

Automatic numerical methods for Feynman integrals through 3-loop

E. de Doncker¹, F. Yuasa², K. Kato³, T. Ishikawa³,
O. Olagbemi¹

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

²High Energy Accelerator Research Organization (KEK), Tsukuba, Japan

³Kogakuin University, Tokyo, Japan

2 Sept. 2014



Outline

1 Introduction

- Automatic Integration
- **PARINT**

2 Parallel multivariate integration

3 Feynman loop integrals

4 Ultraviolet (UV) singularities

5 Conclusions

Outline

1 Introduction

- Automatic Integration
- PARINT

2 Parallel multivariate integration

3 Feynman loop integrals

4 Ultraviolet (UV) singularities

5 Conclusions

Outline

1 Introduction

- Automatic Integration
- PARINT

2 Parallel multivariate integration

3 Feynman loop integrals

4 Ultraviolet (UV) singularities

5 Conclusions

Outline

1 Introduction

- Automatic Integration
- PARINT

2 Parallel multivariate integration

3 Feynman loop integrals

4 Ultraviolet (UV) singularities

5 Conclusions

Outline

1 Introduction

- Automatic Integration
- PARINT

2 Parallel multivariate integration

3 Feynman loop integrals

4 Ultraviolet (UV) singularities

5 Conclusions

Automatic integration

Black box approach

Obtain an approximation $Q(f)$ 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 .

Introduction - 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 SPRNG Pseudo-Random Number Generator (PRNG), *high dimensions, erratic integrand and/or domain*
- (1D) adaptive quadrature methods from QuadPack [12], 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, $\int_D f \approx \sum_k w_k f(\vec{x}_k)$: 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 memory systems)
- In **iterated integration**:
 - On the **rule** level: inner integrals are independent and computed in parallel, e.g., over subregion $S = D_1 \times D_2$ (inner region D_2) $\int_S F(\vec{x}) dx \approx \sum_k w_k F(\vec{x}_k)$, with $F(\vec{x}_k) = \int_{D_2} f(\vec{x}_k, \vec{y}) d\vec{y}$
 - Inner integrations can be performed **adaptively**.

Adaptive partitioning

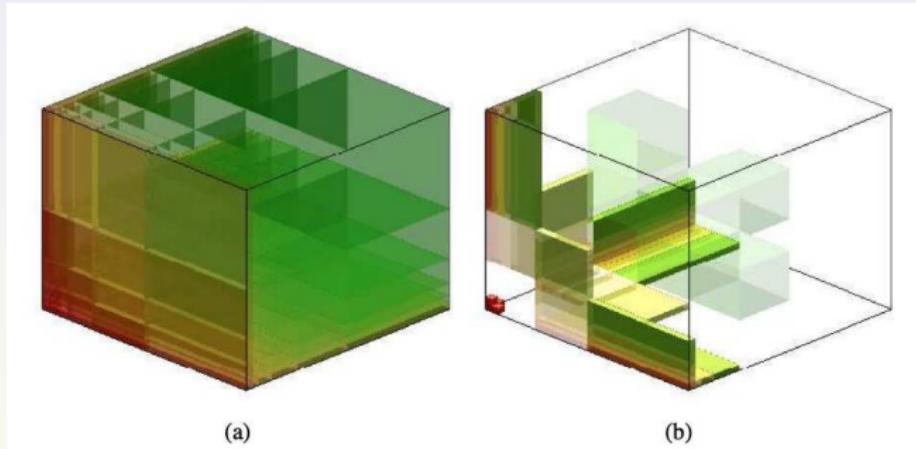


Figure : 1. Adaptive partitioning of the domain (singularity along axes) [11]

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:

$$(2D) \sum_k w_k f(x_k, y_k)$$

$$\text{Iterated (1D)}^2 \sum_i u_i \sum_j v_j f(x_i, y_j)$$

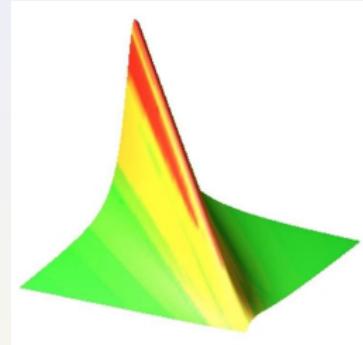
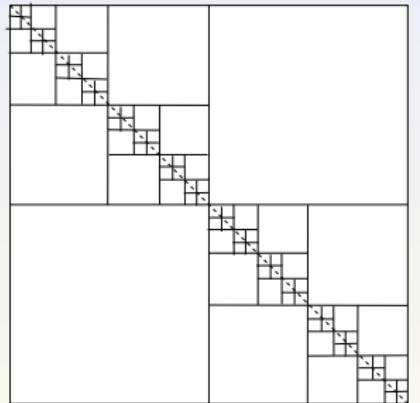


