Package for Calculations and Simplifications of Expressions with Dirac Matrixes (MatrixExp)

V. A. Poghosyan[1]
Yerevan Physics Institute, 2 Alikhanyan Br., 375036 Yerevan, Armenia,

Abstract

This paper describes a package for calculations of expressions with Dirac matrixes. Advantages to existing similar packages are described. MatrixExp package is intended for simplification of complex expressions involving γ-matrixes, providing such tools as automatic Feynman parameterization, integration in d-dimensional space, sorting and grouping of results in a given order. Also, in comparison with existing similar package Tracer, presented package MatrixExp has more enhanced input possibility. User-available functions of MatrixExp package are described in detail. Also an example of calculation of Feynman diagram for process $b \to s\gamma g$ with application of functions of MatrixExp package is presented.

PACS: 13.10.+q, 13.90.+i, 07.05.Bx, 14.40.Nd

[1]email: vpoghos@server.physdep.r.am
The package is available at http://www.yerphi.am/matrixexp.html

PROGRAM SUMMARY

Keywords: Computer algebra, Particle Physics, Computing, Quantum Field Theory, High energy physics, Calculation, Gamma-algebra,Dimensional Regularization
Classification: 11.1 General, High Energy Physics and Computing, 4.4 Feynman diagrams
Nature of problem:
Feynman diagram calculation, simplification of expressions with γ-matrixes
Restrictions:
MatrixExp package works only with single line of expressions (G[l1,...]), in contrary with Tracer package which works with multiple lines, i.e. the following is possible in Tracer, but not in MatrixExp : G[l1,...]**G[l2,...]**G[l3,...]..., which will return the result of G[l1,...]**G[l1,...]**G[l1,...]...
Running time:
Seconds for expressions with several different γ-matrixes on Pentium IV 1.8GHz and of the order of a minute on Pentium II 233MHz. Calculation times rise with the number of matrixes.

LONG WRITE-UP

1 Introduction

Modern calculations of radiative corrections in Relativistic Quantum Field Theory, in particular within the limits of Standard Model, lead to calculations of one-, two- and higher-loop Feynman diagrams. Considering, that calculations of two-loop and furthermore multi-loop diagrams are labor-consuming enough and volumetric, and also that the number of possible diagrams to be evaluated is great, now are widely applied various packages of symbolic manipulations. In particular, Tracer package [1] working in MATHEMATICA [2] environment is known, allowing to operate with expressions of γ-matrixes in d-dimensional space. Following the developers of Tracer package we chose MATHEMATICA environment for the MatrixExp package, since MATHEMATICA is widely used by researchers and is easily programmable, and also, which is more important, unlike similar package MAPLE [3], MATHEMATICA allows to program in "rule-based" style: a special style of programming convenient for implementation (application) of mathematical knowledge into the program (package). From the technical point of view, during calculations of radiative corrections it is necessary to get rid from arising intermediate singularities, inherent to quantum corrections, usually by the method of dimensional regularization; hence, it is necessary to perform the calculations in d-dimensional space. The MatrixExp package uses the so-called Naive Dimensional Regularization scheme (NDR). As it is known, the given scheme leads to some algebraic inconsistencies in d-dimensional space [4, 5]. However, as it has been shown in Ref. [6], in many calculations which don't involve traces with γ_5 matrix, NDR scheme gives correct result. A problem connected with definition of γ_5 matrix in d-dimensional space is solved in the t'Hooft-Veltman regularization scheme [7, 8]. In future, this scheme will also be included in the MatrixExp package.

2 Tracer package

Tracer package is written basically in "rule-based" style which is presented in [1]. We shall briefly describe this style since our developed program also is written in this style. Unlike the procedural style of programming, where

the program is executed consistently from the first command up to the final, realizing the algorithm for solving the task in view, "rule-based" style is setting up a collection of transformation rules which just represent mathematical rules of transformations and properties of the given objects. The decision of a task in view in "rule-based" style is reduced to the following:

1. Setting up of transformation rules (which are applied when expression matches the pattern on the left-hand side (lhs) of the rule, and return the right-hand side (rhs) of the rule)

2. Use of basic rules built-in in MATHEMATICA

3. Consecutive application of corresponding rules (automatically carried out by MATHEMATICA)

4. Last received expression which does not match to any pattern is the received result.

Schematically the solution of a problem in "rule-based" style is presented on Fig. 1.

