

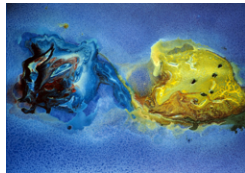
FEYNCALC 9

Vladyslav Shtabovenko

in collaboration with R. Mertig and F. Orellana
based on arXiv:1601.01167

Technische Universität München

ACAT 2016
UTFSM Valparaiso



Physik-Department T30f

OUTLINE

- 1 INTRODUCTION
 - What is FeynCalc?
- 2 NEW DEVELOPMENTS IN FEYN CALC 9
 - Simplification of loop integrals
 - Combining FeynCalc with other tools
- 3 SUMMARY

WHY AUTOMATION?

- Automation of QFT calculations: Crucial to study multi-loop and multi-particle processes.
- Even 1-loop calculations are often hardly feasible by pen and paper.
- Common solution: Private codes on top of a CAS / CAS-like framework (MATHEMATICA, REDUCE, FORM, GiNAC ...)
- Other approach: Use publicly available tools (FIRE, TRACER, QGRAF ...)

MULTI-PURPOSE TOOLS

- Different approaches to QFT automation
 - Fully-automatic: Complete evaluation from Lagrangian till cross-section / decay rate (FORMCALC, CALCHEP, GRACE, ...).
 - Semi-automatic: Collection of useful functions, the user combines them in the most effective way (HEPMATH, FEYNCALC, PACKAGE X, ...)
- The two philosophies are not competitive but rather complementary, e.g.
 - Full automation works perfectly for SM processes
 - Determination of matching coefficients in EFTs is usually too specific to be automatized in full generality.

- FEYNCalc is a free and open source (LGPLv3) MATHEMATICA package for symbolic semi-automatic evaluation of Feynman diagrams and algebraic expressions in QFT.

[Mertig et al., 1991]

[Shtabovenko et al., 2016]



FEATURES

- Suitable for evaluating both single expressions and full Feynman diagrams.
- The calculation can be organized in many different ways (flexibility)
- Extensive typesetting for better readability
- Tools for frequently occurring tasks like Lorentz index contraction, $SU(N)$ algebra, Dirac matrix manipulation and traces, etc.
- Passarino-Veltman reduction of one-loop amplitudes to standard scalar integrals
- Basic support for manipulating multi-loop integrals
- General tools for non-commutative algebra

- Easy installation directly from a MATHEMATICA session

```
Import["https://raw.githubusercontent.com/FeynCalc/feyncalc/master/install.m"]
InstallFeynCalc []
```

- Many examples for QED/QCD calculations already included in *FeynCalc/Examples*
- Documentation integrated into MATHEMATICA's Documentation Center

```
In[1]= << FeynCalc`
FeynCalc 9.0.0. For help, use the documentation center, check out the wiki or write to the mailing list.
```

The screenshot shows the FeynCalc interface within Wolfram Mathematica 10.3. The window title is "FeynCalc - Wolfram Mathematica 10.3". The menu bar includes File, Edit, Insert, Format, Cell, Graphics, Evaluation, Palettes, Window, and Help. The address bar shows "FeynCalc/". The main content area displays the "FeynCalc" guide page, which includes a description of the package and a "Reference" section with various sub-topics and functions.

FeynCalc
 FeynCalc is a Mathematica package for algebraic calculation in Quantum Field Theory and semi-automatic evaluation of Feynman Diagrams

Reference

- Basic objects**
- Basic functions**
- Lorentz tensors**
- Dirac algebra**
- Algebra of non-commutative objects**
- SU(N) algebra**
- CalcColorFactor** — calculates the color factor
- SUNDeltaContract** — contracts Kronecker deltas with adjoint color indices
- SUNFDeltaContract** — contracts Kronecker deltas with fundamental color indices
- SUNSimplify, SUNFSimplify** — simplifies expressions that contain SU(N) matrices
- SUNTrace** — computes traces over SU(N) matrices
- Loop integrals**
- Expand and Insert**

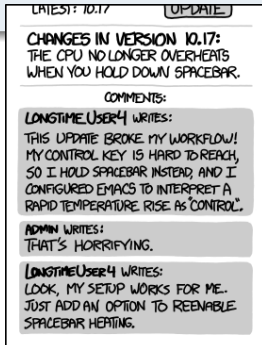
The interface also shows a "Functions" and "URL" dropdown menu on the right side of the content area, and a status bar at the bottom right indicating "100%" zoom.

FEYN CALC 9

- FEYN CALC 9 was officially released this year (see arXiv:1601.01167)
- The source code is now hosted on GitHub: github.com/FeynCalc
- Works with MATHEMATICA versions 8, 9 and 10.
- Test driven development: over 3000 unit and integration tests to prevent new bugs and regressions.

WHAT'S NEW

- Improved 1-loop tensor reduction.
- New partial fractioning algorithm.
- Basic multi-loop tensor reduction.
- Simpler interfacing with other tools (e.g. for IBP-reduction)
- Better interface to FEYNARTS.



