Numerical integration and extrapolation for finite and UV-divergent 3-loop Feynman integrals

E de Doncker¹, F Yuasa², K Kato³ and T Ishikawa²

¹ Department of Computer Science, Western Michigan University, Kalamazoo MI 49008, U. S. A.
² High Energy Accelerator Research Organization (KEK), Oho 1-1, Tsukuba, Ibaraki, 305-0801, Japan
³ Kogakuin University, Shinjuku, Tokyo 163-8677, Japan

E-mail:
elise.dedoncker@wmich.edu, fukuko.yuasa@kek.jp, kato@cc.kogakuin.ac.jp, tadashi.ishikawa@kek.jp

Abstract. Automatic numerical integration attempts to alleviate the detailed application of solution methods to the problem. We use automatic multivariate integration with PARINT combined with numerical extrapolation for classes of 3-loop UV-divergent self-energy diagrams with massive internal lines, and finite or UV-divergent massless vertex diagrams. The extrapolation is performed with respect to the dimensional regularization parameter. It is shown that the PARINT software handles the integrations accurately and efficiently.

1. Introduction

The radiative correction (RC) is highly important to study the physics of high-precision experiments at the current and future colliders. The development of methods to compute the integral in a multi-loop Feynman diagram is indispensable for the higher-order RC.

For a Feynman diagram with $L$ loops and $N$ internal lines, we consider the integral

$$I = \Gamma(N - nL/2)(-1)^N \int_0^1 \prod_{r=1}^N dx_r \delta \left(1 - \sum x_r\right) \frac{C^{N-n(L+1)/2}}{(D - i\varrho C)^{N-nL/2}}$$

where $C$ and $D$ are polynomials determined by the topology of the corresponding diagram and physical parameters, $n = 4 - 2\varepsilon$ to account for UV singularity (where $\varepsilon$ is the dimensional regularization parameter), and

$$S_d = \{x \in \mathbb{R}^d \mid \sum_{r=1}^d x_r \leq 1, x_r \geq 0 \text{ for } 1 \leq r \leq d\}$$

is the $d$-dimensional unit simplex. In the absence of singularities in the interior of the integration domain we set $\varrho = 0$. Feynman loop integrals defined in Eq (1) can be expanded in a Laurent series with respect to the dimensional regularization parameter $\varepsilon$, of the form $I \sim \sum_{k>K} C_k \varepsilon^k$, for an integer $K$. In this paper we obtain the leading coefficients $C_k$ of the expansion numerically using automatic integration and linear extrapolation.
Automatic integration implements a black-box approach to obtain an approximation \( Qf \) to an integral
\[
If = \int_D f(x) \, dx
\]
and an absolute error estimate \( E_a \), in order to satisfy a specified accuracy requirement for the actual error, of the form
\[
| Qf - If | \leq E_a \leq \max \{ t_a, t_r |If| \}
\]
for a given integrand function \( f \), region \( D \) and (absolute/relative) error tolerances \( t_a \) and \( t_r \). We denote the corresponding relative error estimate by \( E_r \), satisfying \( E_a = E_r |Qf| \). The type of accuracy requirement given by Eq (3) is based on [1] and used extensively in QUADPACK [2]. The integral of Eq (1) will be transformed so the domain \( D \) is the \((N - 1)\)-dimensional unit cube.

For the numerical integrations in this paper we apply the global adaptive code in the PARINT package (see, e.g., [3] for a description of PARINT), which implements a region partitioning strategy layered over MPI [4] and generally distributed over a network of computational nodes. The global adaptive strategy sums the total absolute error estimate \( E \) contributed over the region partition. On each subregion we use the integral and error estimates given for the multivariate cube in [5]. The integration and error estimation over the subregions is performed by the worker processes, while giving preference to the highest-error regions for subdivision. The controller process evaluates the overall termination criteria and (by default) also participates in the local integrations. For termination, the second inequality of Eq (3) is approximated by
\[
E_a \leq \max \{ t_a, t_r |Qf| \}
\]
where \( Qf \) is the total integral approximation. Apart from Eq (4), termination is also triggered when the user-specified maximum number of integrand evaluations is reached (whichever comes first). The latter refers to the total number of times the integrand function sub-program is called, and is proportional to the overall number of region subdivisions (also totaled by the controller over all worker processes). The combination of the termination criteria and the importance-driven subdivisions tailored to the specific difficulties in the integrand function lead to a fully automatic and machine-learned integration procedure.

