

Proseminar on Computational Methods in QFT

Berends Giele Recursion

Report

Roman Gruber

ETH Zürich, October 2019

Abstract

QCD tree amplitudes are of great interest. A calculation of tree level scattering amplitudes in QCD with only gluons involved is presented. The calculation is simplified using color decomposition of full amplitudes into color ordered partial amplitudes. The results are formulated using the spinor helicity formalism for massless vector bosons. A concrete implementation of the methods mentioned in the report are introduced and further ways for optimisation are discussed.

This work is licensed under a Creative Commons “Attribution-ShareAlike 4.0 International” license.



Contents

1	Introduction	2
2	The Spinor helicity formalism for massless vector bosons	2
2.1	Important identities and definitions	3
3	Color Ordering	4
3.1	QCD gauge group	4
3.2	Color decomposition	6
4	Berends-Giele current	7
4.1	Recursion idea	7
4.2	Special cases	8
5	Parke Taylor formula	14
6	Implementation	15
6.1	Conventions	15
6.2	Particle class	15
6.3	Berends-Giele current	16
6.4	Parke Taylor formula	18
6.5	Comparison	18
6.6	Further improvements	18
6.6.1	Parallel processing	18
6.6.2	C loops	18
6.6.3	C arrays	18
7	Summary	19

Appendices	21
A QCD Feynman rules	21
B Color ordered QCD Feynman rules	21
C Code	21

1 Introduction

For predictions of multi-jet processes at high-energy colliders, we need the knowledge of scattering amplitudes with multiple final parton states in QCD. They are evident in the prediction of cross sections. Writing down all involved Feynman diagrams in the perturbative expansion for such a process can become very cumbersome, since they are armed with a complicated color algebra any many kinematic variables. The perturbative expansion is in the QCD coupling α_s , which is in the applicable scale at high energies [Deur et al., 2016]. In this report such amplitudes are discussed at tree level with an arbitrary number of gluons.

In section 2 the spinor helicity formalism for massless vector bosons is briefly introduced, such that later in the report, we are able to correctly formulate the problems. It turns out that MHV amplitudes can be simplified massively using that formalism.

In a second preliminary in section 3 a concept called color ordering is introduced. With it, we can decompose tree level amplitudes into parts that are not dependent on color anymore. These parts are then called color ordered amplitudes or partial amplitudes. Such objects have much simpler kinematic properties than the full amplitude, since they only receive contributions from diagrams with a particular ordering of the external legs. The remaining components of the full amplitude consist of pure traces that can be evaluated quite fast.

Section 4 the Berends-Giele current is presented. This is a recursive approach to calculate partial amplitudes. Even if the analytical expressions received by this method are not easy to simplify, the numerical implementation is simple and evaluation is quite efficient.

In section 5 a formula for MHV amplitudes is introduced and proofed - the Parke-Taylor formula. MHV amplitudes are maximally helicity violating amplitudes that are not vanishing. In case of pure gluonic scattering amplitudes, they consist of $n - 2$ gluons with the same helicity and 2 with the opposite helicity.

Eventually in section 6 a concrete implementation of both - the Berends-Giele current and the Parke-Taylor formula - is discussed. Both formulas are compared in terms of performance. The computation time required by these two methods scales with the number of external legs. The linear scaling of the Parke-Taylor formula of course ends up in a faster calculation than with the Berends-Giele recursion which scales with $n!$, because it's recursive.

Finally in the appendices there are - for the sake of completeness - Feynman rules for QCD with and without color ordering in Feynman gauge.

