Application of Gröbner basis method to evaluation of Feynman diagrams

OLEG TARASOV

DESY Zeuthen

Content

- Generalized recurrence relations
- Transformation of recurrence relations to a system of differential equations - 2 methods
- Buchberger's algorithm - short introduction
- Gröbner basis for one-loop self-energy integrals
- Gröbner basis for two-loop sun-set integrals
- Conclusions

Presentation based on my paper *"Reduction of Feynman graph amplitudes to a minimal set of basic integrals"*,

Acta Physica Polonica, v **B29** (1998) 2655

Recurrence relations - main tool to calculate Feynman integrals:

- First relation for Feynman integrals derived from *integration by parts* identities in momentum space was given by G.'tHooft and M. Veltman, Nucl.Phys. **B44**(1972) 189.
- A systematic method was proposed by Chetyrkin and Tkachov (Nucl.Phys **B192** (1981) 159) now called *integration by parts method*

To find relations between Feynman integrals G.'tHooft and M. Veltman used identity:

$$\int d^d k_1 \ldots d^d k_L \; \frac{\partial}{\partial k_{r\mu}} \; f(\{k\}) \equiv 0,$$

For Feynman integral in d dimensions with N lines

$$\int d^d k_1 \ldots d^d k_L \; \frac{\partial}{\partial k_{r\mu}} \; \frac{R_{\{\mu\}}(\{k\},\{p\})}{P_{\bar{k}_1,m_1}^{\nu_1} \ldots P_{\bar{k}_N,m_N}^{\nu_N}} \equiv 0,$$

where R is an arbitrary *tensor* polynomial and

$$P_{\bar{k}_j,m_j} = (\bar{k}_j - p_j)^2 - m_j^2 + i\epsilon$$

After differentiation scalar products can be written in terms of inverse propagators :

$$k_i p_l = \tfrac{1}{2}(k_i^2 + p_l^2 - (k_i - p_l)^2).$$

$==>$ relations connecting integrals with different powers of propagators.

Chetyrkin and Tkachov demonstrated that with the help of these relations Feynman diagrams can be expressed in terms of restricted number of the so-called master integrals.

Not always k_i^2, $(k_i - p_l)^2$ are in denominator $==>$ problem of irreducible numerators. Each irreducible numerator needs additional recurrence relations.

Another representation of scalar products can be used:

$$p_{l\mu}\int k_{i\mu} d^d k_1 \ldots d^d k_L \; \frac{1}{P_{\bar{k}_1,m_1}^{\nu_1} \ldots P_{\bar{k}_N,m_N}^{\nu_N}} =$$

$$p_{l\mu} T_\mu(\{p\},\{\partial\},\mathbf{d}^+) \int d^d k_1 \ldots d^d k_N \; \frac{1}{P_{\bar{k}_1,m_1}^{\nu_1} \ldots P_{\bar{k}_N,m_N}^{\nu_N}},$$

where $\{\partial\}$ are derivatives w.r.t. masses. Such representation was proposed by O. T., (Phys.Rev.**D54** (1996) 6479). It gives another kind of recurrence relations for Feynman integrals.

No problems with irreducible numerators! Only recurrence relation w.r.t. d is needed for all irreducible numerators.

Combination of both representations for scalar products gives many recurrence relations (generalized recurrence relations) connecting integrals with different indices ν_j and shifts of d. They extend number of recurrency parameters: $\{\nu_j\} \to \{\nu_j, d\}$

Problems:

- how to use these identities in a systematic way for any integral?
- how many master integrals are needed for the reduction of any integral?

An approach to solve these problems was formulated in my paper

"Reduction of Feynman graph amplitudes to a minimal set of basic integrals" ,

Acta Physica Polonica, v **B29** (1998) 2655; hep-ph/9812250

The main idea – use of Gröbner basis technique,

i.e. to solve the problem we should find Gröbner bases for the ideal generated by generalized recurrence relations

Different realizations of this idea are possible

For example, set of recurrence relations can be transformed into an overdetermined **system of differential equations**. Then reduction of this system to

involutive form or standard form or to a differential Gröbner basis

solves the problem of finding optimal set of recurrence relations and a minimal set of master integrals

This reduction can be done by using existing standard software.

