Numerical Evaluation of 3-Loop "Baseball" and "Running Shirt" Feynman Diagrams Using GPU-Accelerated QMC Integration and Extrapolation, Based on Sector Decomposition

Elise de Doncker[1(✉)], Hiroshi Daisaka[3], Tadashi Ishikawa[2], Kiyoshi Kato[4], Naohito Nakasato[5], and Fukuko Yuasa[2]

[1] Western Michigan University, Kalamazoo, MI 49008, USA
elise.dedoncker@wmich.edu
[2] High Energy Accelerator Research Organization (KEK), 1-1 OHO, Tsukuba, Ibaraki 305-0801, Japan
{tadashi.ishikawa,fukuko.yuasa}@kek.jp
[3] Graduate School of Business Administration, Hitotsubashi University, Kunitachi, Tokyo 186-8601, Japan
daisaka@phys.science.hit-u.ac.jp
[4] Department of Physics, Kogakuin University, Shinjuku, Tokyo 163-8677, Japan
katok@kute.tokyo
[5] Department of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu, Fukushima 965-8580, Japan
nakasato@u-aizu.ac.jp

Abstract. We consider integrals associated with two-point Feynman diagrams that have three loops and six internal lines. In particular, this paper studies two diagrams that, in view of their shape, are named "Baseball" and "Running Shirt" diagrams. Since the integrals are generally divergent, we transformed the integral into a suitable form using the sector decomposition method, and evaluated the resulting integrals with singular integrands using appropriate numerical methods.The combination of the lattice Quasi-Monte Carlo (QMC) rule and the extrapolation is used for the computation. The integration is accelerated using GPUs.

Keywords: 3-loop Feynman diagrams · QMC integration · Extrapolation · GPU

1 Introduction

Improvements in the technology of high energy physics experiments require theoretical predictions of commensurate accuracy. Higher order corrections are needed for accurate theoretical predictions of the cross-section for particle interactions.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16760, pp. 469–483, 2027.
https://doi.org/10.1007/978-3-032-30521-3_30

The Feynman diagrammatic approach is commonly used to address higher order corrections, and Feynman loop integrals arise in the calculations. Symbolic or symbolic/numerical calculations are done for some challenging problems using established software packages such as pySecDec [4], FIESTA [25] and MBnumerics [26].

A general and analytic treatment is not possible for general mass and momentum values, so we resort to numerical approaches. The number of dimensions and the degree of difficulty of the integration increase as the number of lines and loops increases, emphasizing the use of multivariate integration methods.

A Feynman integral with 3 loops and 6 lines can be represented by

$$\mathbf{I} = (-1)^6 \frac{\Gamma(6-3\nu/2)}{(4\pi)^{3\nu/2}} \int_{\mathcal{K}_6} \prod_{r=1}^{6} dx_r \frac{\delta(1-\sum x_r)}{U^{\nu/2} V^{6-3\nu/2}} = \frac{\Gamma(1+3\varepsilon)}{(4\pi)^{6-3\varepsilon}} \frac{1}{3\varepsilon} I. \qquad (1)$$

where U and V are determined by the topology of the corresponding diagram and physical parameters; $\nu = 4 - 2\varepsilon$ is the space-time dimension; $\mathcal{K}_6$ is the 6-dimensional unit hypercube. V for a two-point function is written in terms of a W function as

$$V = M^2 - s\frac{W}{U}, \quad M^2 = \sum_{r=1}^{6} m_r^2 x_r - i\varrho \qquad (2)$$

where the m_r are the particle masses of internal lines and s is the squared momentum P that flows into the diagram. The U and W functions are polynomials of Feynman parameters, x_r; U is third order and W is of order 4. As shown in Eq. (3), U is semi-positive definite and the zero occurring at the endpoint of the variables can produce the ultraviolet (UV) singularity, which is extracted efficiently through the sector decomposition method. V is positive for $s < 0$, and when we increase the value of s, it becomes zero in $\mathcal{K}_6$ above a certain value of s which we call the threshold. ϱ is a parameter to alleviate the singular behavior above the threshold.

There are eleven diagrams contributing to the 3-loop two-point function with six internal lines when the model for the Feynman diagram generation is restricted to include up to five-point vertices and to diagrams that are neither products of lower-order functions nor energy-independent [17]. Among these eleven diagrams, we already presented the result for the one diagram free from UV singularities in [10], as well as the results for the two diagrams involving the "complete" four-variable U function in [6].

In this paper, we handle the " Baseball (BB)" and " Running Shirt (RS)" diagrams as case studies (cf., Fig. 1). It is worth noting that diagrams (2n) and (2m) in Laporta [15] give rise to the same integrals as BB and RS, respectively.