2 The Spinor helicity formalism for massless vector bosons

In calculating amplitudes in QCD one question that arises is: What are the right kinematic variables for scattering amplitudes? Typically one uses the 4-momenta of the involved particles. For the momenta, specially their Lorentz-invariant products $s_{ij} = (p_i + p_j)^2$ should be considered. Also the helicity basis [Berends et al., 1981] is very convenient, where the spin is quantized along the axis of the momentum p . For positive helicity, the spin points in the same direction as p , for negative helicity in the opposite direction. We have $u_{\pm}(p)$, which is a 4-spinor. We know that for massless vectors there is a smaller representation which is two dimensional. For a four momentum p and a helicity h , we get two 2-spinors:

$$p^\mu, h \rightarrow u_-(p) = \begin{pmatrix} \lambda_{p\alpha} \\ 0 \\ 0 \end{pmatrix} \leftrightarrow |p\rangle = \lambda_{p\alpha} = \begin{pmatrix} \cdot \\ \cdot \end{pmatrix}$$

$$u_+(p) = \begin{pmatrix} 0 \\ 0 \\ \tilde{\lambda}_p^{\dot{\alpha}} \end{pmatrix} \leftrightarrow [p] = \tilde{\lambda}_p^{\dot{\alpha}} = \begin{pmatrix} \cdot \\ \cdot \end{pmatrix}.$$

In this representation, $|p\rangle$ is a column vector with helicity $-1/2$, whereas $[p]$ is a column vector with helicity $+1/2$. The dotted and undotted indices correspond to two different spinors

representations of the Lorentz-group. Positive u_{\pm} and negative v_{\pm} energy solutions are not distinct, since $p^2 = 0$.

Using this formalism, the massless Dirac equation can then be written as

$$\not{p}|p\rangle = \not{p}|p] = 0.$$

Definition 2.1 (Short notation of spinors). As a short notation, when we are dealing with n particles, we have n four momenta labelled p_i , helicities labelled h_i and maybe adjoint colors labelled a_i , where $i \in \{1, \dots, n\}$. The i -th spinor can then be written as

$$\begin{aligned} |p_i\rangle &\equiv |i\rangle, \\ |p_i] &\equiv |i]. \end{aligned}$$

Next, we want to build some Lorentz-invariant quantities and define the spinor product.

Definition 2.2 (Spinor product). The spinor product can be performed between spinors of the same representation

$$\begin{aligned} \langle ij \rangle &\equiv \lambda_i^\alpha \epsilon_{\alpha\beta} \lambda_j^\beta, \\ [ij] &\equiv \tilde{\lambda}_{i\dot{\alpha}} \epsilon^{\dot{\alpha}\dot{\beta}} \tilde{\lambda}_{j\dot{\beta}}, \end{aligned}$$

where the ϵ matrices are anti-symmetric and given by

$$\epsilon_{\alpha\beta} = \epsilon_{\dot{\alpha}\dot{\beta}} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad \epsilon^{\alpha\beta} = \epsilon^{\dot{\alpha}\dot{\beta}} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

Notice that, the spinor products are anti-symmetric by definition, $\langle ii \rangle = [ii] = 0$.

Definition 2.3 (Momentum products and sums). To deal with multiple momenta later, we define some intermediate quantities

$$P_{i,j} = p_i + \dots + p_j, \tag{2.1}$$

$$s_{ij} = (p_i + p_j)^2. \tag{2.2}$$

Definition 2.4 (Pauli matrices). The Pauli matrices are defined as

$$\sigma^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma^1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

and

$$\begin{aligned} \sigma^\mu &= (1, \sigma^1, \sigma^2, \sigma^3), \\ \bar{\sigma}^\mu &= (1, -\sigma^1, -\sigma^2, -\sigma^3). \end{aligned}$$

2.1 Important identities and definitions

Using the formalism defined above, there are now some identities that follow.