The solution of a problem in this style greatly simplifies the programming and does clearer the program code since allows to not supervise all stream of the information during the solution of a problem (at calculation of diagrams very bulky and complex expressions arise), but by means of setting up of all transformation rules (which are the initial mathematical rules, i.e. are clear and natural to the user) directly solves the problem in view (in our case - will transform initial expression to a convenient form, carrying out all necessary transformations). To illustrate the above mentioned, codes for differentiation in procedural style and in "rule-based" style are presented in Table 1.

Note, that if we try to define differentiation for all[2] other functions as well, then the code in procedural style would get very complicated, including special recursive codes for mixed functions, code for parsing expression of functions etc., and in the case of "rule-based" programming, one has just to add all the definitions, rules for differentiation of a sum and product of functions and rules for differentiation of $f(g(x))$.

Tracer package realizes γ-algebra in "rule-based" style and calculates a trace of strings of γ-matrixes [1]. Tracer accepts following types of data in a corresponding format:

[2]e.g. for 5 or 10 more functons

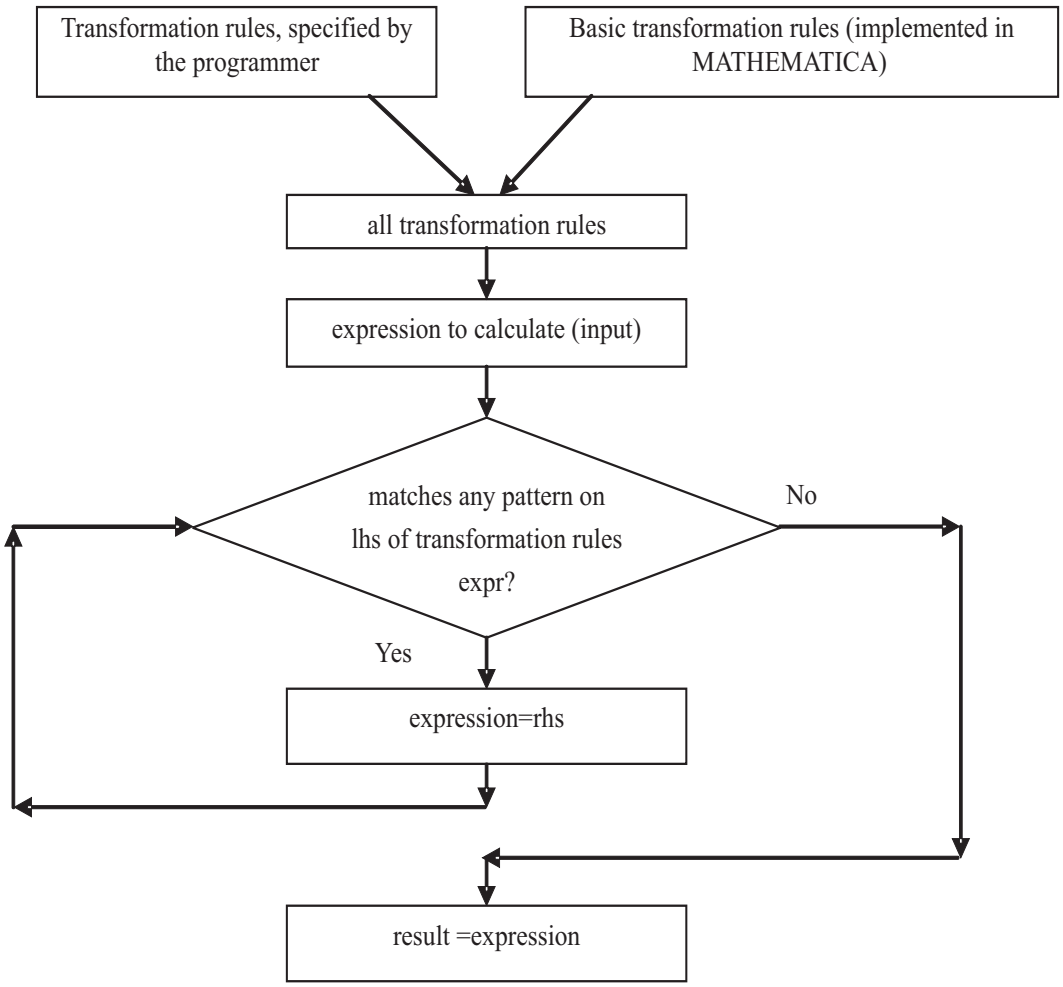

Figure 1: Scheme of problem solution in a "rule-based" style.

a. Procedural style

```
diff[y_, x_] :=
  Block[{n},
    If[TrueQ[Length[y]==2 && y[[0]]==Power && y[[1]]==x],
      n=y[[2]]; n*x^(n-1),
      If[TrueQ[y==x], 1,
        If[TrueQ[Length[y==1] && y[[0]]==Log && y[[1]]==x], 1/x]]];
```

b. "rule-based" style

```
diff[x_^n_, x_] := n*x^(n-1);
diff[Log[x_], x_]:= 1/x;
```

Table 1: Codes for derivative calculation (cases of power function and logarithmic function) in procedural **(a)** and "rule-based" **(b)** styles.

