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]Dept. of Computer Science, W. Michigan Univ, Kalamazoo MI 49008, U.S.A.

[2]High Energy Accelerator Research Organization (KEK), 1-1 OHO Tsukuba,
Ibaraki 305-0801, Japan

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

October 25, 2022

Outline

1. Introduction - Overview
2. Loop Integrals
3. Methods
   - Methods: Iterated Automatic Adaptive Integration
   - Methods: Double Extrapolation
4. Results
5. Conclusions

Outline

1. Introduction - Overview
2. Loop Integrals
3. Methods
   - Methods: Iterated Automatic Adaptive Integration
   - Methods: Double Extrapolation
4. Results
5. Conclusions

Outline

1. Introduction - Overview
2. Loop Integrals
3. Methods
   - Methods: Iterated Automatic Adaptive Integration
   - Methods: Double Extrapolation
4. Results
5. Conclusions

Outline

1. Introduction - Overview
2. Loop Integrals
3. Methods
   - Methods: Iterated Automatic Adaptive Integration
   - Methods: Double Extrapolation
4. Results
5. Conclusions

Outline

1. Introduction - Overview
2. Loop Integrals
3. Methods
   - Methods: Iterated Automatic Adaptive Integration
   - Methods: Double Extrapolation
4. Results
5. Conclusions

Introduction - Overview

– 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).

– 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.

Introduction - Overview

– To handle a singularity in the domain interior, we add a term ($-i\varrho$) in the denominator, and perform a nonlinear extrapolation (as $\varrho \to 0$) to a sequence of integrals obtained for a (geometrically) decreasing sequence of ϱ.

– UV singularities are treated by 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 $\varepsilon \to 0$.

– Presence of both types of singularities warrants a double extrapolation.

– The code is further multi-threaded to run in a shared memory environment. We will demonstrate the combined methods for sample diagrams.

Loop integral - Representation

L-loop integral with *N* internal lines

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

$$= \Gamma(N - \frac{\nu L}{2})\,(-1)^N \int_{\mathcal{S}_{N-1}} \prod_{r=1}^{N-1} dx_r\, U^{-\nu/2}(V - i\varrho)^{\nu L/2-N}$$

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;

$\mathcal{C}_N =$ the N-dimensional unit hypercube; $\mathcal{S}_d = \{\mathbf{x} \in \mathcal{C}_d \mid \sum_{j=1}^{d} x_j \leq 1\}$ is the d-dimensional unit simplex.

Asymptotics

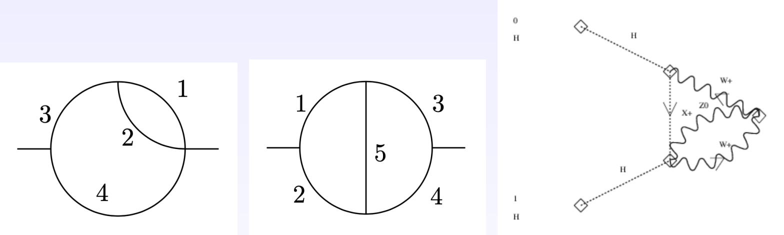
Asymptotic expansion (as $\varepsilon \to 0$)

$$\mathcal{I} = \mathcal{I}(\varepsilon) \sim \sum_{k \geq \kappa} C_k \, \varepsilon^k$$

$\kappa = -2$ for the cases considered.

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

Sample diagrams



(a) (b) (c)

Figure: Sample diagrams (a) Lemon, $N = 4$; (b) Magdeburg, $N = 5$; (c) Graph 2116, $N = 4$ [7]

Specifications [9]

Lemon, Graph 2116

$U = x_{12}x_{34} + x_1 x_2, \quad W = s\,(x_4(x_1 x_2 + x_1 x_3 + x_2 x_3))$
$x_{k\ell\ldots n} = x_k + x_\ell + \ldots + x_n$
Cases:
$m_1 = m_2 = m_3 = m_4 = 1$ (lemon1111)
Graph 2116 (Z, W, W, χ): In units of $M_H = 1$, $M_W = 0.64308$ and $M_Z = 0.7295008$, $M_\chi = M_W$.