With respect to expanding Eq (1) let us first consider a general expansion
\[
S(\varepsilon) \sim C_K \varphi_K(\varepsilon) + C_{K+1} \varphi_{K+1}(\varepsilon) + C_{K+2} \varphi_{K+2}(\varepsilon) + \ldots
\]
where we assume the \( \varphi_k \) functions are known (for example, \( \varphi_k(\varepsilon) = \varepsilon^k \)). To approximate the leading coefficients \( C_k, k = K, K+1, \ldots \), we apply a linear extrapolation, yielding the coefficients by solving a linear system based on the underlying expansion. This is achieved by generating a sequence of \( S(\varepsilon) \) values such that
\[
S(\varepsilon) = C_K \varphi_K(\varepsilon) + C_{K+1} \varphi_{K+1}(\varepsilon) + \ldots C_{K+\nu} \varphi_{K+\nu}(\varepsilon), \quad \ell = k, \ldots, \nu + k
\]
(\( \nu, k \geq 0 \)) and solving the linear system of order \((\nu + 1) \times (\nu + 1)\). For error control this can be done for increasing values of \( \nu \) and decreasing \( \varepsilon = \varepsilon + \varepsilon \). For example, a geometric sequence \( \varepsilon = b^{-\ell}, b > 1 \) can be used, or a Bulirsch [6] type sequence of the form \( \varepsilon = 1/b, b = 2,3,4,6,8,12,16,24,\ldots \).

Non-linear extrapolation with the \( \varepsilon \)-algorithm [7, 8, 9] can be applied (for unknown \( \varphi_k(\varepsilon) \)) under more general conditions with geometric sequences of \( \varepsilon \). In previous work we have made ample use of the \( \varepsilon \)-algorithm (see for instance, [10, 11, 12, 13]) to treat loop integrals in the physical region, with integrand singularities caused by vanishing denominators in the interior of the integration domain. In this paper we employ linear extrapolation to obtain the coefficients of the Laurent series.

The method is demonstrated in Section 2 for two self-energy diagrams, where integrand transformations are further introduced to diminish singularities at the boundaries of the domain. Section 3 presents results for 3-loop vertex diagrams, and Section 4 applies an extrapolation to a 2-fold Mellin-Barnes integral representation.
2. 3-loop self-energy diagrams

![diagram](image)

**Figure 1.** 3-loop UV-divergent self-energy diagrams with massive internal lines \((m_i = 1\) and \(s = p^2 = 1)\), cf., Laporta [14]: (a) Laporta Fig 2(o) \((N = 7, \text{ladybug diagram})\), (b) Laporta Fig 2(p) \((N = 7, \text{cracked Magdeburg diagram})\)

Corresponding to the ladybug diagram of Fig 1(a) and the cracked Magdeburg diagram of Fig 1(b) with \(L = 3\) and \(N = 7\), we consider the 3-loop self-energy integrals

\[
I^{S3}(\varepsilon) = \Gamma(1 + 3\varepsilon) \int \frac{C^{-1+4\varepsilon}}{D^{1+3\varepsilon}} \, dx
\]  

(7)

The \(C\) and \(D\) functions derived from the diagrams are the determinants given in Table (1), where we denote \(x_{j_1,j_2 \ldots j_k} = x_{j_1} + x_{j_2} + \ldots + x_{j_k}\). The corresponding polynomials are given as \((C^{S3}_a, D^{S3}_a)\) in Eq (A.1), and \((C^{S3}_b, D^{S3}_b)\) in Eq (A.2) of Appendix A in this paper.

|                | \(I^{S3}_a\)         | \(I^{S3}_b\)         |
|----------------|-----------------------|-----------------------|
| \(C\)          | \([x_{1257}, -x_7, -x_5; -x_7, x_{3467}, -x_6; -x_5, -x_6, x_{56}]\) | \([x_{126}, -x_6, 0; -x_6, x_{4567}, -x_7; 0, -x_7, x_{37}]\) |
| \(D\)          | \([-x_{1257}, -x_7, -x_5; -x_7, x_{3467}, -x_6; -x_5, -x_6, x_{56}; px_1, px_1, 0]\) | \([-x_{126}, -x_6, 0; -x_6, x_{4567}, -x_7; 0, -x_7, x_{37}; px_1, px_4, px_3]\) |

The integrals (7) satisfy an asymptotic expansion \(\sum_{k \geq -1} C_k \varepsilon^k\) of the form (5), corresponding to [14] (2o) and (2p):

\[
\Gamma^{-3}(1 + \varepsilon) I^{S3}_a(\varepsilon) \sim 0.92363182652 \varepsilon^{-1} - 2.423491634 + 8.38134971 \varepsilon - 26.9936212 \varepsilon^2 \ldots
\]  

(8)

\[
\Gamma^{-3}(1 + \varepsilon) I^{S3}_b(\varepsilon) \sim 0.92363182652 \varepsilon^{-1} - 2.116619719 + 6.92954468 \varepsilon - 21.5032784 \varepsilon^2 \ldots
\]  

(9)