1. a momentum or a sum of momenta p, $p-q$, $(p+k1-k2)$, which correspond to $\hat{p}$, $\hat{p}-\hat{q}$ and $\hat{p}+\hat{k}_1-\hat{k}_2$

2. an open index of a gamma matrix, which is denoted by a symbol in curly brackets {alpha} corresponding to γ_α

3. a linear combination of the unit matrix in the gamma algebra U and the matrix γ_5 denoted by $G5$, with scalar coefficients

4. a linear combination of a momentum p and a mass term mU.

For convenience, we use the same input-format for MatrixExp package[3], for users already familiarized using Tracer package. Also, this way one may directly use results achieved by Tracer as input for MatrixExp and vice versa.

[3]except points (3) and (4). There are special symbols for $(1\pm\gamma_5)/2$ defined as R and L correspondingly. γ_5 also may be used in the same way, i.e. $G5$. For point (4) see sections 3 and 4.

3 MatrixExp Package

When calculating diagrams, usually so-called Feynman parameterization [9] is applied, therefore, in initial expression instead of momenta p arises their linear combinations, for example $p + u_1 * k + u_2 * r...$ and so the direct application of Tracer package is already impossible, i.e. it is necessary to segment (partition) the initial expression to parts, not containing such combinations of momenta $(p, k, r...)$ and scalars $(u_1, u_2...)$.

Developed new package MatrixExp presented in this paper, allows entering data without additional transformations.

Input data for calculation by Tracer and MatrixExp packages correspondingly (b) and (c), is presented on Table 2.

Thus, MatrixExp package allows working with complex expressions, and there is no need for additional manual transformations of initial input expression. Besides, since it is very often applied Feynman parameterization, with the subsequent d-dimensional integration, this opportunity is also included in the MatrixExp package, i.e. it is possible to set both automatic Feynman parameterization, and automatic integration. The package is intended not only for calculation of traces of matrixes, but also for calculations and simplifications (sorting and grouping) of complex expressions received as a result of previous calculations or as output from other packages. During calculations MatrixExp package operates with scalar products (additional conditions imposed on scalar products can be set via appropriate user-function of MatrixExp (see next section)), combines and recombines momenta according to $\hat{p} = \gamma_\alpha p_\alpha$ for performing calculations, integrations and grouping results (for example, $p_\alpha \hat{q} k_\beta (u_1 r_\alpha + \gamma_\alpha)\gamma_\beta ... \to \hat{q}(u_1(p.r) + \hat{p})\hat{k}...$, etc.).

MatrixExp package includes such enhancements as definition of scalars, automatic sorting by the given mask (order), automatic Feynman parameterization (using previously declared scalar symbols), grouping of sorted results, automatic d-dimensional integration over specified momenta, automatic rearrangement of "edge"-momenta (p_b and p_s) and their replacement by corresponding masses according to Dirac equation ($\langle s|p_s = m_s\langle s|$ and $p_b|b\rangle = m_b|b\rangle$), operations with scalar products, etc.

(a) Expression to be calculated

$\langle s|\hat{p}_s, \gamma_\alpha, \hat{r}, \hat{p} - m_b, \hat{k} + \hat{r}, \gamma_\alpha|b\rangle,\ \hat{r} \to \hat{r} - u_1\hat{p} + u_2\hat{p}_s$

(b) Input for Tracer package

```
a11 = G[l1, ps, {alpha}, r, p-U mb, k+r, {alpha}];
a12 = -u1*G[l1, ps, {alpha}, k, p-U mb, k+r, {alpha}];
a13 = u2*G[l1, ps, {alpha}, ps, p-U mb, k+r, {alpha}];
a21 = -u1*G[l1, ps, {alpha}, r, p-U mb, k-p, {alpha}];
a22 = u1^2*G[l1, ps, {alpha}, k, p-U mb, k-p, {alpha}];
a23 = -u1*u2*G[l1, ps, {alpha}, ps, p-U mb, k-p, {alpha}];
a31 = u2*G[l1, ps, {alpha}, r, p-U mb, k+ps, {alpha}];
a32 = -u2*u1*G[l1, ps, {alpha}, k, p-U mb, k+ps, {alpha}];
a33 = u2^2*G[l1, ps, {alpha}, ps, p-U mb, k+ps, {alpha}];
result = a11+a12+a13+a21+a22+a23+a31+a32+a33;
```

