PROCEEDINGS OF SCIENCE

High-Accurate Computation of One-Loop Integrals by Several Hundred Digits Multiple-Precision Arithmetic

Hiroshi Fujiwara[*]
Graduate School of Informatics, Kyoto University
E-mail: `fujiwara@acs.i.kyoto-u.ac.jp`

We present high-accurate numerical results for one-loop integrals by multiple-precision arithmetic, which realizes reliable numerical computation for numerically unstable problems. The quantitative analysis of numerical singularities and the convergence property of one-loop integrals by multiple-precision arithmetic are also shown.

3rd Computational Particle Physics Workshop- CPP2010,
September 23-25, 2010
Kek Japan

[*] Speaker.

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike Licence.

PoS(CPP2010)013

1. Introduction

The purpose of the present paper is to show an effective use of multiple-precision arithmetic in evaluation of one-loop integrals appearing in the higher order correction of the cross section for collisions of elementary particles. Since exact treatments of loop integrals are hard in most cases, development of numerical treatments is important [3].

A one-loop integral is formally defined as the limit of the regularization for a singular integral. In [1], Doncker et al. have introduced an extrapolation method to evaluate the limit. However inappropriate choice of parameters results in wrong evaluation.

Our approach is direct computation of integrals involving sufficiently small regularization parameters by fast multiple-precision arithmetic. In the next section we give a remark on rounding errors in floating-point arithmetic, and we introduce a fast multiple-precision arithmetic in §3. In §4, we describe the one-loop box type integral on infrared divergent physics process. Numerical results with the proposed multiple-precision arithmetic are shown in §5, and its imaginary parts are also discussed in §6.

PoS(CPP2010)013

2. Floating-Point Arithmetic and Rounding Errors

In digital arithmetic, a real number is treated in a fixed-point format or a floating-point format, and the latter is mainly used in scientific computations. A floating-point format consists of three components: a sign part, an exponent part, and a fractional part. Two kinds of floating-point arithmetic are defined as the standard in IEEE754. One is single precision, and the other is double precision. The latter is used in the most scientific computations, and it has approximately 16.0 decimal digits precision.

Since the real values are approximated with floating-point values in numerical computation, we cannot avoid rounding errors. Moreover rounding errors prevent us to treat mathematical theory exactly on digital computers. For instance, we consider the inequality $a^2+b^2 \geq 2ab$ with $a = 1.022$ and $b = 1.038$ by four decimal digits arithmetic. In the example the exact value of a^2 is 1.044484, which is rounded to 1.044. We denote this $a \otimes a = 1.044$ where $\otimes$ means multiplication with rounding. Similarly $b^2 = 1.077444$. Thus $b \otimes b = 1.077$. Hence we get $a \otimes a \oplus b \otimes b = 2.121$. On the other hand, the right hand side of the inequality $2ab = 2.121672$ is rounded to 2.122. Therefore $a \otimes a \oplus b \otimes b$ is smaller than $2 \otimes a \otimes b$ in the precision. Moreover floating-point arithmetic for $1/(xy+zw-2st)$ with $x = z = s, y = w = t$ probably includes large rounding errors with odd sign.

3. Multiple-Precision Arithmetic Environment “exflib”

We introduce a multiple-precision arithmetic environment “exflib” (extended precision floating-point arithmetic library) [4]. It enables us to allocate arbitrary precision to the fractional part of a floating-point number. Thus we can make rounding errors arbitrary small with enough precision and virtually achieve numerical computation without rounding errors. In particular we can perform direct numerical computation for numerically unstable problems.

The proposed environment is designed for scientific computation with one hundred up to several thousand decimal digits, and its advantages lie in fast computation, small memory requirement,

PoS(CPP2010)013

and seamless interface. For fast computation, four basic rules are implemented in assembly languages. It is equipped with polymorphic interface of four basic rules and built-in functions in the programming language C++ and FORTRAN90.

It is implemented on AMD64, EM64T, SPARC V9, and IA32 architectures. It is executable on UNIX-like operating system (Solaris, Linux, FreeBSD, Mac OSX) and Windows operating system with commercial and noncommercial C++ and FORTRAN90 compilers. The proposed environment works with MPI and OpenMP for parallel computations.

