Automatic numerical integration and extrapolation for Feynman loop integrals

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

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

[2]High Energy Accelerator Research Organization (KEK), Tsukuba, Japan
[3]Department of Physics, Kogakuin University, Shinjuku, Tokyo, Japan

Oct. 9, 2016

Outline

1. Loop integral - representation
2. Extrapolation/convergence acceleration methods
3. Automatic integration/ParInt
   - Automatic Integration
   - PARINT
   - Parallel multivariate integration
4. Numerical results
   - 2-loop self-energy/Magdeburg
   - 2-loop vertex, box - PARINT performance
   - 3-loop self-energy
   - 3-loop vertex
   - 4-loop self-energy
5. Conclusions

Outline

1. Loop integral - representation
2. Extrapolation/convergence acceleration methods
3. Automatic integration/ParInt
   - Automatic Integration
   - PARINT
   - Parallel multivariate integration
4. Numerical results
   - 2-loop self-energy/Magdeburg
   - 2-loop vertex, box - PARINT performance
   - 3-loop self-energy
   - 3-loop vertex
   - 4-loop self-energy
5. Conclusions

Outline

1. Loop integral - representation
2. Extrapolation/convergence acceleration methods
3. Automatic integration/ParInt
   - Automatic Integration
   - ParInt
   - Parallel multivariate integration
4. Numerical results
   - 2-loop self-energy/Magdeburg
   - 2-loop vertex, box - ParInt performance
   - 3-loop self-energy
   - 3-loop vertex
   - 4-loop self-energy
5. Conclusions

Outline

1. Loop integral - representation
2. Extrapolation/convergence acceleration methods
3. Automatic integration/ParInt
   - Automatic Integration
   - PARINT
   - Parallel multivariate integration
4. Numerical results
   - 2-loop self-energy/Magdeburg
   - 2-loop vertex, box - PARINT performance
   - 3-loop self-energy
   - 3-loop vertex
   - 4-loop self-energy
5. Conclusions

Outline

1. Loop integral - representation
2. Extrapolation/convergence acceleration methods
3. Automatic integration/ParInt
   - Automatic Integration
   - PARINT
   - Parallel multivariate integration
4. Numerical results
   - 2-loop self-energy/Magdeburg
   - 2-loop vertex, box - PARINT performance
   - 3-loop self-energy
   - 3-loop vertex
   - 4-loop self-energy
5. Conclusions

Loop integral - representation

L-loop integral with N internal lines

$$\mathcal{I} = \frac{\Gamma\left(N - \frac{nL}{2}\right)}{(4\pi)^{nL/2}} (-1)^N \int_0^1 \prod_{r=1}^{N} dx_r \, \delta(1 - \sum x_r) \frac{C^{N-n(L+1)/2}}{(D - i\varrho C)^{N-nL/2}} \qquad (1)$$

- C and D are polynomials determined by the topology of the corresponding diagram and physical parameters
- $n = n(\varepsilon)$ to account for IR or UV singularity (where ε = dimensional regularization parameter); let $n(\varepsilon) = 4 - 2\varepsilon$ for UV singularity

Loop integral - representation

$$\mathcal{I}_{N,L} = \Gamma(N - \frac{nL}{2})(-1)^N \int_0^1 dx_1 \int_0^{1-x_1} dx_2 \ldots \int_0^{1-x_1\ldots-x_{N-2}} \frac{C^{N-n(L+1)/2}}{(D - i\varrho C)^{N-nL/2}}$$

$$= \Gamma(N - \frac{nL}{2})(-1)^N \int_{\mathcal{S}_{N-1}} \frac{C^{N-n(L+1)/2}}{(D - i\varrho C)^{N-nL/2}} \, d\mathbf{x}$$

where $\mathcal{S}_d = \{\mathbf{x} \in \mathbb{R}^{\mathbf{d}} \mid 0 \le \sum_{r=1}^{d} x_r \le 1\}$: d-dimensional unit simplex

Napkin integral

$N = 3, L = 1, C = 1, D = -x_1 x_2 s + (x_1 + x_2)^2 m^2 + (1 - x_1 - x_2) M^2$
Example for $m = 40, M = 93, s = 9000$

$$\mathcal{I}_{3,1}(\rho) = -\int_0^1 dx_1 \int_0^{1-x_1} dx_2 \, \frac{1}{D - i\varrho}$$

$$Re\,\mathcal{I}_{3,1}(\rho) = -\int_0^1 dx_1 \int_0^{1-x_1} dx_2 \, \frac{D}{D^2 + \varrho^2} \approx \; I + C_1 \varrho + C_2 \varrho^2 + \ldots + C_\nu \varrho^\nu$$

Linear extrapolation as $\varrho \to 0$: Let $\varrho = \varrho_\ell = 2^{7-\ell}$, $\ell = 0, 1, \ldots$ [3]
Approximate $Q(\rho_\ell) \approx \mathcal{I}_{3,1}(\rho_\ell)$ and solve $(\nu + 1) \times (\nu + 1)$ linear systems, $\nu = 1, 2, \ldots$

$$Q(\rho_\ell) = \sum_{k=0}^{\nu} C_k {\varrho_\ell}^k, \quad \ell = 0, \ldots, \nu$$