Lemma 2.1 (Fierz identity).

$$\langle i|\sigma^\mu|j\rangle\langle k|\sigma_\mu|l\rangle = 2\langle ik\rangle[lj] \tag{2.3}$$

Lemma 2.2 (Charge conjugation identity).

$$\langle i|\sigma^\mu|j\rangle = [j|\bar{\sigma}^\mu|i\rangle \tag{2.4}$$

Lemma 2.3 (Schouten identity).

$$\langle ij\rangle\langle k| + \langle jk\rangle\langle i| + \langle ki\rangle\langle j| = 0 \tag{2.5}$$

Lemma 2.4 (Squaring identity).

$$\langle ij \rangle [ji] = 2p_i \cdot p_j = s_{ij} \quad (2.6)$$

Definition 2.5 (Feynman slash notation). Feynman slash notation has also a representation in the new formalism

$$\not{p} = \sigma^\mu p_\mu = p^\mu \sigma_\mu = |p\rangle[p] + |p]\langle p|. \quad (2.7)$$

Using the spinor products, we are now able to build polarisation vectors for massless vector bosons with helicity $h = \pm 1$ [Xu et al., 1987],

$$\varepsilon_+^\mu(p, q) = \frac{1}{\sqrt{2}} \frac{\langle q | \sigma^\mu | p \rangle}{\langle qp \rangle} \quad \text{and} \quad \varepsilon_-^\mu(p, q) = -\frac{1}{\sqrt{2}} \frac{[q | \bar{\sigma}^\mu | p \rangle}{[qp]}, \quad (2.8)$$

where q is a reference vector with massless momentum $q^2 = 0$. ε_+ is a state with helicity $h = +1$, because $|p\rangle$ in the numerator gives $+1/2$ and $|p\rangle$ gives $-1/2$, but in the denominator it also gives $+1/2$. The choice of q is a gauge choice and absolutely free as long as $q^2 = 0$ and q is not proportional to p , because else the spinor product in the denominator would vanish. This choice originates from the gauge invariance of the gauge field A_μ^a in the Lagrangian. The results in the end should not depend on any reference momentum q , but intermediate results do in general. When choosing a different q' as reference momentum, the polarisation vectors are shifted by an amount proportional to p^μ .

3 Color Ordering

Color ordering is a concept to treat the color component of scattering amplitudes. We want to decompose the problem into subproblems. This can be achieved by color ordering. In the Feynman rules of QCD, there is a lot of color dependency (see appendix A).

3.1 QCD gauge group

The gauge group of QCD is $SU(N_c)$ - unitary $N_c \times N_c$ matrices with unit determinant - with $N_c = 3$ the number of colors. The Lie algebra of this group is $\mathfrak{su}(N_c)$ - traceless hermitian $N_c \times N_c$ matrices. The unit determinant converts to tracelessness, when going to the Lie algebra. The dimension of both the gauge group and its Lie algebra is $N_c^2 - 1$ (the tracelessness condition removes one dimension). We therefore have a basis of the Lie algebra - the generators of the gauge group - denoted by T^a , where a is the adjoint color index going from 1 to $N_c^2 - 1 = 8$.

Definition 3.1 (Normalisation). We define the normalisation of the generators as

$$\text{tr}(T^a T^b) = \delta^{ab}. \quad (3.1)$$

This normalisation, though uncommon in literature, prevents later expressions from unnecessary $\sqrt{2}$ appearing.

Proposition 3.1 (Fierz identity). The generators T^a of the gauge group satisfy the following identity [Berends and Giele, 1987]:

$$(T^a)_{i_1}^{\bar{j}_1} (T^a)_{i_2}^{\bar{j}_2} = \delta_{i_1}^{\bar{j}_2} \delta_{i_2}^{\bar{j}_1} - \frac{1}{N_c} \delta_{i_1}^{\bar{j}_1} \delta_{i_2}^{\bar{j}_2} \quad (3.2)$$

