High Performance and Increased Precision Techniques for Feynman Loop Integrals

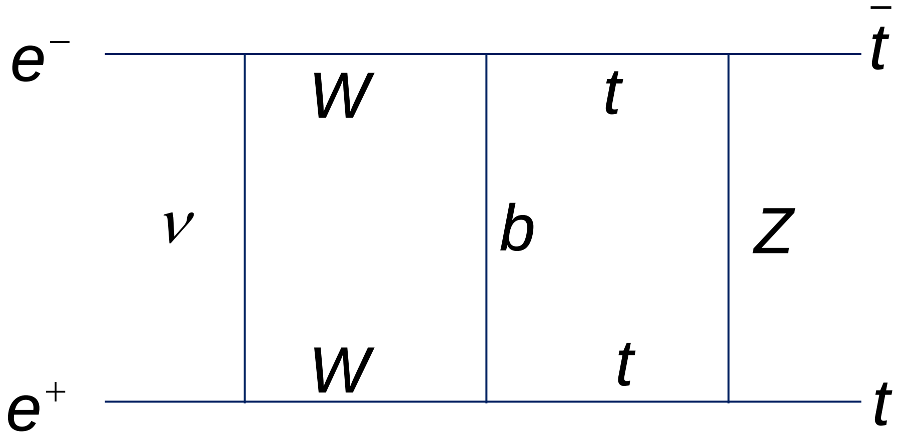
ACAT2016(UTFSM, Valparaiso, 2016.01.21)

K. Kato, E. de Doncker, T. Ishikawa,
J. Kapenga, O. Olagabemi and F.Yuasa

How to handle the large scale computation in QFT?

- Automated systems to perform perturbative calculation in QFT have been developed to handle cases beyond man-power computation.
- Many systems are successfully working in tree and 1-loop level. GRACE, CompHEP, CalcHEP, FeynArt/Calc, FDC,...
- Extension to higher-loop calculation for the future HEP(high accuracy, multi-body states).

- Next generation of systems include general multi-loop calculation library.
- Formulae for 2-loop integrals are given for many cases: However, it seems to be difficult to write ‘general solution’ .
  → Numerical methods
- 2-loop EW, 1-loop EW/MSSM+1-loop QCD,…



In the analytic method, one needs sophisticated variable transformation. In this analysis, the integrand is used ‘*as it is*’.

Target: multi-loop integral(scalar)

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

U,V Polynomials of x's: V depends on masses and momenta

N= number of propagators

L= number of loops

Space-time dimension

$n = 4 - 2\varepsilon$

Numerator function can be introduced.

Singularities of the integral in numerical method

$\rho \to 0$ Numerically danger, analytically integrated

The validity of this regularization is already reported.

$\varepsilon \to 0$ UV(IR) divergence

IR handling is already published. Here, we mainly discuss UV poles.

DCM: Direct computation method

DCM= regularized integration
+ sequence extrapolation
(or linear solver)

- Calculate the integral with finite ε (or ρ) ····· $I(\varepsilon_j)$

For finite values, the integral is convergent numerically.

geometrical

$$\varepsilon_j = \frac{\varepsilon_0}{A^j} \quad 1 < A < 2$$

Bulirsch type sequence

$$\varepsilon_j = \frac{\varepsilon_0}{b_j} \quad b_j = 2, 3, 4, 6, 8, 12, 16, 24, \ldots.$$

- Estimate the integral by extrapolation $I = \lim_{\varepsilon \to 0} I(\varepsilon_j)$

Sometimes, one can put $\varepsilon = 0$. No need for extrapolation.

extrapolation

$$I(\varepsilon) = \cdots + \frac{C_{-2}}{\varepsilon^2} + \frac{C_{-1}}{\varepsilon} + C_0 + C_1\varepsilon + C_2\varepsilon^2 + \cdots$$

After integration

$\{I(\varepsilon_j)\}$

$j = 0, \ldots n$

If you want C_{-1}, apply Wynn to $\varepsilon_j I(\varepsilon_j)$

If you want C_0, apply Wynn to $I(\varepsilon_j)$

:

Wynn's algorithm (Math. magic)

$$a(j, k+1) = a(j+1, k-1) + \frac{1}{a(j+1,k) - a(j,k)}$$

Input

$a(j,0) = I(\varepsilon_j), \quad a(j,-1) = 0$

geometric ε_j

Numerical integration

Numerical integration packages

DQ … DQAGE/DQAGS routine in Quadpack package
(http://www.netlib.org/quadpack/)