Extrapolation/convergence acceleration methods

(i) If denominator vanishes in interior of the integration domain: integral calculated in the limit as $\varrho \to 0$

(ii) Integral with infrared (IR) singularity ($n = 4 + 2\varepsilon$ in (1)): calculated in the limit as $\varepsilon \to 0$ [6]

(iii) Integral with ultraviolet (UV) singularity ($n = 4 - 2\varepsilon$): calculated in the limit as $\varepsilon \to 0$

Asymptotics/Mechanics of extrapolation

Numerical extrapolation (is tailored to an underlying asymptotic expansion):
Linear extrapolation for

$$S(\varepsilon) \sim C_K\varphi_K(\varepsilon) + C_{K+1}\varphi_{K+1}(\varepsilon) + C_{K+2}\varphi_{K+2}(\varepsilon) + \ldots, \quad \text{as } \varepsilon \to 0.$$

assuming the φ_k functions are known, for example, $\varphi_k(\varepsilon) = \varepsilon^k$.
Create sequences of $S(\varepsilon_\ell)$ such that

$$S(\varepsilon_\ell) = C_K\varphi_K(\varepsilon_\ell) + C_{K+1}\varphi_{K+1}(\varepsilon_\ell) + \ldots C_{K+\nu}\varphi_{K+\nu}(\varepsilon_\ell), \quad \ell = 0, \ldots, \nu.$$

Solve linear systems of orders $(\nu+1) \times (\nu+1)$, for increasing values of ν.
and decreasing $\varepsilon = \varepsilon_\ell$ (e.g., geometric sequence $\varepsilon_\ell = b^{-\ell}$, $b > 1$).
Bulirsch [2] type sequences can be used for a sequence of the form
$\varepsilon_\ell = 1/b_\ell,\ b_\ell = 2, 3, 4, 6, 8, 12, 16, 24, \ldots$.
Non-linear extrapolation with the ϵ-algorithm [11, 13, 12] can be applied under more general conditions with geometric sequences of ε_ℓ [3].

Automatic integration

Black box approach

Obtain an approximation Qf to an integral

$$If = \int_{\mathcal{D}} f(\vec{x})\, d\vec{x}$$

and error estimate $\mathcal{E}f$, in order to satisfy a specified accuracy requirement for the actual error, of the form:

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

for given *integrand function* f, region $\mathcal{D}$ and (absolute/relative) error tolerances t_a and t_r.

PARINT package

PARINT (PARallel/distributed INTegration), over MPI (Message Passing Interface), includes:

- Multivariate adaptive code: *low dimensions (say, ≤ 12), deals with non-severe integrand problems*
- Quasi-Monte Carlo (QMC): sequence of Korobov/Richtmyer rules (non-adaptive); randomized copies of each rule are applied for error estimate computation, *high dimensions okay, smooth integrand behavior*
- Monte Carlo (MC): based on (choice of) SPRNG or Random123 Pseudo-Random Number Generators (PRNG), *high dimensions, erratic integrand and/or domain*
- (1D) adaptive quadrature methods from QuadPack [10], can be used in iterated (repeated) integration

Parallel multivariate integration

- On the rule or points level: in non-adaptive algorithms, e.g., Monte-Carlo (MC) algorithms and composite rules using grid or lattice points, $If = \int_{\mathcal{D}} f \approx \sum_k w_k f(\vec{x}_k)$: computation of the $f(\vec{x}_k)$ evaluation points in parallel
- On the region level: in adaptive (region-partitioning) methods, task pool algorithms, load balancing (distributed memory systems); or maintaining shared priority queue (in shared memorory systems)
- In iterated integration:
  - On the rule level: inner integrals are independent and computed in parallel, e.g., over subregion $\mathcal{S} = \mathcal{D}_1 \times \mathcal{D}_2$ (inner region $\mathcal{D}_2$) $\int_{\mathcal{S}} F(\vec{x})dx \approx \sum_k w_k F(\vec{x}_k)$, with $F(\vec{x}_k) = \int_{\mathcal{D}_2} f(\vec{x}_k, \vec{y})d\vec{y}$
  - Inner integrations can be performed adaptively.

Adaptive partitioning

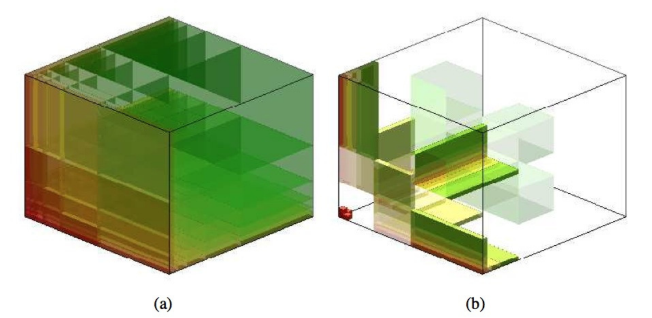


Figure : Adaptive partitioning of the domain (singularity along axes) [9]