Via a transformation $x_j = y_j$, $(1 \le j \le 4)$, $x_5 = y_5 z$, $x_6 = y_5 \bar{z}$, the function U for these diagrams is rewritten as

$$(\mathrm{BB}, \mathrm{RS}) \qquad U = y_1y_2y_3 + y_1y_2y_4 + y_1y_3y_4 + y_2y_3y_4 + y_{12}y_{34}y_5, \qquad (3)$$

and the W functions are rewritten as

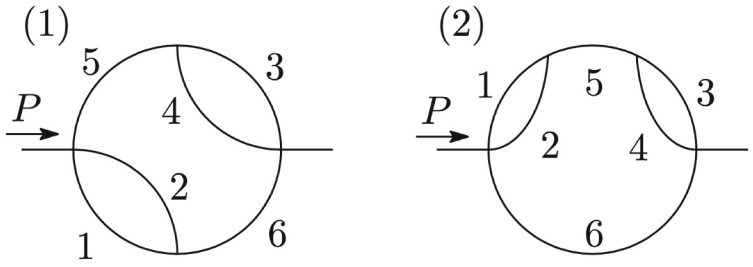

Fig. 1. Baseball (1) and Running Shirt (2) Feynman diagrams. The numbers assigned to the propagator denote the index r in Eq. (2)

$$\text{(BB)} \qquad W = y_1y_2y_3y_4 + y_1y_2y_{34}y_5z + y_{12}y_3y_4y_5\bar{z} + y_{12}y_{34}y_5^2z\bar{z}\,, \tag{4}$$

$$\text{(RS)} \qquad W = (y_1y_2y_{34} + y_{12}y_3y_4)y_5\bar{z} + y_{12}y_{34}y_5^2z\bar{z}\,. \tag{5}$$

where we use the notation $y_{ij} = y_i + y_j$ and $\bar{z} = 1 - z$. In this paper, we consider the case where all masses are equal ($m_r = m$).

While partial results for BB were presented in our previous work [7,9] (for only one or two sectors), in this paper we complete the evaluation of all contributions (15 sectors) to both BB and RS and present the results. In other work, we handled diagrams with up to five loops (e.g., [6–9,11]).

Information on the integrals obtained by the sector decomposition procedure, and the asymptotics of the integral will be given in Sect. 2. Section 3 describes the proposed numerical methods, including a combination of rank-1 lattice QMC integration and extrapolation. Section 4 presents numerical results for the fourth-order coefficients in the expansion of the illustrated three-loop two-point integrals, obtained using a GPU accelerator. Finally, the conclusions are presented in Sect. 5.

2 Formulation

We consider a Laurent expansion of the integral in Eq. (1),

$$I = \frac{1}{\varepsilon^2}C_{-2} + \frac{1}{\varepsilon}C_{-1} + C_0 + \varepsilon C_1 + \cdots\,. \tag{6}$$

The first three terms, i.e., the UV divergent coefficients, are required for the renormalization. As the genuine difference between BB and RS arises from the term $y_1y_2y_3y_4$ in W, it follows from the sector decomposition procedure (see, e.g., [1–3,13,14,16]) described below that the UV divergent coefficients are common

to both. The coefficients $C_k(k=-2,-1,0)$ of these diagrams can be expressed by the following formulae:

$$C_{-2}=1, \quad C_{-1}=2-3j_s^{(1)}(s),$$
$$C_0=8\{c_0+c_1j_s^{(1)}(s)+c_2j_s^{(2)}(s)-3(J_1(s)+J_2(s)+J_3(s))\} \tag{7}$$

where $c_0=-0.210710$, $c_1=(3/4)(\log 2-1)$, $c_2=9/16$,

$$j_s^{(k)}(s)=\int_0^1 dz\,(\log P_0(s))^k, \tag{8}$$

$$J_{1,2}(s)=\frac{1}{2}\int_0^1 dz\int_0^1 dt\int_0^1 du\,(u,1)\frac{\log P_{1,2}(s)}{(1+u+tu)^2}, \tag{9}$$

$$J_3(s)=\frac{1}{2}\int_0^1 dz\int_0^1 dt\int_0^1 du\,\frac{1}{t}\left(\frac{\log P_3(s)}{(1+u+tu)^2}-\frac{\log P_0(s)}{(1+u)^2}\right), \tag{10}$$

$$P_0(s)=m^2-sz\bar{z}, \quad P_{1,2,3}(s)=m^2(1+t+tu)-s\left((tu,t,1)z\bar{z}+\frac{tuz^2}{1+u+tu}\right). \tag{11}$$

The s-dependent functions have a threshold. While $j_s^{(1)}$, $j_s^{(2)}$ and J_3 have the normal threshold at $s=4m^2$, that of J_1 and J_2 is unphysical: The threshold of J_1 is at $s=8m^2$, where $z=3/4$, $t=1$, $u=1$ and that of J_2 is at $s=2(2+\sqrt{3})m^2$ where $z=\sqrt{3}/(\sqrt{3}+1)$, $t=1$, $u=(\sqrt{3}-1)/2$.

The coefficient C_1 is difficult to determine and we calculate C_1 numerically. We confirmed that analytical coefficients $C_k(k=-2,-1,0)$ in Eq. (7) coincide with the results obtained by the numerical calculations.

