

# Vertex functions in QCD - preparation for beyond two loops

J.A. Gracey



**DFG** Mercator Fellow

January, 2020

# Overview

- 1 Introduction
  - Motivation
- 2 Triple gluon vertex
  - Background
  - Example
- 3 Beyond two loops

# Outline

There has been a large industry in computing on-shell multileg multiloop Green's functions for LHC physics

One primary goal is improving cross-sections to high precision in order to extract signals above the QCD background for instance

The situation with completely off-shell Green's functions in QCD is not as advanced

Various theoretical reasons why such computations are needed

Non-perturbative structure can be accessed in the field theory context through solving Schwinger-Dyson equations

This is achieved by using various ansätze for the general form of the propagators and vertex functions

Certain assumptions are made in their solution and so it is important to have independent data on the vertex functions for instance in order to gauge how accurate the approximations are

Also other nonperturbative methods such as lattice computations of vertex functions in the chiral limit require high order perturbative data in order to match to the high energy continuum limit

Such lattice computations are for off-shell momentum configurations

More recently knowing say the off-shell triple gluon vertex to two loops means that the QCD  $\beta$ -function is known to three loops in kinematic schemes such as the momentum subtraction scheme popularized by Celmaster and Gonsalves

Various one and two loop computations available over 40 years with various off-shell momentum configurations

Other motivations for computing off-shell vertex structure in QCD is to exploit the known structure in models of gluon radiation as well as to check that the Slavnov-Taylor identities are correctly implemented ensuring that the strong coupling constant can be measured from any of the vertex processes

A few examples are

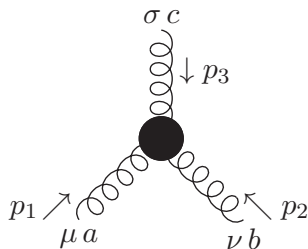
- One loop 3-point QCD vertices at the fully symmetric point [Celmaster, Gonsalves]
- One loop 4-point gluon vertex at the fully symmetric point [Pascual, Tarrach]
- Two loop 3-point QCD vertices at the fully symmetric point and later for the most general off-shell configuration [JAG]
- All other 4-point QCD Green's functions known at one loop at the symmetric point [JAG]
- Related work by a variety of others [Davydychev, Saks, Osland, Tarasov, Chetyrkin, Seidensticker]

There is a clear need to push this formalism to beyond the current two loop order

# Setup

Will focus on the triple gluon vertex as an example

The kinematic setup for the off-shell gluon 3-point function is



where

$$\sum_{n=1}^3 p_n = 0$$

Take  $p_1$  and  $p_2$  as the independent momenta

The independent variables of the vertex function are defined as

$$x = \frac{p_1^2}{p_3^2} \quad , \quad y = \frac{p_2^2}{p_3^2} \quad , \quad p_3^2 = -\mu^2$$

The fully symmetric point of Celmaster and Gonsalves is  $x = y = 1$

In order to compute the Green's function analytically need to decompose into the colour and Lorentz bases

$$\left\langle A_\mu^a(p_1) A_\nu^b(p_2) A_\sigma^c(-p_1 - p_2) \right\rangle = f^{abc} \Sigma_{\mu\nu\sigma}^{\text{ggg}}(p_1, p_2)$$

then

$$\Sigma_{\mu\nu\sigma}^{\text{ggg}}(p_1, p_2) = \sum_{k=1}^{14} \mathcal{P}_{(k)\mu\nu\sigma}^{\text{ggg}}(p_1, p_2) \Sigma_{(k)}^{\text{ggg}}(p_1, p_2)$$