Proof. Equation (3.2) is just the statement that the T^a together with the identity form a basis of the $N_c \times N_c$ hermitian matrices. The T^a are a basis of $N_c \times N_c$ traceless hermitian matrices, since the identity is linearly independent of all T^a . An arbitrary hermitian matrix A can therefore be written as

$$A = \sum_{a=1}^{N_c^2-1} b_a T^a + b_n id,$$

where $n = N_c^2$. We now apply the trace on both sides of that equation:

$$tr(A) = b_n N_c,$$

where we have used the fact that the T^a are traceless and the trace of the identity is its dimension. We obtain the n -th coefficient $b_n = \text{tr}(A)/N_c$. We can also multiply both sides with T^c and then take the trace and obtain the remaining coefficients:

$$tr\left(AT^c\right)=b_c,$$

where we again used the tracelessness of the T^a and their normalisation (definition 3.1). Therefore we can write the i, j -th entry of the matrix A as

$$A_i{}^j = \sum_{a=1}^{N_c^2-1} (T^a)_k{}^l A_l{}^k (T^a)_i{}^j + \frac{\text{tr}(A)}{N_c} \delta_i{}^j.$$

Using $tr(A) = tr(Aid) = A_l^k \delta_k^l$ and $A_i^j = A_l^k \delta_i^l \delta_j^k$, we find

$$A_l{}^k \left[\sum_{a=1}^{N_c^2-1} (T^a)_k{}^l (T^a)_i{}^j + \frac{1}{N_c} \delta_k{}^l \delta_i{}^j - \delta_i{}^l \delta_j{}^k \right] = 0,$$

that holds for all A_l^k , which implies that the term in the square brackets must be 0.

On the other hand, the generators also satisfy the relation

$$[T^a, T^b] = i\sqrt{2} \sum_{c=1}^{N_c^2-1} f^{abc} T^c, \quad (3.3)$$

where the f^{abc} are the anti-symmetric structure constants. The above relation can be solved for the f^{abc} : starting with equation (3.3) and multiplying from the right side with another T^d and taking the trace, we obtain

$$tr(T^a T^b T^d) - tr(T^b T^b T^d) = i\sqrt{2} \sum_{c=1}^{N_c^2-1} f^{abc} tr(T^c T^d). \quad (3.4)$$

Using the definition of the normalisation 3.1, the sum on the right hand side is over a Kronecker delta, therefore only one term is not vanishing

$$f^{abc} = -\frac{i}{\sqrt{2}} \left(\text{tr}(T^a T^b T^c) - \text{tr}(T^b T^a T^c) \right). \quad (3.5)$$

The expression for the structure constants (3.5) can be written pictorially in a very intuitive way taking only the color component from the Feynman rules into account:

$$\begin{aligned}
& \text{Diagram 1} = \text{Diagram 2} - \text{Diagram 3} \\
& i\sqrt{2}f^{abc} = (T^a)_{\bar{k}}(T^b)_i(T^c)_j{}^{\bar{k}} - (T^a)_i{}^{\bar{k}}(T^b)_{\bar{j}}(T^c)_k{}^{\bar{j}} \\
& i\sqrt{2}f^{abc} = \text{tr}(T^a T^b T^c) - \text{tr}(T^a T^c T^b).
\end{aligned}
\tag{3.6}$$

The structure constant f^{abc} is proportional to the 3-gluon vertex according to the Feynman rules (see appendix A), whereas the loop diagrams on the right hand side produce traces of all involved generators T^a using the Feynman rule for the quark-gluon vertex. The Fierz identity (3.2) can also be illustrated in a similar way:

$$\begin{aligned}
\begin{array}{c} \bar{j}_1 \\ i_2 \\ i_1 \\ \bar{j}_2 \end{array} \begin{array}{c} \text{---} a \text{---} \\ \text{---} \end{array} \begin{array}{c} i_2 \\ \bar{j}_2 \\ i_1 \\ \bar{j}_1 \end{array} &= \begin{array}{c} \bar{j}_1 \\ i_1 \\ \bar{j}_2 \\ i_2 \end{array} \begin{array}{c} \text{---} \text{---} \\ \text{---} \text{---} \end{array} \begin{array}{c} i_2 \\ \bar{j}_2 \\ i_1 \\ \bar{j}_1 \end{array} - \frac{1}{N_c} \begin{array}{c} \bar{j}_1 \\ i_1 \\ \bar{j}_2 \\ i_2 \end{array} \begin{array}{c} \text{---} \end{array} \begin{array}{c} i_2 \\ \bar{j}_2 \\ i_1 \\ \bar{j}_1 \end{array} \\
(T^a)_{i_1}{}^{\bar{j}_1} (T^a)_{i_2}{}^{\bar{j}_2} &= \delta_{i_1}{}^{\bar{j}_2} \delta_{i_2}{}^{\bar{j}_1} - \frac{1}{N_c} \delta_{i_1}{}^{\bar{j}_1} \delta_{i_2}{}^{\bar{j}_2}.
\end{aligned} \tag{3.7}$$

Again the Feynman rules from appendix A where used to derive the pictures. The two equations can therefore be used to rewrite some guonic amplitudes. Let's see this in an example.

3.2 Color decomposition

Example 3.1 (4-gluon vertex). In this example, we decompose the 4-gluon vertex into multiple diagrams. We use the two pictorial representations (3.6) and (3.7). The 4-vertex can be decomposed into 3 components of each two 3-vertices by just looking at the Feynman rule:

$$\begin{array}{c} b \\ c \\ a \\ d \end{array} \begin{array}{c} \text{---} \text{---} \\ \text{---} \text{---} \end{array} \begin{array}{c} c \\ b \\ d \\ a \end{array} = \begin{array}{c} b \\ c \\ a \\ d \end{array} \begin{array}{c} \text{---} e \text{---} \\ \text{---} \end{array} \begin{array}{c} c \\ b \\ d \\ a \end{array} + \begin{array}{c} b \\ c \\ a \\ d \end{array} \begin{array}{c} \text{---} e \text{---} \\ \text{---} \end{array} \begin{array}{c} c \\ b \\ d \\ a \end{array} + \begin{array}{c} b \\ c \\ a \\ d \end{array} \begin{array}{c} \text{---} e \text{---} \\ \text{---} \end{array} \begin{array}{c} c \\ b \\ d \\ a \end{array} \\
f^{abe} f^{cde} \quad f^{ade} f^{bce} \quad f^{ace} f^{bde}.
\end{array} \tag{3.8}$$

We decompose the vertex of the first diagram proportional to $f^{abe} f^{cde}$ (the others are analogue)

$$\begin{array}{c} b \\ c \\ a \\ d \end{array} \begin{array}{c} \text{---} e \text{---} \\ \text{---} \end{array} \begin{array}{c} c \\ b \\ d \\ a \end{array} = \begin{array}{c} b \\ c \\ a \\ d \end{array} \begin{array}{c} \text{---} \text{---} \\ \text{---} \end{array} \begin{array}{c} c \\ b \\ d \\ a \end{array} \pm \text{permutations} \\
= \begin{array}{c} b \\ c \\ a \\ d \end{array} \begin{array}{c} \text{---} \text{---} \\ \text{---} \end{array} \begin{array}{c} c \\ b \\ d \\ a \end{array} \pm \text{permutations} \\
= \begin{array}{c} b \\ c \\ a \\ d \end{array} \begin{array}{c} \text{---} \text{---} \\ \text{---} \end{array} \begin{array}{c} c \\ b \\ d \\ a \end{array} \pm \text{permutations},
\end{array}$$

where we have used equation (3.6) in the first step and (3.7) in the second. It can be seen that the 4-gluon vertex decomposes into many permutations of traces of generators.