EVERY CHANGE BREAKS SOMEONE'S WORKFLOW.

[xkcd.com/1172]

TENSOR REDUCTION

- 1-loop tensor reduction is done via Passarino-Veltman technique: TID
- TID received many improvements in FEYNCalc 9.
- Default mode: Reduce each tensor integral to PaVe scalar functions (A_0 , B_0 , C_0 , D_0)

In[1]:= FCI[GAD[μ].(m + GSD[q]).GAD[μ] FAD[{q, m}]]

$$\mathbf{Out[1]} := \frac{\gamma^\mu \cdot (m + \gamma \cdot q) \cdot \gamma^\mu}{(q^2 - m^2) \cdot (q - p)^2}$$

In[2]:= TID[% , -p + q], q]//ToPaVe[# , q]&

$$\mathbf{Out[2]} = \frac{i\pi^2 (D - 2) A_0(m^2) \gamma \cdot p}{2p^2} - \frac{i\pi^2 B_0(p^2, 0, m^2) (Dm^2 \gamma \cdot p - 2Dmp^2 + Dp^2 \gamma \cdot p - 2m^2 \gamma \cdot p - 2p^2 \gamma \cdot p)}{2p^2}$$

TENSOR REDUCTION

- Zero Gram determinants? Detected automatically, reduction get switched to Passarino-Veltman coefficient functions (e.g. B_1 , B_{00} , C_{222} etc.)

```
In[1]:= ScalarProduct[p, p] = 0;
```

```
$LimitTo4=False;
```

```
TID[GAD[μ].(m + GSD[q]).GAD[μ] FAD[{q, m}, -p + q], q];
```

```
Out[2]:= iπ2B0 (0, 0, m2) (Dm - Dγ · p + 2γ · p) - iπ2(D - 2)γ · pB1 (0, 0, m2)
```

- New useful options:
 - **UsePaVeBasis**: Enforces reduction into coefficient functions for any kinematics.
 - **GenPaVe**: Allows define PaVe functions in a different way (standard is the LOOPTOOLS convention)
 - **Isolate**: Kinematic coefficients in front of the loop inetgrals will be abbreviated. Use FRH to recover the original form.

TENSOR REDUCTION

HOW ABOUT MULTI-LOOP TENSOR REDUCTION?

- In general, not very useful above 1-loop, many scalar products in the denominators can't be cancelled against propagators in the numerators.
- Still practical for loop momenta contracted with Dirac matrices and Levi-Civita tensors. FEYNCalc 9 features FCMultiLoopTID:
 - uses the same PaVe algorithm as for 1-loop.
 - currently no proper way to handle zero Gram determinants.

In[1]:= FCI[FVD[q1, μ] FVD[q2, ν] FAD[q1, q2, {q1 - p1}, {q2 - p1}, {q1 - q2}]]

$$\frac{q1^\mu q2^\nu}{q1^2 \cdot q2^2 \cdot (q1 - p1)^2 \cdot (q2 - p1)^2 \cdot (q1 - q2)^2}$$

Out[1]:=

In[2]:= FCMultiLoopTID[%, {q1, q2}]

Out[2]:=
$$\frac{Dp1^\mu p1^\nu - p1^2 g^{\mu\nu}}{4(D - 1)q2^2 \cdot q1^2 \cdot (q2 - p1)^2 \cdot (q1 - q2)^2 \cdot (q1 - p1)^2} -$$

$$\frac{\frac{p1^2 g^{\mu\nu} - p1^\mu p1^\nu}{Dp1^\mu p1^\nu - p1^2 g^{\mu\nu}}}{2(D - 1)p1^2 q2^2 \cdot q1^2 \cdot (q2 - p1)^2 \cdot (q1 - p1)^2} + \frac{p1^2 g^{\mu\nu} - p1^\mu p1^\nu}{(D - 1)p1^2 q2^2 \cdot q1^2 \cdot (q1 - q2)^2 \cdot (q1 - p1)^2} -$$

$$\frac{2(D - 1)p1^4 q1^2 \cdot (q2 - p1)^2 \cdot (q1 - q2)^2}{2(D - 1)p1^4 q1^2 \cdot (q2 - p1)^2 \cdot (q1 - q2)^2}$$

PARTIAL FRACTIONING

- Scalar loop integrals can be often simplified even further by using partial fractioning.
- Well known identities (implemented in SPC and APART2) are

$$q \cdot p = \frac{1}{2} [(q+p)^2 + m_2^2 - (q^2 + m_1^2) - p^2 - m_2^2 + m_1^2],$$
$$\frac{1}{(q^2 - m_1^2)(q^2 - m_2^2)} = \frac{1}{m_1^2 - m_2^2} \left(\frac{1}{q^2 - m_1^2} - \frac{1}{q^2 - m_2^2} \right).$$

- But: Many decompositions, e.g.

$$\int d^D q \frac{1}{q^2(q-p)^2(q+p)^2} = \frac{1}{p^2} \int d^D q \left(\frac{1}{q^2(q-p)^2} - \frac{1}{(q-p)^2(q+p)^2} \right),$$

require more sophisticated algorithms.