Priority driven adaptive algorithm

Adaptive region partitioning

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

Subregion approximations:
(2D) $\sum_k w_k f(x_k, y_k)$
Iterated $(1D)^2$ $\sum_i u_i \sum_j v_j f(x_i, y_j)$

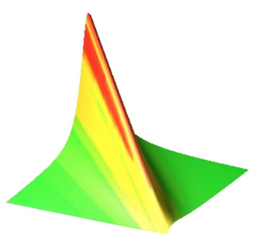
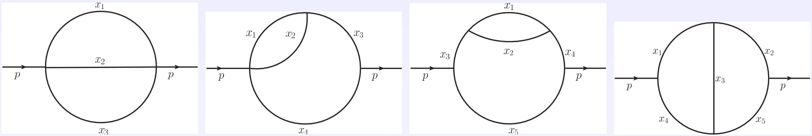


Figure : $\int_0^1 dx \int_0^1 dy \frac{2\varrho y}{(x+y-1)^2+\varrho^2}$
$= \int_0^1 dx \left[\int_0^1 dy \frac{2\varrho y}{(x+y-1)^2+\varrho^2}\right]$

Sample 2-loop self-energy diagrams



(a) (b) (c) (d)

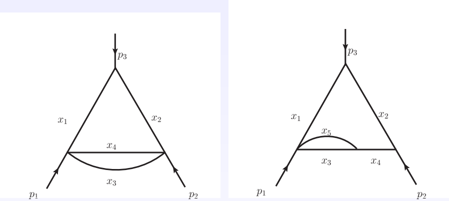
Figure : [2ls] 2-loop self-energy diagrams with massive internal lines: (a) 2-loop *sunrise-sunset* $N = 3$ (Laporta [8] Fig 2(b)), (b) 2-loop *lemon* $N = 4$ (Laporta [8] Fig 2(c)), (c) 2-loop *half-boiled egg* $N = 5$, (d) 2-loop *Magdeburg* $N = 5$ (Laporta [8] Fig 2(d))

Extrapolation results for Magdeburg diagram Fig [2ls] (d)

	Magdeburg Integral		$I_d^{S2}\,\Gamma(1+\varepsilon)^{-2}$	Extrapolation		
ℓ	E_r	T[s]	Res. C_0	Res. C_1	Res. C_2	Res. C_3
0	8.5e-14	0.8				
1	1.0e-13	19.7	0.69130084611470	-0.142989490499		
2	1.6e-13	7.5	0.84949643770104	-0.617576265258	0.3163911832	
3	4.8e-13	6.9	0.90878784906010	-1.032616144771	1.1464709422	-0.47433129
4	8.5e-13	6.6	0.92198476262012	-1.230569848171	2.0702548914	-2.05796092
5	1.2e-12	6.5	0.92353497740505	-1.278626506507	2.5508214747	-3.98022725
6	3.0e-12	6.6	0.92362889210499	-1.284543132600	2.6730984140	-5.02831530
7	2.4e-12	6.6	0.92363178137723	-1.284910070175	2.6885097922	-5.30131686
8	2.8e-12	6.6	0.92363182617006	-1.284921492347	2.6894768694	-5.33613164
9	2.8e-12	6.6	0.92363182651847	-1.284921670382	2.6895071354	-5.33832809
10	2.9e-12	6.6	0.92363182651995	-1.284921671903	2.6895076534	-5.33840357
11	2.9e-12	6.6	0.92363182651990	-1.284921671790	2.6895075765	-5.33838110
12	2.9e-12	6.6	0.92363182651992	-1.284921671898	2.6895077252	-5.33846839
13	3.7e-12	6.6	0.92363182651991	-1.284921671798	2.6895075774	-5.33835823
14	2.9e-12	6.6	0.92363182651991	-1.284921671840	2.6895076182	-5.33839951
		Exact:	0.9236318265199	-1.284921671848	2.6895076265	-5.33839923

Table : *Magdeburg* integral (by ParInt on *thor* cluster, 64 procs., in *long double* precision), $t_r = 10^{-13}$, max. # evals = 1B, $\varepsilon = 2^{-\ell}$, $I_d^{S2}\,\Gamma(1+\varepsilon)^{-2} \sim \sum_{k\geq 0} C_k\,\varepsilon^k \approx 0.9236318265199 - 1.284921671848\,\varepsilon + 2.689507626490\,\varepsilon^2 - 5.338399227511\,\varepsilon^3 \ldots$

Sample 2-loop vertex diagrams



(a) (b)