In MAPLE: package Groebner, DEtools and it's subpackage Rif, package diffalg and it's subpackage Rosenfeld_Grobner

Also, recurrence relations can be written in terms of **difference operators** and therefore packages for **Ore algebra** can be used.

Reduction of generalized recurrence relations

Two algorithms are possible Algorithm I

- Powers of propagator ν_j keep arbitrary
- If term with $\nu_i - 1$ occurs in the recurrence relation then make the substitution $\nu_i \to \nu_i + 1$.
- Propagators with positive 'shifts' of indices represent as:

$$\frac{1}{[(k_i - q_s)^2 - m_l^2]^{\nu_l + r}} =$$
$$\frac{\Gamma(\nu_l)}{\Gamma(\nu_l + r)} \frac{\partial^r}{(\partial m_l^2)^r} \frac{1}{[(k_i - q_s)^2 - m_l^2]^{\nu_l}}.$$

- Integrals with different dimensional shifts represent as different functions

$$I^d = V^1, \quad I^{d+2} = V^2, \quad I^{d+4} = V^3, \ldots$$

- Apply Gröbner basis package for overdetermined systems of differential equations to obtain minimal set of recurrence relations (Gröbner basis !) and set of master integrals

This algorithm can be used to find minimal set of recurrence relations and master integrals for integrals of a given topology

Algorithm II

- Derive recurrence relations for the integral $I^{(d)}_{\nu_1 \ldots \nu_N}$ of given topology with arbitrary ν's.
- From these relations produce relations corresponding to integrals with different values of sum of powers of propagators

$$S = \sum_{i=1}^{N} \nu_i, \quad S = N, N+1, N+2, \ldots.$$

These relations will include integrals with contracted lines !!

- Scalar propagators with $\nu_i > 1$ represent as

$$\frac{1}{[(k_i - q_s)^2 - m_l^2]^{r+1}} = \frac{1}{\Gamma(r+1)} \frac{\partial^r}{(\partial m_l^2)^r} \frac{1}{[(k_i - q_s)^2 - m_l^2]},$$

- Integrals with shifts in d represent as different functions

- First apply Gröbner basis package to solve relations for integrals with minimal number of lines
- Increase number of lines by 1 and produce recurrence relations connecting these integrals and integrals with contracted lines.
- Substitute relations for integrals with minimal number of lines into these relations
- Again apply Gröbner basis package. The minimal set of recurrence relations and set of master integrals will be produced.
- Increase number of lines by 1 and produce different recurrence relations ...

The second algorithm more suitable for considering concrete integrals occurring in calculating Feynman diagrams.

The subject of our condideration will be a set of generalized recurrence relations or integration by parts polynomial relations

$$f_1(x_1, ..., x_n) = 0,$$
$$f_2(x_1, ..., x_n) = 0,$$
$$....$$
$$f_s(x_1, ..., x_n) = 0.$$

Our task will be to find another set of relations with some nice properties and to provide an algorithm allowing to express all required Feynman integrals in terms of "master" integrals by using this set of relations.

This is quite general mathematical problem solvable by Gröbner basis technique.

GRÖBNER BASIS METHOD

To find Gröbner basis Buchberger proposed an algorithm (Buchberger, 1965).

BUCHBERGER ALGORITHM

Term ordering

For polynomials in $x_1, x_2, \ldots x_n$ we define the *pure lexicographic ordering*

$x_1^{i_1}, x_2^{i_2}, \ldots x_n^{i_n} \prec x_1^{j_1}, x_2^{j_2}, \ldots x_n^{j_n}$ if and only if,
for some $l \in \{1, \ldots, n-1\}, i_k = j_k$ for all $k < l$ and $i_l < j_l$.