- New in FEYN CALC 9: ApartFF introduces partial fractioning algorithm from [Feng, 2012]
- Compared to the reference MATHEMATICA implementation (<https://github.com/F-Feng/APart>), it is fully integrated into FEYN CALC

```
In[1]:= ApartFF[FAD[{q}, {q - p}, {q + p}], {q}]
```

```
Out[2]:=  $\frac{1}{p^2 q^2 \cdot (q - p)^2} - \frac{1}{p^2 q^2 \cdot (q - 2p)^2}$ 
```

EXTRACTION OF LOOP INTEGRALS

- To evaluate the loop integrals outside of FEYNCalc, we need to extract all the unique integrals from the given expression
- New in FeynCalc 9: FCLoopIsolate

In[1]:= gse = FCI[FAD[q, -p + q] MTD[Lor3, Lor4] (FVD[-p - q, Lor5] MTD[Lor1, Lor3] + FVD[2 p - q, Lor3] MTD[Lor1, Lor5] + FVD[-p + 2 q, Lor1] MTD[Lor3, Lor5]) (FVD[p + q, Lor6] MTD[Lor2, Lor4] + FVD[-2 p + q, Lor4] MTD[Lor2, Lor6] + FVD[p - 2 q, Lor2] MTD[Lor4, Lor6]) MTD[Lor5, Lor6]]

Out[1]:=
$$\frac{g^{\text{Lor3Lor4}} g^{\text{Lor5Lor6}} \left(g^{\text{Lor3Lor5}} (2q - p)^{\text{Lor1}} + g^{\text{Lor1Lor5}} (2p - q)^{\text{Lor3}} + g^{\text{Lor1Lor3}} (-p - q)^{\text{Lor5}} \right)}{\left(g^{\text{Lor4Lor6}} (p - 2q)^{\text{Lor2}} + g^{\text{Lor2Lor6}} (q - 2p)^{\text{Lor4}} + g^{\text{Lor2Lor4}} (p + q)^{\text{Lor6}} \right) \frac{q^2 \cdot (q - p)^2}{q^{\text{Lor2Lor4}}}}$$

In[2]:= FCLoopIsolate[Contract[gse], {q}, Head -> loop] // Cases2[#, loop] &

Out[2]:=
$$\left\{ \text{loop} \left(\frac{1}{q^2 \cdot (q - p)^2} \right), \text{loop} \left(\frac{q^{\text{Lor1}}}{q^2 \cdot (q - p)^2} \right), \text{loop} \left(\frac{q^{\text{Lor2}}}{q^2 \cdot (q - p)^2} \right), \right.$$

$$\left. \text{loop} \left(\frac{q^{\text{Lor1}} q^{\text{Lor2}}}{q^2 \cdot (q - p)^2} \right), \text{loop} \left(\frac{pq \cdot}{q^2 \cdot (q - p)^2} \right), \text{loop} \left(\frac{q^2}{q^2 \cdot (q - p)^2} \right) \right\}$$

- Furthermore:
 - FCLoopSplit to separate different types of loop integral (free of loops, scalar integrals with and without scalar products in the numerators, tensor integrals)
 - FCLoopExtract for combined application of FCLoopIsolate and FCLoopSplit

TOOLS FOR IBP-REDUCTION

- Reduction of scalar loop integrals using integration-by-parts (IBP) identities [Chetyrkin & Tkachov, 1981] is a standard technique in modern loop calculations.
- Many publicly available IBP-packages on the market: FIRE [Smirnov & Smirnov, 2013], LITERED [Lee, 2012], REDUZE [Studerus, 2009], AIR [Anastasiou & Lazopoulos, 2004], ...
- Expected input: loop integrals with propagators that form a basis.
- What about integrals with an incomplete or overdetermined basis?
 - `FCLoopBasisIncompleteQ` detects integrals that require a basis completion
 - `FCLoopBasisFindCompletion` gives a list of propagators (with zero exponents) required to complete the basis
 - `FCLoopBasisOverdeterminedQ` checks if the propagators are linearly dependent. Such integrals can be decomposed further using `ApartFF`.

```
In[1]:= FCI[FAD[{q1, m, 2}, {q1 + q3, m}, {q2 - q3}, q2]]
```

```
Out[1]:= 
$$\frac{1}{(q1^2 - m^2) \cdot (q1^2 - m^2) \cdot ((q1 + q3)^2 - m^2) \cdot (q2 - q3)^2 \cdot q2^2}$$

```

```
In[2]:= FCLoopBasisIncompleteQ[%, {q1, q2, q3}]
```

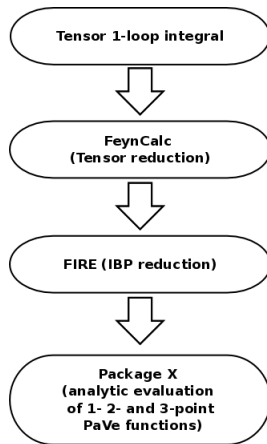
```
Out[2]:= True
```

```
In[3]:= FCLoopBasisFindCompletion[%%, {q1, q2, q3}][[2]]
```

```
Out[3]:= 
$$\left\{ -(q1 \cdot q3) + q2 \cdot q3 + 2q3^2, q1 \cdot q2 \right\}$$