where  $\Sigma_{(k)}^{\text{ggg}}(p_1, p_2)$  are scalar amplitudes

These can be accessed for calculation purposes by a projection method

# Basis

Full basis is

$$\begin{aligned}\mathcal{P}_{(1)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) &= \eta_{\mu\nu} p_{1\sigma} \quad , \quad \mathcal{P}_{(2)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) = \eta_{\nu\sigma} p_{1\mu} \\ \mathcal{P}_{(3)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) &= \eta_{\sigma\mu} p_{1\nu} \quad , \quad \mathcal{P}_{(4)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) = \eta_{\mu\nu} p_{2\sigma} \\ \mathcal{P}_{(5)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) &= \eta_{\nu\sigma} p_{2\mu} \quad , \quad \mathcal{P}_{(6)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) = \eta_{\sigma\mu} p_{2\nu} \\ \mathcal{P}_{(7)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) &= \frac{1}{\mu^2} p_{1\mu} p_{1\nu} p_{1\sigma} \quad , \quad \mathcal{P}_{(8)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) = \frac{1}{\mu^2} p_{1\mu} p_{1\nu} p_{2\sigma} \\ \mathcal{P}_{(9)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) &= \frac{1}{\mu^2} p_{1\mu} p_{2\nu} p_{1\sigma} \quad , \quad \mathcal{P}_{(10)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) = \frac{1}{\mu^2} p_{2\mu} p_{1\nu} p_{1\sigma} \\ \mathcal{P}_{(11)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) &= \frac{1}{\mu^2} p_{1\mu} p_{2\nu} p_{2\sigma} \quad , \quad \mathcal{P}_{(12)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) = \frac{1}{\mu^2} p_{2\mu} p_{1\nu} p_{2\sigma} \\ \mathcal{P}_{(13)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) &= \frac{1}{\mu^2} p_{2\mu} p_{2\nu} p_{1\sigma} \quad , \quad \mathcal{P}_{(14)\mu\nu\sigma}^{\text{gggg}}(p_1, p_2) = \frac{1}{\mu^2} p_{2\mu} p_{2\nu} p_{2\sigma}\end{aligned}$$



# Projection method

From

$$\Sigma_{\mu\nu\sigma}^{\text{ggg}}(p_1, p_2) = \sum_{k=1}^{14} \mathcal{P}_{(k)\mu\nu\sigma}^{\text{ggg}}(p_1, p_2) \Sigma_{(k)}^{\text{ggg}}(p_1, p_2)$$

then

$$\begin{aligned} \mathcal{P}_{(j)}^{\text{ggg}\mu\nu\sigma}(p_1, p_2) \Sigma_{\mu\nu\sigma}^{\text{ggg}}(p_1, p_2) &= \mathcal{P}_{(j)}^{\text{ggg}\mu\nu\sigma}(p_1, p_2) \sum_{k=1}^{14} \mathcal{P}_{(k)\mu\nu\sigma}^{\text{ggg}}(p_1, p_2) \Sigma_{(k)}^{\text{ggg}}(p_1, p_2) \\ &\equiv \sum_{k=1}^{14} \mathcal{N}_{jk}^{\text{ggg}} \Sigma_{(k)}^{\text{ggg}}(p_1, p_2) \end{aligned}$$

which implies

$$\Sigma_{(i)}^{\text{ggg}}(p_1, p_2) = \sum_{k=1}^{14} \mathcal{M}_{ik}^{\text{ggg}} \mathcal{P}_{(k)}^{\text{ggg}\mu\nu\sigma}(p_1, p_2) \Sigma_{\mu\nu\sigma}^{\text{ggg}}(p_1, p_2)$$

where  $\mathcal{M}_{ij}^{\text{ggg}}$  is the inverse of  $\mathcal{N}_{ij}^{\text{ggg}}$

A few of the 196 elements are

$$\mathcal{M}_{413} = -\frac{8}{(d-2)\Delta^3} \left[ x^2 - 2xy - 2x + y^2 - 2y + 1 \right] (x+y-1)x$$

$$\mathcal{M}_{55} = \frac{4}{(d-2)\Delta^3} \left[ x^2 - 2xy - 2x + y^2 - 2y + 1 \right]^2 x$$

$$\mathcal{M}_{89} = \frac{16}{(d-2)\Delta^3} (d+1)(x+y-1)^2 y$$

$$\mathcal{M}_{1113} = \frac{16}{(d-2)\Delta^3} (d+1)(x+y-1)^2 x$$

where

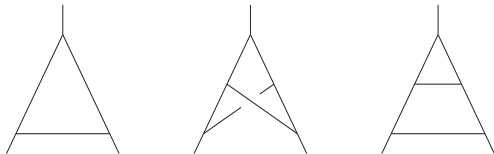
$$\Delta = x^2 - 2xy + y^2 - 2x - 2y + 1$$

is the Gram determinant

# Computation

Procedure to evaluate vertex function is to project out the amplitudes in terms of a very large number of scalar Feynman integrals

Structure of all the integrals to two loops are derived from the graphical representation of the integral families



including two rotations of the last topology

These families are the starting point for relating all the required Feynman integrals to a small set of master integrals using the Laporta algorithm based on integration by parts