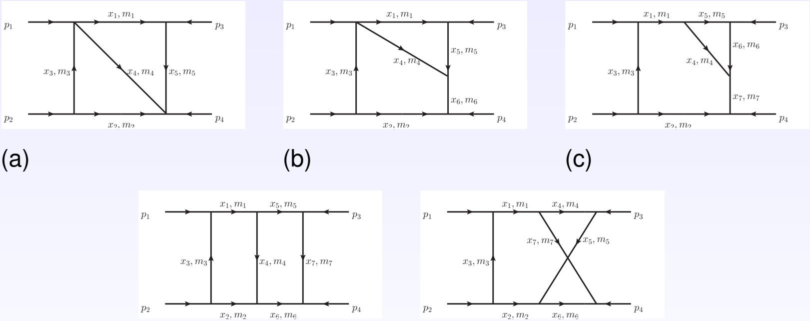
Figure : [2lv] 2-loop vertex (UV-divergent) diagrams with massive internal lines: (a) $N = 4$ (Laporta [8] Fig 3(b)), (b) $N = 5$ (Laporta [8] Fig 3(c))

Extrapolation results for diagram Fig [2lv] (a)

	Integral $I_a^{V2}\,\Gamma(1+\varepsilon)^{-2}$ Extrapolation						
b_ℓ	E_r	T[s]	Res. C_{-2}	Res. C_{-1}	Res. C_0	Res. C_1
3	8.7e-12	1.4				
4	2.3e-11	1.5	0.51489021736	0.535680679		
6	3.1e-10	1.8	0.50586162735	0.598880809	-0.1083431	
8	8.2e-11	5.2	0.50111160609	0.660631086	-0.3648442	0.342002
12	1.2e-09	2.5	0.50015901670	0.680635463	-0.5153534	0.822107
16	1.7e-09	4.2	0.50001764652	0.685300679	-0.5733151	1.161395
24	4.4e-09	20.5	0.50000135665	0.686098882	-0.5885950	1.307352
32	7.7e-09	19.1	0.50000007798	0.686192225	-0.5912981	1.347595
48	3.8e-10	8.6	0.50000000083	0.686200327	-0.5916415	1.355242
		Exact:	0.5	0.686200636	-0.5916667	1.356197

Table : Results 2-loop UV vertex integral [5], I_a^{V2} (on Mac Pro), rel. err. tol. $t_r = 10^{-10}$ (outer), 5×10^{-11} (inner three), $T[s]$ = Time (elapsed user time in s), $\varepsilon = 1/b_\ell$ (starting at 1/3), $E_r =$ outer integration estim. rel. error;
$I_a^{V2}(\varepsilon)\,\Gamma(1+\varepsilon)^{-2} \sim \sum_{k\geq -2} C_k\,\varepsilon^k \approx$
$0.5\,\varepsilon^{-2} + 0.6862006357658\,\varepsilon^{-1} - 0.5916667014024 + 1.356196533114\,\varepsilon \ldots$ [8]

Sample 2-loop box diagrams



(a) (b) (c) (d) (e)

Figure : [2lb] Sample 2-loop box diagrams (a) $N = 5$ (double-triangle), (b) $N = 6$ (tetragon-triangle), (c) $N = 7$ (pentagon-triangle), (d) $N = 7$ ladder, (e) $N = 7$ crossed ladder [8, 4]

Parallel performance of ParInt adaptive integration for 2-loop box integrals on MPI (OpenMPI)

2-LOOP DIAG.	N	REL TOL t_r	MAX EVALS	$T_1\,[s]$	$T_{64}\,[s]$	SPEEDUP
Fig [2lb] (a)	5	10^{-10}	400M	32.6	0.74	44.1
Fig [2lb] (b)	6	10^{-9}	3B	213.6	5.06	42.2
Fig [2lb] (c)	7	10^{-8}	5B	507.9	8.83	57.5
Fig [2lb] (d) ladder	7	10^{-8}	2B	189.9	4.33	43.9
Fig [2lb] (e) crossed	7	10^{-7}	300M	27.6	0.49	56.3
Fig [2lb] (e) crossed	7	10^{-9}	20B	1892.5	34.6	54.7

Table : [2lb] Test specifications and parallel performance (ParInt) for 2-loop box diagrams [4]; Speedup for p procs. $= T_1/T_p$

3-loop self-energy diagrams

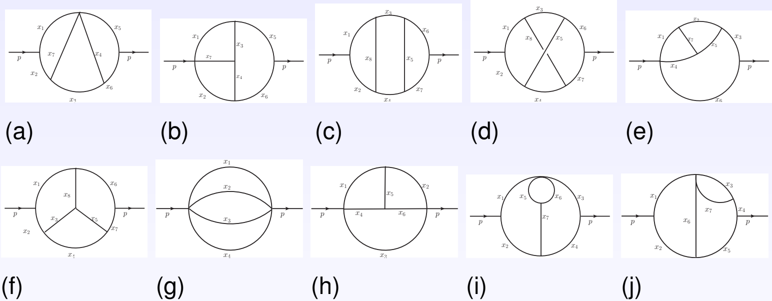


Figure : [3ls] Sample 3-loop self-energy diagrams with massive and massless internal lines (finite and UV-divergent diagrams), cf. Laporta [8], Baikov and Chetyrkin [1]: (a) $N = 7$, (b) $N = 7$, (c) $N = 8$, (d) $N = 8$, (e) $N = 7$, (f) $N = 8$, (g) $N = 4$, (h) $N = 6$, (i) $N = 7$, (j) $N = 7$