4. One-Loop Box Type Integral on Infrared Dirergent Process

We apply multiple-precision arithmetic to numerical evaluation of the one-loop box type loop integral on infrared divergent physics process [2], that turns to $\lim_{\varepsilon\downarrow 0} I(\varepsilon)$, where

$$I(\varepsilon) = \int_0^1 dx \int_0^{1-x} dy \int_0^{1-x-y} dz \frac{D(x,y,z)^2 - \varepsilon^2}{(D(x,y,z)^2 + \varepsilon^2)^2},$$

and

$$\begin{aligned} D(x,y,z) = & -xys - tz(1-x-y-z) + (x+y)\lambda^2 \\ & + (1-x-y-z)(1-x-y)m_e^2 + z(1-x-y)m_f^2. \end{aligned}$$

Here λ is fictious photon mass, m_e is electron mass, m_f is fermion mass, s and t are Mandelstam variables. The square root of s is center-of-mass energy. We suppose that t is negative. In the following numerical computations, values shown in Table 1 are used.

Table 1: Parameters in the integrand [5]

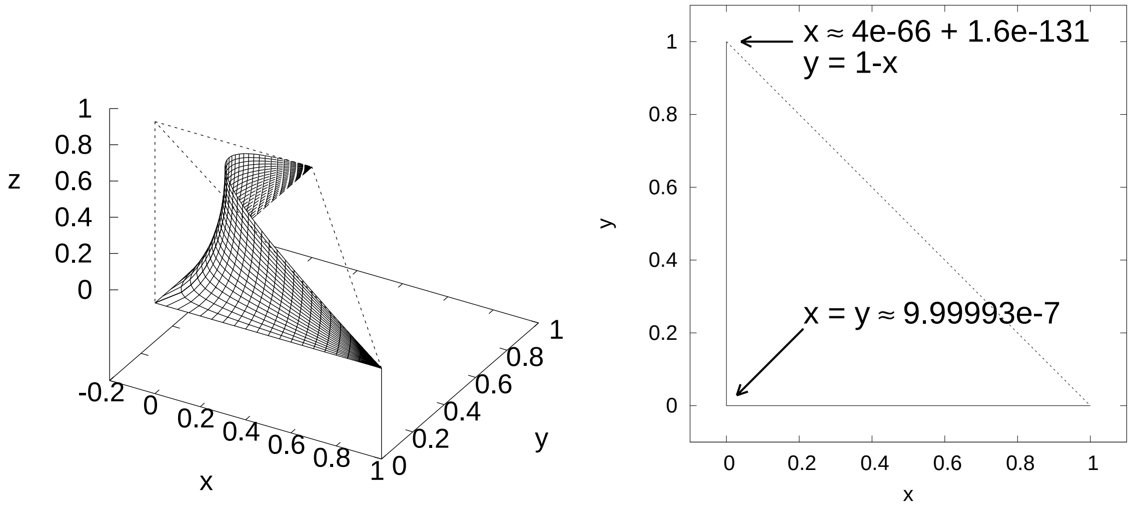
λ	10^{-30}	[GeV]	m_e	0.0005	[GeV]	s	500^2	[GeV^2]
			m_f	150	[GeV]	t	-150^2	[GeV^2]

5. Numerical Evaluation of One-Loop Integral

In this section numerical analysis of $\lim_{\varepsilon\downarrow 0} I(\varepsilon)$ by multiple-precision arithmetic are shown. The proposed method consists of four techniques: preventing cancellations in numerical integrations, decomposition of an integral domain at numerical singularities, the use of a high-accurate quadrature rule and the use of multiple-precision arithmetic.

The function $D(x,y,z)$ has zero in the integral domain, which is a smooth surface shown in Figure 1. Great attention must be paid to change of the sign of the the numerator $D(x,y,z)^2 - \varepsilon^2$ for high accurate integration. Actually some values of the most inner integral $I^{(0)}$ of $I(\varepsilon)$ are shown in Table 2. Such oscillation causes cancellation in floating-point arithmetic and inaccuracy for numerical integration. To prevent cancellation we decompose it into two positive parts as

$$I^{(0)}(x,y;\varepsilon) = \int_0^{1-x-y} \frac{D^2-\varepsilon^2}{(D^2+\varepsilon^2)^2} dz = \int_0^{1-x-y} \frac{dz}{D^2+\varepsilon^2} dz - \int_0^{1-x-y} \frac{2\varepsilon^2}{(D^2+\varepsilon^2)^2} dz, \tag{5.1}$$



(a) Surface $D(x,y,z) = 0$.

