Automatic numerical integration methods for Feynman integrals through 3-loop

E de Doncker\textsuperscript{1}, F Yuasa\textsuperscript{2}, K Kato\textsuperscript{3}, T Ishikawa\textsuperscript{2} and O Olagbemi\textsuperscript{1}

\textsuperscript{1} Department of Computer Science, Western Michigan University, Kalamazoo MI 49008, U. S. A.
\textsuperscript{2} High Energy Accelerator Research Organization (KEK), Oho 1-1, Tsukuba, Ibaraki, 305-0801, Japan
\textsuperscript{3} Department of Physics, 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, omofolakunmiel.olagbemi@wmich.edu

Abstract. We give numerical integration results for Feynman loop diagrams through 3-loop such as those covered by Laporta [1]. The methods are based on automatic adaptive integration, using iterated integration and extrapolation with programs from the QUADPACK package, or multivariate techniques from the ParInt package. The Dqags algorithm from QUADPACK accommodates boundary singularities of fairly general types. PARINT is a package for multivariate integration layered over MPI (Message Passing Interface), which runs on clusters and incorporates advanced parallel/distributed techniques such as load balancing among processes that may be distributed over a network of nodes. Results are included for 3-loop self-energy diagrams without IR (infra-red) or UV (ultra-violet) singularities. A procedure based on iterated integration and extrapolation yields a novel method of numerical regularization for integrals with UV terms, and is applied to a set of 2-loop self-energy diagrams with UV singularities.

1. Introduction

The techniques in this paper are based on automatic integration, which is a black-box approach to generate an approximation $Q(f)$ to an integral $I_f = \int_D f(x) \, dx$ and an absolute error estimate $\epsilon_f$, in order to satisfy a specified accuracy requirement for the actual error. We use an accuracy requirement of the form

$$|Q_f - I_f| \leq \epsilon_f \leq \max \{ t_a, t_r | I_f | \}$$  \hspace{1cm} (1)

for a given integrand function $f$, region $D$ and (absolute/relative) error tolerances $t_a$ and $t_r$, respectively. In order to achieve Eq (1), the actual error should not exceed the error estimate $\epsilon_f$, and the error estimate should not exceed the weakest of the absolute and relative error tolerances (indicated by the maximum taken on the right of Eq (1)). This type of accuracy requirement is based on [2] and used extensively in QUADPACK [3].

Known methods for parallelization of these procedures include:

(i) Parallelization on the rule or points level: typically in non-adaptive algorithms, e.g., for Monte-Carlo (MC) algorithms and composite rules using grid or lattice points. Then in $I_f = \int_D f \approx \sum_k w_k f(\bar{x}_k)$ the function evaluations $f(\bar{x}_k)$ are performed in parallel.

(ii) Parallelization on the region level: in adaptive (region-partitioning) methods. These lead to task pool strategies, which may benefit from load balancing on distributed memory systems; or maintain a shared priority queue on shared memory systems.
(iii) We added multi-threading to iterated integration [4, 5, 6]: the inner integrals are independent and computed in parallel, e.g., over a subregion \( S = D_1 \times D_2 \) (with inner region \( D_2 \)) consider \( \int_S F(x) \, dx \approx \sum_k w_k F(x_k) \), with \( F(x_k) = \int_{D_1} f(x_k, y) \, dy \). The inner integrations can be performed adaptively, which we applied in iterated versions of the 1D QUADPACK program DQAGS [3, 7].

We apply numerical extrapolation to integrals with an asymptotic expansion in the dimensional regularization parameter \( \epsilon \), of the form

\[
S(\epsilon) \sim \sum_{k \geq K} C_k \varphi_k(\epsilon) \quad \text{as } \epsilon \to 0. \tag{2}
\]

For example, the \( \varphi_k(\epsilon) \) functions may be integer powers of \( \epsilon \), \( \varphi_k(\epsilon) = \epsilon^k \). Then for finite integrals \( K = 0 \) and the integral is represented by \( C_0 \). Linear extrapolation can be applied when the \( \varphi_k(\epsilon) \) functions are known. In that case, \( S(\epsilon) \) is approximated for decreasing values of \( \epsilon = \epsilon_j \), and Eq (2) is truncated after 2, 3, \ldots, \( N \) terms to form linear systems of increasing size in the \( C_k \) variables.

