Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Physics Communications 159 (2004) 145–156
Computer Physics Communications
www.elsevier.com/locate/cpc

Computation of loop integrals using extrapolation

Elise de Doncker [a,*], Yoshimitsu Shimizu [b], Junpei Fujimoto [b], Fukuko Yuasa [b]

[a] *Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008, USA*
[b] *High Energy Accelerator Research Organization (KEK), Oho 1-1, Tsukuba, Ibaraki, 305, Japan*

Received 22 July 2003; accepted 29 January 2004

Abstract

We present a class of methods for the evaluation of loop integrals based on extrapolation. The method is based on generating a sequence of approximations which converge to the loop integral value as a parameter ε introduced in the integrand tends to zero. We examine the applicability of linear and non-linear extrapolation processes. Test results are given for one-loop three-point vertex and four-point functions and for a two-loop vertex diagram.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Loop integrals; Extrapolation; Multivariate integration; Vertex diagram; Box diagram

1. Introduction

In high energy physics the computation of loop integrals is required to obtain higher order terms in perturbation calculations of the scattering amplitude. The latter in turn deliver higher order corrections to the cross section for the collision of elementary particles.

As in [9] the scalar one-loop n-point integral is given by

$$\mathcal{I} = \int \frac{\mathrm{d}^4 l}{(2\pi)^4 i} \frac{1}{(l^2 - m_1^2 + i\varepsilon)((l+p_1)^2 - m_2^2 + i\varepsilon)\cdots((l+\sum_{j=1}^{n-1} p_j)^2 - m_n^2 + i\varepsilon)} \tag{1}$$

where l is the loop momentum, p_j the momentum of the jth external particle and m_j the mass carried by the jth internal line. $\varepsilon > 0$ is a real constant which is supplied to prevent the integral from diverging. A physical scattering amplitude contains this type of integrals and its value is defined at $\varepsilon = 0$. Note that the integral defines an analytic function of the external parameters, such as the Lorentz invariants $p_j \cdot p_k$, and generally exhibits a real and imaginary part.

* Corresponding author.
E-mail addresses: elise@cs.wmich.edu (E. de Doncker), yoshimitsu.shimizu@kek.jp (Y. Shimizu), junpei.fujimoto@kek.jp (J. Fujimoto), fukuko.yuasa@kek.jp (F. Yuasa).
URL: http://www.cs.wmich.edu/~elise (E. de Doncker).

0010-4655/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2004.01.004

This integral is equal to $(-1)^n/(16\pi^2)I_n$ where

$$I_n = \int_{S_{n-1}} \frac{1}{(D_n(\mathbf{x}) - i\varepsilon)^{n-2}}\, \mathrm{d}\mathbf{x} \tag{2}$$

is obtained from (1) by introducing Feynman parameters and integrating over the loop momentum l. The term *scalar* refers to the numerator of the integrand $= 1$; in general it is polynomial. The integration region S_{n-1} is the $n-1$ dimensional unit simplex and $D_n(\mathbf{x})$ is a quadratic; the integrand may have a non-integrable singularity if $D_n(\mathbf{x})$ vanishes within the domain of integration. This happens, in general, when the integral is evaluated in the physical region.

For the simplest cases, the results can be calculated analytically. So far, numerical techniques have been successful only after considerable analytic manipulation (see, e.g., [5,6,9,14,18]). In this paper we present a novel method, relying on multivariate integration and extrapolation, which has shown promise for an automatic calculation of loop integrals. We apply several variations of the method to cases of one-loop vertex ($n = 3$), one-loop box ($n = 4$) and two-loop planar vertex diagrams.

2. Numerical extrapolation and three-point vertex function

We consider the loop integral in the limit as $\varepsilon \to 0$. For example, for $n = 3$ in (2), let us consider

$$\lim_{\varepsilon\to 0} I(\varepsilon) = \lim_{\varepsilon\to 0} \int_{S_2} \frac{D_3(x, y)}{D_3(x, y)^2 + \varepsilon^2}\, \mathrm{d}x\, \mathrm{d}y.$$

As an example we take a fermion vertex with Z^0 exchange, which has a simple structure as depicted in Fig. 1. In this case the quadratic in the denominator is

$$D_3(x, y) = -xys + (x + y)^2 m^2 + (1 - x - y)M^2.$$

Here s denotes the squared energy and m and M are particle masses corresponding to the fermion and Z^0 boson, respectively.

It seems natural to construct a sequence of $I(\varepsilon_\iota), \iota = 0, 1, \ldots$, and extrapolate to the limit $I(0)$. Some methods to perform this numerically are outlined in the remainder of this section and in the next section.

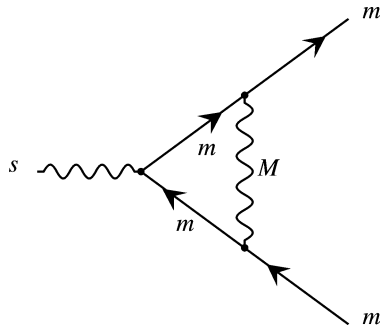

Fig. 1. Vertex example.

Assume we obtain a sequence of approximations $Q(\varepsilon_\iota)$ which satisfy an expansion of the form

$$Q(\varepsilon) \approx I(\varepsilon) = I(0) + \sum_{j=1}^{\nu} a_j \varphi_j(\varepsilon) + R_{\nu+1}(\varepsilon), \tag{3}$$

where $R_{\nu+1}(\varepsilon)$ is a remainder term and it is assumed that the integration error $\Delta(\varepsilon) = Q(\varepsilon) - I(\varepsilon)$ is sufficiently small. We can set $a_0 = I(0)$, $\varphi_0(\varepsilon) = 1$, and consider the φ functions ordered so that $\lim_{\varepsilon\to 0} \varphi_{j+1}(\varepsilon)/\varphi_j(\varepsilon) = 0$.