2.1 Sector Decomposition

The sector decomposition procedure is implemented as follows. We label sectors by $\alpha=(k,l,m,n,p)$, permutation of $(1,2,3,4,5)$. In each sector, we perform the variable transformation.

$$y_k=s_k, \quad y_l=s_kt_l, \quad y_m=s_kt_lu_m, \quad y_n=s_kt_lu_mv_n, \quad y_p=s_kt_lu_mv_nw_p. \tag{12}$$

Though there are $5!=120$ sectors, we only consider 15 sectors due to the symmetry as we study the equal mass case.

The final result is given by

$$I=8\sum_\alpha I(\alpha) \tag{13}$$

where α runs over the 15 permutations:

$$12345, 12354, 12534, 13245, 13254,$$
$$13425, 13452, 13524, 13542, 15234,$$
$$15324, 15342, 51234, 51324, 51342.$$

The U function is a sum of monomials, which are the product of three different y's. While there can be 10 such monomials of $y_1, \cdots, y_5$, U consists of 8 of them. If a $y_k y_l y_m$ is absent in U, i.e., k, l, m is the permutation of $(1, 2, 5)$ or $(3, 4, 5)$, we call the sectors/integrals of Type-II. The rest of sectors/integrals are Type-I.

By this transformation, the U- and W-functions in each sector are expressed as

$$\begin{array}{ll} U = s_k^3 t_l^2 u_m f & W = s_k^4 t_l^2 u_m q \quad (\text{Type} - \text{I}) \\ U = s_k^3 t_l^2 u_m v_n f & W = s_k^4 t_l^2 u_m v_n q \ (\text{Type} - \text{II}) \end{array} \tag{14}$$

$$\frac{V}{s_k} = G = m^2(1 + t_l(1 + u_m(1 + v_n(1 + w_p)))) - s\frac{q}{f}. \tag{15}$$

When it is redundant, the suffix of s, t, u, v, w is omitted. Thus U and W are replaced by f and q. For instance, when $\alpha = 12345$, $f = 1 + v + uv + uvw + uv^2w + tuv + tuvw + tuv^2w$ and $q = tuv(1 + wz + vwz + uvw\bar{z} + uvw^2\bar{z}z + uv^2w^2\bar{z}z + tuvw\bar{z} + tuvw^2\bar{z}z + tuv^2w^2\bar{z}z)$. Please note that s_k appears in neither f nor q and that f is a regular function.

The integral is given by the following formula. Functions f and q are defined in each sector α.

$$(\text{Type} - \text{I}) \qquad I(\alpha) = \int_0^1 dz \int_0^1 dt \int_0^1 du \int_0^1 dv \int_0^1 dw \frac{hv}{t^{1-2\varepsilon} u^{-\varepsilon}} H, \tag{16}$$

$$(\text{Type} - \text{II}) \qquad I(\alpha) = \int_0^1 dz \int_0^1 dt \int_0^1 du \int_0^1 dv \int_0^1 dw \frac{h}{t^{1-2\varepsilon} u^{-\varepsilon} v^{1-\varepsilon}} H, \tag{17}$$

where $h = y_5/s_k$ and $H = G^{-3\varepsilon}/f^{2-\varepsilon}$.

2.2 Integrands

The sector decomposition procedure allows us to easily expand the integrands of the Type-I and Type-II integrals in ε. In this section, we present the resulting integrands, F, for the 15 sectors. Each F corresponds to the integrand of either the integral in Eq. (16) or Eq. (17). These expressions can be used to evaluate each term analytically and/or numerically. We denote $f_0 = f|_{t=0,v=0}$, $f_a = f|_{t=0}$ and $f_b = f|_{v=0}$ and use a similar notation for G.

$I(12345)$, $I(12354)$, $I(13245)$, $I(13254)$, $I(13425)$, $I(13452)$, $I(13524)$, $I(13542)$, $I(15324)$ and $I(15342)$. The integrals are of Type-I.

$$F = \frac{hv}{tf^2} + \frac{hv}{tf^2}(\log f + 2\log t + \log u - 3\log G)\varepsilon + O(\varepsilon^2) \tag{18}$$

and the s-dependent part arises from the G function.

***I*(12534) and *I*(15234).** The integrals are of Type-II and the integrand can further be written as $F = F_0 + F_1$, with

$$F_0 = \frac{h}{tf_b^2}\frac{1}{\varepsilon} + \frac{h}{tf_b^2}\left(\log f_b + 2\log t + \log u - 3\log G_b\right) + \frac{h}{2tf_b^2}(\log f_b + 2\log t + \log u - 3\log G_b)^2\varepsilon + O(\varepsilon^2) \tag{19}$$

and the s-dependent part comes from the G_b function;

$$F_1 = \frac{h}{tv}\left(\frac{1}{f^2} - \frac{1}{f_b^2}\right) + \frac{h}{tv}\left(\frac{\log f + 2\log t + \log u + \log v - 3\log G}{f^2} - \frac{\log f_b + 2\log t + \log u + \log v - 3\log G_b}{f_b^2}\right)\varepsilon + O(\varepsilon^2) \tag{20}$$

and the s-dependent part appears from the G and G_b functions.