As the integral approximation generally becomes harder with smaller \( \epsilon \), we use slowly decreasing sequences \( \{ \epsilon_j \} \), such as a geometric sequence with base 1/1.2. Another sequence of interest is based on the Bulirsch sequence 1, 2, 3, 4, 6, 8, 12, 16, 24, \ldots, (see [8]); we employ \( \{ 1/b_j \} \) from a starting index \( J \) in the Bulirsch sequence. We resort to non-linear extrapolation when the structure of the asymptotic expansion is not known. In previous work we have made ample use of the \( \epsilon \)-algorithm [9, 10], which can be applied with geometric sequences of \( \epsilon \).

Section 2 covers some background and notations for multi-loop integrals, and presents numerical results obtained with the PARINT package [11] for a set of 3-loop self-energy Feynman diagrams. These do not exhibit IR or UV singularities. Integrals with UV singularities are specified in Section 3, with their asymptotic expansions in the dimensional regularization parameter \( \epsilon \). In Section 3 we describe a novel method of numerical regularization for integrals with UV singularities, based on iterated integration and linear extrapolation, which is applied to a set of 1- and 2-loop self-energy diagrams. Section 4 shows numerical results of the regularization procedure.

2. Feynman loop integrals

Higher order corrections are required for accurate theoretical predictions of the cross-section for particle interactions. Loop diagrams are taken into account, leading to the evaluation of loop integrals. The derivation of a closed analytic form is sometimes hard for the higher order loop integrals with arbitrary internal masses and external momenta. Thus we resort to numerical calculations.

A \( L \)-loop integral with \( N \) internal lines can be represented in Feynman parameter space by

\[
\mathcal{I} = \frac{\Gamma(N - \frac{nL}{2})}{(4\pi)^{nL/2}} (-1)^N \int_0^1 \prod_{r=1}^N dx_r \, \delta(1 - \sum x_r) \frac{C^{N-n(L+1)/2}}{(D - i\rho C)^{N-nL/2}}, \tag{3}
\]

where \( C \) and \( D \) are polynomials determined by the topology of the corresponding diagram and physical parameters \( (C = 1 \) for 1-loop \( (L = 1 \) integrals). The integration in Eq (3) is taken over the \( N \)-dimensional unit cube. However, as a result of the \( \delta \)-function one of the \( x_r \) can be expressed in terms of the other ones, which reduces the integral dimension to \( N-1 \) and the domain to the \( (N-1) \)-dimensional unit simplex. In the absence of IR and UV singularities, \( n = 4 \). For dimensional regularization in case of IR singularities we set \( n = 4 + 2\epsilon \) (cf, [12]); and for UV singularities \( n = 4 - 2\epsilon \). We apply the regularization by a numerical extrapolation as \( \epsilon \to 0 \).

The term \( i\rho C \) prevents the integral from diverging if the denominator vanishes in the interior of the domain. Results using iterated integration with QUADPACK programs and extrapolation were given in [13, 7]. However with the parameters of Laporta [1], which we use in this paper, the denominator does not vanish inside the integration domain. In this case we can take \( \rho = 0 \) in the integrand of (3).