Note that numerical results for these integrals were given in [15] using a product trapezoidal rule with double exponential transformation (DE) on Intel(R) Xeon(R) E5-2687W v3 3.10 GHz in IEEE 754-2008 binary128, which required extensive computation times. However the executions with PARINT in long double precision reported later in this section take far less than the DE time (while reaching near comparable accuracy).
For the numerical approximation of the integrals over $S_{N-1}$, a variable transformation is performed that maps the simplex domain to the unit cube and at the same time achieves a smoothening of the integrand boundary singularities. For $I_{a}^{S3}$ the variables are transformed as

\begin{align}
  x_1 &= y_1m y_2 y_4 y_5, \quad x_2 = y_1m y_2 y_4 y_5m, \\
  x_3 &= y_1m y_2 y_4m y_6, \quad x_4 = y_1m y_2 y_4m y_6m, \\
  x_5 &= y_1 y_3, \quad x_6 = y_1 y_3m, \\
  x_7 &= y_1m y_2m
\end{align}

with $y_{1m} = 1 - y_1$ and Jacobian $y_1 y_1^4 y_2^3 y_4 y_4m$. The transformation applied for $I_{b}^{S3}$ is

\begin{align}
  x_1 &= y_1 y_2 y_4, \quad x_2 = y_1 y_2 y_4m, \\
  x_3 &= y_1m y_3 y_6, \quad x_4 = y_1m y_3 y_6m, \\
  x_5 &= y_1 y_2m y_5m, \quad x_6 = y_1m y_3m, \\
  x_7 &= y_1m y_3 y_6m
\end{align}

with $y_{1m} = 1 - y_1$ and Jacobian $y_1^3 y_1^2 y_2m y_2 y_3$. While the above two variable transformations are interchangeable for $I_{a}^{S3}$ and $I_{b}^{S3}$, the transformation (10) yields better results for $I_{a}^{S3}$, and (11) for $I_{b}^{S3}$.

Numerical integration results were obtained with PARINT on the thor cluster at WMU, using 64 processes on four 16-core nodes (each with dual Intel Xeon E5-2670, 8C, 2.6 GHz processors, and 128 GB memory). We employ the adaptive method (over the 6D cube) with the multivariate cubature process on four 16-core nodes (each with dual Intel Xeon E5-2670, 8C, 2.6 GHz processors, and 128 GB memory). We employ the adaptive method (over the 6D cube) with the multivariate cubature process on four 16-core nodes (each with dual Intel Xeon E5-2670, 8C, 2.6 GHz processors, and 128 GB memory). The convergence of linear extrapolation applied to the PARINT sequence as $\varepsilon = \varepsilon_\ell \to 0$ is shown in Table 2 and Table 3 for (8) and (9), respectively. The computation of each integral as a function of $\varepsilon$, corresponding to each row, is thus performed automatically according to the global adaptive strategy. The tables also list the obtained estimated relative error $E_r = E_a/|Qf|$. Table 2 further shows the user-specified maximum number of function evaluations MAX given in billions (B) for each integration. Setting a strict relative error tolerance ($t_r = 10^{-13}$) allows terminating based on MAX, which is the case for the computations in Table 2 and part of Table 3. The latter gives the actual number of function evaluations (#EV) performed for the integration. Since for the UV self-energy problems at hand, the integration becomes harder as $\varepsilon_\ell$ decreases, the relative error estimates $E_r$ tend to increase for successive $\ell$.

**Table 2.** Results 3-loop UV self-energy integral $I_{a,b}^{S3}(\varepsilon) \Gamma^{-3}(1+\varepsilon)$ (on 4 nodes with 16 proc/node of thor cluster, long double precision); integration err. tolerances $t_r = 10^{-13}$, $t_a = 0$; $E_r = E_a/|Qf|$ (= integration estim. rel. error); #EV[B] = # integrand evals. in billions; T[s] = Time (elapsed user time) in seconds; $\varepsilon = \varepsilon_\ell = 1.15^{-\ell}$

| $\ell$ | $E_r$ | MAX[B] | T[s] | RESULT | $C_{-1}$ | RESULT | $C_{0}$ | RESULT | $C_{1}$ | RESULT | $C_{2}$ |
|-------|-------|--------|------|--------|----------|--------|----------|--------|----------|--------|----------|
| 17    | 2.3e-12 | 100    | 682.3 | 0.88532153209 | -1.412495535 | 4.1493435 |
| 18    | 4.2e-12 | 100    | 682.8 | 0.91647848504 | -2.133365157 | 4.1493435 |
| 19    | 6.0e-12 | 100    | 683.8 | 0.91647848504 | -2.133365157 | 4.1493435 |
| 20    | 5.2e-12 | 150    | 1027.6 | 0.92246162843 | -2.356946186 | 6.9162602 | -11.33999 |
| 21    | 2.2e-12 | 350    | 2415.8 | 0.92346458479 | -2.410840073 | 7.9934802 | -20.83203 |
| 22    | 3.6e-12 | 350    | 2419.5 | 0.92361089965 | -2.421546174 | 8.2986081 | -25.17454 |
| 23    | 5.7e-12 | 350    | 2425.8 | 0.92362953012 | -2.423211188 | 8.3667190 | -26.56843 |
| 24    | 8.7e-12 | 350    | 2431.6 | 0.92363160796 | -2.423458644 | 8.3791875 | -26.91297 |
| 25    | 1.3e-11 | 350    | 2437.1 | 0.92363180412 | -2.423487620 | 8.3813166 | -26.99204 |
| 26    | 1.7e-11 | 350    | 2445.7 | 0.92363182397 | -2.423491206 | 8.3813166 | -26.99204 |