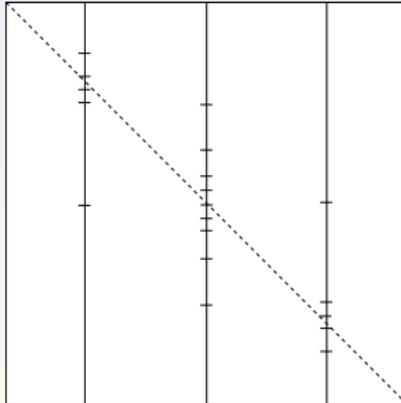
Figure :

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

Standard vs. iterated multivariate integration



(a)



(b)

Figure : 3. (a) Standard subdivision; (b) Iterated adaptive strategy for integrand with ridge on diagonal [10]

Feynman loop integrals

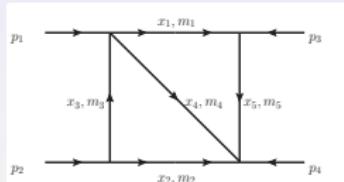
- Higher order corrections are required for accurate theoretical predictions of cross-sections for particle interactions. Loop diagrams are taken into account, leading to the evaluation of loop integrals.

L-loop integral with *N* internal lines

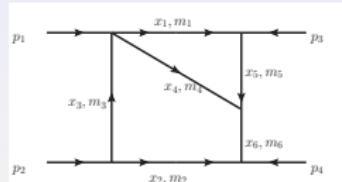
$$\mathcal{I} = \frac{\Gamma(N - \frac{nL}{2})}{(4\pi)^{nL/2}} (-1)^N \int_0^1 \prod_{r=1}^N dx_r \delta(1 - \sum x_r) \frac{C^{N-n(L+1)/2}}{(D - i\rho C)^{N-nL/2}} \quad (1)$$

- C* and *D* are polynomials determined by the topology of the corresponding diagram and physical parameters (*C* = 1 for 1-loop (*L* = 1) integrals).

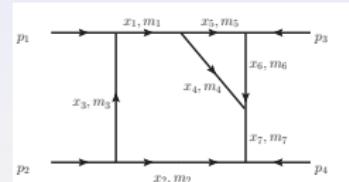
Sample two-loop diagrams



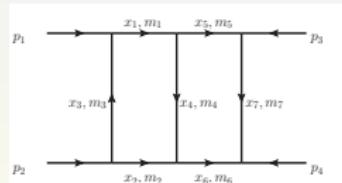
(a)



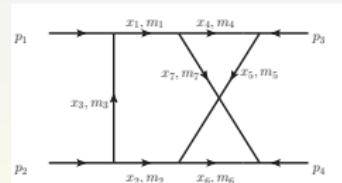
(b)



(c)



(d)



(e)

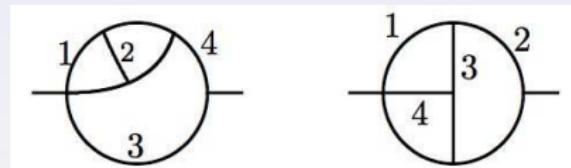
Figure : 4. 2-loop diagrams (a) $N = 5$ (double-triangle), (b) $N = 6$ (tetragon-triangle), (c) $N = 7$ (pentagon-triangle), (d) $N = 7$ ladder, (e) $N = 7$ crossed ladder [9, 4]

Parallel performance of PARINT adaptive integration for two-loop diagrams on MPI (OpenMPI)