```

- Real life computations: combination of different tools is often advantageous.
- Useful tools to be used with FEYNCalc for the evaluation of 1-loop integrals:
 - FIRE [Smirnov, 2014] (fast and powerful tool for IBP-reduction)
 - PACKAGE X [Patel, 2015] (contains library of analytic results for PaVe functions with up to 3-legs)
- Challenges: Need to convert between the conventions used in each package and avoid variable shadowing.
- Work in progress (VS): Interface for seamless integration between these (and possibly) other packages and FEYNCalc.
- Private version available, public release in near future.
- Technical implementation: Packages are executed on MATHEMATICA subkernels.



- Call FIRE from FEYNCalc:

```
In[1]:= FCI[FAD[q1, q2, {q1 - p, 0, 2}, {q1 - q2, 0, 2}]]
```

```
Out[1]:= 
$$\frac{1}{q1^2 \cdot q2^2 \cdot (q1 - p)^2 \cdot (q1 - p)^2 \cdot (q1 - q2)^2 \cdot (q1 - q2)^2}$$

```

```
In[2]:= FIREBurn[%, {q1, q2}, {p}]
```

```
Out[2]:= 
$$\frac{3(D - 5)(D - 3)(3D - 10)(3D - 8)}{(D - 6)(D - 4)p^6 q2^2 \cdot (q1 - q2)^2 \cdot (q1 - p)^2}$$

```

- FIREBurn actually generates the code to run FIRE standalone

```
FIREPath="/home/vs/.Mathematica/Applications/FIRE5/";  
Get["/home/vs/.Mathematica/Applications/FIRE5/FIRE5.m"];  
Internal={q1, q2};  
External={p};  
Propagators={q1^2, (-p + q1)^2, (q1 - q2)^2, q2^2, p*q2};  
Replacements={};  
PrepareBP[];  
Prepare[AutoDetectRestrictions -> True];  
SaveStart["/home/vs/.Mathematica/Applications/FeynCalc/Database/FIREStartFile"];
```

- Easy and convenient evaluation of Passarino-Veltman functions with PACKAGE-X directly from FEYNCalc:

```
In[1]:= FAD[{q, m0}]
```

$$\text{Out}[1] := \frac{1}{q^2 - m_0^2}$$

```
In[2]:= PaXEvaluate[%, q, PaXImplicitPrefactor -> 1/(2 Pi)^D]
```

$$\text{Out}[2] := \frac{im_0^2}{16\pi^2\varepsilon} - \frac{im_0^2 \left(-\log\left(\frac{\mu^2}{m_0^2}\right) + \gamma - 1 - \log(4\pi) \right)}{16\pi^2}$$

```
In[3]:= I Pi PaVe[1, 2, {m^2, 0, m^2}, {0, m^2, m^2}]
```

$$\text{Out}[3] := i\pi C_{12} \left(m^2, 0, m^2, 0, m^2, m^2 \right)$$

```
In[4]:= PaXEvaluate[%, q, PaXImplicitPrefactor -> 1/(2 Pi)^D]
```

$$\text{Out}[4] := -\frac{i}{192\pi^3 m^2}$$

EVALUATING FEYNMAN DIAGRAMS

- To simplest way to generate Feynman diagrams for FEYNCalc is to use FEYNARTS 3 [Hahn, 2001].
- Both tools are not 100% compatible, some auxiliary steps are needed
 - FEYNARTS needs to be patched to avoid variable conflicts with FEYNCalc (now handles by the automatic installer).
 - The output of FEYNARTS needs to be converted into FEYNCalc notation (new in FEYNCalc 9: FCFACONVERT)

```
In[2]:= diags = InsertFields [ CreateTopologies[0, 2 -> 2], {F[3, {1}],
V[1]} -> {V[5], F[3, {1}]}, InsertionLevel -> {Classes},
Model -> "SMQCD";
```

```
In[3]:= FCFACONVERT[CreateFeynAmp[diags], IncomingMomenta -> {p1, kp},
OutgoingMomenta -> {kg, p2}, UndoChiralSplittings -> True,
TransversePolarizationVectors -> {kg}, DropSumOver -> True,
List -> False] // Contract
```

```
Out[3]:=
```

$$\frac{2ELg_s T_{\text{Col4Col1}}^{\text{Glu3}} (\varphi(\overline{p2}, \text{MU})) \cdot (\bar{\gamma} \cdot \bar{\epsilon}^*(\text{kg})) \cdot (\bar{\gamma} \cdot (\overline{\text{kg}} + \overline{p2}) + \text{MU}) \cdot (\bar{\gamma} \cdot \bar{\epsilon}(\text{kp})) \cdot (\varphi(\overline{p1}, \text{MU}))}{3 ((-\text{kg} - \text{p2})^2 - \text{MU}^2)}$$

$$\frac{2ELg_s T_{\text{Col4Col1}}^{\text{Glu3}} (\varphi(\overline{p2}, \text{MU})) \cdot (\bar{\gamma} \cdot \bar{\epsilon}(\text{kp})) \cdot (\bar{\gamma} \cdot (\overline{p2} - \overline{\text{kp}}) + \text{MU}) \cdot (\bar{\gamma} \cdot \bar{\epsilon}^*(\text{kg})) \cdot (\varphi(\overline{p1}, \text{MU}))}{3 ((\text{kp} - \text{p2})^2 - \text{MU}^2)}$$