(b) Slice at $z = 0$.

Figure 1: The surface $D(x,y,z) = 0$ in the integral domain. Dotted lines show the integral domain.

and we let the integrals on the right hand side as $I^{(1)}(x,y;\varepsilon)$ and $I^{(2)}(x,y;\varepsilon)$ respectively.

Figure 2(a) shows the profile of $I^{(1)}(x,y;\varepsilon)$ on $\{(x,y)\,;\,0 < x,y < 1, x+y < 1\}$, and Figure 2(b) shows its slice on $x = 0.5$. From the Figure 2(b), it has some singularities from numerical computation view points though $I^{(1)}$ is mathematically analytic. We call it numerical singularity in the present study. Thus finding numerical singularities and decomposing the integral domain is quite effective for numerical quadratures. For instance, $I^{(1)}(x,y;\varepsilon)$ at $x = 0.5$ has following numerical singularities shown in Figure 3:

$$y_1 \approx 3.836005841927751618102719437216756398164 \times 10^{-2} \approx \mathop{\arg\max}_{0<y<0.5} I^{(1)}(0.5, y; 10^{-60}),$$

$$y_2 \approx 4.999999999990000000000024999999999930000 \times 10^{-13},$$

$$y_3 \approx 3.836005841927751618102617914134273526589 \times 10^{-2}.$$

Moreover Figure 4 shows the profile of $\int_0^{1-x} I^{(1)}(x,y;\varepsilon)\,dy$. From the figure we decompose the domain at

$$x_1 \approx 4 \times 10^{-66} + 1.6 \times 10^{-125}.$$

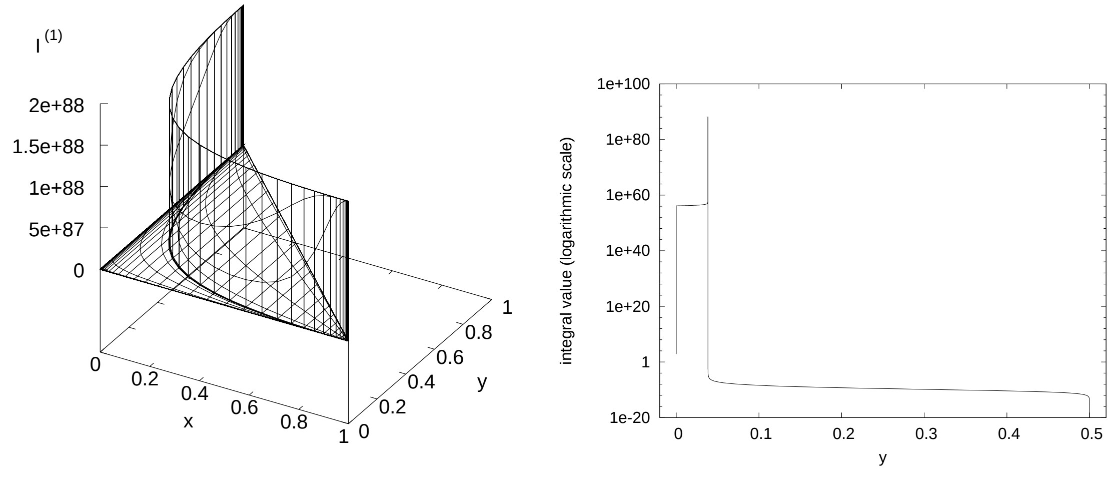
The numerical results for $I^{(1)}(\varepsilon)$ and $I(\varepsilon)$ are shown in Table 3 and in Table 4 respectively. We use the adoptive double exponential rule. In Table 4, N_0 and N_1 are discretization numbers for intervals $[0, x_1]$ and $[x_1, 1]$. Computational time with exflib is also shown in Table 4 on a PC cluster. Each node is equipped with four quad-core Opteron 8350 (2.3GHz). Parallel computation is done with 512 processes by MPI.