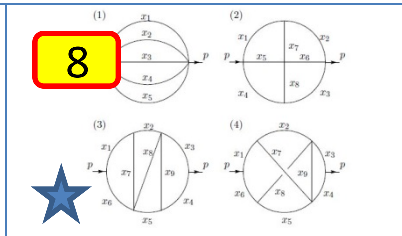
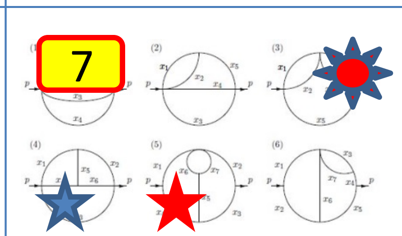
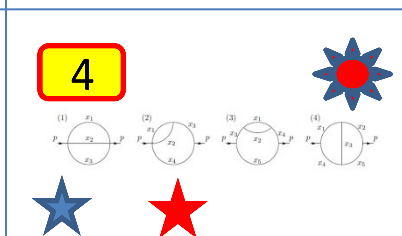
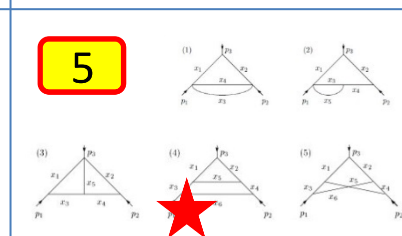
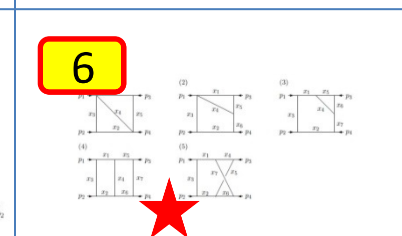
ParInt package … Adaptive method
(https://cs.wmich.edu/parint/)

DE … Double exponential formula
(http://www.sciencedirect.com/science/article/pii/S037704270000501X)

Parallel computing in multi-core environment

MPI(Message Passing Interface) … distributed memory
OpenMP(Multi-Processing) … shared memory

Status of DCM

4			
3			
2			
Loops	Self energy	Vertex	Box

massless

massive

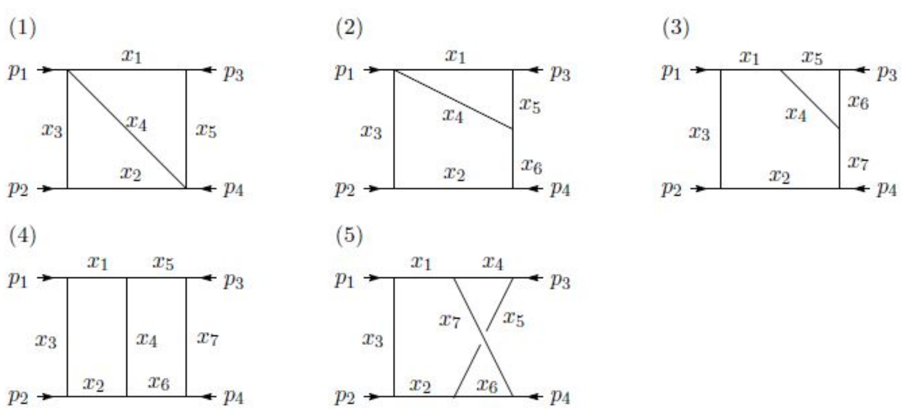
UV divergence in integral part

n (computed) dimension of integral

2-loop box

Finite

	Dim	Rel. Tol.	Max. eval.	T1[s]	T64[s]	T1/T64 (speedup)
(1)	5	10^{-10}	400M	32.6	0.74	44.1
(2)	6	10^{-9}	3B	213.6	5.06	42.2
(3)	7	10^{-8}	5B	507.9	8.83	57.5
(4)	7	10^{-8}	2B	189.9	4.33	43.9
(5)	7	10^{-7}	300M	27.6	0.49	56.3
	7	10^{-9}	20B	1893	34.6	54.7



$p^2=1$, $m=1$, $s=t=1$

By ParInt

thor cluster, MPI

Reported ACAT2013 for comparison with Laporta work and the convergence for $\varepsilon \to 0$