Subsequently we adapt the procedure given in [12] to more general sequences (see also [4]). Denoting $\beta_\iota = Q(\varepsilon_\iota)$ and disregarding the remainder term in (3), we solve a $(\nu+1)\times(\nu+1)$ linear system of equations, $\Phi\alpha = \beta$, of the form

$$\sum_{j=0}^{\nu} \varphi_j(\varepsilon_\iota)\alpha_j^{(\nu)} = \beta_\iota, \quad \iota = 0, \ldots, \nu. \tag{4}$$

For successive $\nu = 1, 2, \ldots$, this delivers $\alpha_j^{(\nu)} \approx a_j$ and, in particular, $\alpha_0^{(\nu)} \approx a_0 = I(0)$.

Considering an example of [13] with $m = 40$ GeV, $M = 93$ GeV, $s = 9000$ GeV^2, and assuming an expansion (3) in integer powers of ε, i.e. $\varphi_j(\varepsilon) = \varepsilon^j$, solving (4) for a geometric sequence (G) with $\varepsilon_\iota = b^{-\iota}\varepsilon_0, 0 \leqslant \iota \leqslant \nu, b = 2$ and $\varepsilon_0 = 256$ gives the results in the first half of the table in Fig. 2. Since in this case $\varphi_j(\varepsilon_\iota) = {\varepsilon_\iota}^j = b^{-\iota j}{\varepsilon_0}^j$, we can also set $\Phi_{\iota j} = b^{-\iota j}$ and solve for $\alpha_j^{(\nu)} \approx {\varepsilon_0}^j a_j$.

Allowing for a sequence of integer powers of ε in (4) corresponds to assuming that $I(\varepsilon)$ satisfies a polynomial approximation. Fig. 3 displays $Q(\varepsilon)$ as a function of ε for this example. The integral approximations $Q(\varepsilon)$ were calculated (in double precision) to a relative tolerated error of 10^{-12} using the multivariate integration routine (for hyper-rectangular regions) DCUHRE [2,3], after a transformation of the triangular domain. DCUHRE is a sequential predecessor of the adaptive algorithms in [1]. We found that similar results can be obtained for small ε more efficiently (i.e. with less function evaluations) via an iterated integration using a one-dimensional integration in both coordinate directions.

It is interesting to note that the sequence $\{\varepsilon_\iota\}$ does not need to be geometric for this procedure. The second half of Fig. 2 shows the results obtained with a harmonic type progression (H) using $\varepsilon_\iota = \varepsilon_0/(\iota+1), 0 \leqslant \iota \leqslant \nu$ and $\varepsilon_0 = 20$. This helps to construct sequences $\{Q(\varepsilon_\iota)\}$ which may be easier to compute since ε_ι decreases slowly.

The number of function evaluations using DCUHRE and H ranges between 491,335 (for $\varepsilon = 20$) and 4.4 million (for $\varepsilon = 20/9$), for a total of 21.7 million, compared to G with 42,965 (for $\varepsilon = 256$) to 9.8 million ($\varepsilon = 1$) and

	Geometric sequence (G)			Harmonic sequence (H)		
nu	eps	Q(eps)	Extrapolated	eps	Q(eps)	Extrapolated
	256	0.2120319070127095E-03		20	0.2421479625475858E-03	
1	128	0.2280362779518554E-03	0.2440406488910013E-03	10	0.2434730201795342E-03	0.2447980778114825E-03
2	64	0.2363491905451424E-03	0.2448692545542390E-03	20/3	0.2439151979858242E-03	0.2448002914918651E-03
3	32	0.2405610711989021E-03	0.2448014224054772E-03	5	0.2441363743331252E-03	0.2448002403715809E-03
4	16	0.2426777113780660E-03	0.2448002064167953E-03	4	0.2442691075095314E-03	0.2448002403543701E-03
5	8	0.2437382985741919E-03	0.2448002401471510E-03	20/6	0.2443576075549443E-03	0.2448002403553846E-03
6	4	0.2442691075095314E-03	0.2448002403568659E-03	20/7	0.2444208273439749E-03	0.2448002403552964E-03
7	2	0.2445346343652990E-03	0.2448002403554218E-03	2.5	0.2444682451604905E-03	0.2448002403558393E-03
8	1	0.2446674275837987E-03	0.2448002403554184E-03	20/9	0.2445051274397318E-03	0.2448002403553095E-03
		Analytic	0.24480024035541454 1E-03			

nu	1	2	3	4	5	6	7	8
cond# G	3	5	6.4	7.3	7.8	8.0	8.1	8.2
cond# H	3	9	28	92	301	1007	3392	11506

Fig. 2. Three-point vertex diagram, extrapolated results for G and H.

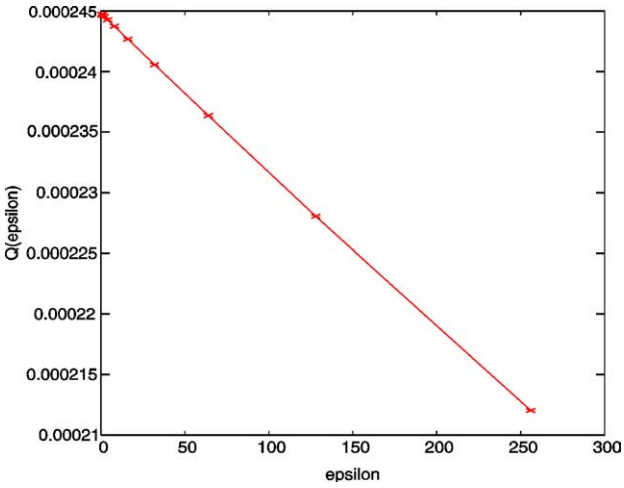

Fig. 3. $Q(\varepsilon)$ vs. ε for $m = 40$ GeV, $M = 93$ GeV, $s = 9000$ GeV^2.