*Exact (8):* $0.92363182652 -2.423491634 8.3813497 -26.99362
Table 3. Results 3-loop UV self-energy integral $I_b^{S3}(\varepsilon) \Gamma^{-3}(1+\varepsilon)$ (on 4 nodes with 16 procs/node of thor cluster, long double precision); integration err. tolerances $t_r = 10^{-13}$, $t_a = 0$; $E_r = E_a/|Qf|$ (= integration estim. rel. error); #Ev[B] = # integrand evals. in billions; T[s] = Time (elapsed user time) in seconds; $\varepsilon = \varepsilon_{\ell} = 1.3 - \ell$

| $\ell$ | $E_r$ | #Ev[B] | T[s] | RESULT $C_{-1}$ | RESULT $C_0$ | RESULT $C_1$ | RESULT $C_2$ |
|-------|-------|--------|------|-----------------|----------------|----------------|----------------|
| 5     | 7.1e-14 | 121     | 915.1| -0.624745066    |                |                |                |
| 6     | 5.7e-14 | 95      | 668.8| 0.79556987526   | -1.356849184   |                |                |
| 7     | 6.4e-14 | 110     | 760.4| 0.88129897281   | -3.7234312     | 1.5364029      |                |
| 8     | 3.9e-13 | 200     | 1444.9| 0.91188370051   | -1.809949406   | 5.4720364      | -9.76427       |
| 9     | 2.2e-13 | 275     | 2108.5| 0.92314713425   | -2.091734298   | 6.4193304      | -15.70840      |
| 10    | 2.4e-12 | 275     | 2018.5| 0.92363108618   | -2.116079491   | 6.9248672      | -21.15424      |
| 11    | 7.0e-12 | 275     | 1978.3| 0.92363177196   | -2.116161010   | 6.9289487      | -21.36736      |
| 12    | 1.3e-11 | 325     | 2288.4| 0.92362368266   | -2.115423673   | 6.9060678      | -20.88190      |
| 13    | 3.2e-11 | 325     | 2309.7| 0.92363108618   | -2.116079491   | 6.9248672      | -21.15424      |
| 14    | 6.7e-11 | 325     | 2331.9| 0.92363177196   | -2.116161010   | 6.9289487      | -21.36736      |
| 15    | 1.4e-10 | 350     | 2534.4| 0.92363182585   | -2.116169538   | 6.9295216      | -21.50172      |

Exact (9): 0.92363182652 -2.116169718 6.9295447 -21.50328

Figure 2. 3-loop vertex diagrams with massless internal lines, cf., Heinrich et al. [16] (a) Diagram $A_{6,2}$ ($N = 6$), (b) Diagram $A_{7,5}$ ($N = 7$). In [16] the incoming momentum is $p_3 = p_1 + p_2$ and outgoing lines are on-shell and massless, $p_1^2 = p_2^2 = 0$.

3. 3-loop vertex diagrams

For the diagrams in Fig 2(a) and (b) with $N = 6$ and $N = 7$, respectively, we denote

$$I_a^{V3}(\varepsilon) = \Gamma(3\varepsilon) \int_{S_5} \frac{C^{2+4\varepsilon}}{D^{3\varepsilon}} \, dx$$ (12)

$$I_b^{V3}(\varepsilon) = -\Gamma(1+3\varepsilon) \int_{S_6} \frac{C^{-1+4\varepsilon}}{D^{1+3\varepsilon}} \, dx$$ (13)

The $C$ and $D$ functions in the integrands are listed as determinants in Table 4 for Eq (12) and in Table 5 for Eq (13). Here $p_1^2 = p_2^2 = 0$, so $-q^2 = -p_3^2 = -(p_1 + p_2)^2 = -2p_1p_2$ (which is written as a separate factor outside of the computed integral). Expanded versions of $C$ and $D$ are given as $(C_a^{V3}, D_a^{V3})$ in Eq (A.3), and $(C_b^{V3}, D_b^{V3})$ in Eq (A.4) of Appendix A.
Table 4. C and D functions of vertex integral (12), \( p_1^2 = p_2^2 = 0 \)

\[
\begin{array}{c|ccc}
I_a^{V^3} & x_{145} & -x_4 & -x_5 \\
& -x_4 & x_{246} & -x_6 \\
& -x_5 & -x_6 & x_{356} \\
\end{array}
\]

\[
C = \begin{vmatrix}
x_{145} & -x_4 & -x_5 & p_1 x_1 \\
-x_4 & x_{246} & -x_6 & -p_2 x_2 \\
-x_5 & -x_6 & x_{356} & 0 \\
p_1 x_1 & -p_2 x_2 & 0 & p_1^2 x_1 + p_2^2 x_2 \\
\end{vmatrix}
\]

