Computation of two-loop integrals with masses by numerical integration and extrapolation

F. Yuasa, T.Ishikawa, Y.Kurihara, N.Hamaguchi, J.Fujimoto and Y.Shimizu, /KEK

E. de Doncker/Western Michigan Univ.

K.Kato/Kogakuin Univ.

LoopFestX, 12-14 May 2011

Plan of talk

- Introduction
- Fully numerical approach
- Two-loop integral with masses
  - Numerical results (two-loop box integral)
  - Computation time
- Cross-check
- Summary

Introduction

- Automatic Computation System in HEP
  - tree level (just few examples)
    - GRACE, CompHEP, CalcHEP, FeynArts/FeynCalc/FormCalc, Madgraph, HELAC, fdc and so on
  - one-loop level (just few examples)
    - FeynArts/FeynCalc/FormCalc with LoopTools, GRACE-1loop, SloopS, xloop, Golem, DIANA, sanC, and so on
- two-loop and more level
  - Developments of tools or systems for multi-loop are ongoing
    - Analytical approach , AMBRE, MB.m and so on..
    - Numerical approach, this talk(up to two-loop level) and so on..

Scalar loop integral in Feynman parameter representation

$$(-1)^N \left(\frac{1}{4\pi}\right)^{nL/2} \Gamma(N - nL/2) \int_0^1 \prod_{i=1}^{N} dx_i \delta(1 - x_1 \cdots - x_N) \frac{C^{N-n(L+1)/2}}{(D - i\varepsilon C)^{N-nL/2}}$$

n: space-time dimension, N: # of internal particles, L: # of loops

D and C are polynomials of $x_1, x_2, \ldots, x_N$. For L=1, C=1.

$n = 4$

L	N	$C^{N-2(L+1)} / D^{N-2L}$
1	3	$1/(D - i\varepsilon)$
	4	$1/(D - i\varepsilon)^2$
	5	$1/(D - i\varepsilon)^3$
2	5	$1/C(D - i\varepsilon C)$
	6	$1/(D - i\varepsilon C)^2$
	7	$C/(D - i\varepsilon C)^3$

D and C are determined by the topology of the diagram

Fully numerical approach: Direct Computation Method

E. De Doncker, Y.Shimizu J.Fujimoto, FY Comput. Phys. Comm. 159('04)145.

In the analytic treatment, ε in the denominator is an infinitesimal number (in complex analysis) while we consider it a finite (sometimes rather large) number

$$I = \lim_{\varepsilon \to 0} \int_0^1 dx \int_0^{1-x} dy \frac{1}{D(x,y) - i\varepsilon}$$

Sometimes denominator becomes 0 in the integration region.

Change ε as $\varepsilon_l = \dfrac{\varepsilon_0}{c^l}, \quad c > 1$

Do integration

and get $I(\varepsilon_l)$, $\varepsilon_l > 0$, $l = 0,1,2,\ldots$

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

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

Extrapolate $I(\varepsilon_l)$

and get the result when ε becomes 0.

$$\Re(I) = \lim_{\varepsilon \to 0} \{\Re(I(\varepsilon_l))\},$$

$$\Im(I) = \lim_{\varepsilon \to 0} \{\Im(I(\varepsilon_l))\}$$

Numerical Integration

- DQAGE

R.Piessens E. De Doncker, C.W.Uberhuber, D.K.Kahaner;
" Quadpack – a subroutine package for automatic integration", Springer-Verlag, 1983

$$I = \int_a^b f(x)dx \approx \sum_{i=1}^{n} \omega_i f(x_i)$$

an adaptive quadrature routine where sampling points are chosen by Gauss-Kronrod quadrature rule

- Double Exponential formulae

H.Takahashi and M.Mori;
"Double Exponential Formulas for Numerical Integration",
Bull.R.I.M.S.,Kyoto Univ.,9,pp.721-741(1974).

$$I = \int_{-1}^{1} f(x)dx = \int_{-\infty}^{\infty} f(g(t))g'(t)dt \approx h \sum_{j=-N}^{N} \omega_j f(x_j)$$

$$x = g(t) \qquad g(t) = \tanh\left(\frac{\pi}{2}\sinh(t)\right) \qquad g'(t) = \frac{\frac{\pi}{2}\cosh(t)}{\cosh^2\left(\frac{\pi}{2}\sinh(t)\right)}$$

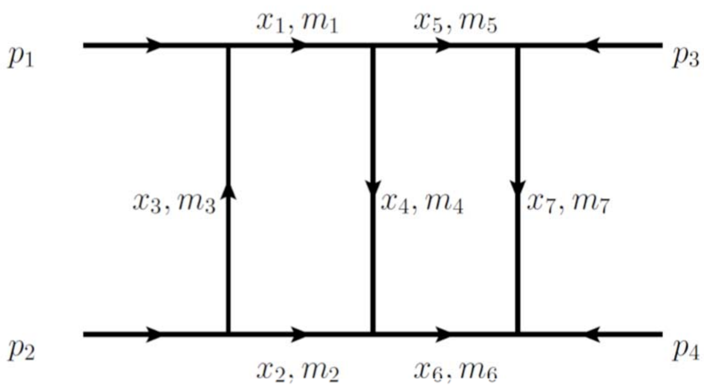
$$x_j = g(hj) \qquad \omega_j = g'(hj)$$

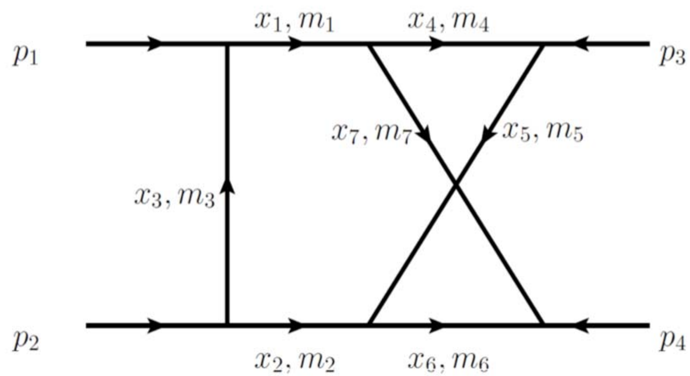
DQAGE and DE (for 1dim) can be used iteratively for multi-dimensional integral.