We choose in x, y, z:

$$1 \prec z \prec z^2 \prec \ldots \prec y \prec yz \prec yz^2 \prec \ldots$$
$$\prec y^2 \prec y^2 z \prec y^2 z^2 \prec \ldots \prec x \prec xz \prec xz^2 \prec \ldots$$
$$\prec xy \prec xy^2 \prec \ldots \prec x^2 \prec \ldots$$

Other orderings are possible: only

(i) $1 \prec t$ for every term $t \neq 1$

(ii) $s \prec t \Longrightarrow s \cdot u \prec t \cdot u$ for all terms s, t, u

For a nonzero polynomial f:

$$\textit{leading term } \mathrm{lt(f)},$$
$$\textit{leading coefficient } \mathrm{lc(f)}$$
$$\textit{leading monomial } \mathrm{lt(f)} = \mathrm{lc(f)} \cdot \mathrm{lm(f)}$$

Reduction, normal form

We say, f *reduces to* $\tilde{f}$ *modulo* g (notation $f \to_g \tilde{f}$) if there exists a term t in f that is divisible by the leading term of g and

$$\tilde{f} = f - \frac{t}{\mathrm{lt}(g)} \cdot g$$

Let $G = g_1, g_2, \ldots, g_m$ be a set of polynomials.

We say f *reduces to* $\tilde{f}$ *modulo* G if there exists a polynomial g_i in G such that $f \longrightarrow_{g_i} \tilde{f}$.

A *normal form of* f *with respect to* G is a polynomial obtained after a finite number of reductions which contains no term anymore that is divisible by leading terms of polynomials of G.

Notation: $\mathrm{normalf}(f, G)$

S-Polynomial

For polynomials f and g we define

$$\mathbf{spoly(f,g) = lcm(lt(f), lt(g)), \left(\frac{f}{lt(f)} - \frac{g}{lt(g)}, \right)}$$

where $\mathrm{lcm}((p),(q))$ denotes the least common multiple of polynomials p and q.

An algorithmic characterization of a Gröbner basis:

A finite set G of polynomials

is a Gröbner basis if and only if

$$\mathrm{normalf}(\mathrm{spoly}(f,g), G) = 0$$

for all pairs (f,g) in G

This definition motivates the **Buchberger algorithm**.

Buchberger algorithm:

```
procedure   grobnerBasis(G)
  GB ← G    B ← {(f,g) | f,g ∈ G, f ≠ g}
while B ≠ 0 do   (f,g) ← select a pair in B
                 B ← B \ {(f,g)}
                 h ← normalf(spoly(f,g), GB)
                 if h ≠ 0 then
                   GB ← GB ∪ {h}
                   B ← B ∪ {(f,g) | f ∈ GB}
                 fi
          od
return(GB)  end
```