Diagrams [3ls] (i) and (j)

$N = 7,\ L = 3$ Massive internal lines [8]

$$I_i^{S3}(\varepsilon)\,\Gamma(1+\varepsilon)^{-3} \sim \sum_{k\geq -1} C_k \varepsilon^k$$

$$\sim 0.92363182652\,\varepsilon^{-1} - 2.423491634 + 8.38134971\,\varepsilon - 26.9936212\,\varepsilon^2 \ldots$$

$$I_j^{S3}(\varepsilon)\,\Gamma(1+\varepsilon)^{-3} \sim \sum_{k\geq -1} C_k \varepsilon^k$$

$$\sim 0.92363182652\,\varepsilon^{-1} - 2.116169719 + 6.92954468\,\varepsilon - 21.5032784\,\varepsilon^2 \ldots$$

Results for diagram [3ls] (i)

ℓ	INTEGRAL Fig [3ls] (i) E_a	T[s]	EXTRAPOLATION RESULT C_{-1}	RESULT C_0	RESULT C_1	RESULT C_2
17	2.3e-12	682.3				
18	4.2e-12	682.8	0.88532153209	-1.412495535		
19	6.0e-12	683.8	0.91647848504	-2.133365157	4.1493435	
20	5.2e-12	1027.6	0.92246162843	-2.356946186	6.9162602	-11.33999
21	2.2e-12	2415.8	0.92346458479	-2.410840073	7.9934802	-20.83203
22	3.6e-12	2419.5	0.92361089965	-2.421456174	8.2986081	-25.17454
23	5.7e-12	2425.8	0.92362953012	-2.423211188	8.3667190	-26.56843
24	8.7e-12	2431.6	0.92363160796	-2.423458644	8.3791875	-26.91297
25	1.3e-11	2437.1	0.92363180412	-2.423487620	8.3810333	-26.97920
26	1.7e-11	2445.7	0.92363182397	-2.423491206	8.3813166	-26.99204
		Exact:	0.92363182652	-2.423491634	8.3813497	-26.99362

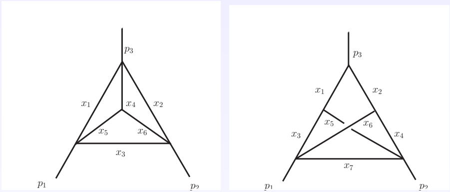
Table : Results 3-loop UV self-energy integral (on 4 nodes with 16 procs per node of *thor* cluster), err. tol. $t_r = 10^{-13}$, $T[s]$ = Time (elapsed user time in s); $\varepsilon = \varepsilon_\ell = 1.15^{-\ell}$, $\ell = 17, 18, \ldots,$ $E_r =$ integration estim. rel. error

Results for diagram [3ls] (j)

ℓ	INTEGRAL Fig [3ls] (j) E_a	T[s]	EXTRAPOLATION RESULT C_{-1}	RESULT C_0	RESULT C_1	RESULT C_2
5	7.1e-14	915.1				
6	5.7e-14	668.8	0.79556987526	-0.624745066		
7	6.4e-14	760.4	0.88129897281	-1.356849184	1.5364029	
8	3.9e-13	1444.9	0.91188370051	-1.809949406	3.7234312	-3.43942
9	2.2e-13	2104.7	0.92097424134	-2.018776382	5.4720364	-9.76427
10	2.4e-12	2018.5	0.92314713425	-2.091734298	6.4193304	-15.70840
11	7.0e-12	1978.3	0.92356124561	-2.111347543	6.7915499	-19.33007
12	1.3e-11	2288.4	0.92362368266	-2.115423673	6.9006678	-20.88190
13	3.2e-11	2309.7	0.92363108618	-2.116079491	6.9248672	-21.15424
14	6.7e-11	2331.9	0.92363177196	-2.116161010	6.9289487	-21.36736
15	1.4e-10	2534.4	0.92363182585	-2.116169538	6.9295216	-21.50172
		Exact:	0.92363182652	-2.116169718	6.9295447	-21.50328

Table : Results 3-loop UV self-energy integral (on 4 nodes with 16 procs per node of *thor* cluster), err. tol. $t_r = 10^{-13}$, $T[s]$ = Time (elapsed user time in s); $\varepsilon = \varepsilon_\ell = 1.3^{-\ell}$, $\ell = 5, 6, \ldots,$ E_r = integration estim. rel. error

Sample 3-loop vertex diagrams



(a) (b)

Figure : [3lv] 3-loop vertex (UV-divergent) diagrams with massless internal lines: (a) $N = 6$ (Heinrich et al. [7] Diag $A_{6,2}$), (b) $N = 7$ (Heinrich et al. [7] Diag $A_{7,5}$)

3-loop vertex diagram [3lv] (a)

