Loop integral computation in the Euclidean or physical kinematical region using numerical integration and extrapolation

E de Doncker[1], F Yuasa[2], T Ishikawa[2] and K Kato[3]

[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] Department of Physics, Kogakuin University, Shinjuku, Tokyo 163-8677, Japan

E-mail:
elise.dedoncker@wmich.edu, fukuko.yuasa@kek.jp, tadashi.ishikawa@kek.jp, katok@kute.tokyo

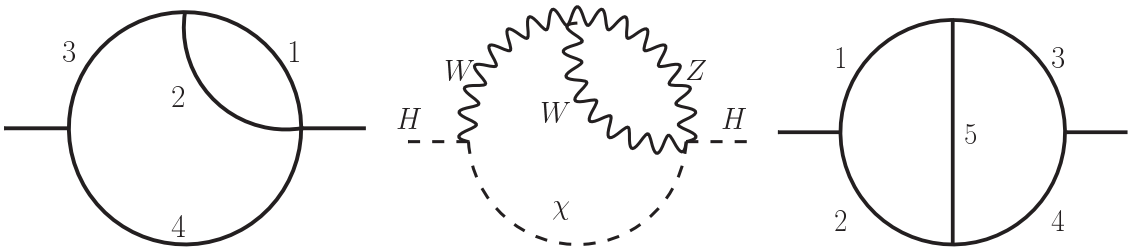
Abstract. The computation of loop integrals is required in high energy physics to account for higher-order corrections of the interaction cross section in perturbative quantum field theory. Depending on internal masses and external momenta, loop integrals may suffer from singularities where the integrand denominator vanishes at the boundaries, and/or in the interior of the integration domain. To handle a threshold singularity originating from a vanishing denominator in the interior of the domain, we add a term (of the form $-i\varrho$) in the denominator, and perform a nonlinear extrapolation to a sequence of integrals obtained for a (geometrically) decreasing sequence of ϱ.

The integral behavior may also be affected by UV singularities, which we treat using dimensional regularization, where the space-time dimension $\nu = 4$ is replaced by $\nu = 4 - 2\varepsilon$ for a sequence of ε values, and a linear extrapolation is applied as ε tends to zero. Presence of both types of singularities may warrant a double extrapolation. In this paper we will devise and apply a strategy for loop integral computations by combining these methods as needed for a set of Feynman diagrams. In view of the compute-intensive nature, the code is further multi-threaded to run in a shared memory environment.

1. Introduction

Accurate theoretical predictions are needed in view of improvements in the technology of high energy physics experiments. Higher order corrections are required for accurate theoretical predictions of the cross-section for particle interactions. The Feynman diagrammatic approach is commonly used to address higher order corrections, and Feynman loop integrals arise in the calculations. Loop integrals may suffer from integrand singularities or irregularities at the boundaries and/or in the interior of the integration domain (for physical kinematics). Symbolic or symbolic/numerical calculations have been performed for challenging problems [1], and there are continuous efforts to develop software packages for the evaluation of loop integrals using numerical approaches, such as pySecDec [2, 3, 4], feyntrop [5], FIESTA [6], and MBnumerics [7].

In previous work we implemented iterated integration numerically using one- or low-dimensional adaptive integration algorithms in subsequent coordinate directions, enabling intensive subdivision in the vicinity of singularities. To handle a singularity in the domain interior, we add a term $-i\varrho$ in the denominator, and perform a nonlinear extrapolation (as $\varrho \to 0$) on a sequence of integrals obtained for a (geometrically) decreasing sequence of ϱ. UV singularities are treated by dimensional regularization,



(a) (b) (c)

Figure 1: Sample 2-loop diagrams (a) Lemon-1 $(1,1,1,1)$, $N=4$; (b) Lemon-2 (M_Z, M_W, M_W, M_χ), $N=4$; (c) Magdeburg, $N=5$

where the space-time dimension $\nu = 4$ is replaced by $\nu = 4-2\varepsilon$ for a sequence of ε values, and a linear extrapolation is applied as $\varepsilon \to 0$. Presence of both types of singularities warrants a double extrapolation. Some of the code is further multi-threaded to run in a shared memory environment. We will demonstrate combinations of the methods for sample diagrams.