Extrapolation

- Extrapolation is used to accelerate convergence of the sequence
  - Wynn's epsilon algorithm
  - Mathematical Tables and Other Aids to Computation, Vol. 10, No. 54 (Apr., 1956), pp.91-96Published (1956) 91-96,
  - SIAM J. Numer. Anal. **3** (1966) 91-122.
- This does not require the specific information of the sequence

Two-loop planar and non-planar box





$$s = (p_1 + p_2)^2 = (p_3 + p_4)^2 \,, \quad t = (p_1 + p_3)^2 = (p_2 + p_4)^2$$

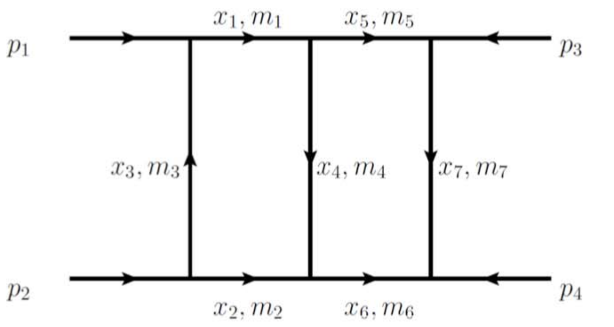
$$p_1 + p_2 + p_3 + p_4 = 0$$

$$J = -\int_0^1 dx_1\, dx_2\, dx_3\, dx_4\, dx_5\, dx_6\, dx_7\, \delta(1 - \sum_{\ell=1}^{7} x_\ell) \frac{\mathcal{C}}{(\mathcal{D} - i\epsilon\mathcal{C})^3}$$

Two-loop planar box

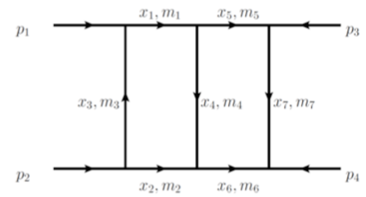
$$J = -\int_0^1 dx_1\, dx_2\, dx_3\, dx_4\, dx_5\, dx_6\, dx_7\, \delta(1 - \sum_{\ell=1}^{7} x_\ell) \frac{\mathcal{C}}{(\mathcal{D} - i\epsilon\mathcal{C})^3}$$

$$\begin{aligned}
\mathcal{D} &= \mathcal{C} \sum x_\ell m_\ell^2 \\
&- \{s(x_1 x_2(x_4 + x_5 + x_6 + x_7) + x_5 x_6(x_1 + x_2 + x_3 + x_4) + x_1 x_4 x_6 + x_2 x_4 x_5) \\
&+ t x_3 x_4 x_7 \\
&+ p_1^2(x_3(x_1 x_4 + x_1 x_5 + x_1 x_6 + x_1 x_7 + x_4 x_5)) \\
&+ p_2^2(x_3(x_2 x_4 + x_2 x_5 + x_2 x_6 + x_2 x_7 + x_4 x_6)) \\
&+ p_3^2(x_7(x_1 x_4 + x_1 x_5 + x_2 x_5 + x_3 x_5 + x_4 x_5)) \\
&+ p_4^2(x_7(x_1 x_6 + x_2 x_4 + x_2 x_6 + x_3 x_6 + x_4 x_6))\},
\end{aligned}$$



$$\mathcal{C} = (x_1 + x_2 + x_3 + x_4)(x_4 + x_5 + x_6 + x_7) - x_4^2$$

Variable transformation for two-loop planar box



$$(x_1, x_2, x_3, x_4, x_5, x_6, x_7) \to (\rho_1, \rho_2, \rho_3, u_1, u_2, u_3, u_4)$$

$$x_1 = \rho_1 u_1, x_2 = \rho_1 u_2, x_3 = \rho_1(1-u_1-u_2), x_4 = \rho_3, x_5 = \rho_2 u_3, x_6 = \rho_2 u_4, x_7 = \rho_2(1-u_3-u_4)$$

$$\to (\rho, \xi, u_1, u_2, u_3, u_4)$$

$$\rho_1 = \rho\xi, \rho_2 = \rho(1-\xi), \rho_3 = 1-\rho$$

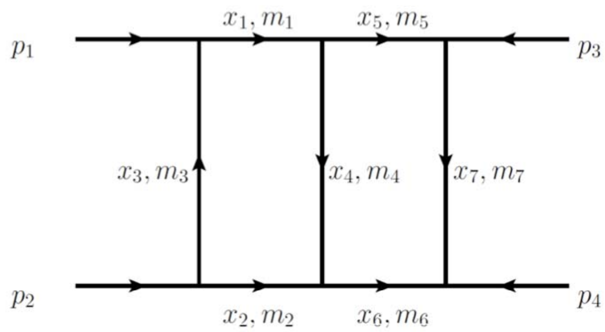
$$\mathcal{D}' = \mathcal{D}/\rho \text{ and } \mathcal{C}' = \mathcal{C}/\rho$$

$$\mathcal{I} = -\int_0^1 d\rho \int_0^1 d\xi \int_0^1 du_1 \int_0^{1-u_1} du_2 \int_0^1 du_3 \int_0^{1-u3} du_4 \, \frac{\mathcal{C}'}{(\mathcal{D}' - i\epsilon\mathcal{C}')^3} \rho^3 \xi^2 (1-\xi)^2$$

where $\mathcal{D}'$ is a quadratic in $\mathbf{u} = (u_1, u_2, u_3, u_4)^T$

$$\mathcal{D}' = \mathbf{u}^T A \mathbf{u} + \mathbf{b}^T \mathbf{u} + c \qquad \mathcal{C}' = \rho\xi(1-\xi) + 1 - \rho$$

$$\begin{cases} p_1^2 = p_2^2 = p_3^2 = p_4^2 = m^2, \\ m_1 = m_2 = m_5 = m_6 = m, \\ m_3 = m_4 = m_7 = M \end{cases}$$