Summary

- FEYNCalc is a very flexible and versatile tool for semi-automatic calculations in QFT.
- FEYNCalc 9 introduces several interesting features that facilitate 1-loop and multi-loop calculations.
- A FEYNCalc add-on that provides direct interfaces to other useful HEP tools is under active development.

- There is no unique way to handle $\gamma^5 = i\gamma^0\gamma^1\gamma^2\gamma^3$ in DR
- In D -dimensions, the relations

$$\{\gamma^5, \gamma^\mu\} = 0$$

and

$$\text{tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma) \neq 0$$

cannot be simultaneously satisfied.

- In other words, there is a conflict between the anticommutativity of γ^5 and the cyclicity property of Dirac traces that involve an odd number of γ^5

[Chanowitz et al., 1979]

[Jegerlehner, 2001]

- We can stick to the anticommuting γ^5 in D -dimensions. This is fine, as long as we have only traces with an even number of γ^5 .
 - ⇒ Naive dimensional regularization (NDR)
- To compute traces with an odd number of γ^5 unambiguously, we need an additional prescription
 - ⇒ Kreimer's prescription [Kreimer, 1990]
 - ⇒ Larin-Gorishny-Akyaempong-Delburgo prescription [Larin, 1993]
- Or we can accept that γ^5 is a purely 4-dimensional object and therefore doesn't anticommute with D -dimensional Dirac matrices
 - [t Hooft & Veltman, 1972]
 - [Breitenlohner & Maison, 1977]
 - ⇒ Breitenlohner-Maison- t'Hooft Veltman scheme (BMHV)

By default, FeynCalc works with an anticommuting γ^5

```
In[1]:= GAD[μ,ν,ρ].GA[5].GAD[σ,τ,κ].GA[5]
Out[1]= γμ.γν.γρ.γ5.γσ.γτ.γκ.γ5
In[2]:= %//DiracSimplify
Out[2]= -γμ.γν.γρ.γσ.γτ.γκ
```

Trying to compute a chiral trace in the naive scheme produces an error message:

```
In[1]:= DiracTrace[GAD[μ,ν,ρ,σ,τ,κ].GA[5]]
Out[1]= tr (γμ.γν.γρ.γσ.γτ.γκ.γ5)
In[2]:= % /. DiracTrace -> Tr
```

DiracTrace::ndranomaly :

You are using naive dimensional regularization (NDR), such that in D dimensions γ^5 anticommutes with all other Dirac matrices. In this scheme (without additional prescriptions) it is not possible to compute traces with an odd number of γ^5 unambiguously. The trace

```
DiracGamma[LorentzIndex[μ, D], D].
DiracGamma[LorentzIndex[ν, D], D].
of DiracGamma[LorentzIndex[ρ, D], D].
DiracGamma[LorentzIndex[σ, D], D].
DiracGamma[LorentzIndex[τ, D], D].
DiracGamma[LorentzIndex[κ, D], D]. DiracGamma[5]
```

is illegal in NDR. Evaluation aborted!

D -dimensional traces with anticommuting γ^5 can be evaluated using Larin-Gorishny-Akyeampong-Delburgo prescription

```
In[1]:= $Larin = True;
In[2]:= $West = False;
In[3]:= $BreitMaison = False;
In[4]:= DiracTrace[GAD[μ, ν, ρ, σ, τ, κ].GA[5]]
Out[4]:= tr (γμ.γν.γρ.γσ.γτ.γκ.γ̄5)
In[5]:= % /. DiracTrace -> Tr
Out[5]:= 4 (igμνεκρστ - igμρεκνστ + igμσεκνρτ - igμτεκνρσ + igνρεκμστ
- igνσεκμρτ + igντεκμρσ + igρσεκμντ - igρτεκμνσ + igστεκμνρ)
```

Or in the BMHV scheme

```
In[6]:= $Larin = False;
In[7]:= $West = True;
In[8]:= $BreitMaison = False;
In[9]:= DiracTrace[GAD[μ, ν, ρ, σ, τ, κ].GA[5]]
Out[9]:= tr (γμ.γν.γρ.γσ.γτ.γκ.γ̄5)
In[10]:= % /. DiracTrace -> Tr
Out[10]:= 4 (-igκμενρστ + igκνεμρστ - igκρεμνστ + igκσεμνρτ - igκτεμνρσ
+ igμνεκρστ - igμρεκνστ + igμσεκνρτ - igμτεκνρσ + igνρεκμστ - igνσεκμρτ
+ igντεκμρσ + igρσεκμντ - igρτεκμνσ + igστεκμνρ)
```