Standard automatic symbolic manipulation tools are used

Graphs generated using QGRAF

Various Laporta integration by parts packages are available but C++ based version REDUZE used (versions 1 and 2)

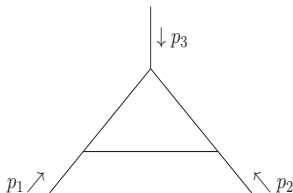
Theory is dimensionally regularized in  $d = 4 - 2\epsilon$  dimensions

Database of relations of all integrals written in terms of the one and two loop masters have been constructed

All routines encoded in symbolic manipulation language FORM

## One loop triangle

One loop master integral  $I_1(x, y)$  is defined graphically by



and has the  $\epsilon$  expansion

$$I_1(x, y) = -\frac{1}{\mu^2} \left[ \Phi_1(x, y) + \Psi_1(x, y)\epsilon + \left[ \frac{\zeta(2)}{2} \Phi_1(x, y) + \chi_1(x, y) \right] \epsilon^2 + O(\epsilon^3) \right]$$

$$\text{with } \Phi_1(x, y) = \frac{1}{\lambda} \left[ 2\text{Li}_2(-\rho x) + 2\text{Li}_2(-\rho y) + \ln\left(\frac{y}{x}\right) \ln\left(\frac{(1+\rho y)}{(1+\rho x)}\right) + \ln(\rho x) \ln(\rho y) + \frac{\pi^2}{3} \right]$$

The functions  $\lambda(x, y)$  and  $\rho(x, y)$  are defined by

$$\begin{aligned}\rho(x, y) &= \frac{2}{[1 - x - y + \lambda(x, y)]} \\ \lambda(x, y) &= \sqrt{[1 - 2x - 2y + x^2 - 2xy + y^2]}\end{aligned}$$

Higher order terms of the  $\epsilon$  expansion involve higher polylogarithms

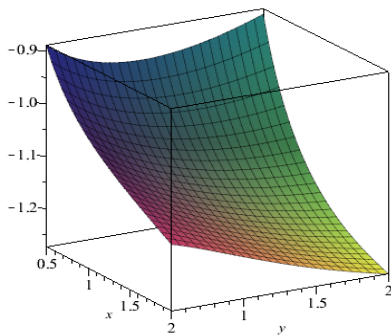
Two loop master integrals involve equally if not more complicated functions involving polylogarithms *and* harmonic polylogarithms

For 3-gluon vertex there are 106 two loop Feynman graphs to be computed

All 3-point off-shell vertex functions are now available at two loops

Three dimensional plot of coefficient of  $\mathcal{P}_{(1)\mu\nu\sigma}^{ggg}(p_1, p_2) = \eta_{\mu\nu} p_{1\sigma}$  at one loop for  $\alpha_S = 0.1$

Two loop correction is around 1% different from one loop at this coupling value



## Three loop strategy

To extend to three loops requires several extensions

First the algorithm will be the same using the projection method to isolate integrals for Laporta reduction

There are more efficient reduction packages available now

At three loops there will be 2382 graphs for triple gluon vertex

Several issues need to be resolved

The master integrals for two loop off-shell computations have been known for over 25 years now [Usyukina, Davydychev; Birthwright et al]

Three loop values need to be determined for arbitrary  $x$  and  $y$

In recent years since the evaluation of the two loop masters, the underlying mathematics is much better understood now from the graph polynomials of the Schwinger parametric representation of Feynman integrals



As the loop order increases the number of parameter integrations of the master increases

In order to perform the parameter integrations the integral has to be linearly reducible

This means there is an order of integration such that at each integration the denominator parameter polynomial can be factorized

Then the integration over that parameter can be evaluated in terms of polylogarithms and their generalizations

It is known that the three loop 3-point off-shell vertex masters are linearly reducible

The explicit values are yet to be determined but methods have since been developed to achieve this

For example the `MAPLE` package `HYPERINT` is available [Panzer]

In order to handle the harmonic polylogarithms that clearly arise need to translate to the word representation

Use notation of zeta\_procedures of Schnetz

$$i[0, \sigma_m, \dots, \sigma_1, z] = \text{Hlog}(z, [\sigma_1, \dots, \sigma_n])$$