2. Feynman Loop Integrals

2.1. Representation

The scalar L-loop integral with N internal lines is represented by

$$\begin{aligned}\mathcal{I} &= \Gamma(N-\frac{\nu L}{2})\,(-1)^N \int_{\mathcal{C}_N} \prod_{r=1}^{N} dx_r\,\delta(1-\sum x_r)\,U^{-\nu/2}(V-i\varrho)^{\nu L/2-N} \\ &= \Gamma(N-\frac{\nu L}{2})\,(-1)^N \int_{\mathcal{S}_{N-1}} \prod_{r=1}^{N-1} dx_r\,U^{-\nu/2}(V-i\varrho)^{\nu L/2-N} \end{aligned} \tag{1}$$

where $V = M^2 - W/U,\ \ M^2 = \sum_r m_r^2 x_r$; U and W are polynomials determined by the topology of the corresponding diagram and physical parameters; $\nu = 4-2\varepsilon$ is the space-time dimension; $\varrho = 0$ unless V vanishes within the domain; $\mathcal{C}_N$ is the N-dimensional unit hypercube and $\mathcal{S}_d = \{\mathbf{x} \in \mathcal{C}_d \mid \sum_{j=1}^{d} x_j \le 1\}$ is the d-dimensional unit simplex.

Note that, whether V or $M^2U - W$ is used for subsequent computations, the extrapolated results will remain similar as long as the numerical integration for each ρ is performed with good accuracy.

2.2. Asymptotics

The integrals computed in this paper satisfy an asymptotic expansion of the form

$$\mathcal{I} = \mathcal{I}(\varepsilon) \sim \sum_{k \ge \kappa} C_k\,\varepsilon^k \qquad \text{as } \varepsilon \to 0$$

For the cases considered here, we have $\kappa = -2$ for Fig. 1 (a) and (b), so that

$$\mathcal{I}(\varepsilon) \sim C_{-2}/\varepsilon^2 + C_{-1}/\varepsilon + C_0 + \ldots \tag{2}$$

Analytic derivations of the asymptotic behavior for 2-loop self-energy integrals were given in [8].

2.3. Sample 2-loop Diagrams

We give results for the diagrams of Fig. 1. For the loop integrals corresponding to the Lemon-1, 2

diagrams of Fig. 1 (a) and (b), respectively, we have

$$U = x_{12}x_{34} + x_1x_2, \quad W = s\,(x_4(x_1x_2 + x_1x_3 + x_2x_3)), \;\; \text{with} \;\; x_{k\ell\ldots n} = x_k + x_\ell + \ldots + x_n$$

Here we assign the masses (m_1, m_2, m_3, m_4) as (M_H, M_H, M_H, M_H) for the Lemon-1 diagram, and as (M_Z, M_W, M_W, M_χ) for Lemon-2. Measuring the particle masses using the Higgs mass as the unit, we have $M_W = 0.64308$, $M_Z = 0.7295008$ and $M_\chi = M_W$, and subsequently we simply express (M_H, M_H, M_H, M_H) as $(1, 1, 1, 1)$.

The Magdeburg integral with the masses $(m_1, m_2, m_3, m_4, m_5)$ satisfies

$$U = x_{12}x_{34} + x_{1234}x_5, \quad W = s\,(x_1x_2x_{34} + x_3x_4x_{12} + x_{13}x_{24}x_5)$$

and we compute the case $(1, 1, 1, 1, 1)$.

3. Methods

3.1. Iterated Automatic Adaptive Integration

3.1.1. Automatic Integration. Automatic integration employs a black-box approach to produce (as outputs) an approximation $Q(f)$ to an integral $If = \int_{\mathcal{D}} f(\vec{x})\, d\vec{x}$, and an error estimate $\mathcal{E}f$ of the actual error $Ef = |Qf - If|$, in order to satisfy an accuracy requirement of the form

$$|\,Qf - If\,| \;\leq\; \mathcal{E}f \;\leq\; \max\{\,t_a\,,\, t_r\,|If|\,\},$$

where the integrand function f, region $\mathcal{D}$ and (absolute/relative) error tolerances t_a and t_r, respectively, are specified as part of the input.