(c) Input for MatrixExp package

```
Vsetscalars[u1, u2, mb];
a = G[l1, ps, {alpha}, r-u1*p+u2*ps, p-mb, k+r-u1*p+u2*ps, {alpha}];
result = Vcalc[a];
```

Table 2: Calculation of expression **(a)** using Tracer package **(b)** (as we see, one needs to segment the initial expression into parts, and if the expression is more complex, then the calculation becomes very complicated) and using MatrixExp package **(c)** (Function Vsetscalars[] declares $u1$, $u2$ and mb as scalars and then function Vcalc automatically performs all corresponding operations).

4 User functions of MatrixExp package

The detailed description of MatrixExp functions and their syntax is included in the package and is accessible via standard method in MATHEMATICA, i.e. ' ?**function_name** '. Call ' **Vhelp**[ ] ' to receive the list of all functions defined in the package.

Here we present basic functions and their purpose.

Vsetscalars[**u1**, **u2**...] - declares symbols **u1**, **u2**... as scalars. By default the package declares as scalars the following symbols: **u1**...**u9** since they are necessary to perform Feynman parameterization. It is possible to rede-

fine scalars calling function **Vsetscalars**[...], or to add other scalars calling function **Vaddscalars**[...].

Vsetintvar[**r**] - declares the symbol **r** as integration variable over which Feynman parameterization and automatic d-dimensional integration will be performed.

Vsetsortlist[**p1**, {**mu1**}, {**mu2**}, **p2**, **p3**...] - sorts momenta (**p1**, **p2**, **p3**...) and γ-matrixes ({**mu1**}, {**mu2**}...) in the designated sequence.

Vcalc[**exp**, **options**] - carries out calculations on expression **exp** and groups (sorts) result. The following may be specified as control parameters (**options**): integration - On/Off (by default it is switched off), a symbol of integration (**integrating** $\rightarrow$ **True**/**False**, **integrating** $\rightarrow$ **r**) (the latter also implies **integration** $\rightarrow$ **True**), Feynman parameterization On/Off (by default it is switched off) (**dofeynman** $\rightarrow$ **True**/**False**), sorting On/Off, sorting order (**sorting** $\rightarrow$ **True**/**False**, **sorting** $\rightarrow$ {**p**, {**mu**}, **k**, {**nu**}, {**sigma**}}) (the latter also implies **sorting** $\rightarrow$ **True**), rearrangement of "edge"-momenta On/Off (by default it is switched off) (**pps** $\rightarrow$ **True**/**False**), calculation of a trace On/Off. (by default it is switched off) (**spur** $\rightarrow$ **True**/**False**)...., etc.

Vfeynman[**x**] - performs Feynman parameterization of expression **x**. Prior to calling of this function it is necessary to define the variable of integration by calling **Vsetintvar**[**r**], with respect to which function **Vfeynman**[**x**] should make parameterization. Gives the output in the following form: {K, $scalar_1^{i_1} scalar_2^{i_2} ... scalar_n^{i_n}$, int_var_0, $power$, $delta$, N}, where K - is the factor arising in a result of parameterization, $scalar_1^{i_1}$, $scalar_2^{i_2}$, $scalar_n^{i_n}$ - are additional multipliers, if any arise, int_var_0 - is the displacement of the integration variable, i.e. in initial expression the integration variable (let it be X for example) should be replaced by $X - int\_var\_0$, $power$ - is the power of the resulting denominator after parameterization, $delta$ - is the parameter of integration, N - is not factored part, i.e. multipliers and factors not dependent on the integration variable, which have not been parameterizated [9]:

$$\frac{1}{D1\,D2\,D3\,D4^{\varepsilon}} = KN \int \frac{\mathrm{d}u1\mathrm{d}u2...\mathrm{d}un\; u1^{i1}...un^{in}\delta(1-u1-...-un)}{[(\mathrm{int\_var}-\mathrm{int\_var\_0})^2+\mathrm{delta}]^{\mathrm{power}}} \quad (1)$$

When specifying Feynman parameterization option in function **Vcalc**, all steps and substitutions are carried out automatically (see expressions (7, 8)).

Vsetrules[{**p1**, **p2**, **c1**}, {**p1**, **c2**}...] - defines rules of scalar product, corresponding to - $((p1, p2) = c1, (p1, p1) = p1^2 = c2)$.