Table 5. C and D functions of vertex integral (13), \( p_1^2 = p_2^2 = 0 \)

\[
\begin{array}{c|ccc}
I_b^{V^4} & x_{1236} & x_{12} & x_3 \\
& x_{12} & x_{1245} & -x_5 \\
& x_3 & -x_5 & x_{357} \\
\end{array}
\]

\[
C = \begin{vmatrix}
x_{1236} & x_{12} & x_3 & p_1 x_{13} - p_2 x_2 \\
x_{12} & x_{1245} & -x_5 & p_1 x_2 - p_2 x_{24} \\
x_3 & -x_5 & x_{357} & p_1 x_3 \\
p_1 x_{13} - p_2 x_2 & p_1 x_1 - p_2 x_{24} & p_1 x_3 & p_1^2 x_{13} + p_2^2 x_{24} \\
\end{vmatrix}
\]

Table 6. Results 3-loop UV vertex integral of Eq (14) by PARINT (on 4 nodes with 16 procs/node of thor cluster, long double precision); integration err. tolerances \( t_r = 10^{-13} \), \( t_a = 0 \); MAX = 30B (= max. # integrand evals.); \( E_c = E_a/|Qf| \) (= integration estim. rel. error); T[s] = Time (elapsed user time) in seconds; \( \varepsilon = \varepsilon_\ell = 1.2^{-\ell} \)

| \( \ell \) | \( E_r \) | T[s] | RES. \( C_{-1} \) | RES. \( C_0 \) | RES. \( C_1 \) | RES. \( C_2 \) |
|---|---|---|---|---|---|---|
| 20 | 6.9e-09 | 171.6 | | | | |
| 21 | 4.7e-09 | 171.6 | -2.2674024847 & -36.22875015 & | | |
| 22 | 3.8e-09 | 171.2 | -2.4184908276 & -23.48554838 & -26.647842 & |
| 23 | 4.2e-09 | 173.2 | -2.4030392631 & -25.64179665 & -16.727996 & -150.451 |
| 24 | 4.0e-09 | 174.5 | -2.4041730407 & -25.40846949 & -185.040255 & -911.900 |
| 25 | 3.5e-09 | 175.9 | -2.4041116670 & -25.42597896 & -183.074773 & -1020.406 |
| 26 | 3.8e-09 | 177.3 | -2.4041138063 & -25.42515557 & -183.204184 & -1009.854 |

Asymptotic expansions are derived analytically for \( I_a^{V^3}(\varepsilon) \) and \( I_b^{V^3}(\varepsilon) \) in [16]:

\[
- \Gamma^3(1 - \varepsilon) I_a^{V^3}(\varepsilon) \sim -2\zeta_3 \varepsilon^{-1} - (18\zeta_3 + \frac{7\pi^4}{180}) \varepsilon + (-122\zeta_3 - \frac{7\pi^4}{20} + \frac{2\pi^2}{3} \zeta_3 - 10\zeta_5) \varepsilon + (-738\zeta_3 - \frac{427\pi^4}{180} + 6\pi^2 \zeta_3 - 90\zeta_5 + \frac{163\pi^6}{7560} + 76\zeta_3^2) \varepsilon^2 + O(\varepsilon^3)
\]

\[
\approx -2.4041138063 \varepsilon^{-1} - 25.42515557 - 183.204184 \varepsilon - 1009.790681 \varepsilon^2 + O(\varepsilon^3)
\]
and

\[- \Gamma^3(1 - \varepsilon) I_b^{V,3} (\varepsilon) \sim 2 \pi^2 \zeta_3 + 10 \zeta_5 + (12 \pi^2 \zeta_3 + 60 \zeta_5 + \frac{11 \pi^6}{162} + 18 \zeta_3^2) \varepsilon + \mathcal{O}(\varepsilon^2)\]

\[\approx 34.0969298 + 295.8700 \varepsilon + \mathcal{O}(\varepsilon^2)\]  \hspace{1cm} (15)

where the \((-\rangle\) sign on the left accounts for a difference in the sign with the definitions of [16].

The integral of Eq (12) is approximated by PARINT and Table 6 lists linear extrapolation results for the sequence (14) as \(\varepsilon = \varepsilon \to 0\). Table 7 shows an extrapolation of a sequence of PARINT results for Eq (15). The integrations are performed using the adaptive method with the rules of polynomial degree 9 in PARINT after mapping the \(d = (N - 1)\)-dimensional simplex to the cube with the transformation

\[(x_1, \ldots, x_d) \rightarrow (y_1, (1 - x_1)y_2, \ldots, (1 - x_1 - \ldots - x_{d-1})y_d)\]

where the coordinates \(y_k\) satisfy \(0 \leq y_k \leq 1\).