Magdeburg

$U = x_{12}x_{34} + x_{1234}x_5, \quad W = s\,(x_1 x_2 x_{34} + x_3 x_4 x_{12} + x_{13}x_{24}x_5)$

Case: $m_1 = m_2 = m_3 = m_4 = m_5 = 1$

Methods: Automatic Integration

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.

Methods: Automatic Adaptive Integration

Applied in consecutive coordinate directions (using Quadpack DQAGE, and DQAGSE (DQAGE strategy + extrapolation) [11, 3, 5]:

```
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: Adaptive Integration Meta-Algorithm

Methods: Iterated Parallel Adaptive Integration

– Integration over a finite d-dimensional product region,

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

– The limits of integration may in general be functions, $\alpha_j = \alpha_j(x_1, x_2, \ldots, x_{j-1})$ and $\beta_j = \beta_j(x_1, x_2, \ldots, x_{j-1})$.
– We can integrate over the interval $[\alpha_j, \beta_j]$ with a 1D adaptive integration code.
– If an interval $[a, b]$ arises in the subdivision of $[\alpha_j, \beta_j]$ for $1 \le j < d$, then the local integral approximation over $[a, b]$ is of the form

$$\int_a^b dx_j \; F(c_1, \ldots, c_{j-1}, x_j) \;\approx\; \sum_{k=1}^{K} w_k F(c_1, \ldots, c_{j-1}, x^{(k)}),$$

where w_k and $x^{(k)}, 1 \le k \le K$, are the weights and abscissae of the local rule scaled to the interval $[a, b]$ and applied in the x_j-direction.

Methods: Iterated Parallel Adaptive Integration

– The function evaluation

$$F(c_1, \ldots, c_{j-1}, x^{(k)}) = \int_{\alpha_{j+1}}^{\beta_{j+1}} dx_{j+1} \ldots \int_{\alpha_d}^{\beta_d} dx_d \; f(c_1, \ldots, c_{j-1}, x^{(k)}, x_{j+1}, \ldots, x_d),$$

is itself an integral in the $x_{j+1}, \ldots, x_d$-directions, and is computed by the method(s) for the inner integrations.
– For $j = d$, this is the evaluation of the integrand function

$$F(c_1, \ldots, c_{d-1}, x^{(k)}) = f(c_1, \ldots, c_{d-1}, x^{(k)}).$$

– Subsequently, we give results obtained with 1D iterated integration by the programs DQAGE and DQAGSE from QUADPACK [11, 3], where the local integration is performed with the (7, 15)- or the (10, 21)-points Gauss-Kronrod pairs.

Methods: Iterated Parallel Adaptive Integration

– The inner integrals are independent and can thus be evaluated in parallel, by multiple threads [4] (using, e.g., OpenMP [10]).

– Important properties of this parallelization include:
(1) large granularity of the parallel integration, involving the inner integrations;