***I*(51234).** The integral is of Type-II and the integrand can be written as $F = F_{00} + F_{01} + F_{10} + F_{11}$, where F_{01} vanishes completely and is therefore omitted.

$$F_{00} = \frac{h}{2f_0^2}\frac{1}{\varepsilon^2} + \frac{h}{2f_0^2}\left(\log f_0 + \log u - 3\log G_0\right)\frac{1}{\varepsilon} + \frac{h}{4f_0^2}\left(\log f_0 + \log u - 3\log G_0\right)^2 + \frac{h}{12f_0^2}\left(\log f_0 + \log u - 3\log G_0\right)^3\varepsilon + O(\varepsilon^2) \tag{21}$$

and the s-dependent part comes from the G_0 function;

$$F_{10} = \frac{h}{t}\left(\frac{1}{f_b^2} - \frac{1}{f_0^2}\right)\frac{1}{\varepsilon} + \frac{h}{t}\left(-\frac{\log f_0 + 2\log t + \log u - 3\log G_0}{f_0^2} + \frac{\log f_b + 2\log t + \log u - 3\log G_b}{f_b^2}\right) + \frac{h}{2t}\left(-\frac{(\log f_0 + 2\log t + \log u - 3\log G_0)^2}{f_0^2} + \frac{(\log f_b + 2\log t + \log u - 3\log G_b)^2}{f_b^2}\right)\varepsilon + O(\varepsilon^2) \tag{22}$$

and the s-dependent part arises from the G_0 and G_b functions;

$$F_{11} = \frac{h}{tv}\left(\frac{1}{f^2} - \frac{1}{f_b^2}\right) + \frac{h}{tv}\left(-\frac{\log f_b + 2\log t + \log u + \log v - 3\log G_b}{f_b^2} + \frac{\log f + 2\log t + \log u + \log v - 3\log G}{f^2}\right)\varepsilon + O(\varepsilon^2) \tag{23}$$

and the s-dependent part comes from the G and G_b functions.

***I*(51324) and *I*(51342).** The integrals are of Type-I and the integrand can be written as $F = F_0 + F_1$, where

$$F_0 = \frac{hv}{2f_a^2}\frac{1}{\varepsilon} + \frac{hv}{2f_a^2}\left(\log f_a + \log u - 3\log G_a\right) + \frac{hv}{4f_a^2}\left(\log f_a + \log u - 3\log G_a\right)^2\varepsilon + O(\varepsilon^2) \tag{24}$$

and the s-dependent part appears from the G_a function;

$$F_1 = \frac{hv}{t}\left(\frac{1}{f^2} - \frac{1}{f_a^2}\right) + \frac{hv}{t}\left(-\frac{\log f_a + 2\log t + \log u - 3\log G_a}{f_a^2} + \frac{\log f + 2\log t + \log u - 3\log G}{f^2}\right)\varepsilon + O(\varepsilon^2) \tag{25}$$

and the s-dependent part comes from the G and G_a functions.

2.3 Asymptotics

In the numerical evaluation, we consider the integrals for the coefficient C_1 as a function of ϱ and construct a sequence of approximations using an appropriate numerical integration method with a decreasing sequence of ϱ. Then we perform an extrapolation to the limit as $\varrho \to 0$. This method for an extrapolation of the sequence relies on the existence of an asymptotic expansion of each integral $\mathcal{I}(\varrho) \sim \mathcal{I} + a_1\phi_1(\varrho) + a_2\phi_2(\varrho) + \cdots$, as $\varrho \to 0$. Although linear or nonlinear extrapolation methods can be explored under certain conditions on the functions $\phi_\ell(\varrho)$, in this paper we use a nonlinear method. The details of the numerical integration and the extrapolation method used in this paper are described in the next section.

3 Numerical Methods

3.1 Lattice QMC Integration

The QMC integration used in this work is based on lattice rules. Rank-1 lattice rules provide an integral approximation over the unit cube of the form

$$\mathcal{Q}f = \mathcal{Q}(\mathbf{z}, N)f = \frac{1}{N}\sum_{j=0}^{N-1} f(\{\frac{j}{N}\mathbf{z}\}) \tag{26}$$

Here, N, the sample size, is a large prime, $\{\mathbf{x}\}$ denotes the vector obtained by taking the fractional part of each component of $\mathbf{x}$ and $\mathbf{z}$ is a generator vector of integer components determined by the specific algorithm for a given N (see [24]). In the current work, N is generally between $\mathcal{O}(10^6)$ and $\mathcal{O}(10^8)$.