nu	eps	Q(eps) Real part	Extrapolated Real part	Q(eps) Imaginary part	Extrapolated Imaginary part
sqrt(s) = 310 GeV					
	256	0.1047173152009391E-03		0.6907578453625810E-04	
1	128	0.1078275992484533E-03	0.1109378832959675E-03	0.7029122369380996E-04	0.7150666285136181E-04
2	64	0.1093974677595504E-03	0.1109771539288741E-03	0.7094032845066672E-04	0.7161702332624406E-04
3	32	0.1101852766410460E-03	0.1109746945128348E-03	0.7127535323263800E-04	0.7161740765850366E-04
4	16	0.1105797964830976E-03	0.1109746867736801E-03	0.7144548959594558E-04	0.7161737550102994E-04
5	8	0.1107771972690049E-03	0.1109746870094049E-03	0.7153121386874132E-04	0.7161737544194998E-04
6	4	0.1108759312523575E-03	0.1109746870096540E-03	0.7157424000016181E-04	0.7161737544287511E-04
sqrt(s) = 500 GeV					
	256	0.1785651265485312E-05		0.4679678048178459E-04	
1	128	0.2032748040828650E-05	0.2279880816171988E-05	0.4705358047333610E-04	0.4731038046488760E-04
2	64	0.2158231498594798E-05	0.2284993003090598E-05	0.4718080844748737E-04	0.4730725507388898E-03
3	32	0.2221447503631560E-05	0.2284977791296521E-05	0.4724411947584019E-04	0.4730722473997146E-04
4	16	0.2253173421175449E-05	0.2284977693474703E-05	0.4727569801868268E-04	0.4730722478064331E-04
5	8	0.2269065775089006E-05	0.2284977693533693E-05	0.4729146789414364E-04	0.4730722478080402E-04
6	4	0.2277019290263020E-05	0.2284977693532246E-05	0.4729934796379781E-04	0.4730722478080829E-04
sqrt(s) = 1000 GeV					
	256	-0.6202809433449274E-05		0.1535547333295052E-04	
1	128	-0.6153587856041095E-05	-0.6104366278632917E-05	0.1543137710561741E-04	0.1550728087828429E-04
2	64	-0.6128438412498485E-05	-0.6102929865730195E-05	0.1546906111539135E-04	0.1550656654079230E-04
3	32	-0.6109342860112937E-05	-0.6102933414928817E-05	0.1548783341649553E-04	0.1550655820664246E-04
4	16	-0.6109342860112937E-05	-0.6102933441501668E-05	0.1549720180233327E-04	0.1550655821659544E-04
5	8	-0.6106140908836059E-05	-0.6102933441486279E-05	0.1550188151193241E-04	0.1550655821664103E-04
6	4	-0.6104537864311489E-05	-0.6102933441487397E-05	0.1550422024064742E-04	0.1550655821664028E-04

Fig. 4. Three-point vertex diagram, real and imaginary parts using G.

a total of 19.4 million. As another comparison, using G and iterated integration with a version of DQAGE from QUADPACK [15], the number of integrand evaluations ranges between 71,085 (for $\varepsilon = 256$) and 661,815 ($\varepsilon = 1$), for a total of 2.9 million. If the integral approximations would be needed through $\varepsilon = 2$ for H and $\varepsilon = 0.5$ for G, the totals listed above increase to 26.6 million (DCUHRE, H), 39.8 million (DCUHRE, G) and 3.7 million (DQAGE, G). With DQAGE it is possible to handle smaller values of ε more efficiently.

As in [12], the stability of the procedure can be checked by computing a condition number which is obtained for each $\nu = 1, 2, \ldots$ by solving the same system as in (4) but with the right hand side replaced by $\beta_\iota = (-1)^\iota$. It is expected that the process is significantly less stable using H; which is acceptable in this case as we do not need

to solve large systems. We can also use sequences with properties in between those of G and H. These condition numbers (in absolute value) are included in Fig. 2. They give an indication of the accuracy which may be lost in the calculation.

Fig. 4 shows results for the example from [13] with $m = m_t = 150$ GeV, $M = M_Z = 91$ GeV, $\sqrt{s} = 310, 500, 1000$ GeV and the G sequence. The integrals $I(\varepsilon)$ were approximated to a 10^{-12} relative tolerance.

It emerges that the final results can be obtained to high accuracy as long as the $Q(\varepsilon)$ calculations can be performed to high accuracy. Note that it is not necessary to solve a system for each ν. We did so to show the improvement in accuracy for increasing ν. The time for solving the linear systems is very small. It may be noted that the computation for $\varphi_j(\varepsilon) = \varepsilon^j$ using G can also be carried out by a recursive procedure (cf., Richardson extrapolation).

3. Extrapolation by the ε-algorithm

In the method of the previous section it is required to know the nature of the functions $\varphi_j(\varepsilon)$. Extrapolation by the ε-algorithm does not require this specific information, as long as the extrapolation method is known to be valid for the class of expansions of interest. It is a recursive implementation by Wynn [19] of a nonlinear sequence to sequence transformation by Shanks [16].

Subsequently we examine the validity of the ε-algorithm for a class of problems under consideration, and relate its results to those of the previous section. Given a sequence $\{\beta_\iota\}$, $\iota = 0, 1, \ldots$ of real numbers, a triangular table is computed as depicted in Fig. 5, according to

$$\begin{aligned}
&\tau_{\iota,-1} = 0,\\
&\tau_{\iota 0} = \beta_\iota,\\
&\tau_{\iota,\kappa+1} = \tau_{\iota+1,\kappa+1} + \frac{1}{\tau_{\iota+1,\kappa} - \tau_{\iota\kappa}}.
\end{aligned}$$

Only the even-numbered columns have meaning; the odd-numbered ones are to store temporary values. The following theorem captures a fundamental property of the transformation (see also [11,15]).

Theorem 1 (Convergence). *If the sequence $\{\sigma_\iota\}$ satisfies a homogeneous linear difference equation of order ν with constant coefficients, $\sum_{\iota=0}^{\nu} c_\iota \sigma_\iota = 0$, and if $\beta_\iota = S + \sigma_\iota$, $\iota = 0, 1, \ldots$, then $\tau_{\iota,2\nu} = S$ (pending $\tau_{\iota,2\nu}$ exists).*

Sufficient conditions validating Theorem 1 are given by the next theorem and its corollary.

$$\begin{array}{cccccc}
 & \tau_{00} & & & & \\
0 & & \tau_{01} & & & \\
 & \tau_{10} & & \tau_{02} & & \\
0 & & \tau_{11} & & \cdots & \\
 & & \cdots & & \cdots & \\
 & & \cdots & & \cdots & \\
0 & & \tau_{\iota-1,1} & & \cdots & \\
 & \tau_{\iota 0} & & \tau_{\iota-1,2} & & \\
0 & & \tau_{\iota 1} & & & \\
 & \tau_{\iota+1,0} & & & &
\end{array}$$

Fig. 5. ε-algorithm table.