Integral approximations obtained with PARINT for 2-loop double-triangle \( (N = 5) \), tetragon-triangle \( (N = 6) \), pentagon-triangle \( (N = 7) \), ladder and crossed ladder \( (N = 7) \) were presented in [6]. A set of
Figure 1. 3-loop diagrams (a) $N = 7$ (Laporta[1] Fig 2(q)), (b) $N = 7$ (Laporta[1] Fig 2(r)), (c) $N = 7$ (Laporta[1] Fig 2(s)), (d) $N = 8$ (Laporta[1] Fig 2(t)), (e) $N = 8$ (Laporta[1] Fig 2(u), (f) $N = 8$ (Laporta[1] Fig 2(v))

Table 1. Parallel performance of PARINT (on MPI) for 3-loop diagrams of Fig 1, abs. tolerance $t_a = 5 \times 10^{-10}$, and max. number of evaluations = 10B

| 3-loop diag. | $N$ | Result $p = 1$ | Result $p = 64$ | $T_1$ | $T_{64}$ | $S_{64}$ |
|--------------|-----|----------------|-----------------|-------|---------|---------|
| Fig 1 (a)    | 7   | 1.32644820827  | 1.326448206     | 902.7 | 15.8    | 57.1    |
| Fig 1 (b)    | 7   | 1.34139924145  | 1.34139924147   | 1026.2| 14.4    | 71.3    |
| Fig 1 (c)    | 7   | 2.00250004111  | 2.00250004113   | 879.3 | 13.4    | 65.6    |
| Fig 1 (d)    | 8   | 0.27960892328  | 0.2796089227    | 1019.7| 15.9    | 64.1    |
| Fig 1 (e)    | 8   | 0.18262723754  | 0.1826272372    | 1018.3| 15.8    | 64.4    |
| Fig 1 (f)    | 8   | 0.14801330396  | 0.1480133036    | 976.6 | 16.4    | 59.5    |

3-loop self-energy diagrams is given in Fig 1, with corresponding PARINT performance results in Table 1. In order to compare our integral approximations with Laporta’s [1], we set all masses $m_r = 1$ and $s = 1$. Note that Laporta’s method is based on the numerical solution of systems of difference equations. For our numerical integration we transform the unit simplex domain of Eq (3) to the $(N-1)$-dimensional unit cube, and apply an integration rule of polynomial degree of precision 9 (from [14]) over the individual subregions resulting in the adaptive partitioning. The approximations thus obtained are more accurate than those generated with the multivariate simplex rules in PARINT, without the transformation. We set the absolute tolerance to $t_a = 5 \times 10^{-10}$ and the maximum number of integrand evaluations to $10B = 10^{10}$ (which was reached in producing the results of Table 1).

The results in Table 1 are given for $p = 1$ and for $p = 64$ MPI processes. $T_1$ is the time with one process and $T_{64}$ is the parallel time on our cluster with $p = 64$ processes (distributed over four 16-core, 2.6 GHz compute nodes). The speedup $S_{64} = T_1/T_{64}$ is (sequential time)/(parallel time). Note that superlinear speedups ($S_{64}$) are obtained in some cases, where the speedup exceeds the number of processes. This is partially due to the fact that the timing is done within PARINT, after the processes are started. The function evaluations are distributed over all the processes. Furthermore, the load balancing option is turned on, as well as letting the controller process act as a worker and participate in the region partitioning. However the adaptive partitioning reaches somewhat more accuracy sequentially. Each
process has its own priority queue, keyed with the absolute error estimates over their region. Thus unnecessary work may be done which increases with the number of processes. The presentation of this paper [15] contains results for $t_a = 5 \times 10^{-8}$ and a maximum of 5B evaluations, which are run in about half the time.

### 3. Ultra-violet (UV) singularities and asymptotics

#### 3.1. Integrals

By replacing $C = U$ and $D = UV$ in Eq (3), the general integral is written as

$$I = \frac{\Gamma \left( N - \frac{nL}{2} \right)}{(4\pi)^{nL/2}} (-1)^N \int_0^1 \prod_{r=1}^N dx_r \frac{\delta(1 - \sum x_r)}{U^{n/2} (V - iq)^{N-nL/2}}. \quad (4)$$

In the form of Eq (4), IR divergence occurs through a singularity arising when $V$ vanishes at the boundaries of the domain. This problem can be addressed by dimensional regularization with $n = 4 + 2\varepsilon$, which we implemented numerically in [13, 12, 7] by an extrapolation as $\varepsilon \to 0$ ($\varepsilon > 0$).