A sequence embedding of a lattice rule is defined in [24], based on a rank-1 rule Q_0 and where each rule of the sequence applies a scaled copy of Q_0 to the subregions obtained by subdividing $\mathcal{K}_d$ into m equal parts in each of r coordinate directions. Denoting

$$\mathcal{Q}_r f = \mathcal{Q}_r^{(m)}(\mathbf{z}, N)f = \frac{1}{m^r N}\sum_{k_r=0}^{m-1}\cdots\sum_{k_1=0}^{m-1}\sum_{j=0}^{N-1} f\left(\{\frac{j}{N}\mathbf{z} + \frac{1}{m}(k_1,\ldots,k_r,0,\ldots,0)\}\right) \tag{27}$$

for N and m relatively prime, the points of Q_r are embedded in Q_{r+1} for $0 \le r < d$. Q_r has $m^r N$ points and is of rank r for $1 \le r \le d$, and Q_d is composed of m^d scaled copies of Q_0. An error estimate for Q_d is calculated based on a sequence of rules $Q^{(i)}$, $1 \le i \le d$, of order $m^{d-1}N$ and embedded in Q_d [24]. The results for the 5-dimensional integrals ($d = 5$) in Sect. 4 of this paper are obtained with $m = 4$.

3.2 Transformation

Since lattice rules are applied with a periodizing transformation of the integrand, we select a transformation that also suppresses singularities on the boundaries of the domain. We make use of Sidi's Ψ_m-transformation [21–23], $\Psi_m(t) = \frac{\int_0^t (\sin\pi u)^m}{\int_0^1 (\sin\pi u)^m}$. In this paper we use the Ψ_6-transformation and it is given as

$$\Psi_6(t) = t - (45\sin(2\pi t) - 9\sin(4\pi t) + \sin(6\pi t))/(60\pi), \quad \Psi_6'(t) = \frac{16}{5}\sin^6(\pi t). \tag{28}$$

3.3 Extrapolation

With the goal of obtaining the extrapolated results numerically, we use the ϵ-algorithm [20,27] with an implementation based on the code of *qext()* from the QuadPack package [19]. The computation produces a triangular table, with a

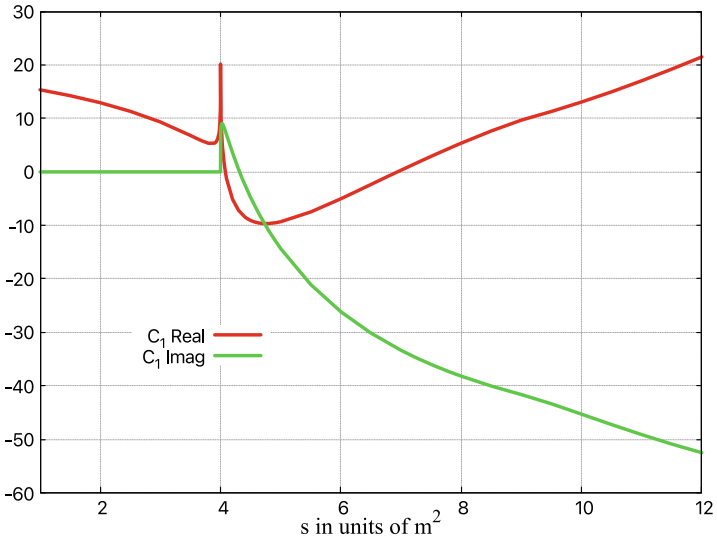

Fig. 2. Baseball C_1 by QMC integration

sequence of integral approximations as the entry sequence in the first column. Each subsequent column is generated from neighboring entries of the previous ones. The *qext()* procedure computes a new lower diagonal of the table when a new entry is available, together with a measure of the distance of each element from preceding neighboring elements. In the construction of the triangular table, a table element *"new"* relative to preceding $e_0,\ e_1,\ e_2,\ e_3$ is obtained as

$$\begin{matrix} & e_0 & \\ e_3 & e_1 & new \\ & e_2 & \end{matrix} = e_1 + 1/(1/(e_1 - e_3) + 1/(e_2 - e_1) - 1/(e_1 - e_0)) \tag{29}$$

and the distance for *new* is set to

$$distance = |e_2 - e_1| + |e_1 - e_0| + |e_2 - new|. \tag{30}$$

Then *qext()* returns the new lower diagonal element with the smallest value of the distance estimate as the extrapolation result.

4 Results

For the numerical results, we focus on the computation of the coefficient C_1 of the linear term in a Laurent expansion. We studied the expansions and their lower-order coefficients in earlier work, e.g., [7,9] for the Baseball and various other diagrams. Figure 2 shows plots of Baseball (BB) C_1 as a function of the squared momentum s and Fig. 3 shows a similar plot for Running Shirt (RS).