ISSUES WITH γ^5 IN NDR

Assuming that both

$$\begin{aligned}\{\gamma^5, \gamma^\mu\} &= 0, \\ \text{Tr}\{\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma\} &\neq 0\end{aligned}$$

hold in D -dimensions leads to a contradiction. The reason is the assumed cyclicity of the Dirac trace

$$\begin{aligned}D \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma) &= \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma \gamma_\tau \gamma^\tau) \\ &= -2g^{\tau\mu} \text{Tr}(\gamma^5 \gamma^\nu \gamma^\rho \gamma^\sigma \gamma_\tau) + 2g^{\tau\nu} \text{Tr}(\gamma^5 \gamma^\mu \gamma^\rho \gamma^\sigma \gamma_\tau) \\ &\quad - 2g^{\tau\rho} \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\sigma \gamma_\tau) + 2g^{\tau\sigma} \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma_\tau) \\ &\quad - D \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma) \\ &= -2 \text{Tr}(\gamma^5 \gamma^\nu \gamma^\rho \gamma^\sigma \gamma^\mu) + 2 \text{Tr}(\gamma^5 \gamma^\mu \gamma^\rho \gamma^\sigma \gamma^\nu) \\ &\quad - 2 \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\sigma \gamma^\rho) + 2 \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma) \\ &\quad - D \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma)\end{aligned}$$

Using that $\text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu) = 0$ we have

$$D \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma) = (8 - D) \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma)$$

or

$$(4 - D) \text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma) = 0.$$

- This implies that $\text{Tr}(\gamma^5 \gamma^\mu \gamma^\nu \gamma^\rho \gamma^\sigma)$ is zero for all $D \neq 4$.
- But if we demand the trace to be meromorphic in D , then the above trace should be zero also for $D = 4$,
- Hence, we cannot recover the 4-dimensional Dirac algebra at $D = 4$.

NON-NAIVE SCHEME FOR γ^5

In the Breitenlohner-Maison-'t Hooft-Veltman scheme we are dealing with matrices in D, 4 and D-4 dimensions. Many identities of the BMHV algebra can be proven by decomposing Dirac matrices into two pieces

$$\begin{aligned} \dim(\gamma^\mu) &= d, & \gamma^\mu &= \bar{\gamma}^\mu + \hat{\gamma}^\mu, \\ \dim(\bar{\gamma}^\mu) &= 4, & g^{\mu\nu} &= \bar{g}^{\mu\nu} + \hat{g}^{\mu\nu}, \\ \dim(\hat{\gamma}^\mu) &= d - 4. & p^\mu &= \bar{p}^\mu + \hat{p}^\mu. \end{aligned}$$

(ANTI)COMMUTATORS BETWEEN γ 'S IN DIFFERENT DIMENSIONS

$$\begin{aligned} \{\gamma^\mu, \gamma^\nu\} &= 2g^{\mu\nu}, \\ \{\bar{\gamma}^\mu, \bar{\gamma}^\nu\} &= \{\gamma^\mu, \bar{\gamma}^\nu\} = 2\bar{g}^{\mu\nu}, \\ \{\hat{\gamma}^\mu, \hat{\gamma}^\nu\} &= \{\gamma^\mu, \hat{\gamma}^\nu\} = 2\hat{g}^{\mu\nu}, \\ \{\bar{\gamma}^\mu, \hat{\gamma}^\nu\} &= 0 \\ \{\bar{\gamma}^\mu, \gamma^5\} &= [\hat{\gamma}^\mu, \gamma^5] = 0, \\ \{\gamma^\mu, \gamma^5\} &= \{\hat{\gamma}^\mu, \gamma^5\} = 2\hat{\gamma}^\mu \gamma^5 = 2\gamma^5 \hat{\gamma}^\mu. \end{aligned}$$

NON-NAIVE SCHEME FOR γ^5

Contractions of Dirac matrices and vectors with the metric

$$\begin{aligned}
 g^{\mu\nu} \gamma_\nu &= \gamma^\mu, & g^{\mu\nu} p_\nu &= p^\mu, \\
 \bar{g}^{\mu\nu} \bar{\gamma}_\nu &= g^{\mu\nu} \bar{\gamma}_\nu = \bar{g}^{\mu\nu} \gamma_\nu = \bar{\gamma}^\mu, & \bar{g}^{\mu\nu} \bar{p}_\nu &= g^{\mu\nu} \bar{p}_\nu = \bar{g}^{\mu\nu} p_\nu = \bar{p}^\mu, \\
 \hat{g}^{\mu\nu} \hat{\gamma}_\nu &= g^{\mu\nu} \hat{\gamma}_\nu = \hat{g}^{\mu\nu} \gamma_\nu = \hat{\gamma}^\mu, & \hat{g}^{\mu\nu} \hat{p}_\nu &= g^{\mu\nu} \hat{p}_\nu = \hat{g}^{\mu\nu} p_\nu = \hat{p}^\mu, \\
 \bar{g}^{\mu\nu} \hat{\gamma}_\nu &= \hat{g}^{\mu\nu} \bar{\gamma}_\nu = 0, & \bar{g}^{\mu\nu} \hat{p}_\nu &= \hat{g}^{\mu\nu} \bar{p}_\nu = 0.
 \end{aligned}$$