UV divergence occurs when $U$ vanishes at the boundaries. The $\Gamma$-function in (4), if it is divergent, may contribute to UV divergence. In this paper we treat UV divergence by a dimensional regularization with $n = 4 - 2\varepsilon$, implemented by a numerical extrapolation as $\varepsilon \to 0$ after an iterated integration with DQAGS from QUADPACK [3, 7] (to handle the boundary singularities).

Fig 2 depicts a 1-loop and three 2-loop self-energy diagrams with $N = 2, 3, 4$ and 5 internal lines. We refer to the 2-loop self energy diagrams (b), (c) and (d) as the sunrise-sunset, lemon and half-boiled egg diagram, respectively. Analytic results for the integrals have been derived by many authors. We use the formulas of Kato [16] for the functions $U$ and $V$ in (4). Let $V = \sum r x_r m^2_r - \frac{s}{\theta} W$, $s = q^2 = 1$, all masses $m_r = 1$ and $n = 4 - 2\varepsilon$.

In the case of the 1-loop self-energy diagram (Fig 2(a)), with $L = 1, N = 2$, the integrand has $U = 1$ and $W = x_1 x_2 = x_1 (1 - x_1)$, so $V = x_1^2 - x_1 + 1$; we denote

$$I_1 = \Gamma(\varepsilon) \int_0^1 \frac{1}{(x_1^2 - x_1 + 1)^\varepsilon} \, dx_1. \quad (5)$$
− For the sunrise-sunset diagram (Fig 2(b)) we have \( L = 2, N = 3 \) and

\[
I_s = (-1)\Gamma(-1 + 2\varepsilon) \int_0^1 dx_1 dx_2 dx_3 \frac{1}{1 - \sum_r x_r} \frac{(V - i\rho)^{1-2\varepsilon}}{U^{2-\varepsilon}}
\]

with \( U = x_1 x_2 + x_2 x_3 + x_3 x_1, \ W = x_1 x_2 x_3 \).

− The integral for the lemon diagram (Fig 2(c)) with \( L = 2, N = 4 \) is

\[
I_l = \Gamma(2\varepsilon) \int_0^1 dx_1 dx_2 dx_3 dx_4 \frac{1}{1 - \sum_r x_r} \frac{1}{U^{2-\varepsilon}(V - i\rho)^{2\varepsilon}}
\]

where \( U = x_1 x_2 + x_2 x_3 + x_3 x_4 \).

− The half-boiled egg diagram (Fig 2(d)) with \( L = 2, N = 5 \) gives rise to

\[
I_h = (-1)\Gamma(1 + 2\varepsilon) \int_0^1 dx_1 dx_2 dx_3 dx_4 dx_5 \frac{1}{1 - \sum_r x_r} \frac{1}{U^{2-\varepsilon}(V - i\rho)^{1+2\varepsilon}}
\]

where \( U = x_1 x_2 + x_2 x_3 + x_3 x_4 + x_4 x_5 \).

3.2. Asymptotic expansions for UV integrals
The integrals in Eqs (5)-(8) are expanded with respect to the dimensional regularization parameter \( \varepsilon \). The expansions are of the form of Eq (2),

\[
S(\varepsilon) \sim \sum_{k \geq K} C_k \varepsilon^k \quad \text{as} \quad \varepsilon \to 0,
\]

and we use linear extrapolation to approximate the coefficients of the leading terms. For the 1-loop, sunrise-sunset, and lemon integrals we expand \( S(\varepsilon) = I(\varepsilon)\Gamma(1 + \varepsilon)^{-1} \) where the integral \( I \) is \( I_1, I_s \) and \( I_l \), respectively. Section 4 compares corresponding extrapolation results with those given by Laporta [1]:

\[
I_1(\varepsilon) \Gamma(1 + \varepsilon)^{-1} \sim \sum_{k \geq 1} C_k \varepsilon^k = \varepsilon^{-1} + 0.186200635766 + 0.021156303568 \varepsilon + 0.0172674535 \varepsilon^2 \ldots
\]

\[
I_s(\varepsilon) \Gamma(1 + \varepsilon)^{-2} \sim \sum_{k \geq 2} C_k \varepsilon^k = -1.5 \varepsilon^{-2} - 4.25 \varepsilon^{-1} - 7.375 - 17.22197253479 \varepsilon \ldots
\]

\[
I_l(\varepsilon) \Gamma(1 + \varepsilon)^{-2} \sim \sum_{k \geq 2} C_k \varepsilon^k = 0.5 \varepsilon^{-2} + 0.6862006357658 \varepsilon^{-1} - 0.6868398873414 + 1.486398391913 \varepsilon \ldots
\]

Note that the value of \( K \) in (9) corresponds with the index of the first coefficient \( C_K \) in the expansion. In that case we find that, if \( K \) is replaced by \( K - 1 \) for the extrapolation, then the first coefficient converges to \( C_{K-1} = 0 \).

The half-boiled egg diagram is not covered in [1]. In order to compare with the leading order terms of [16] we expand

\[
\hat{I}_h(\varepsilon) = I_h(\varepsilon)/((-1)\Gamma(1 + 2\varepsilon)) \sim \sum_{k \geq -1} \hat{C}_k \varepsilon^k
\]

with \( I(\varepsilon) \) as in Eq (8). The analytic results in [16] are:

\[
\hat{C}_{-1} = J_1 = \int_0^1 \frac{\rho'}{M_0^2 - sG_0} \, dp'
\]

\[
\hat{C}_0 = -\frac{3}{2} J_1 - 2 J_2 + I_B
\]
Table 2. Results of UV 1-loop self-energy integral (on Mac Pro, 2.6 GHz Intel Core i7, 16GB memory, OS X), err. tol. $t_r = 10^{-14}$, $T(s)$ = Time (elapsed user time) (s); $\varepsilon = 1/b_l$ (starting at 1/6). $E_r =$ integration estim. rel. error

| $b_l$ | $E_r$ | $T(s)$ | $C_{-1}$ | $\hat{C}_{-1}$ | $C_0$ | $C_1$ | $C_2$ |
|------|-------|--------|----------|----------------|-------|------|------|
| 6    | 5.0e-15 | 1.2e-5 | 0.192483548522 | 0.021156303578 | 0.00172674499 |
| 8    | 5.0e-15 | 2.0e-6 | 0.99954860270457280 | 0.021184445970 | 0.00177553674 |
| 12   | 5.0e-15 | 2.0e-6 | 0.100000000000000000 | 0.021156346896 | 0.0017264999 |
| 16   | 5.0e-15 | 2.0e-6 | 0.99999998779988053 | 0.021156303568 | 0.0017264999 |
| 24   | 5.0e-15 | 2.0e-6 | 0.100000000000000000 | 0.021156346896 | 0.0017264999 |
| 32   | 5.0e-15 | 2.0e-6 | 0.999999999999999125 | 0.021156303578 | 0.0017264999 |
| 48   | 5.0e-15 | 2.0e-6 | 0.100000000000000000 | 0.021156346896 | 0.0017264999 |
| 64   | 5.0e-15 | 2.0e-6 | 0.999999999999999878 | 0.021156303568 | 0.0017264999 |

with

$$J_2 = \int_0^1 \frac{\rho^l \log(M_0^2 - sG_0)}{M_0^2 - sG_0} \, d\rho'$$  \hspace{1cm} (16)