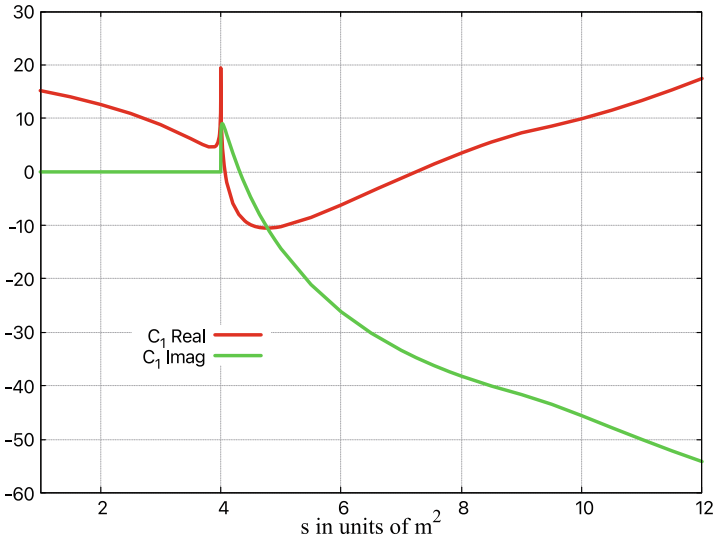

Fig. 3. Running Shirt C_1 by QMC integration

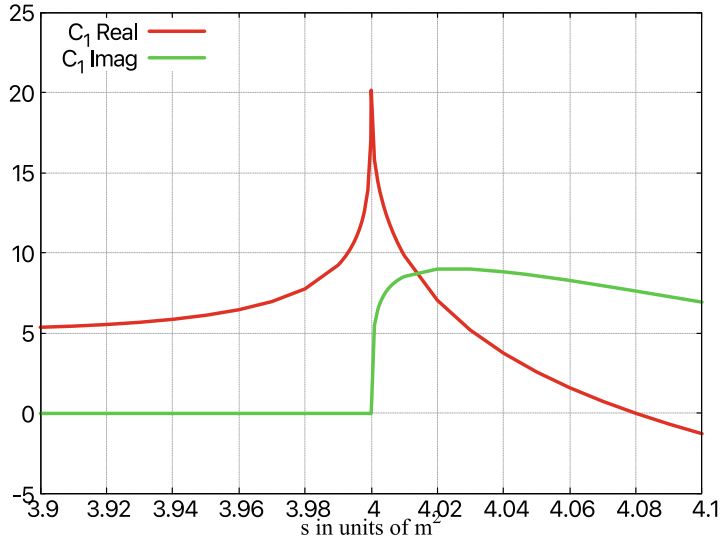

Fig. 4. Baseball C_1 around s = 4.0

As the diagrams have a threshold (singularity) at $s = 4$, we plot the behavior around $s = 4$ for BB and RS in Figs. 4 and 5, respectively. Note that the real part peaks at the threshold and the imaginary part emerges (for $s \geq 4$).

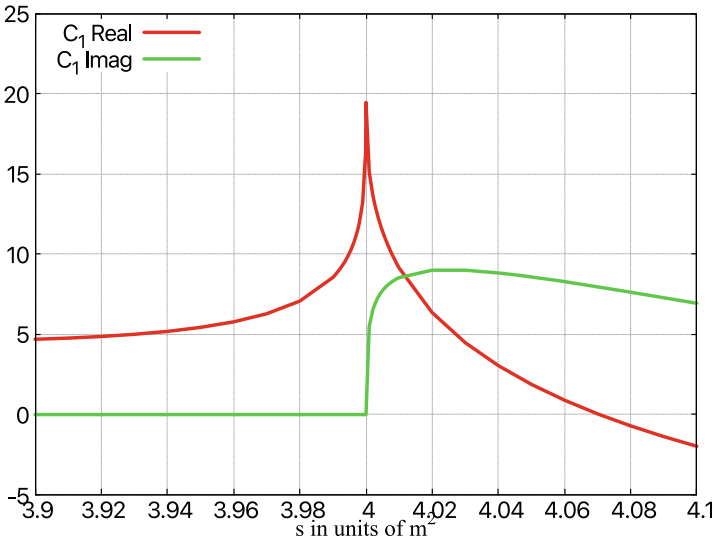

Fig. 5. Running Shirt C_1 around s = 4.0

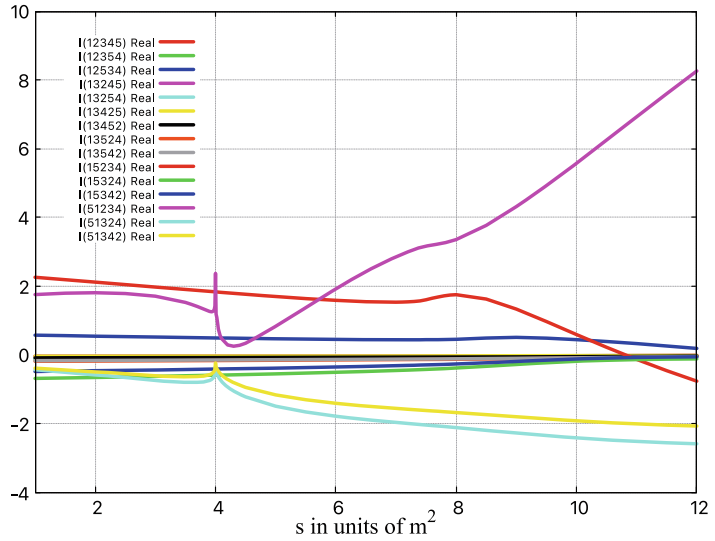

Fig. 6. Baseball C_1 real part 15 sectors by QMC integration

In the numerical computation, we set all masses equal, $m_r = 1, (r = 1, \cdots, 6.)$ For the QMC integration, we use the parameters $N = 400\text{M}$ [5] and $m = 4$, and for the extrapolation we take $\varrho = 1.2^{-24}, \cdots, 1.2^{-38}$ for most values of s. The numerical results are in agreement with those obtained by pySecDec [4], with a