2-loop vertex

UV-div. up to 2nd order

(1) $\varepsilon_j = 1/b_j$

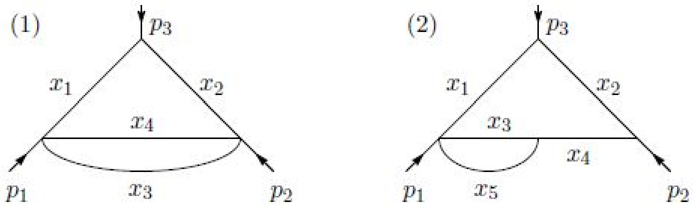
b_j	T[s]	C_{-2}	C_{-1}	C_0	C_1
3	1.4				
4	1.5	0.514890217	0.535680679		
6	1.8	0.505861627	0.598880809	-0.108343080	
8	5.2	0.501111606	0.660631086	-0.364844229	0.342001531
12	2.5	0.500159017	0.680635463	-0.515353351	0.822106580
16	4.2	0.500017647	0.685300679	-0.573315125	1.161395016
24	20.5	0.500001357	0.686098882	-0.588595022	1.307352233
32	19.1	0.500000078	0.686192225	-0.591298134	1.347594589
48	8.6	0.500000001	0.686200327	-0.591641476	1.355242242
Analytic		0.5	0.686200636	-0.591666701	1.356196533

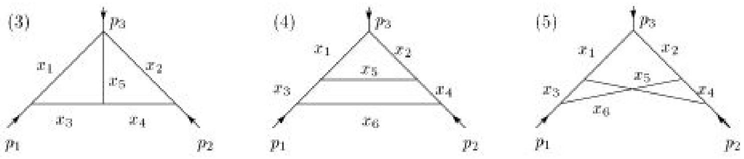
Linear extrapolation

$p^2=1$, $m=1$

(2) Is also computed for the most divergent term C_{-1}= 0.6712531728 $\pm 0.344 \times 10^{-7}$

Laporta 0.67125310574



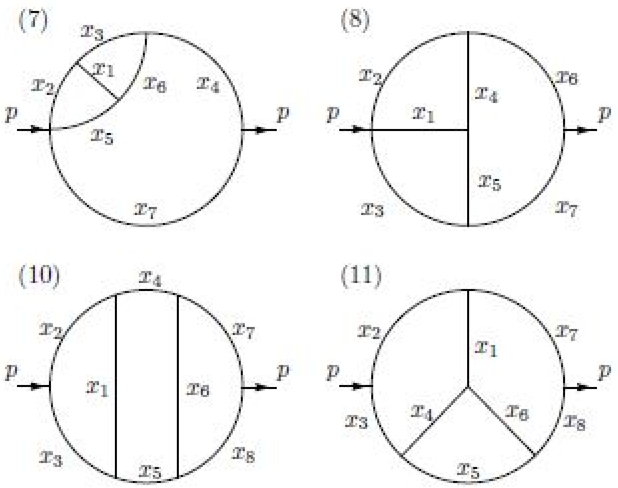


Finite, Reported in ACAT2011

3-loop self-energy

Finite integrals ,no extrapolation

	dim	Result, p=1	Result, p=64	T(1) [s]	T(64) [s]	T1/T64
(7)	7	**1.32644**81	**1.32644**35	529.8	7.90	67.1
(8)	7	**1.341399**23	**1.341399**17	431.6	8.14	53.0
(10)	8	**0.279608**90	**0.279608**4	504.3	7.84	64.3
(11)	8	**0.182627**22	**0.182627**20	423.6	8.17	51.8



Comparison with Laporta(s=1, m=1)

Absolute tolerance= 5 x 10^{-8},
Max evaluations = 5B,
T64 on *thor* cluster with p = 64 processes
(distributed over four 16-core nodes)

3-loop self-energy

UV-div. (up to 3rd order)

E5-2687W v3
@ 3.10GHz
Quadraple prec.
40 thread

(5) Ladybug

Laporta (s=1, m=1)

$$0.923631826\frac{1}{\varepsilon}-2.42349163\ +\cdots$$

DCM (DE)