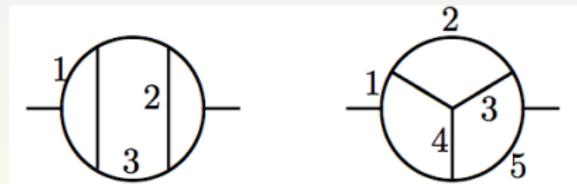
2-LOOP DIAG.	N	REL TOL t_r	MAX EVALS	$T_1[s]$	$T_{64}[s]$	SPEEDUP S_{64}
Fig 4 (a)	5	10^{-10}	400M	32.6	0.74	44.1
Fig 4 (b)	6	10^{-9}	3B	213.6	5.06	42.2
Fig 4 (c)	7	10^{-8}	5B	507.9	8.83	57.5
Fig 4 (d) ladder	7	10^{-8}	2B	189.9	4.33	43.9
Fig 4 (e) crossed	7	10^{-7}	300M	27.6	0.49	56.3
Fig 4 (e) crossed	7	10^{-9}	20B	1892.5	34.6	54.7

Table : 1. Test specifications and parallel performance (PARINT) for 2-loop diagrams [4]

Sample 3-loop self-energy diagrams



(a)



(b)

Figure : 5. 3-loop diagrams (a)(L) $N = 7$ (Laporta[9] Fig 2(q)), (a)(R) $N = 7$ ((Laporta[9] Fig 2(r)), (b)(L) $N = 8$ (Laporta[9] Fig 2(t)), (b)(R) $N = 8$ (Laporta[9] Fig 2(u))

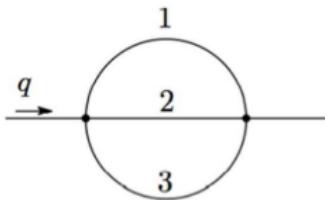
Parallel performance of PARINT integration for 3-loop diagrams on MPI (OpenMPI)

3-loop diag.	N	Result $p = 1$	Result $p = 64$	$T_1[s]$	$T_{64}[s]$	S_{64}
Fig 5 (a)(L), L2(q)	7	1.3264481	1.3264435	529.8*	7.90*	67.1
Fig 5 (a)(R), L2(r)	7	1.34139923	1.34139917	431.6	8.14*	53.0
Fig 5 (b)(L), L2(t)	8	0.27960890	0.2796084	504.3*	7.84*	64.3
Fig 5 (b)(R), L2(u)	8	0.18262722	0.18262720	423.6	8.17*	51.8

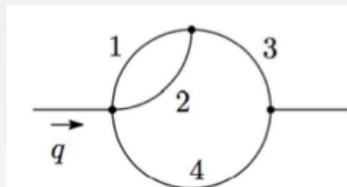
Table : 2. Parallel performance of PARINT for 3-loop diagrams of Fig 5, Abs. tol.

$t_a = 5 \times 10^{-8}$, Max evals = 5B was reached in starred(*) cases, T_{64} is parallel time on *thor* cluster with $p = 64$ processes (distributed over four 16-core nodes), speedup $S_{64} = T_1 / T_{64} = (\text{sequential time}) / (\text{parallel time})$. Note that (1) Timing is done within PARINT, after the processes are started. (2) Superlinear speedup (S_{64}) is obtained in cases where both the sequential and parallel runs reach the user-specified maximum on the number of integrand evaluations (5B); however the adaptive partitioning is somewhat more effective sequentially.

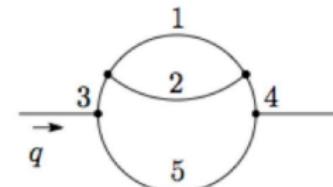
Ultraviolet (UV) singularities



(a)



(b)



(c)

Figure : 6. UV diagrams: (a) *sunrise-sunset* (Laporta [9] Fig 2(b)), (b) *"lemon"* (Laporta [9] Fig 2(c)), (c) *"half-boiled egg"* (Kato [7] Fig 2)

Ultraviolet (UV) singularities - Integral

General form

$$\mathcal{I} = \frac{\Gamma\left(N - \frac{nL}{2}\right)}{(4\pi)^{nL/2}} (-1)^N \int_0^1 \prod_{r=1}^N dx_r \frac{\delta(1 - \sum x_r)}{U^{n/2}(V - i\rho)^{N-nL/2}} \quad (2)$$

Compared to the notation in Eq. (1), the functions C and D correspond to $C = U$ and $D = UV$.