$$\mathcal{D}' = \mathbf{u}^T A \mathbf{u} + \mathbf{b}^T \mathbf{u} + c$$

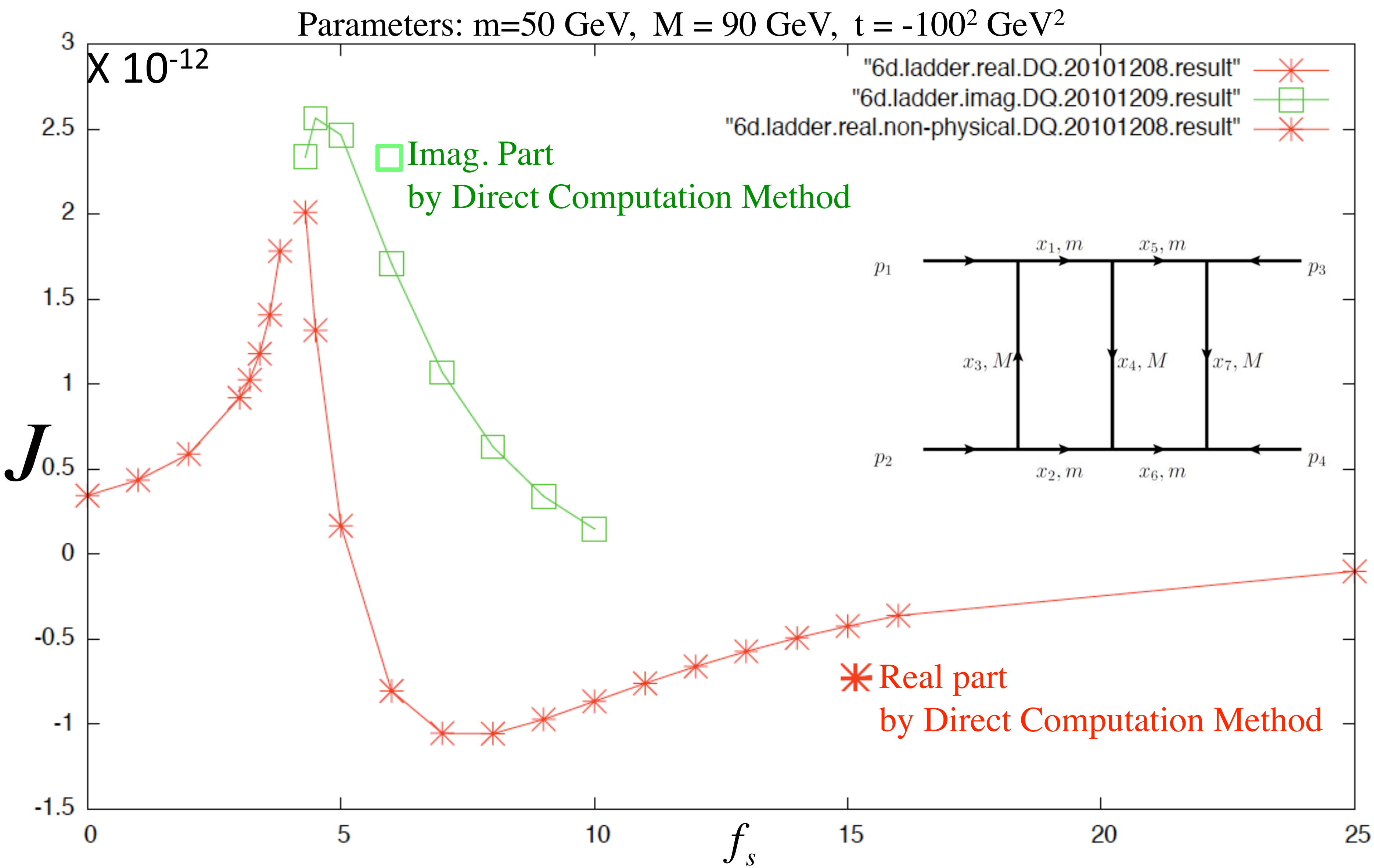
$$A = \begin{pmatrix} (1-\rho)^2 A_1 & \rho\xi(1-\rho)(1-\xi)A_2 \\ \rho\xi(1-\rho)(1-\xi)A_2 & A_3 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} -m^2 & s/2 - m^2 \\ s/2 - m^2 & -m^2 \end{pmatrix}, A_2 = \begin{pmatrix} t/2 - m^2 & 2/s + t/2 - m^2 \\ s/2 + t/2 - m^2 & t/2 - m^2 \end{pmatrix},$$

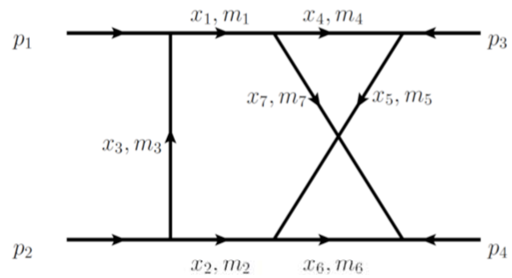
$$A_3 = \begin{pmatrix} -m^2\rho\xi^2(1-\rho\xi) & (-s/2+m^2)\rho(1-\rho)\xi(1-\xi) \\ (-s/2+m^2)\rho(1-\rho)\xi(1-\xi) & -m^2\rho(1-\xi)^2(1-\rho+\rho\xi) \end{pmatrix}.$$

$$\vec{b} = \begin{pmatrix} -t\rho\xi(1-\rho)(1-\xi) + M^2(1-\rho)\mathcal{C}' \\ -t\rho\xi(1-\rho)(1-\xi) + M^2(1-\rho)\mathcal{C}' \\ -t\rho\xi(1-\rho)(1-\xi) + M^2\rho\xi\mathcal{C}' \\ -t\rho\xi(1-\rho)(1-\xi) + M^2\rho(1-\xi)\mathcal{C}' \end{pmatrix} \qquad c = t\rho(1-\rho)\xi(1-\xi) - M^2\mathcal{C}'$$

Numerical results of Two-loop planar box with masses in function of $f_s = s/m^2$



Two-loop non-planar box



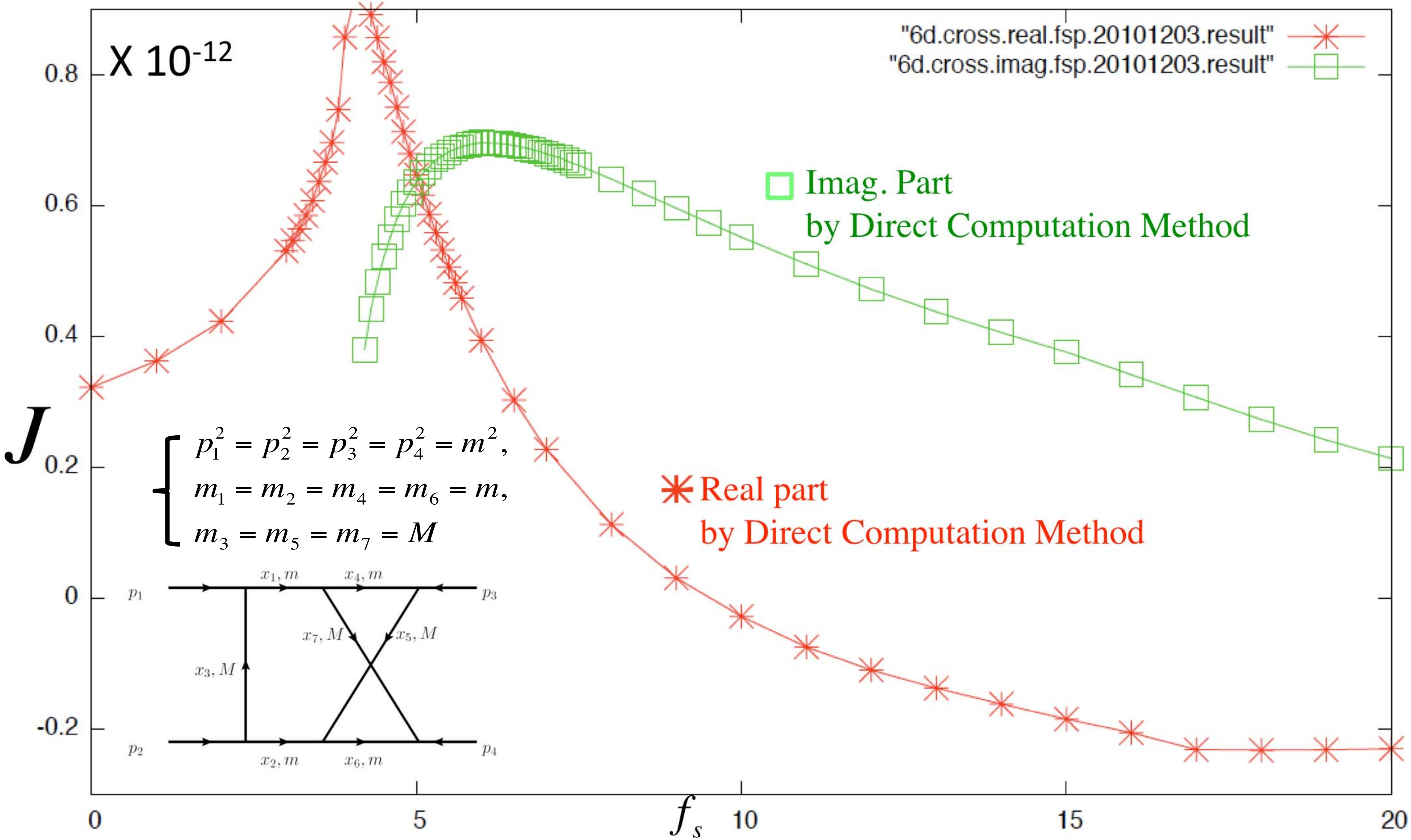
$$J = -\int_0^1 dx_1\, dx_2\, dx_3\, dx_4\, dx_5\, dx_6\, dx_7\, \delta(1 - \sum_{\ell=1}^{7} x_\ell) \frac{\mathcal{C}}{(\mathcal{D} - i\epsilon\mathcal{C})^3}$$