Theorem 2 (Sufficient condition). *The sequence* $\sigma_\iota = u(\iota)\iota^\kappa \varepsilon^\iota$, $\iota = 0, 1, \ldots$, *where* $\kappa \geqslant 0$ *integer,* $\varepsilon \in \Re$ *and the* $u(\iota)$ *are periodic functions with period* ρ*, satisfies a homogeneous linear difference equation with constant coefficients of order* $\nu = (\kappa + 1)\rho$.

Corollary 3 (Extension). *The sequence*

$$\sigma_\iota = \sum_{\kappa=0}^{\mu} \sum_{j=1}^{\nu} u_{j\kappa}(\iota)\iota^\kappa {\varepsilon_j}^\iota, \quad \iota = 0, 1, \ldots,$$

where $\nu \geqslant 1$, $\mu \geqslant 0$ *integer,* $\varepsilon_j \in \Re$*, and* $u_{j\kappa}(\iota)$ *are periodic functions with period* ρ*, satisfies a linear homogeneous difference equation with constant coefficients.*

As a special case of Corollary 3 with $\kappa = 0$ and constant $u_j = u_{j0}(\iota)$ ($\rho = 1$), if furthermore

$$\beta_\iota - S = \sum_{j=1}^{\nu} u_j {\varepsilon_j}^\iota, \tag{5}$$

then $\tau_{\iota,2\nu} = S$ (pending $\tau_{\iota,2\nu}$ exists).

Note the correspondence of (5) with (3) for a geometric sequence $\varepsilon_j = b^{-j}$. Indeed if we let $u_j = a_j {\varepsilon_0}^j$, then $u_j {\varepsilon_j}^\iota = a_j {\varepsilon_0}^j b^{-j\iota} = a_j \varphi_j(\varepsilon_\iota)$. It is clear that for this reasoning to work, the sequence needs to be geometric.

It should be noted that Corollary 3 allows extrapolation with the ε-algorithm for significantly more complicated expansions (3); for instance $\varphi_j(\varepsilon)$ may be of the form $\varepsilon^\gamma \log^\delta \varepsilon$, for $\delta \geqslant 0$ integer and real $\gamma > 0$ (as long as a geometric sequence is used for ε).

For comparison with the previous section we give results of extrapolation with the ε-algorithm for the case of Figs. 2 and 3. A version of the ε-algorithm code from [15] was used. (We should point out that [15] uses extrapolation inside some of its integration algorithms. This differs from what we are doing here in that we are now extrapolating on a sequence of results obtained from an integration code.)

For every $\beta_\iota = \tau_{\iota 0}$, $\iota = 0, 1, \ldots$, the ε-algorithm routine computes the lower diagonal of the table (see Fig. 5) through column ι or $\iota - 1$ if ι is even or odd, respectively. For each element it also calculates an error estimate based on how its value compares with nearby elements. It does not necessarily return the rightmost element computed but rather the one with the least individual error estimate.

Fig. 6 gives results obtained with the sequences $\varepsilon_\iota = \varepsilon_0 b^{-\iota}$, for $b = 2$ and $\varepsilon_0 = 256$ (left), and for $b = 1.2$ and $\varepsilon_0 = 1.2^{40}$ (right). For the latter we display only every fourth result. The integrals $I(\varepsilon)$ were approximated to a relative tolerance of 10^{-12}.

b = 2				b = 1.2 (eps = 1.2^(41-p))			
p	eps	Q(eps)	Extrapolated	p		Q(eps)	Extrapolated
1	256	0.2120319070127095E-03		4		0.1528743894317701E-03	0.4316691758746829E-03
2	128	0.2280362779518554E-03		8		0.1941968510705258E-03	0.2311914999988003E-03
3	64	0.2363491905451424E-03	0.2453337696296586E-03	12		0.2191889439432741E-03	0.2451361661323442E-03
4	32	0.2405610711989021E-03	0.2448867967487399E-03	16		0.2322503791767422E-03	0.2448034893044764E-03
5	16	0.2426777113780660E-03	0.2448017954351509E-03	20		0.2387139036026090E-03	0.2448002168460955E-03
6	8	0.2437382985741919E-03	0.2448006914824931E-03	24		0.2418586009244202E-03	0.2448002405705739E-03
7	4	0.2442691075095314E-03	0.2448001658003823E-03	28		0.2433802831821293E-03	0.2448002405705739E-03
8	2	0.2445346343652990E-03	0.2448002390032324E-03	32		0.2441151680705952E-03	0.2448002403563278E-03
9	1	0.2446674275837987E-03	0.2448002403591474E-03	36		0.2444697959556787E-03	0.2448002403550793E-03
10	0.5	0.2447338315398706E-03	0.2448002403553692E-03	40		0.2446408673684701E-03	0.2448002403552772E-03
		Analytic	0.24480024035541454 1E-03				

Fig. 6. Three-point vertex, ε-algorithm results for G with $b = 2$ (left) and $b = 1.2$ (right).

4. Four-point function

Here the integral is I_4 given by (2), over the three-dimensional unit simplex $\mathcal{S}_3$. According to [9], the quadratic D_4 is expressed as

$$D_4 = {}^{\tau}\mathbf{x}A\mathbf{x} + 2\mathbf{v}\cdot\mathbf{x} + C,$$

where

$$A_{ij} = q_i \cdot q_j,\ q_1 = -p_1, \qquad q_2 = p_2, \qquad q_3 = p_2 + p_3, \qquad C = M_0^2 = m_2^2$$

and

$$v_i = \frac{1}{2}\left(-q_i^2 + M_i^2 - M_0^2\right) \quad \text{with } M_1 = m_1,\ M_2 = m_3,\ M_3 = m_4.$$

Fig. 7 displays the Feynman box diagram for the interaction $e^-e^+ \to t\bar{t}$. Fig. 8 illustrates the use of the ε-algorithm for the integral computation. Here $m_1 = m_3 = M_Z = 91$ GeV, $m_2 = m_e = 0.511$ MeV, $m_4 = m_t = 150$ GeV, $\sqrt{s} = 500$ GeV and $\theta = \angle(\mathbf{p}_1, \mathbf{p}_4)$ [9]. In this table the results are given for $\cos\theta = -0.5$. The integrals were approximated to a requested accuracy of 10^{-7} (in each coordinate direction) using an iterated integration with DQAGE. The extrapolation table is displayed (even-numbered columns), together with the number of function evaluations corresponding to each integral. For $\cos\theta = 0$ and $\cos\theta = 0.5$ as well as for $\cos\theta = -0.5$, the final extrapolation results agree to the 6-digit accuracy of the analytic results reported in [9].