5 Calculation of an expression by means of MatrixExp package

Below we present an example of calculation with use of MatrixExp package. It is calculated a Feynman diagram for process $b \to s\gamma g$ [10] in case of local operators O_1, O_2, having the following form: $O_1 = (\bar{s}_L\gamma_\mu T^a c_L)(\bar{c}_L\gamma^\mu T^a b_L)$, $O_2 = (\bar{s}_L\gamma_\mu c_L)(\bar{c}_L\gamma^\mu b_L)$. On Fig. 2 we present the two Feynman diagrams sum of which in [10] is denoted as $J_{\alpha\beta}$. Here q designates photon momentum, r - gluon momentum.

Figure 2: Bremsstrahlung diagrams induced by O_1 and O_2.

$$M_1 \sim \bar{s}_L \frac{\gamma_\mu, \hat{k}+\hat{q}+m_c, \gamma_\alpha, \hat{k}+m_c, \gamma_\beta, \hat{k}-\hat{r}+m_c, \gamma_\mu}{[(\hat{k}+\hat{q})^2-m_c^2][(\hat{k}-\hat{r})^2-m_c^2][k^2-m_c^2]} b_L, \tag{2}$$

$$M_2 \sim \bar{s}_L \frac{\gamma_\mu, \hat{k}+\hat{r}+m_c, \gamma_\beta, \hat{k}+m_c, \gamma_\alpha, \hat{k}-\hat{q}+m_c, \gamma_\mu}{[(\hat{k}+\hat{r})^2-m_c^2][(\hat{k}-\hat{q})^2-m_c^2][k^2-m_c^2]} b_L, \tag{3}$$

$$J_{\alpha\beta} = M_1 + M_2. \tag{4}$$

On start-up MatrixExp package by default defines some parameters, in particular variables $u1...u9$ are defined as scalars, automatic sorting is switched On, automatic integration is switched Off.

First of all we add mc as scalar

Vaddscalars[**mc**] (5)

Next we define the rules for scalar products $q^2 = 0$, $r^2 = 0$

$$\mathbf{Vsetrules}[\{\mathbf{q},\mathbf{0}\},\{\mathbf{r},\mathbf{0}\}] \tag{6}$$

Finally we call function **Vcalc** for calculation of M_1 and M_2 accordingly

$$\begin{aligned}\mathbf{M1} &= \mathbf{Vcalc}[\frac{\mathbf{G}[\mathbf{l1},\mathbf{R},\{\mathbf{mu}\},\mathbf{k}+\mathbf{q}+\mathbf{mc},\{\mathbf{al}\},\mathbf{k}+\mathbf{mc},\{\mathbf{bet}\},\mathbf{k}-\mathbf{r}+\mathbf{mc},\{\mathbf{mu}\},\mathbf{L}]}{((\mathbf{k}-\mathbf{r})^{\mathbf{2}}-\mathbf{mc^2})((\mathbf{k}+\mathbf{q})^{\mathbf{2}}-\mathbf{mc^2})(\mathbf{k^2}-\mathbf{mc^2})},\\ &\quad \mathbf{integrating}\to\mathbf{k},\mathbf{dofeynman}\to\mathbf{True},\mathbf{sorting}\to\{\{\mathbf{al}\},\{\mathbf{bet}\},\mathbf{r},\mathbf{q}\}]\end{aligned} \tag{7}$$

$$\begin{aligned}\mathbf{M2} &= \mathbf{Vcalc}[\frac{\mathbf{G}[\mathbf{l1},\mathbf{R},\{\mathbf{mu}\},\mathbf{k}+\mathbf{r}+\mathbf{mc},\{\mathbf{bet}\},\mathbf{k}+\mathbf{mc},\{\mathbf{al}\},\mathbf{k}-\mathbf{q}+\mathbf{mc},\{\mathbf{mu}\},\mathbf{L}]}{((\mathbf{k}+\mathbf{r})^{\mathbf{2}}-\mathbf{mc^2})((\mathbf{k}-\mathbf{q})^{\mathbf{2}}-\mathbf{mc^2})(\mathbf{k^2}-\mathbf{mc^2})},\\ &\quad \mathbf{integrating}\to\mathbf{k},\mathbf{dofeynman}\to\mathbf{True},\mathbf{sorting}\to\{\{\mathbf{al}\},\{\mathbf{bet}\},\mathbf{r},\mathbf{q}\}]\end{aligned} \tag{8}$$