$$\begin{aligned}
\mathcal{D} &= -\mathcal{C} \sum x_\ell m_\ell^2 \\
&+ \{s(x_1x_2x_4 + x_1x_2x_5 + x_1x_2x_6 + x_1x_2x_7 + x_1x_5x_6 + x_2x_4x_7 - x_3x_4x_6) \\
&+ t(x_3(-x_4x_6 + x_5x_7)) \\
&+ p_1^2(x_3(x_1x_4 + x_1x_5 + x_1x_6 + x_1x_7 + x_4x_6 + x_4x_7)) \\
&+ p_2^2(x_3(x_2x_4 + x_2x_5 + x_2x_6 + x_2x_7 + x_4x_6 + x_5x_6)) \\
&+ p_3^2(x_1x_4x_5 + x_1x_5x_7 + x_2x_4x_5 + x_2x_4x_6 + x_3x_4x_5 + x_3x_4x_6 + x_4x_5x_6 + x_4x_5x_7) \\
&+ p_4^2(x_1x_4x_6 + x_1x_6x_7 + x_2x_5x_7 + x_2x_6x_7 + x_3x_4x_6 + x_3x_6x_7 + x_4x_6x_7 + x_5x_6x_7)\}
\end{aligned}$$

$$\mathcal{C} = (x_1 + x_2 + x_3 + x_4 + x_5)(x_1 + x_2 + x_3 + x_6 + x_7) - (x_1 + x_2 + x_3)^2$$

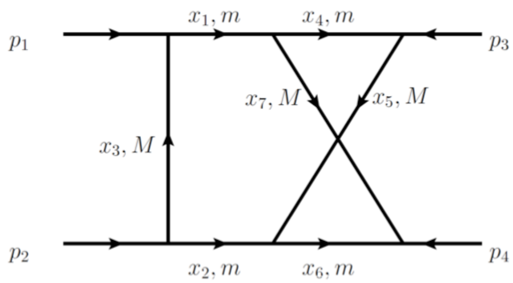
Numerical results of Two-loop non-planar box with masses in function of $f_s = s/m^2$

Parameters: m=50 GeV, M = 90 GeV, t = -100² GeV²



Computation time

Example: two-loop non-planar box in physical region



fs	Computation time	key	Limit
6.0	16 hours	2	10, 20, 10, 10, 10, 10
7.0	2 days	2	10, 20, 10, 10, 10, 10
10.0	1 week	2	10, 20, 20, 10, 10, 10

For f_s = -1, computation time is ~24 sec.

by single CPU: Intel(R) Xeon(R) CPU X5460 @ 3.16GHz

Integration parameter
Key : Gauss-Kronrod rule, 10 - 21 points if key = 2
Limit: an upperbound on the number of subintervals

Parallel computing is required.

Cross-check

1. Comparison with reduction method
2. Self consistency check using Dispersion relation
3. Comparison with another method

Cross-check: Reduction Method

$$D = \mathbf{x}^T A\mathbf{x} + \mathbf{B}^T\mathbf{x} + C$$

$$\mathbf{x}^T = (x_1, x_2, \cdots, x_N), \quad A = (a_{ij}) \quad a_{ij} = a_{ji}$$

$$\mathbf{B}^T = (b_1, b_2, \cdots, b_N)\ , \qquad C \text{ is a constant}$$

Let us assume $\det(A) \neq 0$

$\Delta_N = \mathbf{B}^T A^{-1}\mathbf{B} - 4C$ and let us assume $\Delta_N \neq 0$