ONE-LOOP EXAMPLE

```
    |\^/|     Maple 8 (IBM INTEL LINUX)
._|\|   |/|_. Copyright (c) 2002 by Waterloo Maple Inc.
 \  MAPLE  /  All rights reserved. Maple is a registered trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
#
#           ___k1____
#          /         \
#     ____/           \_____
#         \           /
#          \_________/
#             k1-q1
>
>
> j1:=0:
> j2:=0:
>
#  z1= m_1^2,    z2 = m_2^2,    q1q1 = q_1^2
#
#  i2(a1,a2) =
#       \int d^d k_1 /( k1^2-m1^2)^(j1+a1)/( (k1-q1)^2-m2^2)^(j2+a2)
#
```

```
>
> eq1:=(  - i2(0,2) - i2(0,2)*j2 - 3*i2(1,1) - 2*i2(1,1)*j1
>          - i2(1,1)*j2 + i2(1,1)*d + i2(1,2)*q1q1*j2 + i2(1,2)*q1q1
>          - i2(1,2)*z1*j2 - i2(1,2)*z1 - i2(1,2)*j2*z2 - i2(1,2)*z2
>          - 2*i2(2,1)*j1*z1 - 2*i2(2,1)*z1 ):
>
> eq2 := (  - i2(0,2) - i2(0,2)*j2 - i2(1,1)*j1 + i2(1,1)*j2
>          + i2(1,2)*q1q1*j2 + i2(1,2)*q1q1 - i2(1,2)*z1*j2 - i2(1,2
>         )*z1 + i2(1,2)*j2*z2 + i2(1,2)*z2 + i2(2,0) + i2(2,0)*j1
>          - i2(2,1)*q1q1*j1 - i2(2,1)*q1q1 - i2(2,1)*j1*z1 + i2(2,1
>         )*j1*z2 - i2(2,1)*z1 + i2(2,1)*z2 ):
>
>
> read `i2.mpr`:
>
#################################### Defining system of equations
> eq1 := eq1;
                (-2 + d) T2(z2)
eq1 := 1/2 ---------------- - 3 I2(z1, z2) + I2(z1, z2) d
                     z2

      / d              \          / d              \       / d              \
    + |--- I2(z1, z2)| q1q1 - |--- I2(z1, z2)| z1 - |--- I2(z1, z2)| z2
      \dz2             /          \dz2             /       \dz2             /
```

```
          / d              \
      - 2 |--- I2(z1, z2)| z1
          \dz1             /

> eq2 := eq2;
             (-2 + d) T2(z2)   / d              \         / d              \
eq2 := 1/2 ---------------- + |--- I2(z1, z2)| q1q1 - |--- I2(z1, z2)| z1
                   z2          \dz2             /         \dz2             /

        / d              \           (-2 + d) T1(z1)   / d              \
      + |--- I2(z1, z2)| z2 - 1/2 ---------------- - |--- I2(z1, z2)| q1q1
        \dz2             /                 z1          \dz1             /

        / d              \       / d              \
      - |--- I2(z1, z2)| z1 + |--- I2(z1, z2)| z2
        \dz1             /       \dz1             /

>
# Relations for tadpole integrals
>
> eq3:=z1*diff(T1(z1),z1) - (d/2-1)*T1(z1):
> eq4:=z2*diff(T2(z2),z2) - (d/2-1)*T2(z2):
>
#####################################################################
#
```

```
# We use package 'Rif'    - improved version of 'Standard Form'
>
>
> with(DEtools):
>
#########################################################################
>
>
> bas:=rifsimp([eq1,eq2,eq3,eq4],[I2]):
>
```

```
> basa:=bas[Solved];
          d
basa := [--- I2(z1, z2) = 1/2 (4 T2(z2) z1 - 2 T2(z2) z1 d
          dz1

                                                                        2
     + 6 I2(z1, z2) q1q1 z1 - 2 I2(z1, z2) d z2 z1 + 2 I2(z1, z2) d z1

                                                                      2
     - q1q1 T1(z1) d + z1 T1(z1) d + 6 I2(z1, z2) z2 z1 - 6 I2(z1, z2) z1

     + 2 q1q1 T1(z1) - 2 z1 T1(z1) + T1(z1) z2 d - 2 I2(z1, z2) d q1q1 z1

                     /                                2                  2    2
     - 2 T1(z1) z2) /  ((-2 z2 z1 - 2 q1q1 z1 + z1  - 2 z2 q1q1 + q1q1  + z2 )
                   /

          d                                                             2
     z1), --- I2(z1, z2) = (2 T2(z2) z1 - T2(z2) z1 d - 2 I2(z1, z2) d z2
          dz2