Contractions of the metric with itself

$$\begin{aligned}
 g^{\mu\nu} g_{\nu\rho} &= g^\mu_\rho & g^{\mu\nu} g_{\mu\nu} &= d, \\
 \bar{g}^{\mu\nu} \bar{g}_{\nu\rho} &= g^{\mu\nu} \bar{g}_{\nu\rho} = \bar{g}^{\mu\nu} g_{\nu\rho} = \bar{g}^\mu_\rho & \bar{g}^{\mu\nu} \bar{g}_{\mu\nu} &= g^{\mu\nu} \bar{g}_{\mu\nu} = \bar{g}^{\mu\nu} g_{\mu\nu} = 4, \\
 \hat{g}^{\mu\nu} \hat{g}_{\nu\rho} &= g^{\mu\nu} \hat{g}_{\nu\rho} = \hat{g}^{\mu\nu} g_{\nu\rho} = \hat{g}^\mu_\rho & \hat{g}^{\mu\nu} \hat{g}_{\mu\nu} &= g^{\mu\nu} \hat{g}_{\mu\nu} = \hat{g}^{\mu\nu} g_{\mu\nu} = d - 4, \\
 \bar{g}^{\mu\nu} \hat{g}_{\nu\rho} &= \hat{g}^{\mu\nu} \bar{g}_{\nu\rho} = 0, & \bar{g}^{\mu\nu} \hat{g}_{\mu\nu} &= \hat{g}^{\mu\nu} \bar{g}_{\mu\nu} = 0.
 \end{aligned}$$

Contractions of Dirac matrices and vectors with themselves

$$\begin{aligned}
 \gamma^\mu \gamma_\mu &= D, & p^\mu p_\mu &= p^2, \\
 \bar{\gamma}^\mu \bar{\gamma}_\mu &= \gamma^\mu \bar{\gamma}_\mu = \bar{\gamma}^\mu \gamma_\mu = 4, & \bar{p}^\mu \bar{p}_\mu &= \bar{p}^\mu p_\mu = p^\mu \bar{p}_\mu = \bar{p}^2, \\
 \hat{\gamma}^\mu \hat{\gamma}_\mu &= \gamma^\mu \hat{\gamma}_\mu = \hat{\gamma}^\mu \gamma_\mu = D - 4, & \hat{p}^\mu \hat{p}_\mu &= \hat{p}^\mu p_\mu = p^\mu \hat{p}_\mu = \hat{p}^2, \\
 \bar{\gamma}^\mu \hat{\gamma}_\mu &= \hat{\gamma}^\mu \bar{\gamma}_\mu = 0, & \bar{p}^\mu \hat{p}_\mu &= \hat{p}^\mu \bar{p}_\mu = 0.
 \end{aligned}$$

Larin-Gorishny-Akyaempong-Delburgo prescription allows one to use anticommuting γ^5 in D -dimensions but compute the chiral traces, such, that the result is expected to be equivalent with the BMHV scheme, if we have only one axial-vector current. The prescription is essentially

- Anticommute γ^5 to the right inside the trace
- Replace $\gamma^\mu \gamma^5$ with $-\frac{i}{6} \varepsilon^{\mu\alpha\beta\sigma} \gamma^\alpha \gamma^\beta \gamma^\sigma$
- Treat $\varepsilon^{\mu\alpha\beta\sigma}$ as if it were D -dimensional, i.e.
 $\varepsilon^{\mu\alpha\beta\sigma} \varepsilon_{\mu\alpha\beta\sigma} = -D(D^3 - 6D^2 + 11D - 6)$ instead of -24 .

DEFINITIONS OF THE PAVE SCALAR INTEGRALS (LOOPTOOLS CONVENTION)

$$A_0(m_0) = \mu^{4-D} (4\pi)^{\frac{4-D}{2}} \int \frac{d^D q}{i\pi^{\frac{D}{2}}} \frac{1}{q^2 - m_0^2}$$

$$B_0(p_1, m_0, m_1) = \mu^{4-D} (4\pi)^{\frac{4-D}{2}} \int \frac{d^D q}{i\pi^{\frac{D}{2}}} \frac{1}{(q^2 - m_0^2)((q + p_1)^2 - m_1^2)}$$

$$C_0(p_1, p_2, m_0, m_1, m_2)$$

$$= \mu^{4-D} (4\pi)^{\frac{4-D}{2}} \int \frac{d^D q}{i\pi^{\frac{D}{2}}} \frac{1}{(q^2 - m_0^2)((q + p_1)^2 - m_1^2)((q + p_1 + p_2)^2 - m_2^2)}$$