(2) apart from possibly the order of the summation in the local rule evaluation, the parallel calculation is the same as the sequential evaluation.

Methods: Double Extrapolation

– Linear extrapolation is based on an asymptotic expansion of the form

$$\mathcal{I}(\varepsilon) \sim \sum_{k \geq \kappa} C_k \, \varphi_k(\varepsilon), \qquad \text{as} \;\; \varepsilon \to 0 \tag{1}$$

where the sequence of $\varphi_k(\varepsilon)$ is known.

– Here, $\kappa = -2$ and $\varphi_k(\varepsilon) = \varepsilon^k$.

– 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 [1, 13].

Methods: Double Extrapolation

– For fixed $\varepsilon = \varepsilon_\ell$, the integral

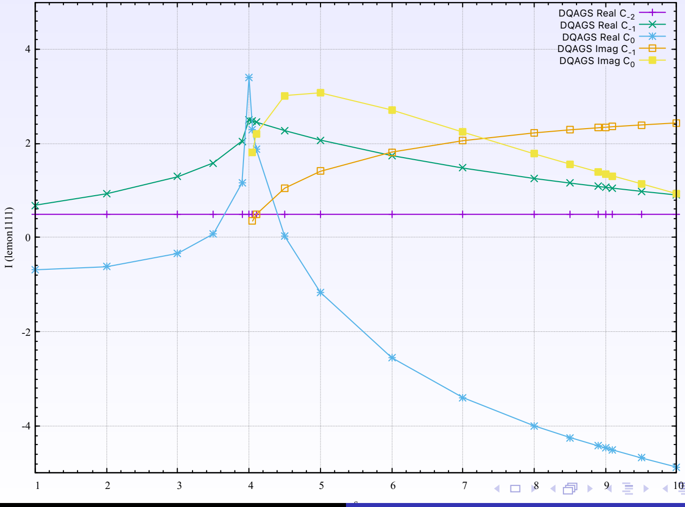
$$\mathcal{I} = \Gamma(N - \frac{\nu L}{2})\,(-1)^N \int_{\mathcal{S}_{N-1}} \prod_{r=1}^{N-1} dx_r\, U^{-\nu/2}(V - i\varrho)^{\nu L/2-N}$$

may have singularities as $\varrho \to 0$.

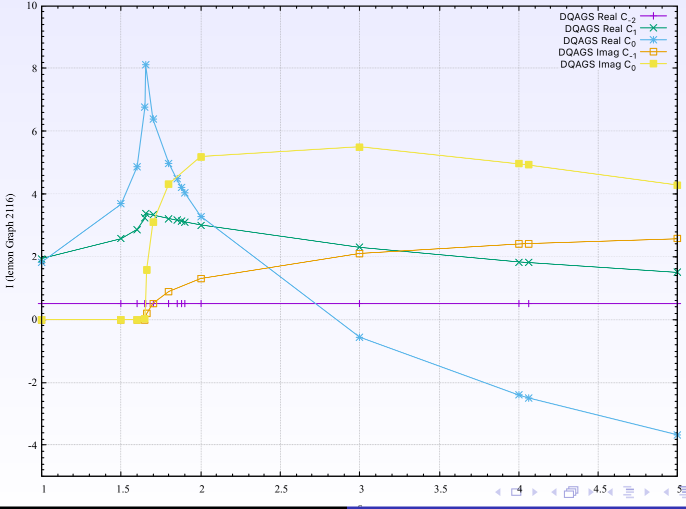
– Since the structure of the expansion in ϱ is unknown, we apply a non-linear extrapolation with the ϵ-algorithm [12, 14] to a sequence of $\mathcal{I}(\varepsilon_\ell, \varrho)$ as $\varrho \to 0$.

– The combined ε and ϱ extrapolations constitute a double extrapolation [2, 6, 15, 8].

Results lemon1111: C_{-2}, C_{-1}, C_0 as a function of s



Results Graph 2116: C_{-2}, C_{-1}, C_0 as a function of s



Results notes

– lemon1111: Pronounced threshold at $s = (1+1)^2 = 4$.
– Threshold at $s = (1+1+1)^2 = 9$ is not pronounced.
– Good comparison with pySecDec results, except for $s = 4$ (where pySecDec returns 0).

– Graph 2116: Pronounced threshold at $s = (2 \times 0.64308)^2 = 1.6542075456$
– Threshold at $s = (2 * 0.64308 + 0.7295008)^2 = 4.06288846065664$ is not pronounced.
– Results were obtained with the particle masses and s in the unit of $M_H^2 = 1$.

– Results for Magdeburg (coefficients) for mass assignments $1111x$ with $x = 0, 1, 2$ were given in [15].

Timing Results DQAGE for Magdeburg as a function of #threads

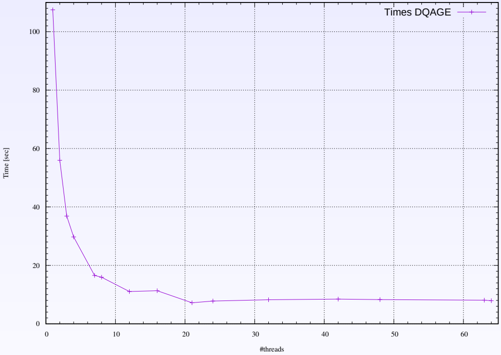


Figure: Magdeburg11111: Parallel times [sec] of DQAGE as a function of #threads, with DQAGE keys 2, 2, 2, 2 (Gauss-Kronrod rule pair with 10-21 points for local integrations), $m_1 = m_2 = m_3 = m_4 = m_5$, $s = 1$, Extrapolation for $\varrho = 2^{-17}, 2^{-18}, \ldots, 2^{-31}$

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 integration, multithreading with OpenMP, and double extrapolation.