3.1.2. Automatic Adaptive Integration. This is a category of Automatic Integration (cf., Section 3.1.1), applied adaptively (by region partitioning). The meta-algorithm of Fig. 2 is implemented in the Quadpack programs DQAGE and DQAGSE. The latter applies a nonlinear extrapolation in addition to the DQAGE global adaptive strategy [9, 10, 11].

```
Evaluate initial region and update results
Initialize priority queue with initial region
while (evaluation limit not reached and
        estimated error too large)
    Retrieve region from priority queue
    Split region into subregions
    Evaluate new subregions and update results
    Insert new subregions into priority queue
```

Figure 2: Adaptive Integration Meta-Algorithm

3.1.3. [Parallel] Iterated Adaptive Integration. This is a category of Automatic Adaptive Integration (cf., Section 3.1.2), applied in consecutive coordinate directions. Integration over a finite d-dimensional product region can be written as

$$\mathcal{I} = \int_{\alpha_1}^{\beta_1} dx_1 \int_{\alpha_2}^{\beta_2} dx_2 \ldots \int_{\alpha_d}^{\beta_d} dx_d\, f\,(x_1, x_2, \ldots, x_d)$$

The limits of integration may in general be functions, $\alpha_j = \alpha_j(x_1, x_2, \ldots, x_{j-1})$ and $\beta_j = \beta_j(x_1, x_2, \ldots, x_{j-1})$. For relatively small d, we can integrate over the interval $[\alpha_j, \beta_j]$ with a 1D adaptive integration code.

Subsequently, we give results obtained with adaptive iterated or parallel adaptive iterated integration (cf., Section 3.1.3) by versions of the programs DQAGE and DQAGSE from QUADPACK [9, 10], where the local integration is performed with the (7, 15)- or the (10, 21)-points Gauss-Kronrod pairs. The inner integrals are independent and can be evaluated in parallel by multiple threads [12] (using, e.g., OpenMP [13]). Important properties of this parallelization include: (1) a large granularity of the parallel integration, involving the inner integrations; (2) possibly apart from numerical implications resulting from the order of the summation in the local rule evaluation, the parallel calculation is the same as the sequential evaluation.

3.2. Double Extrapolation

Linear extrapolation is based on an asymptotic expansion of the form

$$\mathcal{I}(\varepsilon) \sim \sum_{k \geq \kappa} C_k \, \varphi_k(\varepsilon), \qquad \text{as } \varepsilon \to 0 \tag{3}$$

where the sequence of $\varphi_k(\varepsilon)$ is known. For the diagrams of Fig. 1, $\varphi_k(\varepsilon) = \varepsilon^k$. The expansion is truncated after $2, 3, \ldots, n$ terms to form linear systems of increasing size in the C_k variables. This is a generalized form of Richardson extrapolation [14, 15].

For fixed $\varepsilon = \varepsilon_\ell$, the integral (1) may have singularities as $\varrho \to 0$. Since an asymptotic expansion in ϱ is unknown, we apply a nonlinear extrapolation with the ϵ-algorithm [16] to a sequence of $\mathcal{I}(\varepsilon_\ell, \varrho)$ as $\varrho \to 0$. The combined ε and ϱ extrapolations constitute a double extrapolation [17, 18, 19, 20].

The nonlinear extrapolation results given in this paper are achieved with an implementation of the ϵ-algorithm from QUADPACK [9].

4. Computational Results

4.1. Asymptotic coefficients for Lemon-1, 2

The computations were performed under GNU/Linux on a multicore x86_64 machine with Intel(R) Xeon(R) Gold (dual) 6230 CPU@2.10 GHz.