with $\mathbf{X} = 2\mathbf{x} + A^{-1}\mathbf{B} = A^{-1}\nabla D(\mathbf{x})$

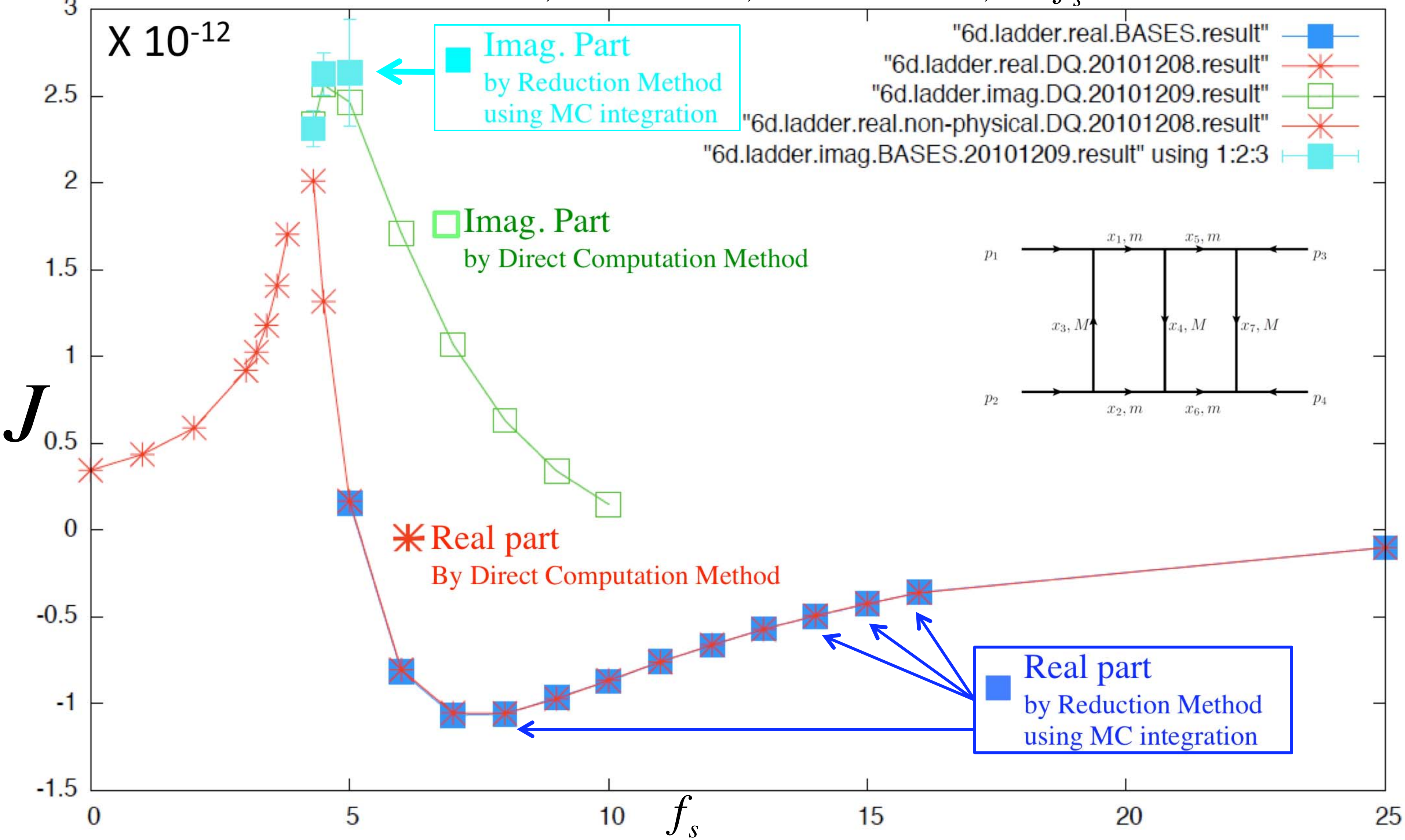
$$\frac{\Delta_N}{D^{n+1}} = \boxed{\frac{-4 + 2N/n}{D^n}} - \frac{1}{n}\nabla^T\left(\frac{\mathbf{X}}{D^n}\right) \quad , \ \nabla^T = (\partial_1, \partial_2, \cdots, \partial_N)$$

This term vanishes with $N=2n$.

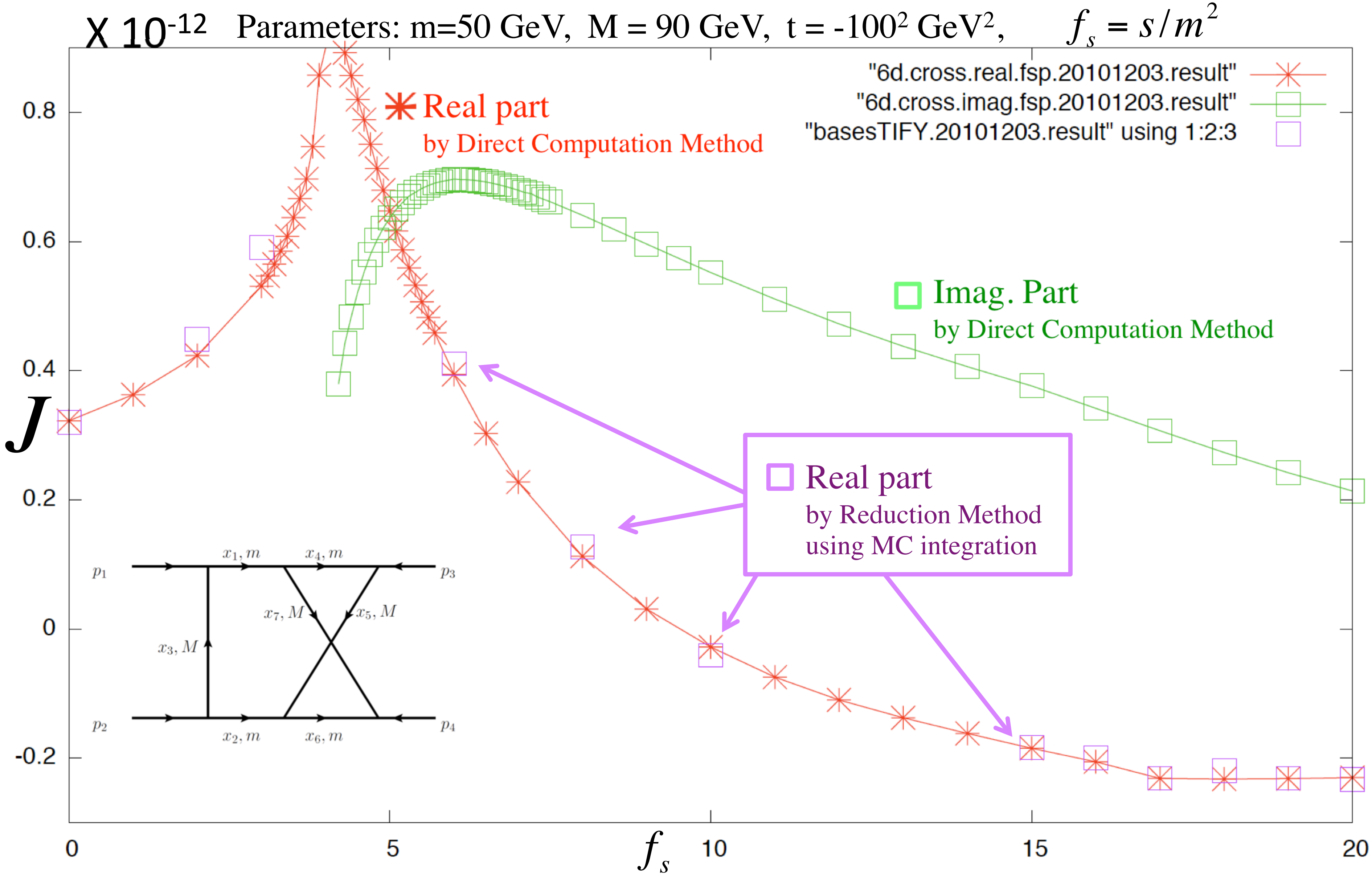
$$n = 0 \qquad \frac{\Delta_N}{D} = (-4 - 2N\log D) + \nabla^T(\mathbf{X}\log D)$$

