` are Mandelstam invariants $(p_i + p_j)^2$,
`m1s,m2s,...` are squared internal masses m_i^2,
`ep=0,-1,-2` selects the coefficient of the ϵ-expansion.

The average evaluation time per phase-space point on a 2 GHz Core 2 laptop for the evaluation of all 81 rank 5 tensor form-factors amounts to 2 ms.

A numerical example is shown, for a configuration as in figure 1, in figures 2 and 3 for a five-point rank $R = 4$ tensor coefficient in a region, where one of the 4-point sub-Gram determinants vanishes [at $x = 0$]:

$$E_{3333}(0, 0, -6\times10^4(x+1), 0, 0, 10^4, -3.5\times10^4, 2\times10^4, -4\times10^4, 1.5\times10^4, 0, 6550, 0, 0, 8315)$$

The red curve is produced with standard PJFry, and the blue one with Passarino-Veltman [PV] reduction [9]; we mention that for the case treated here ($x \to 0$), the PV reduction is no standard option. Our expansion in terms of higher dimensional scalar 3-point functions in case of vanishing 4-point sub-Gram determinants uses only functions I_3^{d+2l} [2]. These are tensor coefficients of the pure $g^{\mu\nu}$ type [10], and so our method is different from others with a mixed numerical approach [11] or with use of additional tensor coefficients [12].

Tensor reduction by PJFry is used as one option of the GoSam package [13]. An older version of the algorithm, as described in [14], has been implemented independently in [15].

[2]One has to carefully control accuracies; e.g. the on-shell conditions for massless particles have to be fulfilled with a numerical precision expected by the scalar functions library in use; for QCDLoop this means on default at least 10 digits.

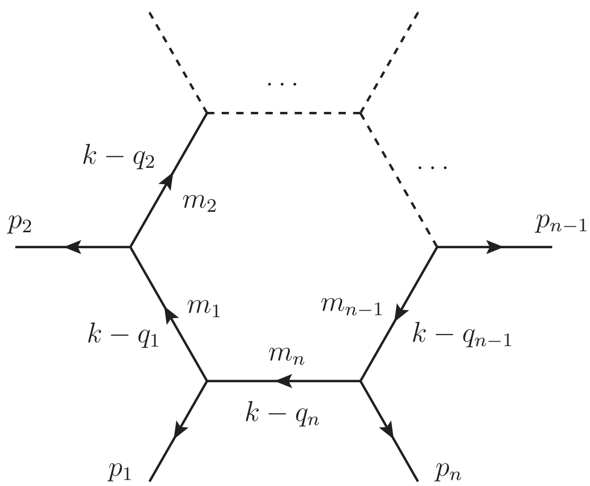

Fig. 1: Momenta definitions for PJFry.

2. POTENTIAL UPGRADES

2.1 Tensor reduction for higher-point functions

So far, PJFry is foreseen for 5-point functions and simpler ones. The extension to 6-point functions is known from e.g. [11, 12, 14]. In [4] we solve analytically generalized recursions for $n \geq 6$, derived in [11]:

$$I_n^{\mu_1\mu_2\ldots\mu_R} = -\sum_{r=1}^{n} C_r^{\mu_1}(n) I_{n-1}^{\mu_2\cdots\mu_R,r}, \tag{2}$$

where in $I_{n-1}^{\mu,\cdots,r}$ the line r is scratched. The coefficients for 6-point functions are:

$$C_r^{s,\mu}(6) = \sum_{i=1}^{5} \frac{1}{\binom{0}{s}_6} \binom{0r}{si}_6 q_i^{\mu_1}, \quad s = 0\ldots 6, \tag{3}$$

where the q_i are chords, and $\binom{0r}{si}_6$ etc. are signed minors with arbitrary s. For the 7-point and 8-point functions, we found several representations, among them

$$C_r^{st,\mu}(7) = \sum_{i=1}^{6} \frac{1}{\binom{st}{st}_7} \binom{sti}{str}_7 q_i^{\mu} \tag{4}$$

and

$$C_r^{stu,\mu}(8) = \sum_{i=1}^{7} \frac{1}{\binom{stu}{stu}_8} \binom{stui}{stur}_8 q_i^{\mu} \tag{5}$$

The upper indices s, t and u stand for the redundancy of the solutions and can be freely chosen.