This also holds for n gluons,

$$\begin{array}{c} a_2 \\ a_3 \\ \vdots \\ a_1 \\ a_n \end{array} \begin{array}{c} \text{---} \text{---} \\ \text{---} \end{array} \begin{array}{c} a_3 \\ a_2 \\ \vdots \\ a_n \\ a_1 \end{array} = \begin{array}{c} a_2 \\ a_3 \\ \vdots \\ a_1 \\ a_n \end{array} \begin{array}{c} \text{---} \text{---} \\ \text{---} \end{array} \begin{array}{c} a_3 \\ a_2 \\ \vdots \\ a_n \\ a_1 \end{array} \pm \text{permutations},$$

because the scattering amplitude of n gluons consists of 3- and 4-vertices. These vertices can be decomposed according to equation (3.6), equation (3.7) and the above example. In the end, one always ends up in many diagrams whose color part to the amplitude can be written in traces.

Definition 3.2 (Color decomposition). This gives motivation to the following formula for a full pure gluonic tree amplitude:

$$\mathcal{A}_n^{tree}(\{p_i, h_i, a_i\}) = g^{n-2} \sum_{\sigma \in S_n/Z_n} \text{tr}(T^{a_{\sigma(1)}} \dots T^{a_{\sigma(n)}}) A_n^{tree}(p_{\sigma(1)}, \dots, p_{\sigma(n)}, h_{\sigma(1)}, \dots, h_{\sigma(n)}), \quad (3.9)$$

where n is the number of involved gluons, g is the strong coupling constant, S_n is the symmetric group consisting of all permutations of n objects, Z_n is the cyclic group consisting of only cyclic permutations of n objects, S_n/Z_n sweeps out all distinct cyclic permutations and consists of $(n-1)!$ elements, T^a are the generators of the Lie algebra. As seen, the full amplitude depends on the momenta p_i , helicities h_i and color configurations a_i of all n gluons. The A_n^{tree} objects are not defined yet, these are the so called partial amplitudes and it has to be noticed that they do not depend on any color anymore. The whole color part of the amplitude is carried out into traces of generators and their permutations. Indeed the partial amplitudes still depend on the momenta and helicities, but in an ordered fashion. They receive contributions only from Feynman diagrams with a particular ordering of gluons. This means that the partial amplitudes are color ordered by construction, they need a fixed ordering of external legs. In literature they are also called color ordered amplitudes, dual amplitudes or primitive amplitudes (if one does not consider loops).

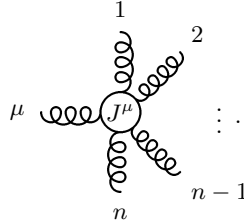
To construct the partial amplitudes, there is now a new set of Feynman rules for this: the color ordered Feynman rules (see section B). Notice that, the new Feynman rules now do not depend on any color anymore (no f^{abc} or T^a).

4 Berends-Giele current

We are now ready to formulate the problem. The goal is to calculate the partial amplitude of n gluons scattering at tree level. The recursion idea is the following.

4.1 Recursion idea

We define an auxiliary quantity called J^μ which we can contract with an appropriate polarisation vector to obtain the partial amplitude. J^μ is called the Berends-Giele current or off-shell current and it consists of the sum of all color ordered Feynman graphs



The off-shell propagator of the external leg μ is also included in J^μ . A pictorial representation of J^μ can be seen in figure 1.

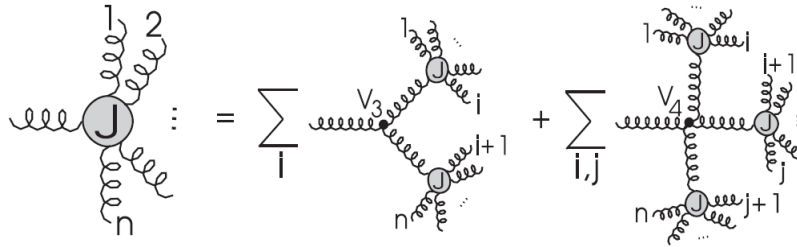


Figure 1: Graphical representation of the recursion relation for the off-shell current J^μ .

The image can be understood as follows. When we follow the off-shell leg μ into the current, we could approach either a 3-vertex or a 4-vertex.