     + 2 I2(z1, z2) d z2 q1q1 - 2 T2(z2) q1q1 - 4 T1(z1) z2 + 2 T1(z1) z2 d

                                    2
     + 2 T2(z2) z2 + 6 I2(z1, z2) z2  - T2(z2) d z2 + T2(z2) d q1q1

                                                                      /
     - 6 I2(z1, z2) z2 z1 - 6 I2(z1, z2) z2 q1q1 + 2 I2(z1, z2) d z2 z1)  /  (
                                                                     /

        2                           2                2         2          3
     4 z2  z1 + 4 q1q1 z2 z1 - 2 z1  z2 + 4 q1q1 z2  - 2 q1q1  z2 - 2 z2 ),
```

```
    d                  T1(z1) d - 2 T1(z1)    d                  T2(z2) d - 2 T2(z2)
    --- T1(z1) = 1/2 --------------------, --- T2(z2) = 1/2 --------------------]
    dz1                        z1           dz2                        z2

>
>
>
> Basis:={basa[1],basa[2],basa[3],basa[4]}:
>
>
########################   Reducing integral to master integrals
#
#          __o___o__
#         /         \
#    ____/           \_____
#        \           /
#         \____o____/
>
> FindI21 := {I21(z1,z2)= diff(I2(z1,z2),z1$2,z2)/2 }:
>
>
>
> EquI21 := convert(Basis union FindI21,list) :
> findI21 := rifsimp(EquI21,[I21,I2]):
```

```
bytes used=4000332, alloc=3407248, time=0.18
>
> findi21 :=subs(convert( findI21[Solved],set),I21(z1,z2)):
> funs := convert(selectfun(findi21,[I2,T1,T2]),list):
bytes used=8000580, alloc=4848776, time=0.35
> findi21 :=collect(findi21,funs,factor);
                                    2   2         2  2    2   3        2     2
findi21 := 1/2 (-2 + d) (-2 z1 d  z2  + z2 z1  d  + d  z2  - z2 d  q1q1

                2           2                                      2          3
    - 10 z2 z1  d - 4 d z2  q1q1 - 4 z2 z1 d q1q1 + 14 z2 d q1q1  - 10 d  z2

                2       2          3       3       3            2        2
    + 20 z1 d z2  - 3 z1  q1q1 + z1  + 26 z2  - q1q1  + 3 z1 q1q1  + 24 z1  z2

                2                          2          2                /
    + 17 q1q1 z2  + 18 q1q1 z2 z1 - 42 q1q1  z2 - 51 z2  z1) T2(z2)  /  (z2
                                                                   /

     3                         2     2        4        4             3
   %1 ) - 1/8 (-2 + d) (-24 z2  q1q1  + 96 z1  - 39 z1  d - 4 d q1q1  z2

              2     2         3                3          4       4
    + 6 d z2  q1q1  - 4 d z2  q1q1 - 4 z1 z2  d + d q1q1  + d z2
```

```
          3                        2                      2       2  2
- 8 z1 q1q1  d + 12 z1 z2 d q1q1  - 100 d z2 q1q1 z1  + 8 z1  d  z2 q1q1

      3  2             2  2     2        2       2        3
- 8 z1  d  q1q1 + 4 z1  d  q1q1  - 26 z1  d q1q1  + 72 z1  d q1q1

      3  2           2  2   2        3                2   2       4  2
- 8 z1  d  z2 + 4 z1  d  z2  + 76 z1  d z2 - 34 d z2  z1  + 4 z1  d

        3             3              2   2          2   2        3
- 176 z1  z2 - 164 z1  q1q1 + 60 z2  z1  + 36 q1q1  z1  + 16 z2  q1q1

           3       4         4        3                     2
+ 16 z2 q1q1  - 4 z2  - 4 q1q1  + 24 z2  z1 + 288 z2 q1q1 z1

           2              2                3                /    2   3
- 12 q1q1 z2  z1 - 48 q1q1  z2 z1 + 36 q1q1  z1) T1(z1)  /  (z1  %1 ) -
                                                       /

                                 2       2
1/2 (d - 3) (d - 5) (3 z1 d z2  - d z1  q1q1 + 2 z2 z1 d q1q1 + d q1q1

        2             2              3          2         3             2
- d q1q1  z1 - d z2  q1q1 + d z1  - 3 z2 z1  d - d z2  + z2 d q1q1
```