In this case, DCUHRE was not able to approximate the integrals to the desired accuracy. For the kind of singularity involved it is expected that multivariate adaptive integration programs may neglect significant portions of the domain, which is often due to inaccurate error estimates near the singularity. As a future integration project, it could be considered to furnish the adaptive multivariate integration algorithm with better error estimates near singularities and/or a technique to force subdividing the neglected subregions.

It is interesting to observe that the integration errors corresponding to the $Q(\varepsilon)$ column of Fig. 8 for the real part decrease by a factor of 2 from one entry to the next. In particular, for $\varepsilon = 64$ through 0.5 they are: 0.064e-09, 0.032e-09, 0.016e-09, 0.008e-09, 0.004e-09, 0.002e-09, 0.001e-09 and 0.0005e-09. This corresponds to the dominant error term being of order $\Theta(\varepsilon)$ since the ratio of the error from one term to the next is 2. Indeed, denoting the absolute error by $E(\varepsilon) = |Q(\varepsilon) - I(\varepsilon)|$, the error ratio is $\frac{E(2\varepsilon)}{E(\varepsilon)} \sim \frac{2\varepsilon}{\varepsilon} = 2$.

The integrals for the imaginary part behave, however, quite differently. The error ratio sequence for the $Q(\varepsilon)$ column of Fig. 8 for the imaginary part is: 1.20, 1.50, 1.63, 1.70, 1.75, 1.77 and 1.79, indicating that the error does not appear to be of order $\Theta(\varepsilon)$. The ε-algorithm succeeds in the extrapolation to the limit.

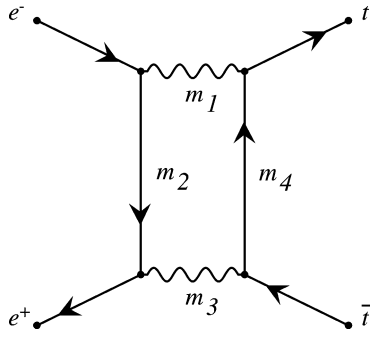

Fig. 7. Feynman diagram for $e^-e^+ \to t\bar{t}$.

eps	Q(eps)	extrapolated results			analytic	#evals
Real part						
64	-0.6144285e-09					5.9e07
32	-0.5822585e-09	-0.5497390e-09				8.7e07
16	-0.5660866e-09	-0.5502165e-09	-0.5502169e-09			1.2e08
8	-0.5580768e-09	-0.5502169e-09	-0.5502165e-09	-0.5501814e-09		1.6e08
4	-0.5541097e-09	-0.5501958e-09	-0.5501784e-09	-0.5501814e-09	-0.550181e-09	2.2e08
2	-0.5521396e-09	-0.5501862e-09	-0.5501811e-09			2.8e08
1	-0.5511587e-09	-0.5501829e-09				3.5e08
0.5	-0.5506695e-09					4.4e08
Imaginary part						
64	0.1212204e-08					6.3e07
32	0.1209017e-08	0.1217354e-08				9.1e07
16	0.1203859e-08	0.1185690e-08	0.1193325e-08			1.3e08
8	0.1199841e-08	0.1192165e-08	0.1193381e-08	0.1193449e-08		1.7e08
4	0.1197203e-08	0.1193069e-08	0.1193413e-08	0.1193430e-08	0.119343e-08	2.3e08
2	0.1195593e-08	0.1193304e-08	0.1193424e-08			2.9e08
1	0.1194648e-08	0.1193381e-08				3.6e08
0.5	0.1194106e-08					4.5e08

Fig. 8. $e^- e^+ \to t\bar{t}$ extrapolation table for $\cos\theta = -0.5$.

5. Two-loop vertex

A general Feynman loop diagram for n external particles, N internal lines and L independent loops is proportional to

$$\mathcal{I}[\wp] = \int \prod_{a=1}^{L} \left(\frac{\mathrm{d}^4 l_a}{(2\pi)^4 i} \right)^L \prod_{\iota=1}^{N} \frac{1}{k_\iota^2 - m_\iota^2 + i\varepsilon} \wp(k_1, \ldots, k_N) = \left(\frac{1}{16\pi^2} \right)^L I[\wp], \tag{6}$$

where the momenta on the ιth propagator and the according masses are denoted by k_ι and m_ι, respectively, for $\iota = 1, \ldots, N$; the independent loop momenta are $l_a, a = 1, \ldots, L$; the external momenta are p_j, $j = 1, \ldots, n$; and the internal momentum k_ι is a linear combination of the l_a and the external momenta.

In this section we will treat the scalar integral (with $\wp = 1$) for the planar two-loop vertex diagram, which corresponds to (6) with $L = 2$, $N = 6$ internal lines and $n = 3$ external lines. The Feynman diagram is shown in Fig. 9.

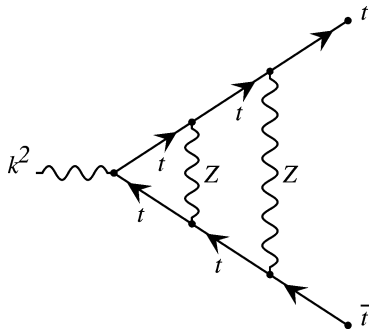

Fig. 9. Feynman diagram for vertex correction of top quark with two Z^0 boson exchanges.