$N = 6,\ L = 3$ - Extrapolation applied to $\Gamma(1-\varepsilon)^3 I_a^{V3}$, where

$$I_a^{V3} = \Gamma(N - nL/2)(-1)^N \int_0^1 \prod_{r=1}^N dx_r\, \delta(1 - \sum x_r)\, \frac{C^{N-n(L+1)/2}}{D^{N-nL/2}}$$

$$= \mathcal{I}_{6,3} = \Gamma(3\varepsilon) \int_{\mathcal{S}_5} \frac{C^{-2+4\varepsilon}}{D^{3\varepsilon}}$$

$$\Gamma(1-\varepsilon)^3 I_a^{V3} \sim \sum_{k \geq -1} C_k\, \varepsilon^k \approx -2.4041138063\, \frac{1}{\varepsilon} - 25.42515557 - 183.204184\, \varepsilon + \ldots$$

	3-loop vertex Integral I_a^{V3}			Extrapolation		
ℓ	E_r	T[s]	Res. C_{-1}	Res. C_0	Res. C_1	Res. C_2
20	6.9e-09	171.6				
21	4.7e-09	171.6	-2.2674024847	-36.22875015		
22	3.8e-09	171.2	-2.4184908276	-23.48554838	-26.647842	
23	4.2e-09	173.2	-2.4030392631	-25.64179665	-16.727996	-150.451
24	4.0e-09	174.5	2.4041730407	-25.40846949	-185.040255	-911.900
25	3.5e-09	175.9	-2.4041116670	-25.42597896	-183.074773	-1020.406
26	3.8e-09	177.3	-2.4041138550	-25.42514605	-183.204386	-1009.854
		Exact:	-2.4041138063	-25.42515557	-183.204184	-1009.791

Table : Results 3-loop UV vertex integral I_a^{V3} by ParInt on 4 nodes/16 procs. per node of *thor* cluster (in *long double* precision), $t_r = 10^{-13}$, max. # evals = 30B

3-loop vertex diagram [3lv] (a) Mellin-Barnes

Using Heinrich et al. [7] 2-fold Mellin-Barnes representation:

$$\Gamma^3(1-\varepsilon) I_a^{V3} = -\frac{\Gamma^3(1-\varepsilon)\,\Gamma(3\varepsilon)\,\Gamma^2(1-3\varepsilon)}{\Gamma(1-2\varepsilon)\,\Gamma(2-4\varepsilon)} \int_{c_1-i\infty}^{c_1+i\infty} \frac{dw_1}{2\pi i} \int_{c_2-i\infty}^{c_2+i\infty} \frac{dw_2}{2\pi i}$$
$$\times \frac{\Gamma(-1+3\varepsilon-w_1)\,\Gamma(-1+2\varepsilon-w_1)\,\Gamma(2-4\varepsilon+w_1)\Gamma(-w_2)\,\Gamma(w_2-w_1)}{\Gamma(3\varepsilon-w_1)\,\Gamma(2-4\varepsilon+w_2)\,\Gamma(2-4\varepsilon+w_1-w_2)}$$
$$\times\, \Gamma(1-\varepsilon+w_2)\,\Gamma(1-\varepsilon-+w_1-w_2)\,\Gamma(1-2\varepsilon+w_2)\,\Gamma(1-2\varepsilon+w_1-w_2)$$

where $c_1 = -6/5,\ c_2 = -1/2,\ -1/15 < \varepsilon < 3/20$ (to insure that the contours separate left poles and right poles of Γ functions in the complex plane).
We use a transformation $w_1 = c_1 + i\,t_1$ and $w_2 = c_2 + i\,t_2$ resulting in an integral over $\mathbb{R}^2$, i. e., $\int_{c_1-i\infty}^{c_1+i\infty} \frac{dw_1}{2\pi i} \int_{c_2-i\infty}^{c_2+i\infty} \frac{dw_2}{2\pi i} \to \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dt_1\, dt_2$ and brings complex arguments in the *Gamma*-functions of the integrand, e.g., $\Gamma(-1+3\varepsilon-w_1) \to \Gamma(-1+3\varepsilon-c_1-i\,t_1)$.

Finally a transformation $t_1 = \tan(x_1)$ with $dt_1 = dx_1/\cos^2(x_1)$ and $t_2 = \tan(x_2)$ with $dt_2 = dx_2/\cos^2(x_2)$ maps $\mathbb{R}^2 \to (-\pi/2, \pi/2) \times (-\pi/2, \pi/2)$. We can use DQAGS×DQAGS from QUADPACK [10] to approximate the integral numerically.

Extrapolation results 3-loop vertex diagram [3lv] (a) Mellin-Barnes

$C_{-1} \approx -2.40411380631918857086,\ C_0 \approx -25.4251557486168080 79,\ C_1 \approx -183.204184197615870658,\ C_2 \approx -1009.79068071241986670$

Extrapolation Mellin-Barnes $\Gamma^3(1-\varepsilon)\ I_a^{V3} \sim \sum_{k\geq -1} C_k \varepsilon^k$