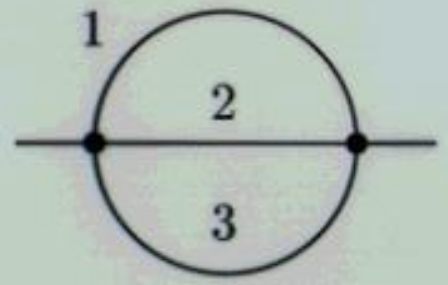
```
                                3                2          2          3                2
     - 12 q1q1 z2 z1 - 8 q1q1  + 12 z1 q1q1  - 8 q1q1  z2 + 4 z2  + 12 q1q1 z2

           2            2          3                  /   3
     - 12 z2  z1 + 12 z1  z2 - 4 z1 ) I2(z1, z2)  /  %1
                                                 /

                                2                     2     2
%1 := -2 z2 z1 - 2 q1q1 z1 + z1  - 2 z2 q1q1 + q1q1  + z2

>
>
> quit;
bytes used=8918292, alloc=4848776, time=0.39
```

Two-loop example



$$Q_2 = x_1 k_{1\nu} k_{1\rho} + x_2 k_{1\nu} k_{2\rho} + x_3 k_{1\rho} k_{2\nu} + x_4 k_{2\nu} k_{2\rho} + x_5 k_{1\nu} q_{1\rho}$$
$$+ x_6 k_{1\rho} q_{1\nu} + x_7 k_{2\nu} q_{1\rho} + x_8 k_{2\rho} q_{1\nu} + x_9 q_{1\rho} q_{1\nu},$$
$$S = w_1 g_{\mu,\nu} q_{1\rho} + w_2 g_{\rho,\nu} q_{1\mu} + w_3 d_{\mu,\rho} q_{1\nu} + w_4 q_{1\mu} q_{1\nu} q_{1\rho},$$

$$\int d^d k_1 d^d k_2 \; \frac{\partial}{\partial k_{1\mu}} \; \frac{S Q_2}{c_1^{\nu_1} c_2^{\nu_2} c_3^{\nu_3}} \equiv 0, \quad \int d^d k_1 d^d k_2 \; \frac{\partial}{\partial k_{2\mu}} \; \frac{S Q_2}{c_1^{\nu_1} c_2^{\nu_2} c_3^{\nu_3}} \equiv 0$$

where

$$c_1 = k_1^2 - m_1^2, \;\; c_2 = (k_1 - k_2)^2 - m_2^2, \;\; c_3 = (k_2 - q)^2 - m_3^2$$

Integrals with irreducible numerators

$$\int\int d^dk_1 d^dk_2 \frac{(k_2q)^\alpha (k_1k_2)^\beta}{c_1^{\nu_1} c_2^{\nu_2} c_3^{\nu_3}}$$

were expressed in terms of integrals with shifted dimension.

```
eq1:=(  - sun(0,0,2,1) - 3*sun(0,1,1,1) + sun(0,1,1,1)*d +
        sun(0,1,2,0) - sun(0,1,2,1)*q1q1 - sun(0,1,2,1)*z1
 - sun(0,1,2,1)*z2 + sun(0,1,2,1)*z3 - 2*sun(0,2,1,1)*z1
 + 4*sun(1,1,3,2)*q1q1 + 2*sun(1,2,2,2)*q1q1 ):

eq2:=(  - sun(0,0,2,1) + sun(0,1,2,0) - sun(0,1,2,1)*q1q1 -
        sun(0,1,2,1)*z1 + sun(0,1,2,1)*z2 + sun(0,1,2,1)*z3
 + sun(0,2,0,1) - sun(0,2,1,0) + sun(0,2,1,1)*q1q1
 - sun(0,2,1,1)*z1 + sun(0,2,1,1)*z2 - sun(0,2,1,1)*z3
 + 4*sun(1,1,3,2)*q1q1 - 4*sun(1,3,1,2)*q1q1 );

eq3:=  .  .  .  .  .  .
```

$z_1 \equiv m_1^2$

$z_2 \equiv m_2^2$

$z_3 \equiv m_3^2$

sun1(...) $\equiv$ sun(1,...)