sunrise-sunset diagram 6 (a): $N = 3, L = 2, n = 4 - 2\varepsilon$

$$\text{Let } I_s = (-1)\Gamma(-1 + 2\varepsilon) \int dx_1 dx_2 dx_3 \delta(1 - \sum_r x_r) \frac{(V - i\rho)^{1-2\varepsilon}}{U^{2-\varepsilon}}$$

$$V = \sum_r x_r m_r^2 - \frac{s}{U} W, \quad s = q^2, \quad s = 1, \quad \text{all masses } m_r = 1$$

$$U = x_1 x_2 + x_2 x_3 + x_3 x_1, \quad W = x_1 x_2 x_3$$

Ultraviolet (UV) singularities - Integral

$$V = \sum_r x_r m_r^2 - \frac{s}{U} W, \quad s = q^2$$

lemon diagram 6 (b): $N = 4, L = 2, n = 4 - 2\varepsilon$

$$I_l = \Gamma(2\varepsilon) \int dx_1 dx_2 dx_3 dx_4 \delta(1 - \sum_r x_r) \frac{1}{U^{2-\varepsilon} (V - i\rho)^{2\varepsilon}}$$

$$U = x_{12}x_{34} + x_1x_2, \quad W = x_4(x_1x_2 + x_2x_3 + x_3x_1)$$

half-boiled egg diagram 6 (c): $N = 5, L = 2, n = 4 - 2\varepsilon$

$$I_h = (-1)\Gamma(1 + 2\varepsilon) \int dx_1 dx_2 dx_3 dx_4 dx_5 \delta(1 - \sum_r x_r) \frac{1}{U^{2-\varepsilon} (V - i\rho)^{1+2\varepsilon}}$$

$$U = x_{12}x_{345} + x_1x_2, \quad W = x_5(x_{12}x_{34} + x_1x_2)$$

Note: $s = 1$, all masses $m_r = 1$

Analytic results for the integrals are studied by many authors
(see also Kato [7]).

Iterated numerical integration: using **DQAGS** from **QUADPACK** [12, 2].

Singularities - extrapolation/convergence acceleration methods

- (i) If denominator vanishes in interior of the integration domain: integral calculated in the limit as $\varrho \rightarrow 0$
- (ii) Integral with infrared (IR) singularity ($n = 4 + 2\varepsilon$ in (1)): calculated in the limit as $\varepsilon \rightarrow 0$ (dimensional regularization) [5]
- (iii) Integral with ultraviolet (UV) singularity ($n = 4 - 2\varepsilon$): calculations in the limit as $\varepsilon \rightarrow 0$
e.g., sunrise-sunset integral [9]:
 $I_s(\varepsilon) \Gamma(1 + \varepsilon)^{-2} = -1.5\varepsilon^{-2} - 4.25\varepsilon^{-1} - 7.375 - 17.22197253479\varepsilon\ldots$

Asymptotics/Mechanics of extrapolation

Extrapolation (is tailored to an underlying asymptotic expansion):

Linear extrapolation for

$$S(\varepsilon) \sim S + c_1\varphi_1(\varepsilon) + c_2\varphi_2(\varepsilon) + \dots, \text{ as } \varepsilon \rightarrow 0,$$

create sequences of $S(\varepsilon_\ell)$ such that

$$S(\varepsilon_\ell) = c_0 + c_1\varphi_1(\varepsilon_\ell) + \dots c_\nu\varphi_\nu(\varepsilon_\ell), \quad \ell = 1, \dots, \nu + 1.$$

Solve linear systems of orders $(\nu + 1) \times (\nu + 1)$, for increasing values of ν and decreasing $\varepsilon = \varepsilon_\ell$.

For example, $\varphi_k(\varepsilon) = \varepsilon^k$.

Bulirsch [1] type sequences can be used for a sequence of the form

$$\varepsilon_\ell = 1/b_\ell, \quad b_\ell = 2, 3, 4, 6, 8, 12, 16, 24, \dots$$

Non-linear extrapolation with the ϵ -algorithm can be applied with geometric sequences, e.g., $\varepsilon_\ell = 1.2^{-\ell}$.