3-LOOP VERTEX INTEGRAL $\Gamma^3(1-\varepsilon)\ I_a^{V3}$ EXTRAPOLATION							
ℓ	E_r	T[s]	RES. C_{-1}	RES. C_0	RES. C_1	RES. C_2	RES. C_3
3	4.6e-10	0.123					
4	7.2e-09	0.065	2.71270658920750307	-124.29196699004			
5	7.1e-09	0.089	-3.26671737757164271	19.214208212662	-765.366267748		
6	6.4e-11	0.103	-2.34303998712649708	-32.511725652267	62.2486740911	-3783.382591	
7	3.2e-10	0.103	-2.40606427439256088	-24.948811180339	-220.100132861	88.82961836	-16521.43
8	7.2e-10	0.105	-2.40408459536474339	-25.439771579237	-180.823300949	-1168.029003	-433.6484
9	1.1e-09	0.101	-2.40411401541469472	-25.424943874060	-183.274814872	-999.9251910	-5454.348
10	1.3e-09	0.100	-2.40411380559384602	-25.425157052041	-183.203187071	-1010.075874	-4804.705
11	1.4e-09	0.100	-2.40411380632043015	-25.425155569821	-183.204191028	-1009.786734	-4842.949
12	1.5e-09	0.100	-2.40411380631917693	-25.425155574937	-183.204184058	-1009.790784	-4841.860
13	1.5e-09	0.100	-2.40411380631918892	-25.425155574852	-183.204184293	-1009.790511	-4842.006
		Exact:	-2.40411380631918857	-25.425155574862	-183.204184198	-1009.790681	—

Table : Results 3-loop UV vertex integral (by DQAGS×DQAGS on Mac-Pro in *double* precision), $t_r = 10^{-8}$, max. # subdivisions = 100 in each direction, $\varepsilon = 2^{-\ell}, \ell \geq 2$

3-loop vertex diagram [3lv] (b)

$N = 7,\; L = 3$ - Extrapolation applied to $\Gamma(1-\varepsilon)^3 I_b^{V3}$, where

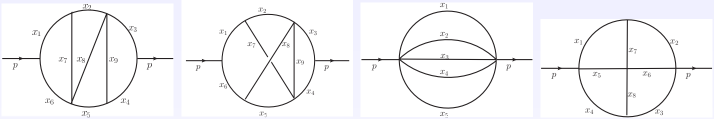
$$I_b^{V3} = \mathcal{I}_{7,5} = \Gamma(1+3\varepsilon)\int_{\mathcal{S}_6} \frac{C^{-1+4\varepsilon}}{D^{1+3\varepsilon}}$$

$$\Gamma(1-\varepsilon)^3 I_b^{V3} \sim \sum_{k\geq 0} C_k\, \varepsilon^k \approx 34.0969298 + 295.8700\,\varepsilon + \ldots$$

	3-loop vertex integral I_b^{V3}			Extrapolation
ℓ	E_r	T[s]	Res. C_0	Res. C_1
25	7.8e-07	667.5		
26	6.8e-07	667.4	33.8889738	338.4560
27	6.0e-07	667.5	34.1049542	293.1279
28	5.6e-07	667.4	34.0967447	295.9785
29	5.3e-07	667.4	34.0969222	295.8877
		Exact:	34.0969298	295.8700

Table : Results 3-loop UV vertex integral I_b^{V3} by ParInt on 4 nodes/16 procs. per node of *thor* cluster (in *long double* precision), $t_r = 10^{-13}$, max. # evals = 100B

Sample 4-loop self-energy diagrams



(a) (b) (c) (d)

Figure : [4ls] 4-loop self-energy diagrams with massless internal lines, cf., Baikov and Chetyrkin [1]: (a) $N = 9$, (b) $N = 9$, (c) *sunrise/sunset* $N = 5$, (d) *Shimadzu* $N = 8$

Results for 4-loop sunrise/sunset diagram

$$n(\varepsilon)^4 I_c^{S4} \sim -\frac{1}{576}\frac{1}{\varepsilon} - \frac{13}{768} - \frac{9823}{82944}\varepsilon + \ldots \approx -0.001736111111\frac{1}{\varepsilon} - 0.016927083333 - 0.118429301698\,\varepsilon + \ldots$$

where $n(\varepsilon) = \dfrac{\Gamma(2-2\varepsilon)}{\Gamma(1+\varepsilon)\,\Gamma(1-\varepsilon)^2}$ (Baikov and Chetyrkin [1])

ℓ	INTEGRAL I_c^{S4} E_a	T[S]	EXTRAPOLATION RESULT C_{-1}	RESULT C_0	RESULT C_1
8	3.9e-14	30.2			
9	3.8e-14	34.3	-0.001735179254977		
10	3.6e-14	34.1	-0.001736115881839	-0.016918557081	-0.012276556
11	4.1e-14	50.8	-0.001736111099910	-0.016927126297	-0.011837812
12	4.2e-14	58.5	-0.001736111111130	-0.016927083216	-0.011842959
13	1.5e-14	48.1	-0.001736111111109	-0.016927083381	-0.011842916
	e	mphExact:	-0.001736111111111	-0.016927083333	-0.011842930