Figure 5 (a) and (b) show the convergence of $I(\varepsilon)$. They show the integral value $I(\varepsilon)$ monotonically decreases as ε tends to zero. From Table 4 and Figure 5, the one-loop box integral under

PoS(CPP2010)013

Table 2: $I^{(0)}(x,y;\varepsilon)$ with $x=0.5, \varepsilon=10^{-60}$, underlines indicate same or different digits

y	$4.99999999999900000000000249999999999930000000000\underline{09999959999340}\times 10^{-13}$
$I^{(0)}$	$+2.222222222236790123456868834019204831300106693496111703852787345\times 10^{55}$
y	$4.99999999999900000000000249999999999930000000000\underline{21000011999934}\times 10^{-13}$
$I^{(0)}$	$-2.222222222236790123456868834019204831300106693496111703852787230\times 10^{55}$
y	$\underline{0.03836005841927751618102}617914134273526588981395223780308332958 61\underline{813831}$
$I^{(0)}$	$+4.332380757938120188349661151018233230611306340936221713787407882\times 10^{66}$
y	$\underline{0.03836005841927751618102}61791413427352658898139522378030833295861\underline{951030}$
$I^{(0)}$	$-4.332380757938120188349661151018233230605812326972555849772597514\times 10^{66}$
y	$\underline{0.03836005841927751618102}71943721675639816420468162279467522727339\underline{786472}$
$I^{(0)}$	$-9.237286884991645593316210503216171926139508812544397833762826889\times 10^{87}$
y	$\underline{0.03836005841927751618102}71943721675639816420468162279467522727339\underline{903181}$
$I^{(0)}$	$+2.773801196845539109945635643808527035032151854445043279740096923\times 10^{87}$



(a) whole profile, linear scale

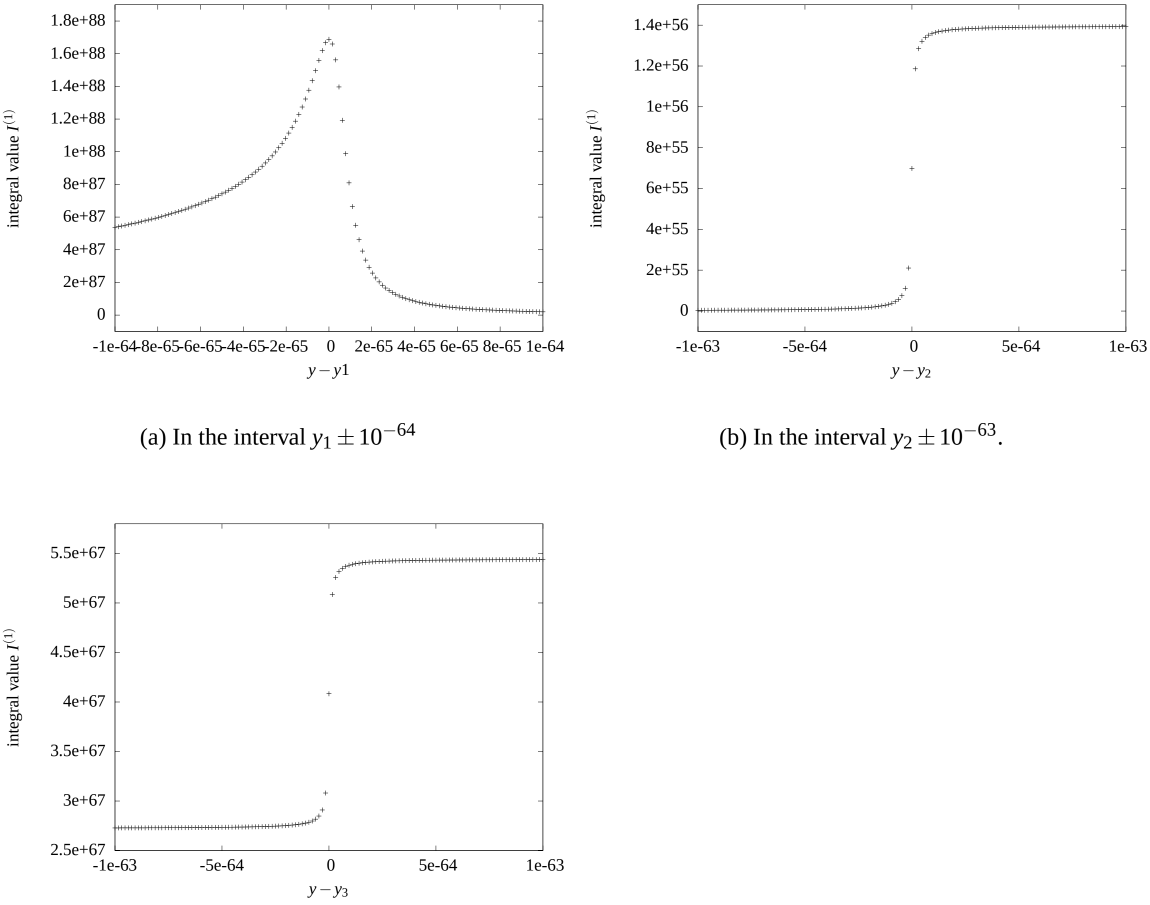
(b) slice at $x=0.5$, semilogarithmic scale