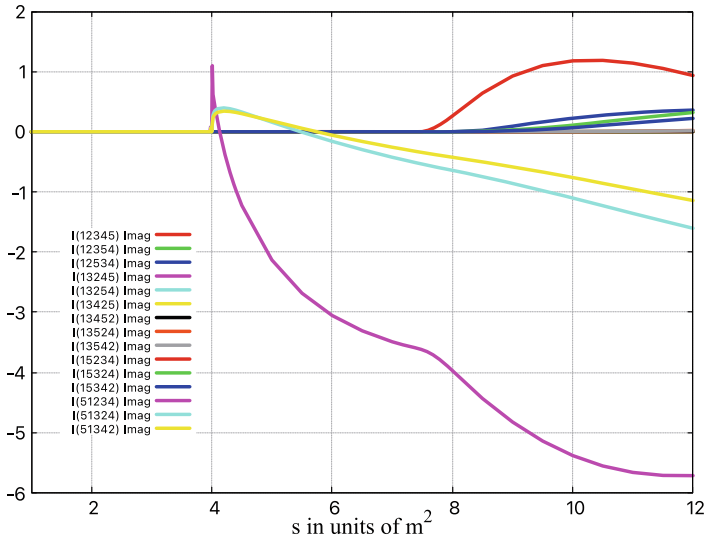

Fig. 7. Baseball C_1 imag part 15 sectors by QMC integration

precision of about five to six digits. For example, for $s = 1$ the coefficient C_1 of the real part of BB is 15.351339, while the corresponding result from pySecDec is 15.351340 $\pm 3.8 \times 10^{-7}$, and C_1 of the real part of RS is 15.200684 and from pySecDec it is 15.200685 $\pm 3.9 \times 10^{-7}$.

Figures 6 and 7 give plots of BB C_1 (real and imaginary) for the 15 sectors. The 15 solid curves correspond to the contributions from individual sectors. The sector decomposition method provides an effective means of isolating UV divergences and improving the convergence of individual integrands. However, because the integration domain is partitioned, the energy dependence of each component may exhibit unphysical behavior associated with artificial thresholds. As illustrated in Fig. 2, once the contributions from all sectors are summed, the final result shows a natural and physically consistent behavior.

The QMC integrations were performed using an NVIDIA GH200 Grace Hopper Superchip [18] equipped on the Miyabi supercomputer system [12]. The extrapolations are performed using Intel(R) Xeon(R) Gold 6230 CPU @ 2.10 GHz. The integrations and extrapolations were computed in double precision.

The elapsed time for the QMC integration in each s value with $m = 4$ ($400M \times 4^5$ sample points) was around 11900 s using a GH200 and that for the extrapolation was negligible. The elapsed time for our method depends on the total number of sampling points. In contrast, the elapsed time for pySecDec depends on the value of s. For values of s in the range $1.0 < s < 12.0$, the observed elapsed time ranged from 36 s to 3026 s on an NVIDIA GV100. Further

improvements to reduce the elapsed time of our method will be pursued by refining the CUDA code and implementing MPI parallelization.

5 Conclusions

In this paper, we studied two three-loop two-point Feynman diagrams. In particular, we evaluated the fourth term in the Laurent expansion for ε, where the difference between the two diagrams appears.

The Feynman integral becomes singular above the threshold, and some other methods use a deformation of the contour to avoid the zero of V, while the control of the deformation is sometimes complicated and tedious. In our numerical method, the zero of V is avoided by taking the regularization parameter ρ as a finite number. When one can use a good precision integration and extrapolation, it is possible to obtain the value of the integral, though the cost of computation increases for the evaluation of the sequence of approximations.

We obtained the energy dependence of the loop integrals. The methods employed in this work, the combination of the GPU-accelerated QMC integration and the numerical extrapolation, have been shown to be effective. We also obtained results for values of s beyond the threshold. The numerical results achieve an accuracy of five to six digits, which is sufficient for phenomenological applications. As a future direction, the present approach can be extended to other three-loop diagrams and higher order loop integrals.