4. Mellin-Barnes representation

Using a 2-fold Mellin-Barnes representation from Heinrich et al. [16], we consider the integral

\[- \Gamma^3(1 - \varepsilon) I_a^{V,3} (\varepsilon) = - \frac{\Gamma^3(1 - \varepsilon) \Gamma(3\varepsilon) \Gamma(1 - 3\varepsilon)}{\Gamma(1 - 2\varepsilon) \Gamma(2 - 4\varepsilon)} \int_{c_1 - i\infty}^{c_1 + i\infty} \frac{dw_1}{2\pi i} \int_{c_2 - i\infty}^{c_2 + i\infty} \frac{dw_2}{2\pi i} \]

\[\times \frac{\Gamma(-1 + 3\varepsilon - w_1) \Gamma(-1 + 2\varepsilon - w_1) \Gamma(2 - 4\varepsilon + w_1) \Gamma(-w_2) \Gamma(w_2 - w_1)}{\Gamma(3\varepsilon - w_1) \Gamma(2 - 4\varepsilon + w_1) \Gamma(2 - 4\varepsilon + w_1 - w_2)} \times \Gamma(1 - \varepsilon + w_2) \Gamma(1 - \varepsilon + w_1 - w_2) \Gamma(1 - 2\varepsilon + w_2) \Gamma(1 - 2\varepsilon + w_1 - w_2)\]

\[\times \frac{\Gamma(-1 + 3\varepsilon - w_1) \Gamma(-1 + 2\varepsilon - w_1) \Gamma(2 - 4\varepsilon + w_1) \Gamma(-w_2) \Gamma(w_2 - w_1)}{\Gamma(3\varepsilon - w_1) \Gamma(2 - 4\varepsilon + w_1) \Gamma(2 - 4\varepsilon + w_1 - w_2)} \times \Gamma(1 - \varepsilon + w_2) \Gamma(1 - \varepsilon + w_1 - w_2) \Gamma(1 - 2\varepsilon + w_2) \Gamma(1 - 2\varepsilon + w_1 - w_2)\]

\[\times \frac{\Gamma(-1 + 3\varepsilon - w_1) \Gamma(-1 + 2\varepsilon - w_1) \Gamma(2 - 4\varepsilon + w_1) \Gamma(-w_2) \Gamma(w_2 - w_1)}{\Gamma(3\varepsilon - w_1) \Gamma(2 - 4\varepsilon + w_1) \Gamma(2 - 4\varepsilon + w_1 - w_2)} \times \Gamma(1 - \varepsilon + w_2) \Gamma(1 - \varepsilon + w_1 - w_2) \Gamma(1 - 2\varepsilon + w_2) \Gamma(1 - 2\varepsilon + w_1 - w_2)\]

\[\times \frac{\Gamma(-1 + 3\varepsilon - w_1) \Gamma(-1 + 2\varepsilon - w_1) \Gamma(2 - 4\varepsilon + w_1) \Gamma(-w_2) \Gamma(w_2 - w_1)}{\Gamma(3\varepsilon - w_1) \Gamma(2 - 4\varepsilon + w_1) \Gamma(2 - 4\varepsilon + w_1 - w_2)} \times \Gamma(1 - \varepsilon + w_2) \Gamma(1 - \varepsilon + w_1 - w_2) \Gamma(1 - 2\varepsilon + w_2) \Gamma(1 - 2\varepsilon + w_1 - w_2)\]

with \(c_1 = -6/5\), \(c_2 = -1/2\), \(-1/15 < \varepsilon < 3/20\) (to ensure that the contours separate left poles and right poles of \(\Gamma\) functions in the complex plane).

A transformation \(w_1 = c_1 + i t_1\) and \(w_2 = c_2 + i t_2\) gives an integral over \(\mathbb{R}^2\),

\[\int_{c_1 - i\infty}^{c_1 + i\infty} \frac{dw_1}{2\pi i} \int_{c_2 - i\infty}^{c_2 + i\infty} \frac{dw_2}{2\pi i} \rightarrow \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dt_1 dt_2\]

\[\text{(18)}\]

Note that this introduces complex arguments into the \(\Gamma\)-functions of the integrand, e.g., it maps \(\Gamma(-1 + 3\varepsilon - w_1)\) to \(\Gamma(-1 + 3\varepsilon - c_1 - i t_1)\). Finally a transformation \(t_1 = \tan(x_1)\) with \(dt_1 =\]
Table 8. Results 3-loop UV vertex integral (by DQAGSE×DQAGSE on Mac-Pro in double precision); integration err. tolerances $\ell_\tau = 10^{-9}$, $t_a = 0$ in each coord. dir.; MAXDIV = 100 (= max. # subdivisions in each coord. direction); $E_r = E_a/[Qf]$ (= integration estim. rel. error); $T[s]$ = Time (elapsed user time) in seconds; $\epsilon = \epsilon_{\ell} = 2^{-\ell}$