Introducing of the Feynman parameters and integrating over the loop momenta then leads to the 5-dimensional integral given in [10]:

$$I[1] = \int_{\mathcal{S}_5} \frac{1}{(D(\mathbf{x}) + i\varepsilon)^2}\, \mathrm{d}\mathbf{x}, \tag{7}$$

with

$$\begin{aligned} D = {} & C\big(x_1(p_1^2 - m_1^2) + x_2(p_2^2 - m_2^2) - x_3 m_3^2 + x_4(p_1^2 - m_4^2) + x_5(p_2^2 - m_5^2) - x_6 m_6^2\big) \\ & - C_1\big(x_5^2 p_2^2 + x_4^2 p_1^2 - x_4 x_5(p_3^2 - p_1^2 - p_2^2)\big) - C_2\big(x_2^2 p_2^2 + x_1^2 p_1^2 - x_1 x_2(p_3^2 - p_1^2 - p_2^2)\big) \\ & - 2x_2 x_3 x_5 p_2^2 - 2x_1 x_3 x_4 p_1^2 + x_3(x_2 x_4 + x_1 x_5)\big(p_3^2 - p_1^2 - p_2^2\big), \end{aligned}$$

and where

$$\begin{aligned} x_6 &= 1 - x_1 - x_2 - x_3 - x_4 - x_5, \\ C_1 &= x_1 + x_2 + x_3, \\ C_2 &= 1 - x_1 - x_2, \\ C &= x_3(1 - x_1 - x_2 - x_3) + (x_1 + x_2)(1 - x_1 - x_2). \end{aligned}$$

In view of the fact that D in (7) vanishes for $x_1 = x_2 = x_3 = 0$ we perform a transformation,

$$\begin{aligned} x_1 &= x_1(r_0, \tilde{x}_1) = r_0 \tilde{x}_1, \\ x_2 &= x_2(r_0, \tilde{x}_1) = r_0(1 - \tilde{x}_1), \\ x_3 &= x_3(r_0, \tilde{x}_1, \tilde{x}_2) = r_0(1 - \tilde{x}_1 - \tilde{x}_2). \end{aligned} \tag{8}$$

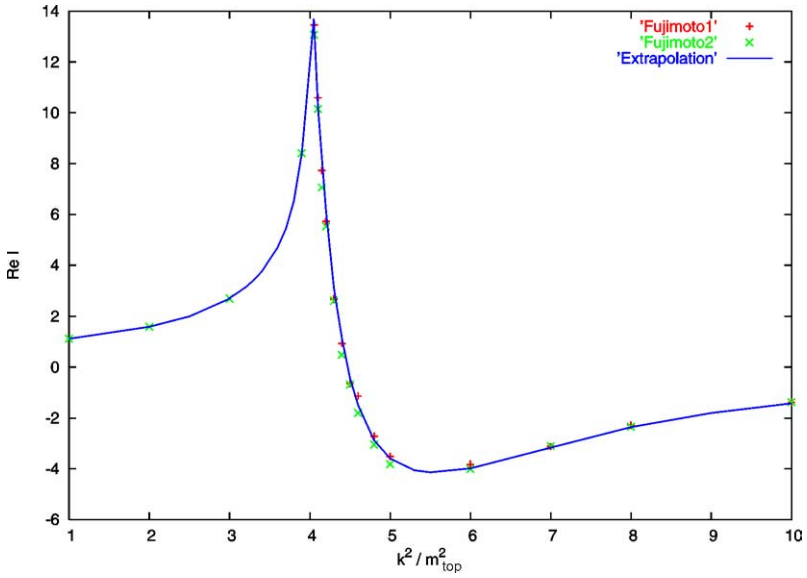

Fig. 10. Approximations to $\Re eI$ as a function of k^2/m_{top}^2, $m_{\mathrm{top}} = 150$ GeV and $M_Z = 91.17$ GeV.

Denoting the integrand of (7) by $f(\mathbf{x})$, the transformation (8) results in

$$I = \int_0^1 \mathrm{d}r_0\, r_0^2 \int_0^1 \mathrm{d}\tilde{x}_1 \int_{2-\frac{1}{r_0}-\tilde{x}_1}^{1-\tilde{x}_1} \int_0^{1-x_1-x_2-x_3} \mathrm{d}x_4 \int_0^{1-x_1-x_2-x_3-x_4} \mathrm{d}x_5\, f.$$

Using $\tilde{x}_2 = 2 - \frac{1}{r_0} - \tilde{x}_1 + (\frac{1}{r_0} - 1)t_3$ to map the outer 3-dimensional integral to the 3d unit hypercube we obtain

$$I = \int_0^1 \mathrm{d}r_0\, r_0(1-r_0) \int_0^1 \mathrm{d}\tilde{x}_1 \int_0^1 \mathrm{d}t_3 \int_0^{1-x_1-x_2-x_3} \mathrm{d}x_4 \int_0^{1-x_1-x_2-x_3-x_4} \mathrm{d}x_5\, f. \tag{9}$$

To approximate the integral (9) we can transform it to the 5-dimensional unit hypercube. Alternatively, since the inner two dimensions are the most difficult, we can perform the outer 3-dimensional integration of the inner 2d integral and use DQAGE for the inner two dimensions. Using the latter method we generated the results displayed on the curve labeled "Extrapolation" in Fig. 10, which represents the integral approximations to (9) as a function of k^2/m_{top}^2 pertaining to the vertex correction of the top quark with two Z^0 boson exchanges. Also shown (as points) are corresponding results from [10].

We applied the ε-algorithm for the extrapolation, using integral approximations corresponding to a geometric progression of ε, with base 1.2 or 2. While the integrals on the left of the k^2/m_{top}^2 threshhold are easy to compute, those on the right are considerably hard and computation intensive, even for fairly large values of ε.

6. Remarks regarding large mass ratios

It is clear that the success (or otherwise) of an extrapolation procedure will depend on whether it is able to intrinsically model the behavior of the entry sequence. We have observed that, for the three-point vertex function, the ratio $\frac{M}{m}$ is a determining factor. The problem is more pronounced for higher values of the energy $\sqrt{s}$. In this case, while the ε-algorithm appears to become unstable when ε is relatively large, the polynomial model delivers an extrapolated sequence only a little "ahead" of the entry sequence but with a behavior similar to the latter.

As an example, Fig. 11(a) displays $Q(\varepsilon)$ as a function of ε $(= 1.2^{20-j}, j = 0, 1, \ldots)$ for $m = 150$ GeV, $M = 10^{-3} m_e = 0.511\text{e-}6$ GeV (fictitious photon mass) and $\sqrt{s} = 310$ GeV, which pertains to the *infrared singularity* arising from the photon exchange. Here M is used as the infrared cutoff usually denoted by λ.