sun2(...) $\equiv$ sun(2,...)

```
read 'equations':

eq100:=z1*diff(T1(z1),z1) - (d/2-1)*T1(z1)=0:
eq101:=z2*diff(T2(z2),z2) - (d/2-1)*T2(z2)=0:
eq102:=z3*diff(T3(z3),z3) - (d/2-1)*T3(z3)=0:
syste:={seq(eq||j,j=1..66)}:
syste:= syste union {eq100,eq101,eq102}:
syste:=convert(syste,list):

#############################################################

with(DEtools):

bas := rifsimp(syste,[[sun2,sun1],[sun0,T1,T2,T3]]);

#############################################################
basa:= bas[Solved];

nb:=nops(basa);

quit;
```

```
%6 := T1(z1) T2(z2)

######################   Gram determinant

          2   2     4        2             4        2   2        3            3
%7 := 6 z1  z3  + z1  + 4 z2  z3 z1 + z3  + 6 z1  z2  - 4 z1  z3 - 4 z1  z2

            3           3           2           3        4       2     2
    - 4 z1 z2  - 4 z1 z3  + 4 z1 z3  z2 - 4 z3  z2 + z2  + 6 z2  q1q1

         3              3             2     2       3                3
    - 4 z3  q1q1 - 4 z2  q1q1 + 6 z1  q1q1  - 4 z1  q1q1 - 4 q1q1  z1

           3           2                          2                2
    - 4 q1q1  z2 + 4 z3  z1 q1q1 + 4 z3 q1q1 z1  + 4 q1q1 z2 z1

              2                2                2                2
    + 4 z2 q1q1  z1 + 4 z3 z2  q1q1 + 4 z1 z2  q1q1 + 4 q1q1  z3 z1

         2                        2                2          4       2     2
    + 4 z3  z2 q1q1 + 4 z2 q1q1  z3 + 4 z2 z1  z3 + q1q1  + 6 z3  q1q1

           3                                    3       2   2
    - 4 q1q1  z3 - 40 z2 z3 q1q1 z1 - 4 z3 z2  + 6 z3  z2
```

```
                6
               d                                                  4     3
Solved = [--------- sun2(z1, z2, z3) = 1/16 (-48 %11 z2  q1q1
              2    4
           dz3  dz2

             5     2            2     5            3     4            6
 + 32 %11 z2  q1q1  - 8 %11 z2  q1q1  + 32 %11 z2  q1q1  - 8 %11 z2  q1q1

             5                   3     2
 + 48 z1 z3  %3 z2 + 528 z1 z2  q1q1  sun0(z1, z2, z3)

             5                                                      4
 + 26 z1 z2  sun0(z1, z2, z3) d - 48 z1 sun0(z1, z2, z3) z2 q1q1

. . . . . .
```

```
          4          4    d                  T1(z1) d - 2 T1(z1)
 + 2 z3 z1  + 2 z3 z2 ), --- T1(z1) = 1/2 --------------------,
                        dz1                        z1

 d                T2(z2) d - 2 T2(z2)   d                T3(z3) d - 2 T3(z3)
--- T2(z2) = 1/2 --------------------, --- T3(z3) = 1/2 --------------------]
dz2                       z2           dz3                       z3

     d
%1 := --- sun0(z1, z2, z3)
     dz3

     d
%2 := --- sun0(z1, z2, z3)
     dz1

     d
%3 := --- sun0(z1, z2, z3)
     dz2

%4 := T2(z2) T3(z3)

%5 := T1(z1) T2(z2)
```

```
read basis:

with(DEtools);

#          ___0__0__
#         /         \
#        /           \
#     -----------------
#        \           /
#         \____0____/

diag:= diff(diff(diff(sun0(z1,z2,z3),z1),z1),z3)-ff(z1,z2,z3);

syste:= basis union {diag}:

#####################################################################

bas := rifsimp(syste,[[ff,sun2,sun1],[sun0,T1,T2,T3]]);

#####################################################################
basa:= bas[Solved];
nb:=nops(basa);
quit;
```

Concluding remarks

- Powerful and universal approach for evaluation of Feynman diagrams is proposed.
- All steps of the reduction of integrals can be implemented in computer algebra packages
- Existing software is still not efficient enough, further improvements in efficiency are needed thought two-loop integrals can be already treated on powerful computers