| $\ell$ | $E_r$ | $T[s]$ | Res. $C_{-1}$ | Res. $C_0$ | Res. $C_1$ | Res. $C_2$ | Res. $C_3$ |
|---|---|---|---|---|---|---|---|
| 1 | 4.6e-10 | 0.123 | 2.71270658920750307 | -124.29196699004 |
| 2 | 7.2e-09 | 0.065 | -3.2667137757164271 | 19.214208212662 | -765.366267748 |
| 3 | 7.1e-09 | 0.089 | -2.34303998712649708 | 62.2486740911 | -3783.82591 |
| 4 | 6.4e-11 | 0.103 | -2.40606427439256088 | -220.100132861 | 88.82961836 | -5454.348 |
| 5 | 3.2e-10 | 0.103 | -2.40408459536474339 | -1168.029003 | -4842.949 |
| 6 | 7.2e-10 | 0.089 | -2.40411401541469472 | -999.9251910 | -4804.705 |
| 7 | 7.1e-09 | 0.065 | -2.34303998712649708 | -5454.348 |
| 8 | 7.0e-10 | 0.101 | -2.40408459536474339 | -4842.949 |
| 9 | 1.1e-09 | 0.101 | -2.40411401541469472 | -4842.949 |
| 10 | 3.2e-10 | 0.103 | -2.40408459536474339 | -4842.949 |
| 11 | 1.3e-09 | 0.100 | -2.40411380559384602 | -4842.949 |
| 12 | 1.4e-09 | 0.100 | -2.404113806302403015 | -4842.949 |
| 13 | 1.5e-09 | 0.100 | -2.404113806301917693 | -4842.949 |

$dx_1/\cos^2(x_1)$, and $t_2 = \tan(x_2)$ with $dt_2 = dx_2/\cos^2(x_2)$ maps $\mathbb{R}^2 \to (-\pi/2, \pi/2) \times (-\pi/2, \pi/2)$. For this example the resulting form is used to generate a sequence of

$$-\Gamma^3(1 - \epsilon) I_a^{V^3}(\epsilon) \sim \sum_{k \geq -1} C_k \epsilon^k$$

by a linear extrapolation as $\epsilon = \epsilon_{\ell} \to 0$, to approximate the expansion coefficients as shown in Table 8.

In order to evaluate the (2D) integral numerically, we apply the (one-dimensional) DQAGSE integration program from QUADPACK [2] in both coordinate directions. As part of its regular strategy, DQAGSE performs a global adaptive subdivision and extrapolation with the $\epsilon$-algorithm [7, 8] to treat certain types of end-point singularities. The input parameters of DQAGSE include the requested absolute and relative accuracies, $t_a$ and $t_r$, respectively, as well as the maximum allowed number of subdivisions, MAXDIV (set as $t_r = 10^{-9}, t_a = 0$, and MAXDIV = 100 for the experiments). Table 8 shows the convergence of the expansion coefficients, as well as the estimated relative error $E_r$ of the outer integration only, and execution time $T$ (in seconds) for each double integration.

5. Conclusions
We presented applications of the PARINT and QUADPACK packages for a direct integration of Feynman loop integrals, with linear extrapolation to compute the expansion coefficients and treat singularities. The methods are fully numerical and viable across the problem types. While the integration strategy is automatic, more work is needed to determine the sequence of $\epsilon$ needed for the extrapolation.

New results using multivariate adaptive integration with PARINT are derived for 3-loop UV-divergent self-energy diagrams with massive internal lines and UV-divergent vertex diagrams with massless propagators. An approach using iterated integration with DQAGSE from QUADPACK combined with linear extrapolation is further shown to provide accurate results based on a 2D Mellin-Barnes representation for a UV-divergent vertex diagram with massless internal lines.

The demonstrated methods provide considerable accuracy within the allotted execution time, compared to other numerical methods that utilize minimal information about the underlying problem. In future work we plan to apply fully numerical methods to various problem classes and widen their
domain of applicability for the computation of Feynman loop integrals, for example by the development of suitable transformations to relieve integrand singularities.

**Acknowledgments**

We acknowledge the support from the National Science Foundation under Award Number 1126438, and the Center for High Performance Computing and Big Data at Western Michigan University. This work is further supported by Grant-in-Aid for Scientific Research (15H03668) of JSPS, and the Large Scale Simulation Program No. 15/16-06 and 16/17-21 of KEK.

**Appendix A. C and D functions**

Below we list the C and D functions \( (C^{S3}_a, D^{S3}_a), (C^{S3}_b, D^{S3}_b) \) of the integrals \( I^{S3}_a, I^{S3}_b \), and \( (C^{V3}_a, D^{V3}_a), (C^{V3}_b, D^{V3}_b) \) of the integrals \( I^{V3}_a, I^{V3}_b \), as derived from the corresponding diagrams in Fig 1 and Fig 2. In \( D^{S3}_a \) and \( D^{S3}_b \) we put \( p^2 = 1 \) and \( m_i^2 = 1 \), and in \( D^{V3}_a \) and \( D^{V3}_b \) the factor \( (-p^2) \) is not shown.