We report sample results, for the diagrams of Fig. 1 with masses as specified in Section 2.3 using iterated adaptive or parallel iterated adaptive integration, encompassing the methods of Sections 3.1.1 - 3.1.3 and double extrapolation (Section 3.2). Fig. 3 (a) plots the asymptotic expansion coefficients C_{-2}, C_{-1}, C_0 as a function of s for the Lemon-1 $(1, 1, 1, 1)$ integral. A pronounced 2-particle threshold is shown at $s = (1 + 1)^2 = 4$. The 3-particle threshold at $s = (1 + 1 + 1)^2 = 9$ is not pronounced.

The computation involves several approximation procedures: the numerical integrations, the nonlinear extrapolation in ϱ for each value of ε, and the linear extrapolations in ε. A fixed number of extrapolations is performed in ϱ. For the linear extrapolation, a sequence of linear systems is solved and error estimates are produced based on the convergence of the sequence, which allows selecting the final result and giving a range for the error.

We found good agreement with results from pySecDec 1.4.5 [3], as the graphs appear indistinguishable (except for $s = 4$, where pySecDec returns 0). Furthermore for $s = 1$, we compared our results with the known values from [22] and confirmed their agreement. While a single extrapolation suffices for $s < 4$, a double extrapolation is necessary when imaginary parts enter at the 2-particle threshold.

Similar results are displayed in Fig. 3 (b) for Lemon-2 with the masses given in Section 2.3. A pronounced 2-particle threshold is revealed at $s = (2 \times 0.64308)^2 = 1.6542075456$. No 3-particle threshold is shown at $s = (2 \times 0.64308 + 0.7295008)^2 = 4.06288846065664$.

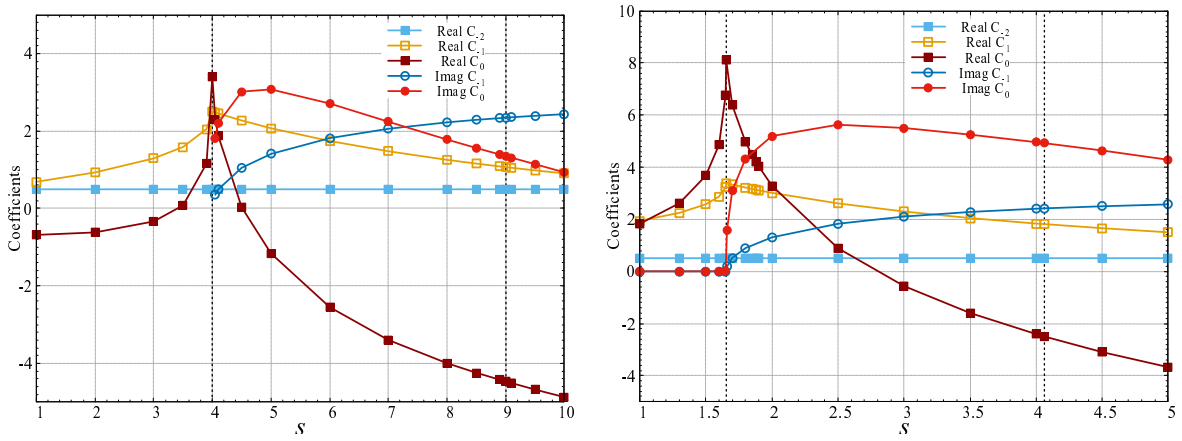


Figure 3: (a) Lemon-1 $(1,1,1,1)$ expansion coefficients C_{-2}, C_{-1}, C_0 as a function of s, results are multiplied with $\frac{1}{\Gamma(1+\varepsilon)^2}$; (b) Lemon-2 (M_Z, M_W, M_W, M_χ) expansion coefficients C_{-2}, C_{-1}, C_0 as a function of s, $M_W = 0.64308$, $M_Z = 0.7295008$, $M_\chi = M_W$ in the unit of the Higgs mass. In the computations for (a) and (b), DQAGSE is used for the integration.