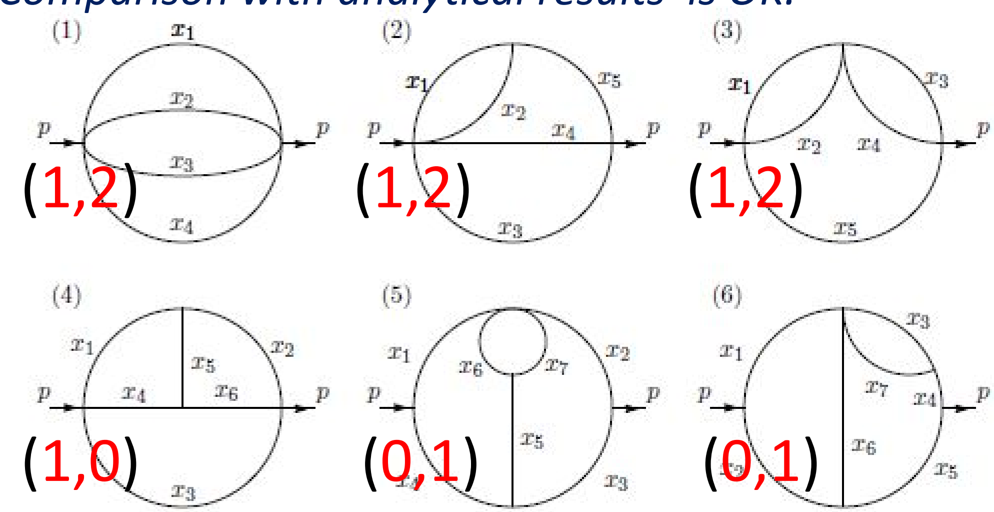
$C_{-1}= 0.92370 \pm 0.434 \times 10^{-3}$

$C_0=-2.4201 \pm 0.424 \times 10^{-1}$

Neval	mesh	Elapsed (Ks)	CPU(Ms)
105	0.1265988	65	2.1
95	0.1253191	37	1.2
83	0.1267232	18	0.57

6-dim.
Max eval
$=(10^2)^6$
$=10^{12}$

Other diagrams are also computed.
Comparison with analytical results is OK.



Divergence order
(Gamma, Integral)

extrapolation

0.49493857234
0.53603234731
0.57465585827
0.6106239846
0.6438487350
0.6743208746
0.702092730
0.7272628131
0.749962533
0.7703450682
0.7885762731
0.804827478
0.819269932
0.832070673
0.843389581
0.85337740
0.8621745
0.869910

linear solver

0.92370 0.9236528

4-loop self-energy

massless, finite and UV div.

$$BC = -0.001736111111111/\varepsilon - 0.016927083333 - 0.01184293016\varepsilon + \cdots$$

(1) -0.001736111111109 -0.016927083381 -0.011842916
Elapsed : 48s : ParInt, thor cluster 64 threads

(2) $BC = 5.184638776/\varepsilon - 2.582436090 + 70.39915145\varepsilon + \cdots$

BC

5.1846392 -2.582434 70.39877
Elapsed : 4.8h : DE, CPU KEKSC(SR-16000,64thread)

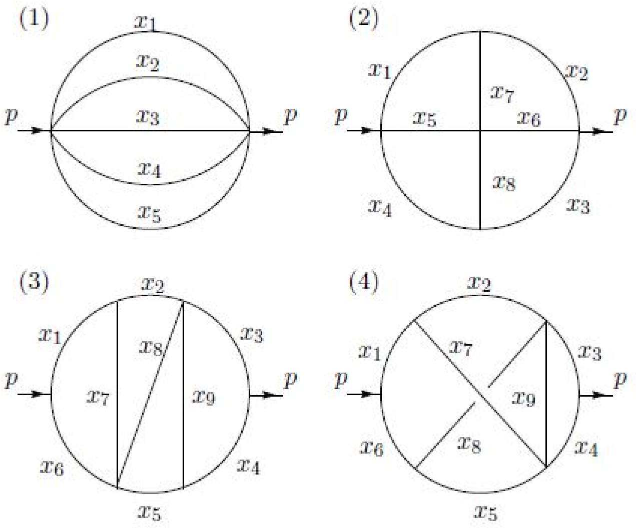
Analytic results: P.A. Baikov and K.G.Chetrykin NPB 837 (2010) 186-220

(3) $BC = 55.58525391567 + O(\varepsilon)$

55.585150 Elapsed : 554 s
eval. 300B, ParInt, thor cluster 64 threads

(4) $BC = 52.01786874361 + O(\varepsilon)$

52.017714 Elapsed : 659 s
eval.275B, ParInt, thor cluster 64 threads



summary

- Numerical method is tested to calculate the loop integrals up to 4-loops and to 4-point functions.
- DCM method works to estimate both UV-divergent and finite terms in dimensional regularization.
- Expecting to be a part of automated systems, no special handling of the integrand is not used in this approach. Automatic decision of the variable transformation might improve the status.
- The large CPU-time problem can be avoided using appropriate software and platform.
- Further study should be extended to the general cases with physical masses and external momenta including numerator part.


Thank you!