In (b) $Q(\varepsilon)$ appears linear as a function of $\log_{1.2}\varepsilon$, indicating that $Q(\varepsilon)$ is logarithmic over the given range. The value of $Q(\varepsilon) - I(0)$ is shown in (c) as a function of $\log_{1.2}(\varepsilon)$, where $I(0) \approx -0.7451\text{e-}2$ according to [8]. The data of (d) are as in (b), but for $M = 0.1$ GeV.

Fig. 12 shows $Q(\varepsilon)$ as a function of ε for $s = 9000$ GeV^2, $m = 150$ GeV, $M = 0.1$ GeV. These integrals are easy, allowing the integration for arbitrarily small ε. It emerges that the graph follows a logarithmic behavior over a large range (see (a)). However, note how in (b) and (c) the curve gradually becomes as that of Fig. 3 over ranges of smaller ε. This change of behavior for very small ε needs to be accurately modeled in order to treat small values of M. For $s = 9000$ GeV^2, $m = 150$ GeV, (d) depicts the integral as a function of ε and M, showing the divergence of the integral as M decreases.

With DQAGE we are able to calculate the integrals for the smaller values of ε needed to handle fairly considerable mass ratios and energy, for example, $m = 150$ GeV, $M = 0.01$ GeV and $\sqrt{s} = 310$ GeV. Convergence results for this case are given in the table of Fig. 13, based on the expansion (4) in integer powers of ε and using a geometric sequence with $b = 1.3$ and $\varepsilon_0 = 1.3^{-30}$. For this case, the method used yields results for mass ratios through about five orders of magnitude.

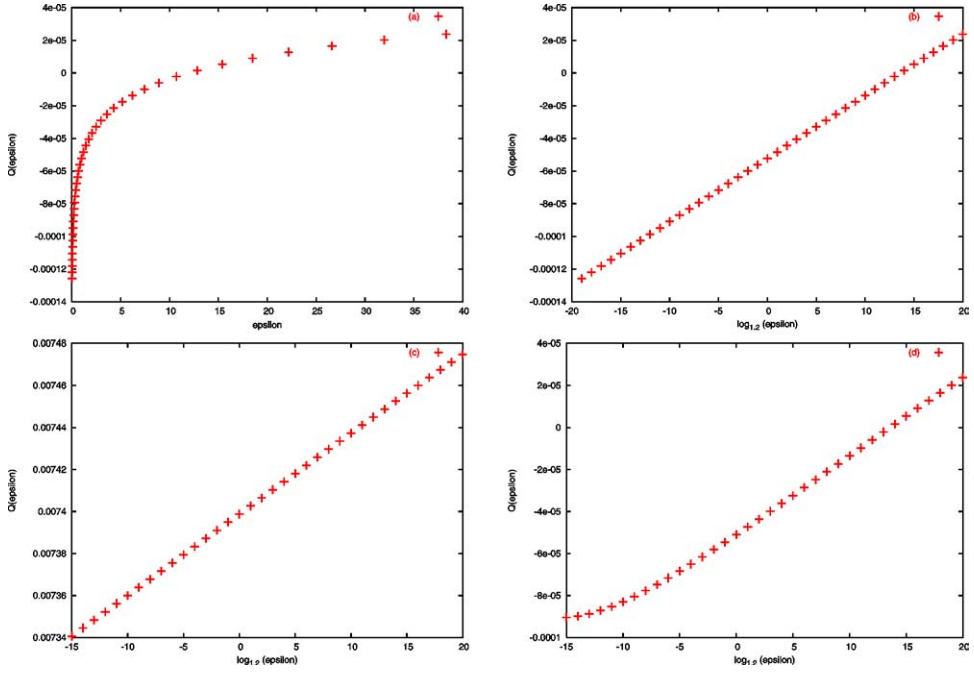

Fig. 11. (a) $Q(\varepsilon)$ vs. ε and (b) vs. $\log_{1.2}\varepsilon$ for $\sqrt{s} = 310$ GeV, $m = 150$ GeV and $M = 10^{-3}m_e$; (c) $Q(\varepsilon) - I(0)$ vs. $\log_{1.2}(\varepsilon)$; (d) as in (b) for $M = 0.1$ GeV.

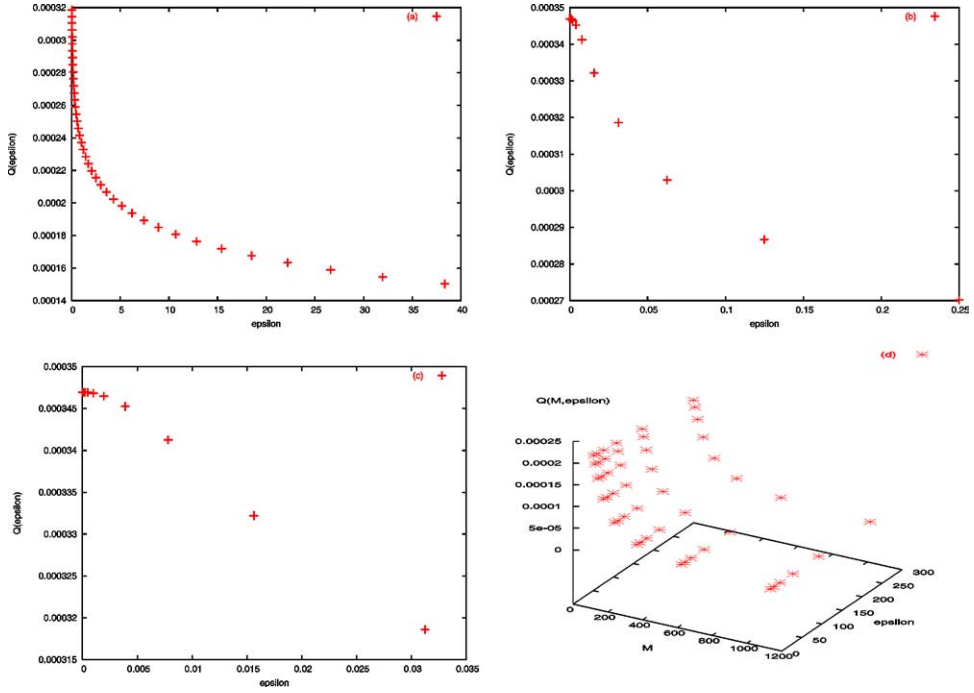

Fig. 12. (a)–(c) $Q(\varepsilon)$ vs. ε for $s = 9000$ GeV^2, $m = 150$ GeV, $M = 0.1$ GeV and various ranges of ε; (d) $Q(M, \varepsilon)$ for $s = 9000$ GeV^2, $m = 40$ GeV.