$$D_0(p_1, p_2, p_3, m_0, m_1, m_2, m_3)$$

$$= \mu^{4-D} (4\pi)^{\frac{4-D}{2}} \int \frac{d^D q}{i\pi^{\frac{D}{2}}} \frac{1}{(q^2 - m_0^2)((q + p_1)^2 - m_1^2)((q + p_1 + p_2)^2 - m_2^2)}$$

$$\times \frac{1}{((q + p_1 + p_2 + p_3)^2 - m_3^2)}$$

NORMALIZATION OF THE PAVE SCALAR INTEGRALS

Passarino-Veltman scalar functions are normally related to the text book integrals by a factor of $(16\pi^2)/i$, e.g.

$$\mu^{4-D} \int \frac{d^D q}{(2\pi)^D} \frac{1}{q^2 - m_0^2} = \frac{i}{16\pi^2} A_0(m_0)$$

To see this observe that

$$\frac{1}{(2\pi)^D} = \frac{1}{16\pi^2} \frac{1}{2^{D-4} \pi^{D-2}} = \frac{1}{16\pi^2} \frac{4^{\frac{4-D}{2}}}{\pi^{D-2}} = \frac{1}{16\pi^2} \frac{(4\pi)^{\frac{4-D}{2}}}{\pi^{\frac{D}{2}}}$$







However, in FeynCalc the PaVe functions are normalized as







$\mu^{4-D} \frac{1}{i\pi^2} \int d^D q(\dots)$. Hence, we have






$$A_{0,FC}(m_0) = (2\pi)^{D-4} A_0(m_0) = \frac{(2\pi)^D}{i\pi^2} \mu^{4-D} \int \frac{d^D q}{(2\pi)^D} \frac{1}{q^2 - m_0^2}$$

On the other hand, if the prefactor $\frac{1}{(2\pi)^D}$ is implicit (i.e. it is understood but not written down explicitly) in the calculation, then it is enough to perform the replacement

$$A_{0,FC}(m_0) \rightarrow \frac{1}{i\pi^2} \mu^{4-D} \int \frac{d^D q}{(2\pi)^D} \frac{1}{q^2 - m_0^2}$$

-  Anastasiou, C. & Lazopoulos, A. (2004).
Automatic Integral Reduction for Higher Order Perturbative Calculations.
JHEP, 0407, 046.
-  Breitenlohner, P. & Maison, D. (1977).
Dimensional renormalization and the action principle.
Communications in Mathematical Physics, 52, 11–38.
-  Chanowitz, M., Furman, M., & Hinchliffe, I. (1979).
The axial current in dimensional regularization.
Nucl. Phys. B, 159(1-2), 225–243.
-  Chetyrkin, K. & Tkachov, F. (1981).
Integration by parts: The algorithm to calculate β -functions in 4 loops.
Nucl. Phys. B, 192(1), 159–204.
-  Feng, F. (2012).
\$Apart: A Generalized Mathematica Apart Function.
Comput. Phys. Commun., 183, 2158–2164.
-  Hahn, T. (2001).
Generating Feynman Diagrams and Amplitudes with FeynArts 3.
Comput. Phys. Commun., 140, 418–431.

-  Jegerlehner, F. (2001).
Facts of life with $\gamma(5)$.
Eur.Phys.J.C, 18:673-679,2001.
-  Kreimer, D. (1990).
The γ^5 -problem and anomalies — A Clifford algebra approach.
Phys. Lett. B, 237(1), 59–62.
-  Larin, S. A. (1993).
The renormalization of the axial anomaly in dimensional regularization.
Phys. Lett. B, 303, 113–118.
-  Lee, R. N. (2012).
Presenting LiteRed: a tool for the Loop InTEgrals REDuction.
-  Mertig, R., Böhm, M., & Denner, A. (1991).
Feyn Calc - Computer-algebraic calculation of Feynman amplitudes.
Comput. Phys. Commun., 64(3), 345–359.
-  Patel, H. H. (2015).
Package-X: A Mathematica package for the analytic calculation of one-loop integrals.
Comput. Phys. Commun., 197, 276–290.

-  Shtabovenko, V., Mertig, R., & Orellana, F. (2016).
New Developments in FeynCalc 9.0.
-  Smirnov, A. V. (2014).
FIRE5: a C++ implementation of Feynman Integral REduction.
-  Smirnov, A. V. & Smirnov, V. A. (2013).
FIRE4, LiteRed and accompanying tools to solve integration by parts relations.
Comput. Phys. Commun., 184(12), 2820–2827.
-  Studerus, C. (2009).
Reduze - Feynman Integral Reduction in C++.
Comput. Phys. Commun., 181, 1293–1300.
-  't Hooft, G. & Veltman, M. (1972).
Regularization and renormalization of gauge fields.
Nucl. Phys. B, 44(1), 189–213.