Table : Results 4-loop UV *sunrise-sunset* integral [5], Fig [4ls] (c) (on 4 nodes/64 procs. *thor* cluster), err. tol. $t_a = 10^{-12}$, $T[s]$ = Time (elapsed user time in seconds); $\varepsilon = \varepsilon_\ell = 2^{-\ell}$, $\ell = 8, 9, \ldots,$ E_a = integration estim. abs. error [5]

Results for 4-loop Shimadzu diagram

$$n(\varepsilon)^4 I_d^{S4} \sim \frac{5\zeta_5}{\varepsilon} - 5\zeta_5 - 7\zeta_3^2 + \frac{25}{2}\zeta_6 + (35\zeta_5 + 7\zeta_3^2 - \frac{25}{2}\zeta_6 - 21\zeta_3\zeta_4 + \frac{127}{2}\zeta_7)\,\varepsilon + \ldots,$$

$$\sim \frac{5.184638776}{\varepsilon} - 2.582436090 + 70.39915145\,\varepsilon + \ldots$$

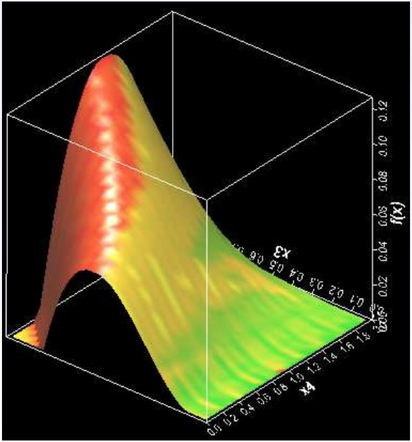
INTEGRAL I_d^{S4} ℓ	EXTRAPOLATION RESULT C_{-1}	RESULT C_0	RESULT C_1
10			
11	5.18460577	-2.47956688	
12	5.18463921	-2.58230627	70.1367604
13	5.18463922	-2.58243393	70.3982056
14	5.18463923	-2.58243413	70.3991764
Exact:	5.18463878	-2.58243609	70.3991515

Table : Results 4-loop UV *Shimadzu* integral [5], I_d^{S4}, Fig [4ls] (d), (on KEKSC 64 threads); $\varepsilon = \varepsilon_\ell = 2^{-\ell}$, $\ell = 10, 11, \ldots$; iteration time is below 20 min. [5]

Conclusions

- Methods from PARINT and QUADPACK are applied to Feynman loop integrals.
- Extrapolation can be used to treat singularities.
- The methods are fully numerical and viable without change across many problem types.
- New results using multivariate adaptive integration with PARINT include: Magdeburg diagram (2-loop finite with massive internal lines), 3-loop UV-divergent self-energy diagrams with massive internal lines, 3-loop finite and 3-loop UV-divergent vertex diagrams with massless propagators, 4-loop self-energy diagrams with massless internal lines.
- Iterated integration with QUADPACK [10] and extrapolation were applied to a Mellin-Barnes representation for a 3-loop UV-divergent vertex diagram with massless internal lines.

Questions?



Bibliography

Baikov, B. A., and Chetyrkin, K. G.
Four loop massless propagators: An algebraic evaluation of all master integrals.
Nuclear Physics B 837 (2010), 186–220.

Bulirsch, R.
Bemerkungen zur Romberg-Integration.
Numerische Mathematik 6 (1964), 6–16.

de Doncker, E., Shimizu, Y., Fujimoto, J., and Yuasa, F.
Computation of loop integrals using extrapolation.
Computer Physics Communications 159 (2004), 145–156.

de Doncker, E., and Yuasa, F.
Distributed and multi-core computation of 2-loop integrals.
XV Adv. Comp. and Anal. Tech. in Phys. Res., Journal of Physics: Conference Series (ACAT 2013) 523, 012052 (2014).
doi:10.1088/1742-6596/523/1/012052.

de Doncker, E., Yuasa, F., Kato, K., Ishikawa, T., Kapenga, J., and Olagbemi, O.
Regularization with numerical extrapolation for finite and uv-divergent multi-loop integrals, 2016.
In preparation.

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

Heinrich, G., Huber, T., and Maître, D.
Master integrals for fermionic contributions to massless three-loop form factors.
Physics Letters B 662 (2008), 344–352.

Laporta, S.
High-precision calculation of multi-loop Feynman integrals by difference equations.
Int. J. Mod. Phys. A 15 (2000), 5087–5159.
arXiv:hep-ph/0102033v1.

Li, S., Kaugars, K., and de Doncker, E.
Distributed adaptive multivariate function visualization.

International Journal of Computational Intelligence and Applications (IJCIA) 6, 2 (2006), 273–288.

PIESSENS, R., DE DONCKER, E., ÜBERHUBER, C. W., AND KAHANER, D. K.
QUADPACK, A Subroutine Package for Automatic Integration, vol. 1 of *Springer Series in Computational Mathematics*.
Springer-Verlag, 1983.

SHANKS, D.
Non-linear transformations of divergent and slowly convergent sequences.
J. Math. and Phys. 34 (1955), 1–42.

SIDI, A.
Practical Extrapolation Methods - Theory and Applications.
Cambridge University Press, 2003.
ISBN 0-521-66159-5.

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