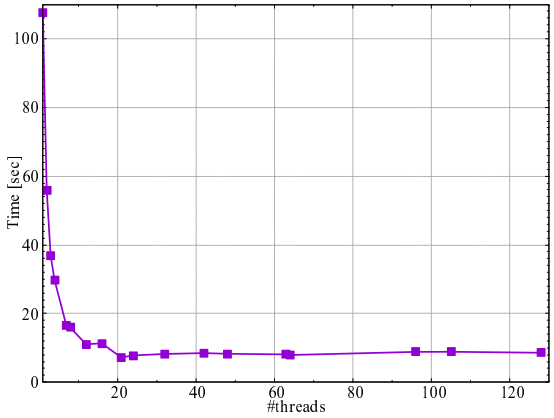


Figure 4: Magdeburg (1,1,1,1,1) parallel times [sec] of DQAGE as a function of #threads, with DQAGE keys 2, 2, 2, 2 (Gauss-Kronrod rule pair with 10-21 points for the local integrations), $s = 1$, $\varepsilon = 0$, extrapolation for $\varrho = 1.2^{-17}, 1.2^{-18}, \ldots, 1.2^{-31}$.

For the Lemon $(1,1,1,1)$ diagram as an example, the real parts of C_{-2}, C_{-1}, and C_0 incur the largest relative errors near the thresholds of $s = 4$ and 9. Apart from these, the ranges of accuracy are generally about 9 to 11 digits for C_{-2}, 6 to 9 digits for C_{-1}, and about 4 to 7 digits for C_0. For the imaginary parts, in the range from $s = 4.05$ to 10, the largest relative errors are also near the thresholds (although less pronounced than for the real parts). Excluding around the thresholds, in the range from $s = 4.5$ to 10, the accuracy is about 8 to 10 digits for C_{-1}, and about 6 to 9 digits for C_0.

In the case of Magdeburg, we have $\kappa = 0$ in the asymptotic series, and results for the coefficient C_0, for mass assignments $(1,1,1,1,M_x)$ with $M_x = 0, 1, 2$ were given in [19].

The execution time is dominated by the integration times, which are in turn determined by parameters such as the total number of integration points and/or subdivisions allowed, and the target accuracy. Below we discuss timings (from sequential runs) for the Lemon-1 real parts. For the integrations, the absolute error tolerance is set as $t_a = 0$; the relative tolerance is $t_r = 10^{-12}$ for the outer

integration, and 5×10^{-13} for the inner integrations. Using a Bulirsch type sequence [21] for ε such as $(1/4, 1/6, 1/8, 1/12, 1/16, 1/24, \ldots, 1/384, 1/512)$ and the geometric sequence $2^{-1}, \ldots, 2^{-13}$ for ρ, we report the maximum times over the ranges of ε and ρ for $s = 1$, and the minimum and maximum times for $s = 8$ (even though the sequence for ε is only used up to convergence of the extrapolation). With $s = 1$, the maximum times occur at various values of ρ; with $s = 8$, they almost always occur at the smallest values of ρ (where they are also the highest). For $s = 1$, the maximum times vary between 1.2 and 4.4 seconds, whereas for $s = 8$, they vary between 35.3 and 172.8 seconds. For $s = 8$, the minimum times vary between 2 and 4.4 seconds.

4.2. Parallelization Results for Magdeburg

Parallel timing results (in seconds) of the DQAGE program using OpenMP are given for Magdeburg as a function of the #threads. The meta-algorithm of Fig. 2 is applied in consecutive coordinate directions, and the outer direction is parallelized in this case. DQAGE incorporates several Gauss-Kronrod rule pairs for its local integral approximation over an interval $[a, b]$. For this instance we used the Gauss 10-point - Kronrod 21-point integration formulas (specified by input parameter *key* = 2 for DQAGE). The absolute and relative error tolerances were set to $t_a = 0$ and $t_r = 10^{-13}$, respectively. With a limit of 30 on the number of subdivisions allowed, no error flags were given, corresponding to reporting a regular termination of the algorithm for the target accuracy in each coordinate direction.

Furthermore, for $s = 1$ and $\varepsilon = 0$, a single nonlinear extrapolation was performed with $\varrho = 1.2^{-17}, 1.2^{-18}, \ldots, 1.2^{-31}$. Fig. 4 shows a significant decrease in time from 1 to about 20 threads, after which it flattens out (without, however, incurring a considerable increase due to parallel overhead).