If it's a 3-vertex, then the remaining external legs from 1 to n must be split into two subsets to connect the next current to it. Since the diagrams are color ordered, we cannot split the set of n legs into two arbitrary sets. The sets must not be empty and must consist of only consequent numbers - this follows from the color ordering. So there are n possibilities to split the set. If we therefore sum over these n possibilities we get the first sum in the term of figure 1.

If it's a 4-vertex, then the remaining external legs must be split into three subsets. The rules for the subsets are the same as for the 3-vertex. We therefore have $n(n-1)$ possibilities. This can be written in terms of a double sum over two indices i and j and we obtain the second sum in figure 1.

Therefore the current can be written as

$$J^\mu(1, \dots, n) = \frac{-i}{P_{1,n}^2} \left[\sum_{i=1}^{n-1} V_3^{\mu\nu\rho}(P_{1,i}, P_{i+1,n}) J_\nu(1, \dots, i) J_\rho(i+1, \dots, n) + \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} V_4^{\mu\nu\rho\sigma} J_\nu(1, \dots, i) J_\rho(i+1, \dots, j) J_\sigma(j+1, \dots, n) \right] \quad (4.1)$$

where the V_i are the colour ordered gluon self interaction vertices (see appendix B),

$$V_3^{\mu\nu\rho}(P, Q) = \frac{i}{\sqrt{2}} (\eta^{\nu\rho} (P - Q)^\mu + 2\eta^{\rho\mu} Q^\nu - 2\eta^{\mu\nu} P^\rho), \quad (4.2)$$

$$V_4^{\mu\nu\rho\sigma} = \frac{i}{2} (2\eta^{\mu\rho} \eta^{\nu\sigma} - \eta^{\mu\nu} \eta^{\rho\sigma} - \eta^{\mu\sigma} \eta^{\nu\rho}), \quad (4.3)$$

and the currents with only one gluon are the corresponding polarisation vectors (see equation (2.8))

$$J^\mu(i^+) = \varepsilon_+^\mu(p_i, q_i) = \frac{1}{\sqrt{2}} \frac{\langle q_i | \sigma^\mu | p_i \rangle}{\langle q_i p_i \rangle}, \quad J_\mu(i^+) = \varepsilon_{+\mu}(p_i, q_i) = \frac{1}{\sqrt{2}} \frac{\langle q_i | \sigma_\mu | p_i \rangle}{\langle q_i p_i \rangle}, \quad (4.4a)$$

$$J^\mu(i^-) = \varepsilon_-^\mu(p_i, q_i) = -\frac{1}{\sqrt{2}} \frac{[q_i | \bar{\sigma}^\mu | p_i \rangle}{[q_i p_i]}, \quad J_\mu(i^-) = \varepsilon_{-\mu}(p_i, q_i) = -\frac{1}{\sqrt{2}} \frac{[q_i | \bar{\sigma}_\mu | p_i \rangle}{[q_i p_i]}. \quad (4.4b)$$

Notice that, the off-shell current J^μ is not gauge invariant - it depends on the reference momenta q_i . The question now is how to get the partial amplitude with this current? Partial amplitudes with $n+1$ external legs are obtained by calculating $J_\mu(1^\pm, \dots, n^\pm)$ for n external legs. Then the off-shell propagator $\frac{-i}{P_{1,n}^2}$ is cut off. The remaining expression is contracted with the $(n+1)$ -th polarisation vectors $\epsilon_\pm^\mu(p_{n+1}, q_{n+1})$. This process is called LSZ-reduction [Lehmann et al., 1955]. The full amplitude can be obtained by applying equation (3.9).

4.2 Special cases

Proposition 4.1. If the helicities of all participating gluons are equal, then equation (4.1) reduces to the following form [Berends and Giele, 1988]:

$$J^\mu(1^+, 2^+, \dots, n^+) = \frac{\langle q | \sigma^\mu \not{P}_{1,n} | q \rangle}{\sqrt{2} \langle q1 \rangle \langle 12 \rangle \cdots \langle n-1, n \rangle \langle nq \rangle}. \quad (4.5)$$

where the reference momentum q is the same for all gluons.