where the option **dofeynman** $\to$ **True** turns On Feynman parameterization, and the option **integrating** $\to$ **k** defines the integration variable (and turns On integration). Received **M1** and **M2** are grouped in sorted resulting matrix expressions (e.g. $\mathbf{G}[\mathbf{l1},\{\mathbf{al}\},\{\mathbf{bet}\},\mathbf{q}]$, $\mathbf{G}[\mathbf{l1},\{\mathbf{al}\},\{\mathbf{bet}\},\mathbf{r}]$, $\mathbf{G}[\mathbf{l1},\{\mathbf{al}\},\{\mathbf{bet}\}]$, $\mathbf{G}[\mathbf{l1},\{\mathbf{bet}\},\mathbf{r},\mathbf{q}]$...) and include scalars $u1$, $u2$ and $u3$, which arose due to Feynman parameterization. (actually one of the scalars will be missing (in this case - $u1$) since during Feynman parameterization, MatrixExp package automatically carries out integration ($\int X\cdot\delta(1-u1-u2-u3)\mathrm{d}u1$), replacing $u1$ by $1-u2-u3$). After that we consider the sum $J_{\alpha\beta} = M1 + M2$

$$\mathbf{Jab} = \mathbf{Vcalc}[\mathbf{M1}+\mathbf{M2},\mathbf{sorting}\to\{\{\mathbf{al}\},\{\mathbf{bet}\},\mathbf{r},\mathbf{q}\}] \tag{9}$$

or, using the built in function of MATHEMATICA (since **M1** and **M2** are already sorted in the identical order)

$$\mathbf{Jab} = \mathbf{Simplify}[\mathbf{M1}+\mathbf{M2}] \tag{10}$$

In the further we need to simplify expression (9, 10), using transversality of photon and gluon, Ward identities, etc., and after representing through $E[\alpha,\beta,r]$ we receive the form presented in [10].

Acknowledgements

The work was partially supported by the ANSEF-05-PS-hepth-813-100 program.

References

[1] Matthias Jamin, Markus E. Lautenbacher, "Tracer version 1.1: A mathematica package for gamma algebra in arbitrary dimensions," Comput. Phys. Commun. **74**: (1993) 265–288.

[2] S. Wolfram, Mathematica - A System for Doing Mathematics by Computer, (Addison-Wesley Publishing Company Inc., 1988).

[3] http://www.maplesoft.com/; B. Char, O. Goddies, G. Gonnet, B. Leong, M. Monagan, and S. Watt, Maple V Language Reference Manual (Springer-Verlag, 1991).

[4] P. Breitenlohner and D. Maison, Comm. Math. Phys. **52** (1977) 11, 39, 55.

[5] G. Bonneau, "Consistency in dimensional regularization with γ_5," Phys. Lett. **B96** (1980) 147–150; "Preserving canonical Ward identities in dimensional regularization with a non-anticommuting γ_5," Nucl. Phys. **B177** (1981) 523–527.

[6] A.J. Buras and P.H. Weisz, "QCD nonleading corrections to weak decays in dimensional regularization and 't Hooft-Veltman schemes," Nucl. Phys. **B333** (1990) 66–99.

[7] G. t'Hooft and M. Veltman, "Regularization and renormalization of gauge fields," Nucl. Phys. **B44** (1972) 189–213.

[8] D.A. Akyeampong and R. Delbourgo, Nuovo Cim. **17A** (1973) 578; **18A** (1973) 94; **19A** (1974) 219.

[9] M.E. Peskin, D.V. Schroeder, Introduction to quantum field theory, (Addison-Wesley Publishing Company).

[10] C. Greub, T. Hurth, and D. Wyler, "Virtual corrections to the decay $b \to s\gamma$," Phys. Lett. **B380** (1996) 385–392, Phys. Rev. **D 54** (1996) 3350–3364.

TEST RUN OUTPUT

In the package $d = 4 - 2\varepsilon$ denotes the space-dimension, so avoid its usage as momenta.[4]

$$\begin{aligned} \text{In[361]} &:= \mathbf{Vcalc}[\mathbf{G}[\mathbf{l1}, \mathbf{a}, \{\mathbf{mu}\}, \mathbf{b}, \{\mathbf{mu}\}, \mathbf{a}]] \\ \text{Out[361]} &= (-2 + \mathrm{d})\, \mathrm{G[l1, b]}\, \mathrm{a.a} - 2\,(-2 + \mathrm{d})\, \mathrm{G[l1, a]}\, \mathrm{a.b} \\ \text{Timing} &:= \mathbf{0.05} \text{ seconds} \end{aligned} \tag{11}$$