Asymptotic expansions for UV integrals

sunrise-sunset integral expansion [9]:

$$I_s(\varepsilon) \Gamma(1 + \varepsilon)^{-2} = \sum_{k \geq -2} C_k \varepsilon^k = \\ -1.5\varepsilon^{-2} - 4.25\varepsilon^{-1} - 7.375 - 17.22197253479\varepsilon\dots$$

lemon integral expansion [9]: $I_l(\varepsilon) \Gamma(1 + \varepsilon)^{-2} = \sum_{k \geq -2} C_k \varepsilon^k =$

$$0.5\varepsilon^{-2} + 0.6862006357658\varepsilon^{-1} - 0.6868398873414 + 1.486398391913\varepsilon\dots$$

half-boiled egg integral expansion [7]: $I_h(\varepsilon) = \Gamma(1 + 2\varepsilon) \sum_k C_k \varepsilon^k$, where

$$C_{-1} = J_1 = \int_0^1 \frac{\rho'}{M_0^2 - sG_0} d\rho', \quad C_0 = -\frac{3}{2}J_1 - 2J_2 + I_B \text{ with}$$

$$J_2 = \int_0^1 \frac{\rho' \log(M_0^2 - sG_0)}{M_0^2 - sG_0} d\rho'$$

$$I_B = \int_0^1 d\rho \int_0^1 d\xi \int_0^1 d\rho' (1 - \rho)^2 \rho' \frac{M_0^2 - F^2 M^2 - s(G_0 - FG)}{\rho F(FM^2 - sG)(M_0^2 - sG_0)}$$

Here $s = 1$, $F = 1 - \rho + \rho\xi(1 - \xi)$, $F_0 = F(\rho = 0) = 1$,

$G = (1 - \rho)(1 - \rho')((1 - \rho)\rho' + \rho\xi(1 - \xi))$, $G_0 = G(\rho = 0) = (1 - \rho')\rho'$,

$M^2 = \sum_r x_r m_r^2$, $M_0^2 = M^2(\rho = 0) = \rho' m_3^2 + (1 - \rho')m_5^2$ (pending $m_3 = m_4$),

and $M_0^2 = M^2 = 1$ (in view of $m_r = 1$, $1 \leq r \leq 5$)

Asymptotic expansions for UV integrals

Numerical approximations for half-boiled egg integral expansion: Let $m_r^2 = 1$ and $s = 1$.

Then $J_1 = 0.6045997880780727$ with abs. error estimate $1.55e-15$,
and $J_2 = -0.1170816559877808$ with abs. error estimate $5.85e-18$,
obtained with DQAGE from QUADPACK [12, 2].

For I_B we find, with iterated integration by DQAGS (used in view of boundary singularities):

$I_B \approx 0.4970393699155826$ (with abs. err. est. $E_a = 1.22 \times 10^{-15}$).

Then, $C_0 = -\frac{3}{2}J_1 - 2J_2 + I_B \approx -0.175697000225964850$.

C_{-1} and C_0 computed according to the analytic formulas agree with the extrapolation results!

Using iterated integration with DQAGS (to rel. tol. $t_r = 10^{-12}$) and linear extrapolation based on a Bulirsch sequence we obtain the coefficients:

$C_{-1} = J_1 \approx 0.604599788078210687$

$C_0 \approx -0.175697000334047643; C_1 \approx -0.2977242; C_2 \approx 0.414015$

Numerical (linear) extrapolation results **lemon** integral

b_ℓ	INTEGRAL I_ℓ		EXTRAPOLATION			
	E_r	T(s)	RESULT C_{-2}	RESULT C_{-1}	RESULT C_0	RESULT C_1
4	3.5e-11	0.36				
6	8.8e-11	0.34	0.5130221162587	0.52467607220		
8	2.9e-12	0.34	0.5031467341833	0.62342989295	-0.237009170	
12	3.4e-12	0.40	0.5004379328119	0.67218831764	-0.518724512	0.5200899
16	1.5e-11	0.41	0.5000485801347	0.68386889795	-0.643317369	1.0807577
24	4.7e-11	0.39	0.5000037328535	0.68593187289	-0.679195194	1.3749558
32	4.1e-11	0.38	0.5000002195177	0.68617780639	-0.685884585	1.4654594
48	1.4e-11	0.43	0.5000000087538	0.68619930431	-0.686757991	1.4837301
64	1.3e-11	0.44	0.5000000002937	0.68620057333	-0.686834471	1.4861463
96	3.2e-11	0.31	0.5000000000039	0.68620063534	-0.686839872	1.4863967
Analytic:		0.5	0.68620063577	-0.686839887	1.4863984	