Comparison with numerical results by reduction method

Parameters: m=50 GeV, M = 90 GeV, t = -100^2 GeV^2, $f_s = s/m^2$

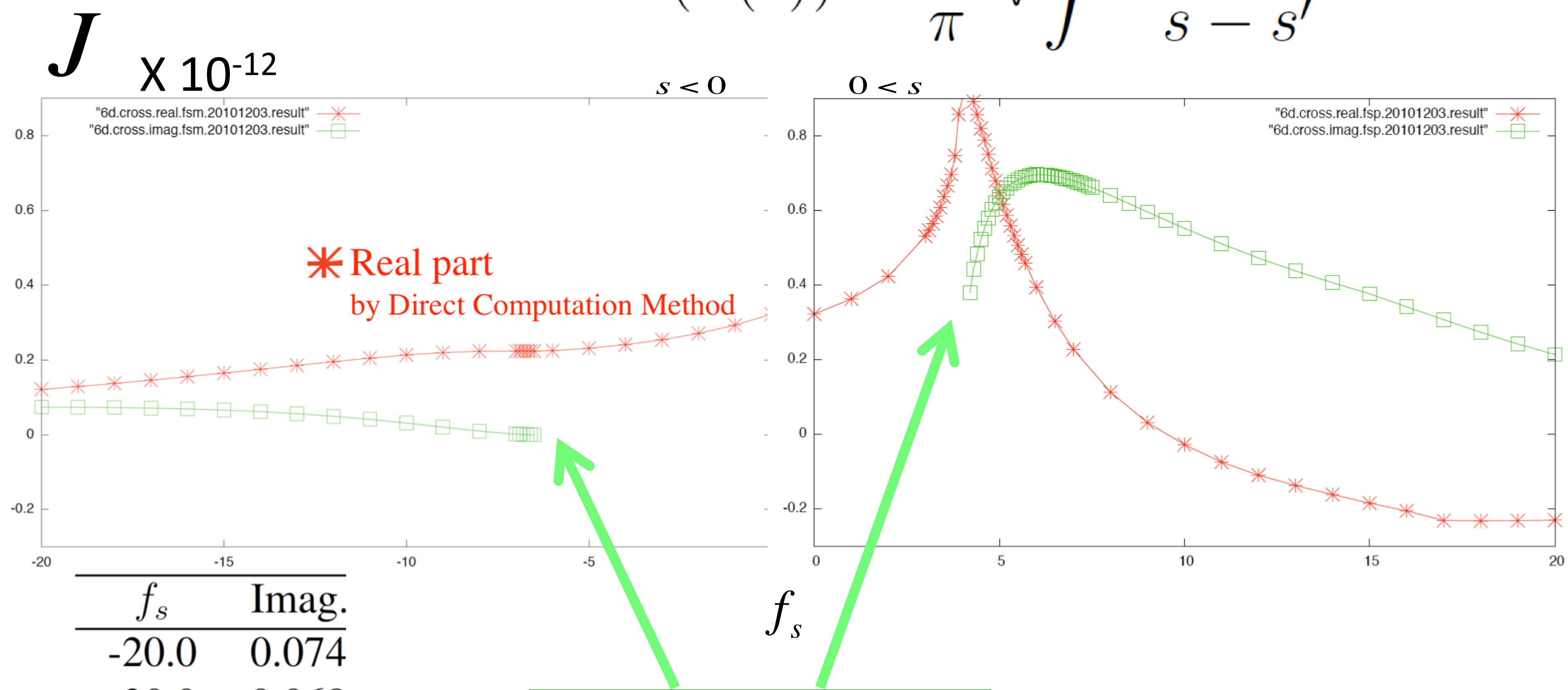


Comparison with numerical results by reduction method



Cross-check: Dispersion Relation

$$\Re(J(s)) = \frac{1}{\pi} \mathrm{P_v} \int \frac{\Im(J(s'))}{s - s'} ds$$



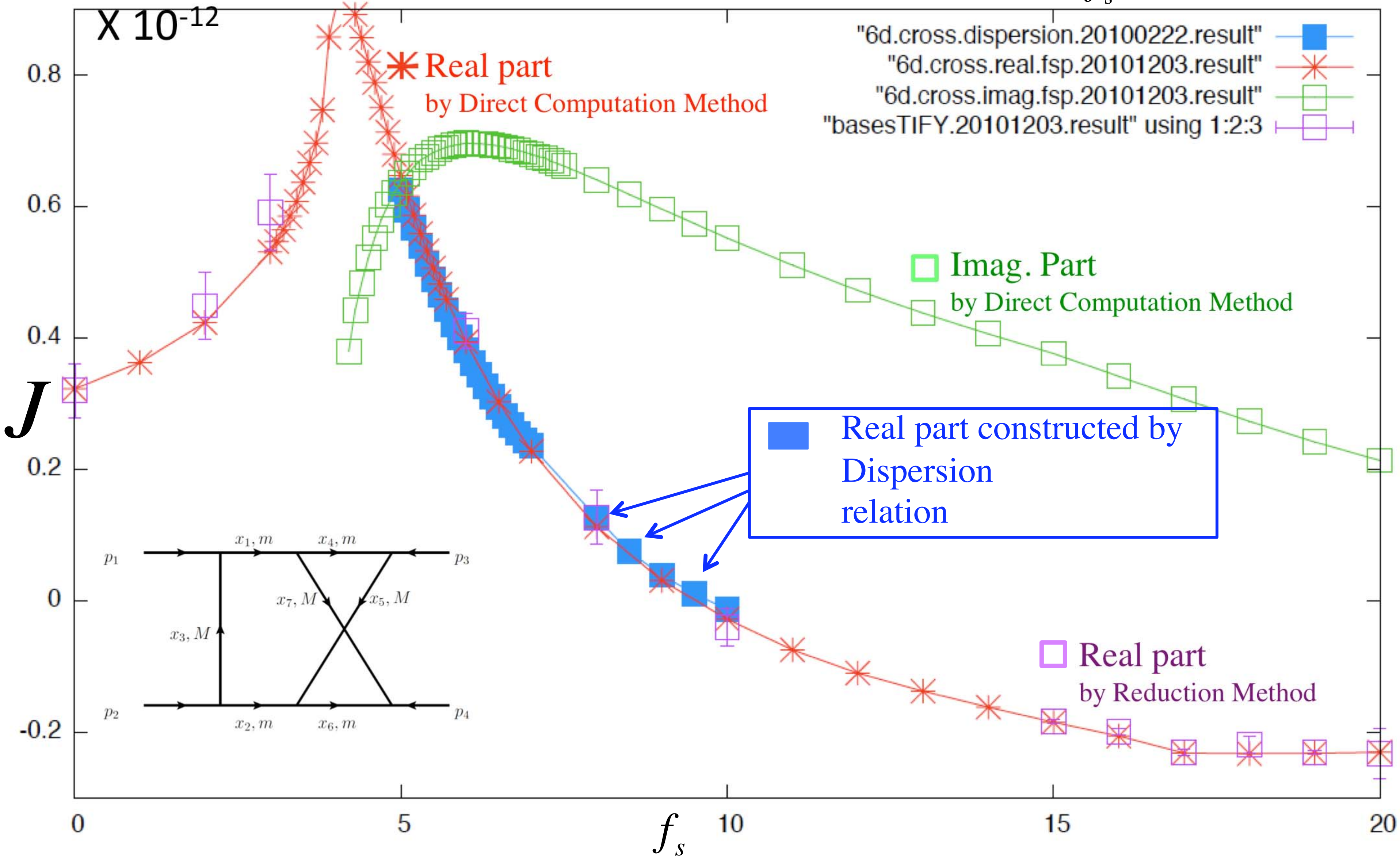
f_s	Imag.
-20.0	0.074
-30.0	0.068
-40.0	0.053
-50.0	0.042
-100.0	0.017

□ Imag. Part
by Direct Computation Method

Real part are constructed by imag. part.