5. Conclusions

Whereas symbolic or symbolic/numerical calculations are performed for some challenging problems using existing software packages, we focus on the development of fully numerical methods for the evaluation of Feynman loop integrals. The integration strategies adhere to automatic adaptive integration, which is a black-box approach for generating an approximation, assuming little or no knowledge of the problem, apart from the specification of the integrand function. We demonstrated efficient strategies based on iterated adaptive integration, multithreading with OpenMP, and double extrapolation.

Acknowledgments

We acknowledge the support by JSPS KAKENHI Grant Number JP20K11858 and JP20K03941, and the National Science Foundation Award Number 1126438 that funded work on multivariate integration.

References

[1] Freitas A 2016 *Progress in Particle and Nuclear Physics* **90** 201–240 https://doi.org/10.1016/j.ppnp.2016.06.004

[2] Heinrich G, Jones S P, , Kerner M, Magerya V, Olsson A and Schlenk J 2024 *Computer Physics Communications* **295** https://doi.org/10.1016/j.cpc.2023.108956; arXiv:2305.19768v2 [hep-ph]

[3] Borowka S, Heinrich G, Jahn S, Jones S P, Kerner M and Schlenk J 2019 *Computer Physics Communications* **240** 120–137 DOI:10.1016/j.cpc.2019.02.015, arXiv:1811.11720v1 [hep-ph], https://arxiv.org/abs/1811.11720

[4] Borowka S, Heinrich G, Jahn S, Jones S P, Kerner M, Schlenk J and Zirke T 2018 *Computer Physics Communications* **222** 313–326 https://doi.org/10.1016/j.cpc.2017.09.015

[5] Borinsky M, Munch H J and Tellander F 2023 *Computer Physics Communications* **292** 108874, arXiv:2302.08955v2 [hep-ph], https://doi.org/10.1016/j.cpc.2023.108874

[6] Smirnov A V, Shapurov N D and Vysotsky L I 2021 *Computer Physics Communications* **277** 108386, https://doi.org/10.1016/j.cpc.2022.108386, arXiv:2110.11660

[7] Usovitsch J, Dubovyk I and Riemann T 2018 *PoS LL2018 046* 1810.04580 [hep-ph], DOI: 10.22323/1.303.0046

[8] Kato K March 2022 Note on 2-loop self-energy scalar integrals Department of Physics, Kogakuin University Shinjuku, Japan, Private communication

[9] 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)

[10] 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
[11] de Doncker E, Yuasa F, Kato K, Ishikawa T, Kapenga J and Olagbemi O 2018 *Computer Physics Communications* **224** 164–185 https://doi.org/10.1016/j.cpc.2017.11.001
[12] de Doncker E, Yuasa F and Assaf R 2013 *Journal of Physics: Conf. Ser.* **454** DOI:10.1088/1742-6596/454/1/012082
[13] OpenMP http://www.openmp.org
[14] Brezinski C 1980 *Numerische Mathematik* **35** 175–187
[15] Sidi A 2003 *Practical Extrapolation Methods - Theory and Applications* (Cambridge Univ. Press) iSBN 0-521-66159-5
[16] Wynn P 1956 *Mathematical Tables and Aids to Computing* **10** 91–96
[17] 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
[18] de Doncker E, Yuasa F and Kurihara Y 2012 *Journal of Physics: Conf. Ser.* **368**
[19] Yuasa F, de Doncker E, Ishikawa T, Kato K, Daisaka H and Nakasato N September 2022 Numerical method for Feynman integrals: Electroweak high-order correction calculation by DCM IV Fall Meeting of the Physical Society of Japan, 8aA133-6
[20] Kato K April 2022 Note on the double-extrapolation method for the evaluation of Feynman integrals Department of Physics, Kogakuin University Shinjuku, Japan, Private communication
[21] Bulirsch R 1964 *Numerische Mathematik* **6** 6–16
[22] Laporta S 2000 *Int. J. Mod. Phys. A* **15** 5087–5159 arXiv:hep-ph/0102033v1