For instance one loop triangle graph will have the form

$$\begin{aligned} I_1(x, y) = & \left[ \left[ \frac{2}{3} i[0, \omega, 0, 1] - \frac{2}{3} i[0, \bar{\omega}, 0, 1] \right] \right. \\ & + \left[ \frac{4}{3} i[0, \omega, 0, 0, 1] - \frac{2}{3} i[0, \omega, \omega, 0, 1] - \frac{2}{3} i[0, \omega, 0, \omega, 1] \right. \\ & + \frac{4}{3} i[0, \omega, 0, 1] i[0, \bar{\omega}^2, 1] - \frac{2}{3} i[0, \omega, 0, 1] i[0, \bar{\omega}, 1] \\ & + \frac{2}{3} i[0, \omega, 1] i[0, \bar{\omega}^2, 0, 1] - \frac{2}{3} i[0, \bar{\omega}, \bar{\omega}^2, 0, 1] \\ & + \frac{2}{3} i[0, \bar{\omega}, \bar{\omega}, 0, 1] - \frac{4}{3} i[0, \bar{\omega}, 0, \bar{\omega}^2, 1] \\ & \left. \left. + \frac{2}{3} i[0, \bar{\omega}, 0, \bar{\omega}, 1] + \frac{2}{3} i[0, \bar{\omega}, 0, 0, 1] \right] \epsilon + O(\epsilon^2) \right] \frac{i\sqrt{3}}{\mu^2} \end{aligned}$$

Expansion available to  $O(\epsilon^4)$

$\omega$  and  $\bar{\omega}$  parametrization from Chavez and Duhr [1209.2722]

$$x = \omega\bar{\omega} \quad , \quad y = (1 - \omega)(1 - \bar{\omega})$$

where  $\omega$  and  $\bar{\omega}$  are roots of unity

Notation is more compact and useful for fully off-shell Green's functions especially in evaluating higher order masters

To determine the three loop masters is a balancing act between minimizing the number of terms in the  $\epsilon$  expansion of the masters that need to be computed

This is in order to reduce computation time which will need to avoid spurious divergences deriving from the Laporta reduction

Current best strategy is to isolate the complicated higher order hyperlogarithms in masters which are ultraviolet finite

This can be achieved with the Tarasov method which is implemented in various reduction packages

# Tarasov method

This relates integrals in  $d$ -dimensions to integrals in  $(d + 2)$ -dimensions with the same or lower topology

The connection is straightforward to derive from Schwinger representation of Feynman integrals in  $d$ -dimensions

If  $\alpha_i$  are the parameters then the Feynman integral representation contains a graph polynomial  $D(\alpha_i)$  that is the only source of  $d$ -dependence

It appears as  $\frac{1}{D(\alpha_i)^{d/2}}$  which can then be rewritten

$$\frac{1}{D(\alpha_i)^{d/2}} = \frac{D(\alpha_i)}{D(\alpha_i)^{(d+2)/2}}$$

## One loop example

Applying the method to the one loop triangle as a simple example gives

$$\text{Triangle}(d) = \text{Triangle}(d+2, \text{dot on bottom}) + \text{Triangle}(d+2, \text{dot on left}) + \text{Triangle}(d+2, \text{dot on right})$$

where label indicates the integral dimension and one dot increases the propagator power by unity

Applying the Laporta algorithm gives

$$\text{Triangle}(d) = \frac{3}{2}(d-4) \text{Triangle}(d+2) + (3-d) \text{Bubble}(d+2) + \dots$$

So  $d$ -dimensional triangle is given by  $(d + 2)$ -dimensional one plus lower topology

Same structure is present at higher loops but the higher dimensional topology will be less ultraviolet divergent and thereby few terms of its  $\epsilon$  expansion will be required

In other words the divergences are located in the lower topologies

Can re-apply method in order to reduce to primitives; possibly 10 dimensions for three loop off-shell masters

## Discussion

To go beyond three loops would be a mammoth task

For triple gluon vertex there are 63992 four loop graphs

Not clear if linear reducibility has been established for four loop masters

There is also a lack of linear reducibility for off-shell 4-point functions beyond one loop

While in principle the three loop 3-point vertex functions ought to be feasible the full Laporta approach may not be the most efficient one

At four loops in the `FORCER` algorithm for 2-point functions a new integration rule was developed that went beyond the star-triangle rule that is the basis for the predecessor three loop `MINCER` algorithm

This rule was derived from examining relations between integrals of the more difficult topologies

It is possible that an adaptation of that could be used for the parallel three loop 3-point function topologies which would be produced from cutting a four loop 2-point function