To be able to use scalars in expressions we must declare scalars, otherwise by default they will be treated as momenta

$$\begin{aligned} \text{In[362]} &:= \mathbf{Vaddscalars}[\mathbf{ma}, \mathbf{mb}, \mathbf{mc}] \\ \text{Out[362]} &= \{\mathrm{u1}, \mathrm{u2}, \mathrm{u3}, \mathrm{u4}, \mathrm{u5}, \mathrm{u6}, \mathrm{u7}, \mathrm{u8}, \mathrm{u9}, \mathrm{ma}, \mathrm{mb}, \mathrm{mc}\} \\ \text{Timing} &:= \mathbf{0.00} \text{ seconds} \end{aligned} \tag{12}$$

We can use either global sorting (via $\mathbf{Vsorton}[\mathbf{True}]$ and $\mathbf{Vsetsortlist}[\mathbf{a}, \mathbf{b}, \mathbf{c}]$, which will be automatically the default sorting for all expressions) or local sorting

$$\begin{aligned} \text{In[363]} &:= \mathbf{r0} = \mathbf{Vcalc}[\mathbf{G}[\mathbf{l1}, \mathbf{R}, \mathbf{a} + \mathbf{ma}, \mathbf{b}, \mathbf{c} + \mathbf{mc}, \mathbf{L}]] \\ \text{Out[363]} &= \mathrm{ma\,mc\,G[l1, b, L]} + \mathrm{G[l1, a, b, c, L]} \\ \text{Timing} &:= \mathbf{0.00} \text{ seconds} \end{aligned} \tag{13}$$

$$\begin{aligned} \text{In[364]} &:= \mathbf{r1} = \mathbf{Vcalc}[\mathbf{G}[\mathbf{l1}, \mathbf{R}, \mathbf{a} + \mathbf{ma}, \mathbf{b}, \mathbf{c} + \mathbf{mc}, \mathbf{L}], \\ &\qquad \mathbf{sorting} \to \{\mathbf{c}, \mathbf{b}, \mathbf{a}\}] \\ \text{Out[364]} &= -\mathrm{G[l1, c, b, a, L]} + 2\,\mathrm{G[l1, c, L]}\,\mathrm{a.b} + \\ &\qquad \mathrm{G[l1, b, L]}\,(\mathrm{ma\,mc} - 2\,\mathrm{a.c}) + 2\,\mathrm{G[l1, a, L]}\,\mathrm{b.c} \\ \text{Timing} &:= \mathbf{0.02} \text{ seconds} \end{aligned} \tag{14}$$

$$\begin{aligned} \text{In[365]} &:= \mathbf{r2} = \mathbf{Vcalc}[\mathbf{r1}, \mathbf{sorting} \to \{\mathbf{b}, \mathbf{c}, \mathbf{a}\}] \\ \text{Out[365]} &= \mathrm{G[l1, b, c, a, L]} + \mathrm{G[l1, c, L]}\,\mathrm{a.b} + \\ &\qquad \mathrm{G[l1, b, L]}\,(\mathrm{ma\,mc} - 2\,\mathrm{a.c}) \\ \text{Timing} &:= \mathbf{0.00} \text{ seconds} \end{aligned} \tag{15}$$

[4]The last row "Timing: " shows the tested time of the calculation. Results for Pentium IV 2.8GHz.

$$\begin{aligned}
\text{In[366]} &:= \mathbf{r3 = Vcalc[r2, sorting \to \{a, b, c\}]} \\
\text{Out[366]} &= \text{ma mc G[l1, b, L]} + \text{G[l1, a, b, c, L]} \\
\text{Timing} &:= \mathbf{0.02}\ \text{seconds}
\end{aligned} \tag{16}$$

Here L and R denote $(1-\gamma_5)/2$ and $(1+\gamma_5)/2$ respectively. As we see, the Output (including numerical coefficients and scalars) may be used as Input without any additional transformations (e.g. $r1$, which is Out[364] is used as input in (15)).

To use integration and Feynman parameterization, lets define some rules for scalar product (we use arbitrary expressions and definitions here, just as mathematical expressions)

$$\begin{aligned}
\text{In[367]} &:= \mathbf{Vsetrules[\{a, ma^2\}, \{b, mb^2\}, \{a, b, ma * mb/2\}]} \\
\text{Out[367]} &= \{\{\text{a}, \text{ma}^2\}, \{\text{b}, \text{mb}^2\}, \{\text{a}, \text{b}, \frac{\text{ma mb}}{2}\}\} \\
\text{Timing} &:= \mathbf{0.02}\ \text{seconds}
\end{aligned} \tag{17}$$