$$I_B = \int_0^1 \int_0^1 \int_0^1 \frac{\rho^l (1-\rho)^2 \rho^2 M_0^2 - F^2 M^2 - s(G_0 - FG)}{\rho F(FM^2 - sG)(M_0^2 - sG)} \, d\rho \, d\rho' \, d\xi$$  \hspace{1cm} (17)

where $s = 1$ and

$$F = 1 - \rho + \rho \xi (1 - \xi), \quad F_0 = F(\rho = 0) = 1$$

$$G = (1-\rho)(1-\rho')(1-\rho')(1-\rho' \xi + \rho \xi (1 - \xi)), \quad G_0 = G(\rho = 0) = (1-\rho')\rho'$$

$$M^2 = \sum_{r} x_r m_r^2, \quad M_0^2 = M^2(\rho = 0) = \rho' m_3^2 + (1-\rho') m_2^2$$  \hspace{1cm} (18)

The latter holds under the assumption $m_3 = m_4$. We further have $M_r^2 = M^2 = 1$ in (18) in view of $m_r = 1$, $1 \leq r \leq 5$. Note that $\hat{C}_{-1} = C_{-1}$ in the expansion of $I_B = \sum_{k \geq -1} C_k \varepsilon^k$, but $\hat{C}_0 \neq C_0$.

4. Numerical extrapolation results for UV singularities

Tables 2 and 3 show the convergence of the extrapolation method for the integrals $I_1$ and $I_l$ of Eqs (5) and (7), respectively. These were run on a Mac Pro, 2.6 GHz Intel Core i7, with 16 GB memory, under OS X. The elapsed time $T(s)$ (in seconds) is listed for each integration. The time for the extrapolation is negligible compared to that of the integration. We use a standard linear system solver to solve very small systems (of sizes $2 \times 2$ up to around $15 \times 15$ for the cases in this paper). As the sequence of the extrapolation parameter $\varepsilon$ we used $\{1/b_l\}$ where $\{b_l\}$ is the Bulirsch sequence [8] started at an early index. Alternatively a slowly decreasing geometric sequence could be used - we plan on further testing with different sequences.

The convergence results in Tables 2 and 3 show excellent agreement with the expansion coefficients in [1] (see Eqs (10) and (12)). Throughout the extrapolation we keep track of the difference with the previous result as a measure of convergence. Increases of the distance between successive extrapolation results are an indicator that the convergence is no longer improving and the process can be terminated.

For the half-boiled egg integral we can compute $J_1$ in the analytic expressions of Eq (14) and $J_2$ of Eq (16) using DQAGE from QUADPACK [3, 7], yielding $J_1 \approx 0.6045997880780727$ with absolute error estimate 1.6e-15, and $J_2 \approx -0.11708165598778085$ with absolute error estimate 1.3e-17 on Mac Pro. In view of boundary singularities we use iterated integration by DQAGS for $I_B$ of Eq (17) and obtain $I_B \approx 0.4970393699155826$ with outer absolute error estimate 1.22e-15 (note that this error estimate does not include contributions from the inner integral error estimates). Then, using Eq (14) and Eq (15) we find $\hat{C}_{-1} = J_1 \approx 0.60459978807807210687$ and $\hat{C}_0 \approx -0.175697000225964850$ for the first two coefficients of the expansion in Eq (13). As shown in Table 4, good approximations are generated by linear extrapolation.
Table 3. Results UV lemon integral (on Mac Pro), err. tol. $t_r = 10^{-10}$ (outer), $5 \times 10^{-11}$ (inner two), $T(s) = \text{Time (elapsed user time)} (s)$; $\varepsilon = 1/b\ell$ (starting at 0.25). $E_r = \text{outer integration estim. rel. error}$