2.2 Evaluation of contracted tensor integrals using sums over signed minors

The contraction of a tensor integral with chords may be written as a sum over basic scalar integrals (at a stage where they are free of tensor coefficient indices), multiplied by (multiple) sums over chords times

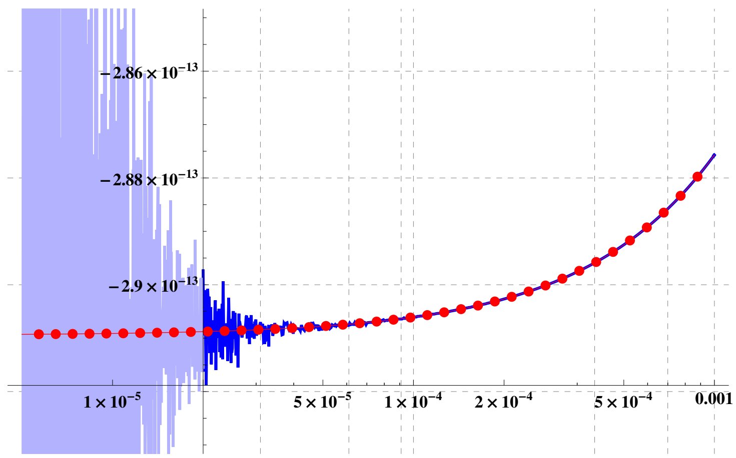

Fig. 2: Absolute accuracy of E_{3333} in the region of vanishing sub-Gram determinant. Blue curve: conventional Passarino-Veltman reduction, red curve: PJFry.

signed minors. If one may perform these sums algebraically, the method becomes very efficient. And this has been systematically worked out in [3], see also [5].

We reproduce here two 7-point examples.

The rank $R = 2, 3$ integrals become by contraction

$$q_{a,\mu} q_{b,\nu}\, I_7^{\mu\nu} = \sum_{r,t=1}^{7} K^{ab,rt} I_5^{rt}, \tag{6}$$

$$q_{a,\mu} q_{b,\nu} q_{c,\lambda}\, I_7^{\mu\nu\lambda} = \sum_{r,t,u=1}^{7} K^{abc,rtu} I_4^{rtu}, \tag{7}$$

where I_5^{rt} and I_4^{rtu} are scalar 5- and 4-point functions, arising from the 7-point function by scratching lines $r, t, \ldots$ In the general case, we have at this stage higher-dimensional integrals $I_n^{d+2l}, n = 2, \ldots, 5$, to be further reduced following the known scheme, if needed. Here, the I_5^{rt} have to be expressed by 4-point functions.

The expansion coefficients are factorizing here,

$$K^{ab,rt} = K^{a,r} K^{b,rt}, \tag{8}$$

$$K^{abc,rtu} = -K^{a,r} K^{b,rt} K^{c,rtu}, \tag{9}$$

and the sums over signed minors have been performed analytically:

$$K^{a,r} = \frac{1}{2}\left(\delta_{ar} - \delta_{7r}\right), \tag{10}$$

$$K^{b,rt} = \sum_{j=1}^{6} (q_b q_j) \frac{\binom{rst}{rsj}_7}{\binom{rs}{rs}_7} \equiv \frac{\Sigma_b^{1,stu}}{\binom{rs}{rs}_7} = \frac{1}{2}\left(\delta_{bt} - \delta_{7t}\right) - \frac{1}{2}\frac{\binom{rs}{ts}_7}{\binom{rs}{rs}_7}\left(\delta_{br} - \delta_{7r}\right), \tag{11}$$

$$K^{a,stu} = \sum_{i=1}^{6} (q_a q_i) \begin{pmatrix} 0stu \\ 0sti \end{pmatrix}_7 \equiv \Sigma_a^{2,stu} \tag{12}$$

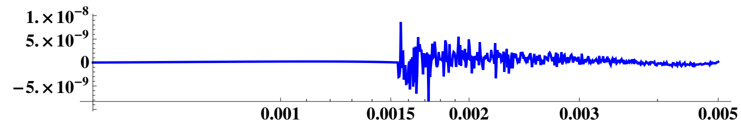

Fig. 3: Relative accuracy of E_{3333} in the region of vanishing sub-Gram determinant. At $x \sim 0.0015$, PJFry switched to the asymptotic expansion.

$$= \frac{1}{2}\left\{\begin{pmatrix} stu \\ st0 \end{pmatrix}_7 (Y_{a7} - Y_{77}) + \begin{pmatrix} 0st \\ 0st \end{pmatrix}_7 (\delta_{au} - \delta_{7u}) - \begin{pmatrix} 0st \\ 0su \end{pmatrix}_7 (\delta_{at} - \delta_{7t}) - \begin{pmatrix} 0ts \\ 0tu \end{pmatrix}_7 (\delta_{as} - \delta_{7s})\right\},$$

with

$$Y_{jk} = -(q_j - q_k)^2 + m_j^2 + m_k^2. \tag{13}$$

Conventionally, $q_7 = 0$.