eps	Q(eps)	Extrapolated	#evals.	eps	Q(eps)	Extrapolated	#evals
0.381680e-03	-0.185652e-03		2.32e06				
0.293600e-03	-0.181426e-03	-0.167341e-03	2.37e06	0.46790e-04	-0.989755e-04	-0.446396e-04	2.88e06
0.225847e-03	-0.174782e-03	-0.131318e-03	2.40e06	0.35992e-04	-0.878292e-04	-0.447056e-04	2.94e06
0.173728e-03	-0.165541e-03	-0.899977e-04	2.49e06	0.27686e-04	-0.785577e-04	-0.443406e-04	3.01e06
0.133637e-03	-0.153854e-03	-0.549082e-04	2.57e06	0.21297e-04	-0.710526e-04	-0.442841e-04	3.10e06
0.102798e-03	-0.140320e-03	-0.363915e-04	2.63e06	0.16382e-04	-0.650854e-04	-0.442955e-04	3.15e06
0.790751e-04	-0.125943e-03	-0.353284e-04	2.74e06	0.12602e-04	-0.603956e-04	-0.442976e-04	3.21e06
0.608270e-04	-0.111854e-03	-0.413304e-04	2.77e06	0.96938e-05	-0.567369e-04	-0.442978e-04	3.36e06

Fig. 13. One-loop vertex illustration of large m/M ratio: $\sqrt{s} = 310$ GeV, $m = 150$ GeV, $M = 0.01$ GeV. This should be compared with the analytic value -0.44297521952810759e-04.

7. Conclusions and future work

In this paper we presented a class of methods for the evaluation of loop integrals based on extrapolation. The extrapolation is based on generating a sequence of approximations which converge to the loop integral value as a parameter ε introduced in the integrand tends to zero.

Using the ε-algorithm for extrapolation this delivers an automatic method. Further work is needed to establish properties of the transient behavior as a function of ε as it relates to the convergence properties of the extrapolation process. As an alternative method, the generalized Richardson extrapolation process by Sidi et al. [7,17] should also be investigated. Further work is also needed to address the infrared singularity.

In view of the importance of the application, tailoring of multivariate integration codes is warranted to make these calculations more efficient. Furthermore, the large granularity of the integrands as well as the number of integrals involved makes the application a reasonable candidate for parallel approaches.

References

[1] http://www.cs.wmich.edu/parint, PARINT web site.
[2] J. Berntsen, T.O. Espelid, A. Genz, An adaptive algorithm for the approximate calculation of multiple integrals, ACM Trans. Math. Softw. 17 (1991) 437–451.
[3] J. Berntsen, T.O. Espelid, A. Genz, Algorithm 698: DCUHRE—an adaptive multidimensional integration routine for a vector of integrals, ACM Trans. Math. Softw. 17 (1991) 452–456, Available from http://www.sci.wsu.edu/math/faculty/genz/homepage.
[4] C. Brezinski, A general extrapolation algorithm, Numer. Math. 35 (1980) 175–187.
[5] A. Ferroglia, G. Passarino, M. Passera, S. Uccirati, All-purpose numerical evaluation of one-loop multi-leg Feynman diagrams, hep-ph/0209219.
[6] J. Fleischer, O.V. Tarasov, Calculation of Feynman diagrams from their small momentum expansion, Z. Phys. C 64 (1994) 413–425.
[7] W. Ford, A. Sidi, An algorithm for the generalization of the Richardson extrapolation process, SIAM J. Numer. Anal. 24 (1987) 1212–1232.
[8] J. Fujimoto, Y. Shimizu, K. Kato, Y. Oyanagi, Numerical approach to one-loop integrals, in: Computing in High Energy Physics '91, 1991, pp. 407–411.
[9] J. Fujimoto, Y. Shimizu, K. Kato, Y. Oyanagi, Numerical approach to one-loop integrals, Progr. Theoret. Phys. 87 (5) (1992) 1233–1247.
[10] J. Fujimoto, Y. Shimizu, K. Kato, Y. Oyanagi, Numerical approach to two-loop integrals, in: Proc. of the VII Workshop on High Energy Physics and Quantum Field Theory, 1992.
[11] A. Genz, The approximate calculation of multidimensional integrals using extrapolation methods, PhD thesis, University of Kent at Canterbury, 1975.
[12] J.N. Lyness, Applications of extrapolation techniques to multidimensional quadrature of some integrand functions with a singularity, J. Comput. Phys. 20 (1976) 346–364.
[13] Y. Oyanagi, T. Kaneko, T. Sasaki, S. Kawabata, Y. Shimizu, How to calculate one-loop diagrams, in: Perspectives of Particle Physics, 1988.
[14] G. Passarino, An approach toward the numerical evaluation of multiloop Feynman diagrams, Nucl. Phys. B 619 (2001) 257.
[15] R. Piessens, E. de Doncker, C.W. Überhuber, D.K. Kahaner, A Subroutine Package for Automatic Integration, in: Springer Series in Computational Mathematics, Springer-Verlag, 1983.
[16] D. Shanks, Non-linear transformations of divergent and slowly convergent sequences, J. Math. Phys. 34 (1955) 1–42.
[17] A. Sidi, Some properties of a generalization of the Richardson extrapolation process, JIMA 24 (1979) 327–346.
[18] F.V. Tkachov, Algebraic algorithms for multiloop calculations: The first 15 years. What's next?, Nucl. Phys. B 389 (1997) 309.
[19] P. Wynn, On a device for computing the $e_m(s_n)$ transformation, Mathematical Tables and Aids to Computing 10 (1956) 91–96.