Figure 2: Profile of $I^{(1)}(x,y;\varepsilon)$ with $\varepsilon=10^{-60}$

parameters in Table 1 are concluded as

$$\lim_{\varepsilon\downarrow 0} I(\varepsilon)\approx -3.561736812918245378434750660221565736544\times 10^{-7}. \quad \text{(40 decimal digits)}$$

PoS(CPP2010)013

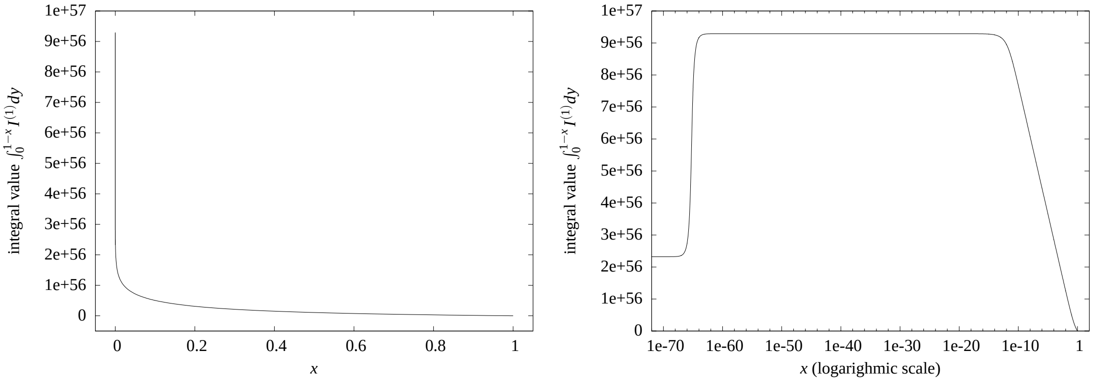


(a) In the interval $y_1 \pm 10^{-64}$

(b) In the interval $y_2 \pm 10^{-63}$.

(c) In the interval $y_3 \pm 10^{-63}$

Figure 3: Profile of $I^{(1)}(x,y;\varepsilon)$, $x = 0.5, \varepsilon = 10^{-60}$



(a) linear scale

(b) semilogarithmic scale

Figure 4: Profile of $\int_0^{1-x} I^{(1)}(x,y;\varepsilon)\,dy$ with $\varepsilon = 10^{-60}$

PoS(CPP2010)013

Table 3: Cancellation in numerical computation of (5.1)

ε	$\iiint \frac{dxdydz}{D^2+\varepsilon^2}$	$\iiint \frac{dxdydz}{D^2+\varepsilon^2} - \iiint \frac{2\varepsilon^2}{(D^2+\varepsilon^2)^2} dxdydz$
10^{-60}	$+2.00601 \times 10^{55}$	-3.55354×10^{-7}
10^{-65}	$+2.00601 \times 10^{60}$	-3.56173×10^{-7}
10^{-70}	$+2.00601 \times 10^{65}$	-3.56174×10^{-7}
10^{-75}	$+2.00601 \times 10^{70}$	-3.56174×10^{-7}
10^{-80}	$+2.00601 \times 10^{75}$	-3.56174×10^{-7}

Table 4: Numerical results of $I(\varepsilon)$ (digit: computation precision in decimal digits, min: computation time in minutes)