The sums may be found in eqns. (A.15) and (A.16) of [3]. The s is redundant and fulfils $s \neq r, b, 7$ in $K^{b,rt}$. In $K_0^{a,stu}$ it is $s, t, u = 1, \ldots 7$ with $s \neq u, t \neq u$.

ACKNOWLEDGEMENTS

J.F. thanks DESY for kind hospitality. Work is supported in part by Sonderforschungsbereich/Transregio SFB/TRR 9 of DFG "Computergestützte Theoretische Teilchenphysik" and European Initial Training Network LHCPHENOnet PITN-GA-2010-264564.

References

[1] V. Yundin. C++ package PJFry, available at the webpage `https://github.com/Vayu/PJFry/`.

[2] J. Fleischer and T. Riemann, *Complete algebraic reduction of one-loop tensor Feynman integrals*, *Phys. Rev.* **D83** (2011) 073004, [`arXiv:1009.4436`].

[3] J. Fleischer and T. Riemann, *Calculating contracted tensor Feynman integrals*, *Phys.Lett.* **B701** (2011) 646–653, [`arXiv:1104.4067`].

[4] J. Fleischer and T. Riemann, *A solution for tensor reduction of one-loop n-point functions with* $n \geq 6$, *Physics Letters B* **707** (2012) 375 – 380, [`arXiv:1111.5821`].

[5] J. Fleischer, T. Riemann, and V. Yundin, *One-loop tensor Feynman integral reduction with signed minors*, `1112.0500`. Talk held at ACAT 2011, to appear in JPCS.

[6] R. K. Ellis and G. Zanderighi, *Scalar one-loop integrals for QCD*, *JHEP* **02** (2008) 002, [arXiv:0712.1851].

[7] G. J. van Oldenborgh, *FF: A Package to evaluate one loop Feynman diagrams*, *Comput. Phys. Commun.* **66** (1991) 1–15. doi:10.1016/0010-4655(91)90002-3, scanned version at http://ccdb3fs.kek.jp/cgi-bin/img_index?9004168.

[8] A. van Hameren, *Oneloop: For the evaluation of one-loop scalar functions*, *Comput. Phys. Commun.* **182** (2011) 2427–2438, [arXiv:1007.4716].

[9] G. Passarino and M. Veltman, *One loop corrections for e^+e^- annihilation into $\mu^+\mu^-$ in the Weinberg model*, *Nucl. Phys.* **B160** (1979) 151. doi:10.1016/0550-3213(79)90234-7.

[10] A. I. Davydychev, *A simple formula for reducing Feynman diagrams to scalar integrals*, *Phys. Lett.* **B263** (1991) 107–111. doi:10.1016/0370-2693(91)91715-8.

[11] T. Binoth, J. Guillet, G. Heinrich, E. Pilon, and C. Schubert, *An algebraic / numerical formalism for one-loop multi-leg amplitudes*, *JHEP* **10** (2005) 015, [hep-ph/0504267].

[12] A. Denner and S. Dittmaier, *Reduction schemes for one-loop tensor integrals*, *Nucl. Phys.* **B734** (2006) 62–115, [hep-ph/0509141].

[13] G. Cullen, N. Greiner, G. Heinrich, G. Luisoni, P. Mastrolia, G. Ossola, T. Reiter, and F. Tramontano, *Automated One-Loop Calculations with GoSam.* , 1111.2034.

[14] T. Diakonidis, J. Fleischer, J. Gluza, K. Kajda, T. Riemann, and J. Tausk, *A complete reduction of one-loop tensor 5- and 6-point integrals*, *Phys. Rev.* **D80** (2009) 036003, [arXiv:0812.2134].

[15] L. Reina and T. Schutzmeier, *Towards $Wb\bar{b}+j$ at NLO with an automatized approach to one-loop computations.* , 1110.4438.