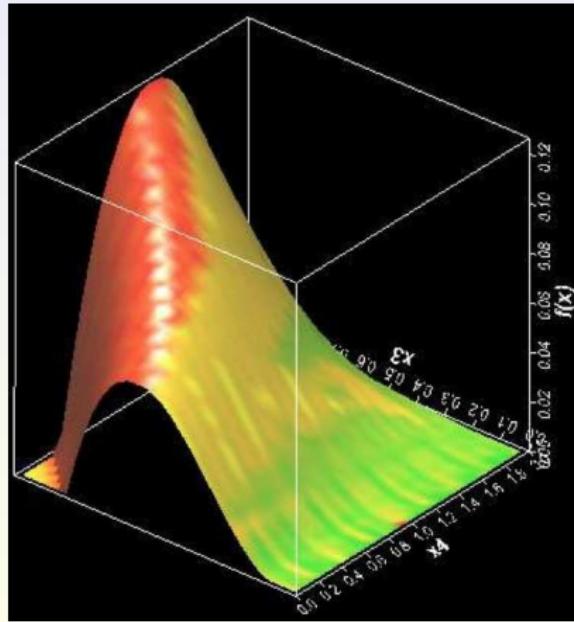
Table : 3. Results UV *lemon* integral (on Mac Pro, 2.6GHz Intel Core i7, 16GB memory, OS X), tolerances $t_r = 10^{-10}$ (outer), 5×10^{-11} (inner two), $T(s) = \text{Time (s)}$; $\varepsilon = 1/b_\ell$ (starting at 0.25), $E_r = \text{outer integration estim. rel. error}$

Conclusions

Extensions to PARINT have been/are being developed for incorporation in the package:

- A **multicore** technique for individual integral parallelization on the function evaluation level, for iterated integration, which is successful in many cases with singularities, where all other automatic methods fail. The parallel version preserves the procedure of the sequential version exactly, i.e., the parallel runs give the same results when repeated.
- CUDA is used to implement **large-scale function sampling on GPUs** (so far MC [3]; to be done for QMC), including error estimate and condition number.
- Summations of very large numbers of terms require special summation techniques (such as **Kahan summation** [8, 6, 3]) to guard against roundoff error (needed not only in MC, but also in adaptive code).
- New results include **3-loop self-energy** diagrams without IR or UV singularities, and **2-loop self-energy** diagrams (with 3, 4 and 5 internal lines) with **UV terms**. The latter are computed with **iterated integration** using **DQAGS** from **QUADPACK**. DQAGS treats types of end-point singularities.

Questions?



BIBLIOGRAPHY



[1] BULIRSCH, R.

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



[2] DE DONCKER, E., FUJIMOTO, J., HAMAGUCHI, N., ISHIKAWA, T., KURIHARA, Y., SHIMIZU, Y., AND YUASA, F.
Quadpack computation of Feynman loop integrals.

Journal of Computational Science (JoCS) 3, 3 (2011), 102–112.
doi:10.1016/j.jocs.2011.06.003.



[3] DE DONCKER, E., KAPENGA, J., AND ASSAF, R.

Monte carlo automatic integration with dynamic parallelism in cuda.
In *Numerical Computations with GPUs* (2014), Springer.
ISBN 978-3-319-06548-9.



[4] 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.



[5] DE DONCKER, E., YUASA, F., AND KURIHARA, Y.

Regularization of IR-divergent loop integrals.

Journal of Physics: Conf. Ser. 368, 012060 (2012).



[6] HIGHAM, N. J.

Accuracy and Stability of Numerical Algorithms 2nd ed.

SIAM, 2002.

ISBN 978-0-898715-21-7.



[7] KATO, K.

Note on 2-loop self-energy scalar integrals.

Tech. rep., Department of Physics, Kogakuin University, Shinjuku, Tokyo
163-8677, Japan, July 2010.

Private communication.



[8] KNUTH, D. E.

The Art of Computer Programming, Volume 2, Seminumerical Algorithms,
third ed.

Addison Wesley, 1998.



[9] 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.



[10] LI, S.

Online Support for Multivariate Integration.

PhD dissertation, Western Michigan University, December 2005.



[11] 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.



[12] PIJSSENS, 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.