ε	digits	$I(\varepsilon)$	N_0	N_1	min.
10^{-40}	180	$-2.4724863525991758999 \times 10^{-7}$	32768	32768	52
10^{-50}	180	$-3.0171115827587106392 \times 10^{-7}$	32768	65536	115
10^{-50}	200	$-3.0171115827587106392 \times 10^{-7}$	32768	65536	142
10^{-60}	200	$-3.5535393863845744033 \times 10^{-7}$	16384	65536	118
10^{-65}	220	$-3.5617368129170627398 \times 10^{-7}$	16384	65536	135
10^{-70}	240	$-3.5617368129182453784 \times 10^{-7}$	16384	131072	547
10^{-80}	300	$-3.5617368129182453784347506602215657365443894 1\times 10^{-7}$			
10^{-85}	300	$-3.56173681291824537843475066022156573654439053 \times 10^{-7}$			

6. Imaginary Parts

The corresponding imaginary part of $I(\varepsilon)$ is

$$J_1(\varepsilon) = \int_0^1 dx \int_0^{1-x} dy \int_0^{1-x-y} dz \frac{2\varepsilon D(x,y,z)}{(D(x,y,z)^2+\varepsilon^2)^2}.$$

On the other hand, replacing s in $D(x,y,z)$ by $s+\sqrt{-1}\varepsilon$ gives another integral [5]

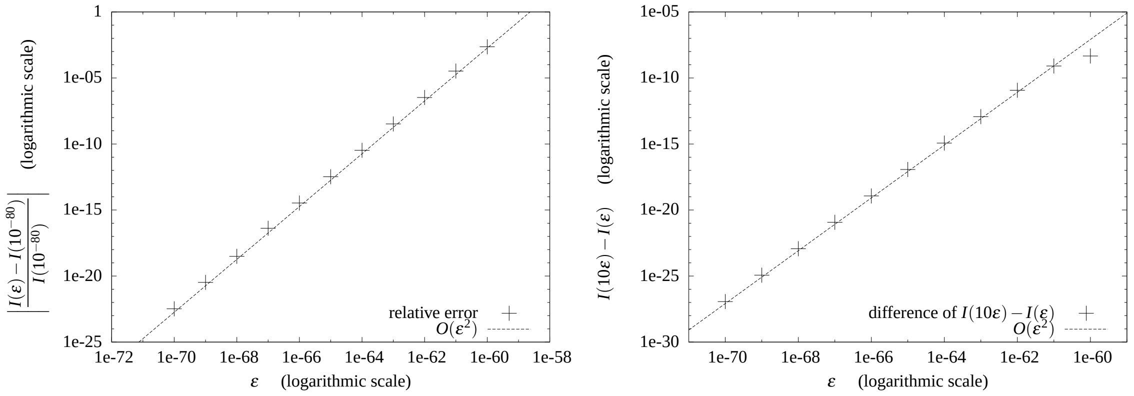
$$J_2(\varepsilon) = \int_0^1 dx \int_0^{1-x} dy \int_0^{1-x-y} dz \frac{-2xy\varepsilon D(x,y,z)}{(D(x,y,z)^2+x^2y^2\varepsilon^2)^2}.$$

Numerical evaluations of J_1 and J_2 by the proposed method with multiple-precision arithmetic exflib are shown in Table 5 and Table 6. From the results, limits of $J_1(\varepsilon)$ and $J_2(\varepsilon)$ under parameters in Table 1 are

$$\lim_{\varepsilon\downarrow 0} J_1(\varepsilon) \approx \lim_{\varepsilon\downarrow 0} J_2(\varepsilon) \approx 7.430737857353396533447836617291272198709 \times 10^{-9}.$$

(40 decimal digits)

PoS(CPP2010)013



(a) Relative Errors

(b) Difference of $I(10\varepsilon)-I(\varepsilon)$

Figure 5: Convergence of $I(\varepsilon)$

Table 5: Numerical Results of $J_1(\varepsilon)$ with 300 decimal digits precision

ε	$J_1(\varepsilon)$
10^{-25}	$3.715368928426927073870743435703398581626791 1049 \times 10^{-9}$
10^{-35}	$3.7153689286766957690120141886898227752366637941 \times 10^{-9}$
10^{-45}	$3.7153689286767006319762628253462875987835294078 \times 10^{-9}$
10^{-55}	$3.7153925814499138357402752142847182064980045570 \times 10^{-9}$
10^{-60}	$5.5730533930150474000858774622803884754816076552 \times 10^{-9}$
10^{-65}	$7.4307142045801809644314794618809912814913846718 \times 10^{-9}$
10^{-75}	$7.4307378573533941681705149815470591000403736745 \times 10^{-9}$
10^{-100}	$7.4307378573533965334478366172912721987088430722 \times 10^{-9}$
10^{-110}	$7.4307378573533965334478366172912721987090795999 \times 10^{-9}$

Our numerical results shows that the limits of $J_1(\varepsilon)$ and $J_2(\varepsilon)$ as ε tends to zero coinsides in 40 decimal digits accuracy.