BIBLIOGRAPHY

[1] C. Brezinski.
A general extrapolation algorithm.
Numerische Mathematik, 35:175–187, 1980.

[2] E. de Doncker, J. Fujimoto, N. Hamaguchi, T. Ishikawa, Y. Kurihara, M. Ljucovic, Y. Shimizu, and F. Yuasa.
Extrapolation algorithms for infrared divergent integrals, 2010.
arXiv:hep-ph/1110.3587; PoS (CPP2010)011.

[3] E. de Doncker, J. Fujimoto, N. Hamaguchi, T. Ishikawa, Y. Kurihara, Y. Shimizu, and F. Yuasa.
Quadpack computation of Feynman loop integrals.
Journal of Computational Science (JoCS), 3(3):102–112, 2011.
doi:10.1016/j.jocs.2011.06.003.

[4] E. de Doncker, F. Yuasa, and R. Assaf.
Multi-threaded adaptive extrapolation procedure for Feynman loop integrals in the physical region.
Journal of Physics: Conf. Ser., 454(012082), 2013.
doi:10.1088/1742-6596/454/1/012082.

[5] E. de Doncker, F. Yuasa, K. Kato, T. Ishikawa, J. Kapenga, and O. Olagbemi.
Regularization with numerical extrapolation for finite and UV-divergent multi-loop integrals.
Computer Physics Communications, 224:164–185, 2018.
https://doi.org/10.1016/j.cpc.2017.11.001.

[6] E. de Doncker, F. Yuasa, and Y. Kurihara.
Regularization of IR-divergent loop integrals.
Journal of Physics: Conf. Ser., 368(012060), 2012.

[7] feynman integral.org.

[8] K. Kato.
Note on the double-extrapolation method for the evaluation of Feynman integrals, April 2022.
Department of Physics, Kogakuin University Shinjuku, Japan.

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

[10] OpenMP.
http://www.openmp.org.

[11] R. Piessens, E. de Doncker, C. W. Überhuber, and D. K. Kahaner.
QUADPACK, A Subroutine Package for Automatic Integration, volume 1 of *Springer Series in Computational Mathematics*.
Springer-Verlag, 1983.

[12] D. Shanks.
Non-linear transformations of divergent and slowly convergent sequences.
J. Math. and Phys., 34:1–42, 1955.

[13] A. Sidi.
Practical Extrapolation Methods - Theory and Applications.
Cambridge Univ. Press, 2003.
ISBN 0-521-66159-5.

[14] P. Wynn.
On a device for computing the $e_m(s_n)$ transformation.
Mathematical Tables and Aids to Computing, 10:91–96, 1956.

[15] F. Yuasa, E. de Doncker, T. Ishikawa, K. Kato, H. Daisaka, and N. Nakasato.
Numerical method for Feynman integrals: Electroweak high-order correction calculation by DCM IV, September 2022.
Fall Meeting of the Physical Society of Japan, 8aA133-6.