$$\begin{aligned}
\text{In[368]} &:= \mathbf{res = Vcalc[}\frac{\mathbf{G[l1, R, a, k, \{mu\}, b + mb, k, \{mu\}, L}}{\mathbf{((k-a)^2 - ma^2)((k-b)^2 - mb^2)}}, \\
&\quad \mathbf{integrating \to k, dofeynman \to True]} \\
\text{Out[368]} &= -((-2+\text{d})\text{mb}(4\pi)^{-2+\text{eps}} \\
&\quad \left(-\frac{1}{-\text{ma}^2(-1+\text{u2})^2 + \text{ma mb}\,(-1+\text{u2}) - \text{mb}^2\text{u2}^2}\right)^{-1+\text{eps}} \\
&\quad (-(-2+\text{eps})\,(-\text{ma}^2(-1+\text{u2})^2 + \text{ma mb}\,(-1+\text{u2})\,\text{u2} - \\
&\quad \text{mb}^2\,\text{u2}^2)) + (-1+\text{eps})\,(\text{ma}^2\,(-1+\text{u2})^2 + \\
&\quad \text{ma mb}\,(-1+\text{u2})\,\text{u2}) - \text{mb}^2\,\text{u2}^2) + (-1+\text{eps}) \\
&\quad (\text{ma}^2(-1+\text{u2})^2 - \text{ma mb}\,(-1+\text{u2})\,\text{u2} + \text{mb}^2\,\text{u2}^2)) \\
&\quad \text{G[l1, a, L]}\,\text{Gamma}[-1+\text{eps}])/(-\text{ma}^2\,(-1+\text{u2})^2 + \\
&\quad \text{ma mb}\,(-1+\text{u2})\,\text{u2} - \text{mb}^2\,\text{u2}^2) \\
\text{Timing} &:= \mathbf{0.13}\ \text{seconds}
\end{aligned} \tag{18}$$

We see, that all the substitutions after parameterization are done, a Dirac-delta integration over parameter $u1$ is done, integration over k is done, and we have a final result.

Here we calculate the spur in two ways. First we calculate the expression $r0$, followed by calculation of the spur of the result, and then we directly calculate the spur of the initial expression $r0$.

$$
\begin{aligned}
\text{In}[369] := \; & \mathbf{r0} = \mathbf{G}[\mathbf{l1}, \mathbf{a} + \mathbf{ma}, \mathbf{b} + \mathbf{mb}, \mathbf{a} + \mathbf{ma}, \mathbf{c} + \mathbf{mc}, \mathbf{f} + \mathbf{ma}, \mathbf{L}]; \\
& \mathbf{rr} = \mathbf{Vcalc}[\mathbf{r0}] \\
& \mathbf{Vcalc}[\mathbf{rr}, \mathbf{spur} \to \mathbf{True}] \\
& \mathbf{Vcalc}[\mathbf{r0}, \mathbf{spur} \to \mathbf{True}] \\
\text{Out}[370] = \; & 3\,\text{ma}^3\,\text{mb}\,\text{mc}\,\text{G}[\text{l1}, \text{L}] + 3\,\text{ma}^2\,\text{mb}\,\text{mc}\,\text{G}[\text{l1}, \text{a}, \text{L}] + \\
& 3\,\text{ma}^3\,\text{mb}\,\text{G}[\text{l1}, \text{c}, \text{L}] + 3\,\text{ma}^2\,\text{mb}\,\text{mc}\,\text{G}[\text{l1}, \text{f}, \text{L}] + \\
& 3\,\text{ma}^2\,\text{mb}\,\text{G}[\text{l1}, \text{a}, \text{c}, \text{L}] + 3\,\text{ma}\,\text{mb}\,\text{mc}\,\text{G}[\text{l1}, \text{a}, \text{f}, \text{L}] + \\
& 3\,\text{ma}^2\,\text{mb}\,\text{G}[\text{l1}, \text{c}, \text{f}, \text{L}] + 3\,\text{ma}\,\text{mb}\,\text{G}[\text{l1}, \text{a}, \text{c}, \text{f}, \text{L}] \\
\text{Out}[371] = \; & 6\,\text{ma}\,\text{mb}\,(\text{ma}\,\text{a.c} + \text{mc}\,\text{a.f} + \text{ma}\,(\text{ma}\,\text{mc} + \text{c.f})) \\
\text{Out}[372] = \; & 6\,\text{ma}\,\text{mb}\,(\text{ma}\,\text{a.c} + \text{mc}\,\text{a.f} + \text{ma}\,(\text{ma}\,\text{mc} + \text{c.f})) \qquad (19) \\
\text{Timing} := \; & \mathbf{0.09}\ \text{seconds}
\end{aligned}
$$

And finally here are the timings for calculations of (7–9): **1.86**, **1.64** and **6.00** seconds respectively.