| $b_{\ell}$ | $T(s)$ | $C_{-2}$ | $C_{-1}$ | $C_{0}$ | $C_{1}$ |
|----------|--------|--------|--------|--------|--------|
| 4        | 3.5e-11 | 0.36   |         |         |        |
| 6        | 8.8e-11 | 0.34   | 0.513022162587 | 0.52467607220 |
| 8        | 2.9e-12 | 0.34   | 0.5031467341833 | 0.623429982959 | -0.237009170 |
| 12       | 3.4e-12 | 0.40   | 0.5004379328119 | 0.67218831764 | -0.5182745124 | 0.5208999 |
| 16       | 1.5e-11 | 0.41   | 0.500485801347 | 0.68386889795 | -0.643317369 | 1.0807577 |
| 24       | 4.7e-11 | 0.39   | 0.500037328535 | 0.68593187289 | -0.679195194 | 1.3495858 |
| 32       | 4.1e-11 | 0.38   | 0.5000000195177 | 0.68617780639 | -0.685884565 | 1.4654594 |
| 48       | 4.4e-11 | 0.37   | 0.5000000087538 | 0.68619903431 | -0.686759791 | 1.4837301 |
| 64       | 1.3e-11 | 0.36   | 0.5000000002937 | 0.68620057333 | -0.686834471 | 1.4864643 |
| 96       | 3.2e-11 | 0.31   | 0.5000000000339 | 0.68620057333 | -0.686839872 | 1.4863967 |

Eq (12): 0.5 0.68620063577 -0.686839887 1.4863984

Table 4. Results UV half-boiled egg integral (on Mac Pro), err. tol. $t_r = 10^{-12}$ (outer), $5 \times 10^{-13}$ (inner three), $T(s) = \text{Time (elapsed user time)} (s)$; $\varepsilon = 1/b\ell$ (starting at 1.0). $E_r = \text{outer integration estim. rel. error}$

| $b_{\ell}$ | $T(s)$ | $C_{-1}$ | $C_{0}$ | $C_{1}$ | $C_{2}$ |
|----------|--------|--------|--------|--------|--------|
| 1        | 4.2e-13 | 7.3    |        |        |        |
| 2        | 3.9e-14 | 11.4   | 0.6121795327003 | -0.26893337928 |
| 3        | 2.6e-13 | 10.8   | 0.62192201629541 | -0.29816083105 | -0.019484967 |
| 4        | 5.2e-13 | 10.3   | 0.60843456496759 | -0.21723612309 | -0.128876997 | 0.0809247 |
| 6        | 1.1e-13 | 14.9   | 0.605066159977971 | -0.1835206939 | -0.246771185 | 0.2493450 |
| 8        | 7.9e-13 | 9.2    | 0.60469393463841 | -0.1767647004 | -0.286828256 | 0.3591235 |
| 12       | 9.6e-13 | 12.5   | 0.60460287243867 | -0.17581097725 | -0.296034926 | 0.4010069 |
| 16       | 2.3e-13 | 19.4   | 0.604599696124784 | -0.17570617438 | -0.297527045 | 0.4117667 |
| 24       | 8.7e-13 | 19.1   | 0.60459979484875 | -0.17569752162 | -0.297707921 | 0.4137422 |
| 32       | 7.7e-13 | 24.9   | 0.60459978828848 | -0.17569702304 | -0.297723238 | 0.4139912 |
| 48       | 3.1e-13 | 33.2   | 0.60459978807821 | -0.17569700033 | -0.2977224241 | 0.4140148 |
| 64       | 6.9e-13 | 20.7   | 0.60459978807543 | -0.17569699900 | -0.297724269 | 0.4140158 |

Eq (14)-(15): 0.60459978807807 -0.17569700023

5. Conclusions

This paper presents new results for 2-loop self-energy diagrams (with 2, 3, 4 and 5 internal lines) with UV terms. These are computed with iterated integration using DQAGS from QUADPACK, and linear extrapolation, which delivers a novel numerical method for dimensional regularization of UV singularities. DQAGS is successful at treating the boundary singularities. The extrapolation yields accurate approximations for the leading term coefficients of the asymptotic expansion in the regularization parameter. The results so far have been verified with expansions by Laporta [1] and Kato [16], and we plan on testing the procedure for more complex diagrams.