\[
C^{S3}_a = x_{1257} x_{3467} x_{56} - 2 x_5 x_6 x_7 - x_{3467} x_5^2 - x_{1257} x_6^2 - x_{56} x_7^2
\]

\[
D^{S3}_a = -x_3^2 x_5 - x_{3467} x_{24567} x_5^2 + x_{1257} x_3^2 x_5^2 + x_1^3 x_{3467} x_5^2 + x_{1257} x_{3467} x_{24567} x_5^6
+ 2 x_1 x_3 x_7 x_{56} - x_{24567} x_7^2 x_5^6 + 2 x_1 x_3 x_5 x_6 - 2 x_{24567} x_7 x_5 x_6 - x_1^2 x_6^2 - x_{1257} x_{24567} x_6^2
\]

\[
C^{S3}_b = x_{126} x_{4567} x_3^7 - x_{126} x_7^2 - x_6^2 x_3^7
\]

\[
D^{S3}_b = x_{126} x_{37} x_4^2 + x_{126} x_3^2 x_{4567}^2 + x_1^2 x_3^7 x_{4567} + x_{126} x_{2567} x_3^7 x_{4567} + 2 x_1 x_3 x_7 x_4 x_6
- x_3^2 x_6 - x_{2567} x_3^7 x_6^2 + 2 x_{126} x_3 x_4 x_7 + 2 x_1 x_3 x_6 x_7 - x_1 x_7^2 - x_{126} x_{2567} x_7^2
\]

\[
C^{V3}_a = x_{145} x_{246} x_{356} - 2 x_4 x_5 x_6 - x_{246} x_5^2 - x_{356} x_4^2 - x_{145} x_6^2
\]

\[
D^{V3}_a = x_1 x_2 x_4 x_{356} + x_1 x_2 x_5 x_6
\]

\[
C^{V3}_b = x_{1236} x_{1245} x_{357} - 2 x_{12} x_3 x_5 - x_{1245} x_3^2 - x_{1236} x_5^2 - x_{12} x_{357}
\]

\[
D^{V3}_b = x_1 x_2 x_3 x_4 + x_1 x_2 x_4 x_5 + x_2 x_3 x_4 x_5 + x_1 x_2 x_3 x_6 + x_1 x_3 x_4 x_6
+ x_1 x_2 x_5 x_6 + x_2 x_3 x_5 x_6 + x_1 x_4 x_5 x_6 + x_3 x_4 x_5 x_6 + x_1 x_2 x_3 x_7
+ x_1 x_2 x_4 x_7 + x_1 x_2 x_5 x_7 + x_2 x_3 x_5 x_7 + x_1 x_2 x_6 x_7 + x_1 x_4 x_6 x_7
\]

**References**

[1] de Boor C 1971 CADRE: An algorithm for numerical quadrature *Mathematical Software* ed Rice J R (Academic Press, New York) pp 417–449

[2] Piessens R, de Doncker E, Überhuber C W and Kahaner D K 1983 QUADPACK, A Subroutine Package for Automatic Integration (Springer Series in Computational Mathematics vol 1) (Springer-Verlag)

[3] de Doncker E and Yuasa F 2014 Distributed and multi-core computation of 2-loop integrals doi:10.1088/1742-6596/523/1/012052

[4] Open-MPI http://www.open-mpi.org
[5] Berntsen J, Espelid T O and Genz A 1991 ACM Trans. Math. Softw. 17 437–451
[6] Bulirsch R 1964 Numerische Mathematik 6 6–16
[7] Shanks D 1955 J. Math. and Phys. 34 1–42
[8] Wynn P 1956 Mathematical Tables and Aids to Computing 10 91–96
[9] Sidi A 2003 Practical Extrapolation Methods - Theory and Applications (Cambridge University Press) ISBN 0-521-66159-5
[10] de Doncker E, Shimizu Y, Fujimoto J, Yuasa F, Cucos L and Van Voorst J 2004 Nuclear Instruments and Methods in Physics Research A 539 269–273 hep-ph/0405098
[11] de Doncker E, Shimizu Y, Fujimoto J and Yuasa F 2004 Computer Physics Communications 159 145–156
[12] Yuasa F, de Doncker E, Hamaguchi N, Ishikawa T, Kato K, Kurihara Y and Shimizu Y 2012 Computer Physics Communications 183 2136–2144
[13] de Doncker E, Fujimoto J, Hamaguchi N, Ishikawa T, Kurihara Y, Shimizu Y and Yuasa F 2011 Journal of Computational Science (JoCS) 3 102–112 doi:10.1016/j.jocs.2011.06.003
[14] Laporta S 2000 Int. J. Mod. Phys. A 15 5087–5159 arXiv:hep-ph/0102033v1
[15] de Doncker E, Yuasa F, Kato K, Ishikawa T, Kapenga J and Olagbemi O 2016 Regularization with numerical extrapolation for finite and UV-divergent multi-loop integrals Submitted
[16] Heinrich G, Huber T and Maître D 2008 Physics Letters B 662 344–352