Self consistency check by dispersion relation

Parameters: m=50 GeV, M = 90 GeV, t = -100^2 GeV^2, $f_s = s/m^2$



Comparison with Laporta's program SYS

- S. Laporta, Int. J. Mod. Phys. A15 (2000) 5087
- The program SYS
  - High-precision calculation of multi-loop Feynman integrals by difference equations
- Parameters:
  - $m_1 = \ldots = m_N = 1$
  - $s = t = 1$ and $u = 2$

selfenergy L=1,2 N=2,3,4,5

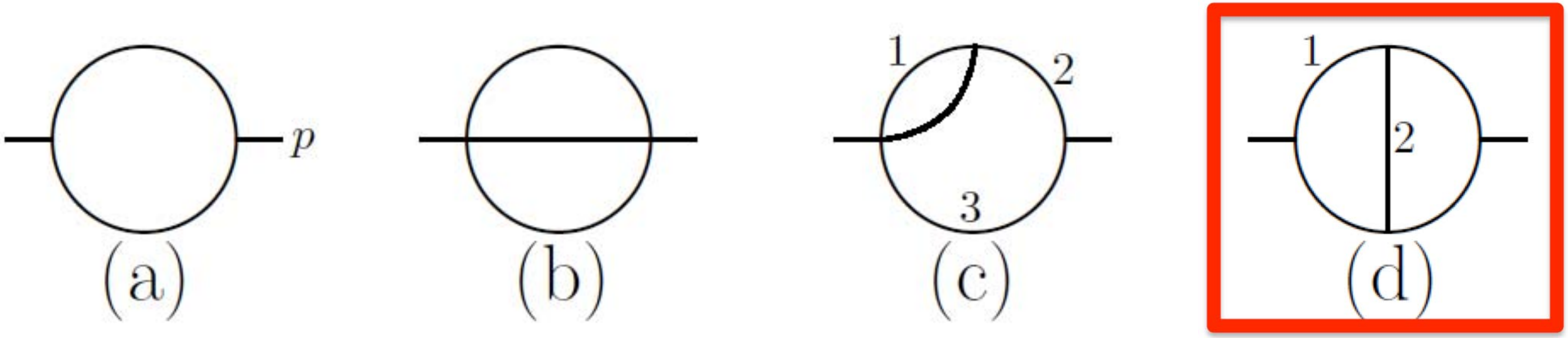


Fig.	L	N		finite term
(d)	2	5	Laporta	0.9236318265199
			DQ-Direct Computation Method	0.92363182651986‌4
			DE-Direct Computation Method	0.9236