New results are further obtained with the adaptive multivariate integration code from the parallel/distributed PARI NT package, for 3-loop self-energy diagrams without IR or UV singularities. The adaptive partitioning strategy is capable of dealing with higher dimensions than the iterated strategies. On the other hand, while it handles irregular integrand behavior to some extent, it cannot be expected to adequately partition higher-dimensional spaces in the vicinity of severe singularities. Future work on PARI NT includes testing and incorporation of special summation methods such as Kahan summation [17, 18, 19] (in view of the large numbers of function evaluations that can be performed especially on distributed processors). Other work for both the iterated and standard multivariate integration will be on further efficient parallelizations of the integration work performed throughout the extrapolation sequence.
Acknowledgments
We acknowledge the support from the National Science Foundation under Award Number 1126438. This work is further supported by Grant-in-Aid for Scientific Research (24540292) of JSPS, and the Large Scale Simulation Program No. 13/14-13 of KEK.

References
[1] Laporta S 2000 Int. J. Mod. Phys. A 15 5087–5159 arXiv:hep-ph/0102033v1
[2] de Boor C 1971 Mathematical Software ed Rice J R (Academic Press, New York) pp 417–449
[3] 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)
[4] de Doncker E and Yuasa F 2013 Journal of Physics: Conf. Series 410 012047 doi:10.1088/1742-6596/410/1/012047
[5] de Doncker E, Yuasa F and Assaf R 2013 Journal of Physics: Conf. Ser. 454 doi:10.1088/1742-6596/454/1/012082
[6] de Doncker E and Yuasa F 2014 XV Adv. Comp. and Anal. Tech. in Phys. Res., Journal of Physics: Conference Series (ACAT 2013) 523 doi:10.1088/1742-6596/523/1/012052
[7] 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
[8] Bulirsch R 1964 Numerische Mathematik 6 6–16
[9] Wynn P 1956 Mathematical Tables and Aids to Computing 10 91–96
[10] Shanks D 1955 J. Math. and Phys. 34 1–42
[11] de Doncker E, Kaugars K, Cucos L and Zanny R 2001 Proc. of Computational Particle Physics Symposium (CPP 2001) pp 110–119
[12] de Doncker E, Yuasa F and Kurihara Y 2012 Journal of Physics: Conf. Ser. 368
[13] de Doncker E, Fujimoto J, Hamaguchi N, Ishikawa T, Kurihara Y, Ljucovic M, Shimizu Y and Yuasa F 2010 Extrapolation algorithms for infrared divergent integrals arXiv:hep-ph/1110.3587; PoS (CPP2010)011
[14] Berntsen J, Espelid T O and Genz A 1991 ACM Trans. Math. Sotfw. 17 437–451
[15] de Doncker E, Yuasa F, Kato K, Ishikawa T and Olagbemi O Automatic numerical integration methods for Feynman integrals through 3-loop XV Adv. Comp. and Anal. Tech. in Phys. Res. (ACAT 2014), https://indico.cern.ch/event/258092/timetable/#all.detailed
[16] Kato K 2010 Note on 2-loop self-energy scalar integrals Tech. rep. Department of Physics, Kogakuin University, Japan Private communication
[17] Knuth D E 1998 The Art of Computer Programming, Volume 2, Seminumerical Algorithms 3rd ed (Addison Wesley) ISBN 0-201-89684-2
[18] Higham N J 2002 Accuracy and Stability of Numerical Algorithms 2nd ed. (SIAM) iSBN 978-0-898715-21-7
[19] de Doncker E, Kapenga J and Assaf R 2014 Numerical Computations with GPUs ed Kindratenko V (Springer) ISBN 978-3-319-06548-9