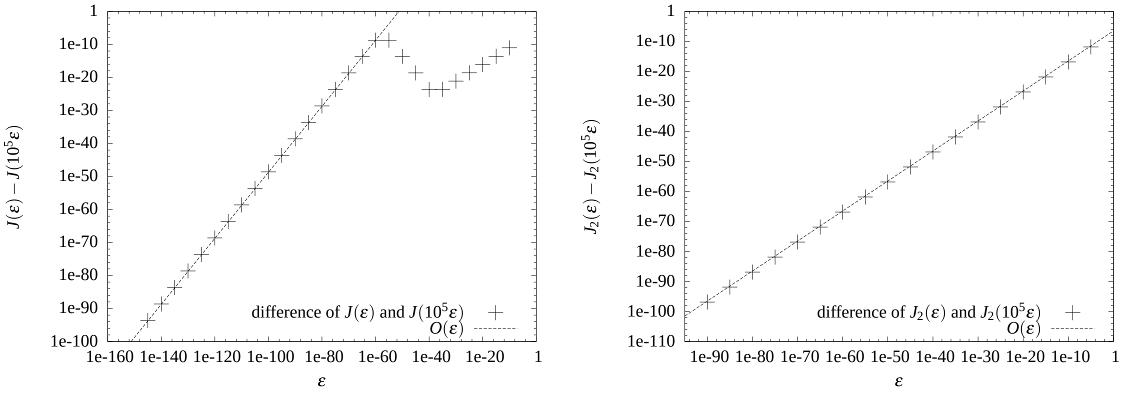
Acknowledgments

The author thanks Professor Fukuko Yuasa and Professor Tadashi Ishikawa (High Energy Accelerator Research Organisation) for their kindly advice. This works is supported in part by JSPS Grant-in-Aid for Young Scientists (B) No. 20740057 and Grant-in-Aid for Scientific Research (B) No. 22340018.

PoS(CPP2010)013

Table 6: Numerical Results of $J_2(\varepsilon)$ with 300 decimal digits precision

ε	$J_2(\varepsilon)$
10^{-10}	$7.4307378573535380568094246928075354312108666776 \times 10^{-9}$
10^{-20}	$7.4307378573533965334478507696274310062608248150 \times 10^{-9}$
10^{-30}	$7.4307378573533965334478366172912736139426954807 \times 10^{-9}$
10^{-60}	$7.4307378573533965334478366172912721987090795999 \times 10^{-9}$
10^{-70}	$7.4307378573533965334478366172912721987090795999 \times 10^{-9}$



(a) Difference of $J_1(\varepsilon)$ and $J_1(10^5\varepsilon)$

(b) Difference of $J_2(\varepsilon)$ and $J_2(10^5\varepsilon)$

Figure 6: Convergence of J_1 and J_2

PoS(CPP2010)013

References

[1] E. Doncker, Y. Shimizu, J. Fujimoto and F. Yuasa, *Computation of loop integrals using extrapolation*, *Comput. Phys. Comm.* **159** (2004) 145–156.

[2] J. Fujimoto, M. Igarashi, N. Nakazawa, Y. Shimizu and K. Tobimatsu, *Radiative corrections to e^+e^- reactions in electroweak theory*, *Prog. Theor. Phys. Suppl.* **100** (1990) 1–379.

[3] J. Fujimoto, Y. Shimizu, K. Kato and Y. Oyanagi, *Numerical approach to one-loop integrals*, *Prog. Theor. Phys.* **87(5)** (1992) 1233–1247.

[4] H. Fujiwara, `http://www-an.acs.i.kyoto-u.ac.jp/˜fujiwara/exflib`

[5] F. Yuasa, E. Doncker, J. Fujimoto, N. Hamaguchi, T. Ishikawa and Y. Shimizu, *Precise Numerical Evaluation of the Scalar One-Loop Integrals with the Infrared Divergence*, in proceedings of *ACAT2007*, `PoS(ACAT)087`.