DQ-Direct Computation Method: DQAGE routine is used

DE-Direct Computation Method: Double Exp. Formulae is used

Cross-check: Comparison with Laporta's program SYS

vertex L=1,2 N=3,4,5,6

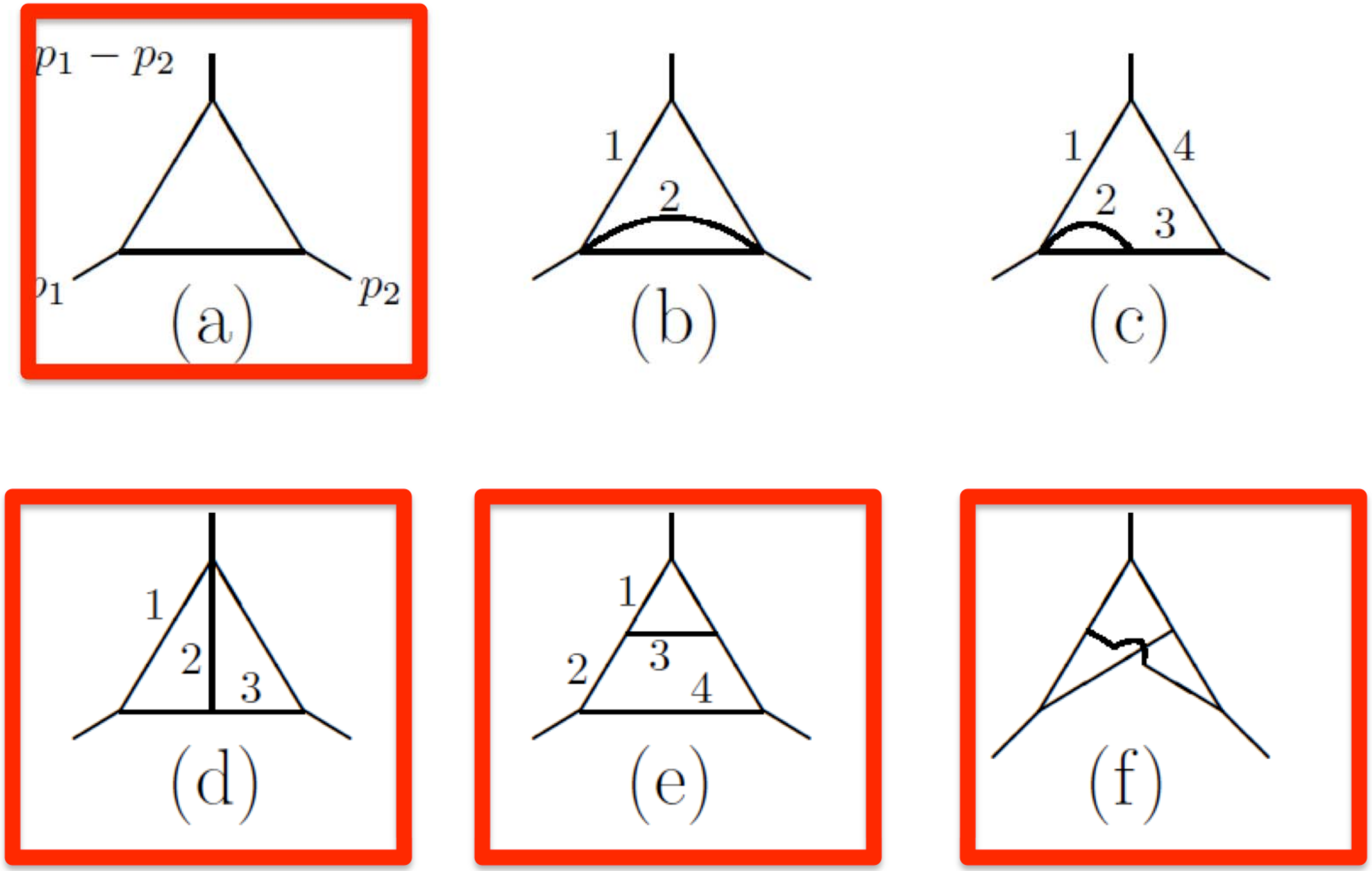
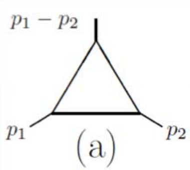
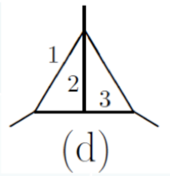
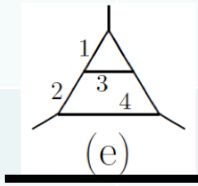
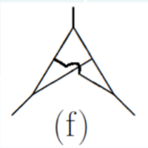
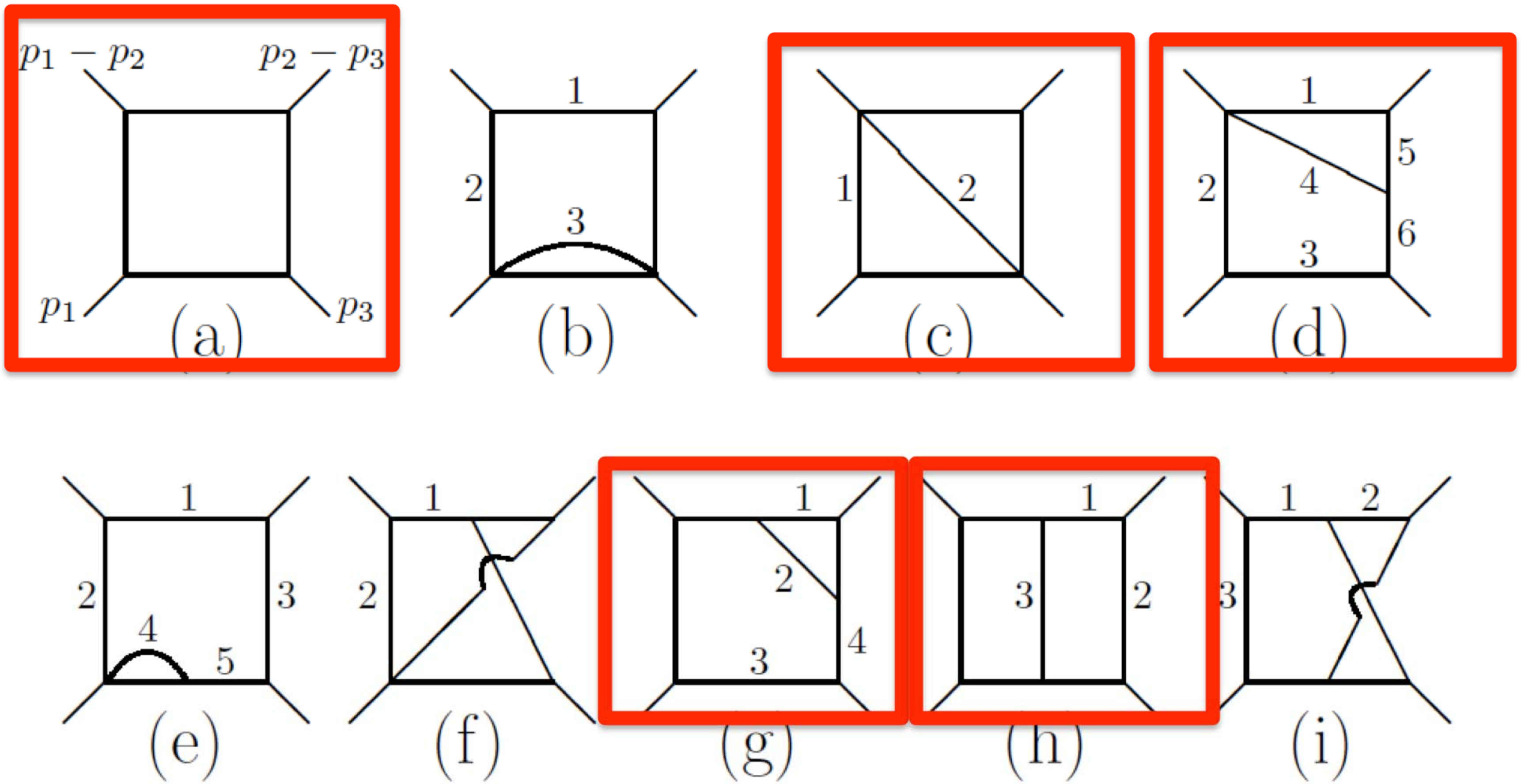
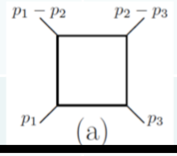
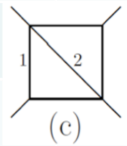
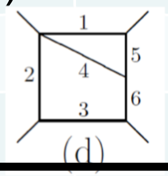
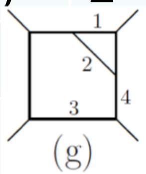
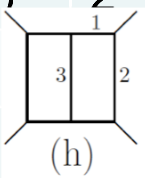


Fig.	L	N		finite term
(a)	1	3	Laporta	0.671253105748
 (a)			DQ-Direct Computation Method	0.67125310574800
			DE-Direct Computation Method	0.67125310574800**5**
(d)	2	5	Laporta	0.937139527315
 (d)			DQ-Direct Computation Method	0.937139
			DE-Direct Computation Method	0.93713952731498**4**
(e)	2	6	Laporta	0.271156349402**2**
 (e)			DQ-Direct Computation Method	0.27116
			DE-Direct Computation Method	0.2711559
(f)	2	6	Laporta	0.173896742268
 (f)			DQ-Direct Computation Method	0.173432
			DE-Direct Computation Method	0.17390

box L=1,2 and N=4,5,6,7



	L	N		finite term
(a)	1	4	Laporta	0.3455029252972
			DQ-Direct Computation Method	0.345502925297 18
			DE-Direct Computation Method	0.34550292528 9537
(c)	2	5	Laporta	0.9509235623171
			DQ-Direct Computation Method	0.95092
			DE-Direct Computation Method	0.95092
(d)	2	6	Laporta	0.276209225359
			DQ-Direct Computation Method	Not yet
			DE-Direct Computation Method	0.276
(g)	2	7	Laporta	0.1723367907503
			DQ-Direct Computation Method	Not yet
			DE-Direct Computation Method	0.17232
(h)	2	7	Laporta	0.1036407209893
			DQ-Direct Computation Method	0.1036408479
			DE-Direct Computation Method	0.103643

Summary

- By fully numerical method, Direct Computation Method, we computed two-loop planar and non-planar box integrals with masses.
- The method is based on the combination of numerical integration and numerical extrapolation.
- The quality of numerical results are checked in three ways; (a) a comparison with reduction method, (b) a self consistency check using dispersion relation and (c) a comparison with Laporta's method. We find agreement in all cases.
- To reduce the computation time, parallel computing is required.