Acknowledgements. We acknowledge the support by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant Numbers 25K07331. This research used computational resources of Miyabi-G provided by the Multidisciplinary Cooperative Research Program in the Center for Computational Sciences, University of Tsukuba.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Binoth, T., Heinrich, G.: An automized algorithm to compute infrared divergent multi-loop integrals. Nucl. Phys. B **585**, 741–759 (2000). arXiv:hep-ph/0004013
2. Binoth, T., Heinrich, G.: Numerical evaluation of multi-loop integrals by sector decomposition. Nucl. Phys. B **680**, 375 (2004). arXiv:hep-ph/0305234
3. Binoth, T., Heinrich, G.: Numerical evaluation of phase space integrals by sector decomposition. Nucl. Phys. B **693**, 134 (2004). arXiv:hep-ph/0402265
4. Borowka, S., Heinrich, G., Jahn, S., Jones, S.P., Kerner, M., Schlenk, J.: A GPU compatible quasi-Monte Carlo integrator interfaced to pySecDec. Comput. Phys. Commun. **240**, 120–137 (2019). arXiv:1811.11720
5. de Doncker, E., Almulihi, A., Yuasa, F.: Transformed lattice rules for Feynman loop integrals. IOP Ser. J. Phys. Conf. Ser. (JPCS) **1136**(012002) (2018). https://doi.org/10.1088/1742-6596/1136/1/012002

6. de Doncker, E., Ishikawa, T., Kato, K., Yuasa, F.: Analytic and numerical approaches for depictive 3-loop integrals using sector decomposition. Prog. Theor. Exp. Phys. (2024). arXiv:2405.13286v1 [hep-ph]. https://doi.org/10.48550/arXiv.2405.13286
7. de Doncker, E., Yuasa, F., Ishikawa, T., Kato, K.: 3-loop Feynman integral extrapolations for the baseball diagram. In: The 22nd International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2024) (2024 To appear)
8. de Doncker, E., Yuasa, F.: Self-energy Feynman diagrams with four loops and 11 internal lines. In: Lecture Notes in Computer Science (LNCS), vol. 12953, pp. 160–175. Springer (2021)
9. de Doncker, E., Yuasa, F., Ishikawa, T., Kato, K.: Methods for extrapolation of 3-loop 2-point Feynman integrals. In: Lecture Notes in Computer Science (LNCS), vol. 15890, pp. 20–30. Springer (2025). https://doi.org/10.1007/978-3-031-97606-3_2
10. de Doncker, E., Yuasa, F., Kato, K., Ishikawa, T., Kapenga, J., Olagbemi, O.: Regularization with numerical extrapolation for finite and UV-divergent multi-loop integrals. Comput. Phys. Commun. **224**, 164–185 (2018). arXiv:1702.04904
11. de Doncker, E., Yuasa, F., Olagbemi, O., Ishikawa, T.: Large scale automatic computations for Feynman diagrams with up to five loops. In: Lecture Notes in Computer science (LNCS), vol. 12253, pp. 145–162. Springer (2020)
12. Hanawa, T., et al.: Miyabi: new supercomputer system of JCAHPC (2024). https://doi.org/10.24669/axies.2024.0_154. (in Japanese)
13. Heinrich, G.: Sector decomposition. Int. J. Mod. Phys. A **23**, 1457 (2008). arXiv:0803.4177
14. Heinrich, G.: Collider physics at the precision frontier. Phys. Rep. **922**, 1 (2021). arXiv:2009.00516
15. Laporta, S.: High-precision calculation of multi-loop Feynman integrals by difference equations. Int. J. Mod. Phys. A **15**, 5087–5159 (2000). arXiv:hep-ph/0102033
16. Nakanishi, N.: Graph Theory and Feynman Integrals. Gordon and Breach, New York (1971)
17. Nogueira, P.: Automatic Feynman graph. J. Comput. Phys. **105**, 279–289 (1993). https://doi.org/10.1006/jcph.1993.1074
18. NVIDIA. https://www.nvidia.com/en-us/data-center/grace-hopper-superchip/
19. Piessens, R., de Doncker, E., Überhuber, C.W., Kahaner, D.K.: QUADPACK, A Subroutine Package for Automatic Integration. Springer Series in Computational Mathematics, vol. 1. Springer-Verlag (1983)
20. Shanks, D.: Non-linear transformations of divergent and slowly convergent sequences. J. Math. Phys. **34**, 1–42 (1955)
21. Sidi, A.: A new variable transformation for numerical integration. Int. Ser. Numer. Math. **112**, 359–373 (1993)
22. Sidi, A.: Extension of a class of periodizing transformations for numerical integration. Math. Comp. **75**(253), 327–343 (2005)
23. Sidi, A.: Further extension of a class of periodizing variable transformations for numerical integration. J. Comput. Appl. Math. **221**, 132–149 (2008)
24. Sloan, I., Joe, S.: Lattice Methods for Multiple Integration. Oxford University Press (1994)
25. Smirnov, A.V.: FIESTA4: optimized Feynman integral calculations with GPU support. Comput. Phys. Commun. **204**, 189–199 (2016). https://doi.org/10.1016/j.cpc.2016.03.013. arXiv:1511.03614

26. Usovitsch, J., Dubovyk, I., Riemann, T.: MBnumerics: numerical integration of Mellin-Barnes integrals in physical regions. PoS LL **046**, 1810.04580 (2018). [hep-ph]. https://doi.org/10.22323/1.303.0046
27. Wynn, P.: On a device for computing the $e_m(s_n)$ transformation. Math. Tables Aids Comput. **10**, 91–96 (1956)