Proof. The proof is given by induction. We begin with the base case $n = 2$:

$$\begin{aligned} J^\mu(1^+, 2^+) &= \frac{-i}{P_{1,2}^2} V_3^{\mu\nu\rho}(P_{1,1}, P_{2,2}) J_\nu(1^+) J_\rho(2^+) \\ &= \frac{-i}{P_{1,2}^2} \frac{i}{\sqrt{2}} (\eta^{\nu\rho} (p_1 - p_2)^\mu + 2\eta^{\rho\mu} p_2^\nu - 2\eta^{\mu\nu} p_1^\rho) \frac{1}{\sqrt{2}} \frac{\langle q | \sigma_\nu | 1 \rangle}{\langle q1 \rangle} \frac{1}{\sqrt{2}} \frac{\langle q | \sigma_\rho | 2 \rangle}{\langle q2 \rangle} \\ &= \frac{-i}{P_{1,2}^2} \frac{i}{\sqrt{2}} (\eta^{\nu\rho} (p_1 - p_2)^\mu + 2\eta^{\rho\mu} p_2^\nu - 2\eta^{\mu\nu} p_1^\rho) \varepsilon_{+\nu}(p_1, q) \varepsilon_{+\rho}(p_2, q) \end{aligned}$$

We will obtain 3 terms. The first one will vanish, after applying the Fierz identity (2.3), since $\langle qq \rangle = [qq] = 0$,

$$\begin{aligned}\varepsilon_+(p_1, q) \cdot \varepsilon_+(p_2, q) (p_1 - p_2)^\mu &= \frac{1}{2} \frac{\langle q|\sigma^\rho|1\rangle\langle q|\sigma_\rho|2\rangle}{\langle q1\rangle\langle q2\rangle} (p_1 - p_2)^\mu \\ &= \frac{\langle qq\rangle[12]}{\langle q1\rangle\langle q2\rangle} (p_1 - p_2)^\mu \\ &= 0.\end{aligned}$$

The second term evaluates to

$$\begin{aligned}2\varepsilon_+^\mu(p_2, q) p_2 \cdot \varepsilon_+(p_1, q) &= 2 \frac{1}{\sqrt{2}} \frac{\langle q|\sigma^\mu|2\rangle}{\langle q2\rangle} \frac{1}{\sqrt{2}} \frac{p_2^\nu \langle q|\sigma_\nu|1\rangle}{\langle q1\rangle} \\ &= \frac{\langle q|\sigma^\mu|2\rangle}{\langle q1\rangle\langle q2\rangle} \frac{1}{2} \langle 2|\sigma^\nu|2\rangle \langle q|\sigma_\nu|1\rangle \\ &= \frac{\langle q|\sigma^\mu|2\rangle}{\langle q1\rangle\langle q2\rangle} \langle 2q\rangle[12] \\ &= \frac{\langle q|\sigma^\mu|2\rangle\langle 2q\rangle}{\langle q1\rangle\langle q2\rangle} \frac{P_{1,2}^2}{\langle 21\rangle} \\ &= \frac{\langle q|\sigma^\mu \not{p}_2|q\rangle P_{1,2}^2}{\langle q1\rangle\langle 12\rangle\langle 2q\rangle}.\end{aligned}$$

Using this result for the third term gives (exchanging 1 and 2 and adding an overall minus)

$$\begin{aligned}-2\varepsilon_+^\mu(p_1, q) p_1 \cdot \varepsilon_+(p_2, q) &= -\frac{\langle q|\sigma^\mu \not{p}_1|q\rangle P_{1,2}^2}{\langle q2\rangle\langle 21\rangle\langle 1q\rangle} \\ &= \frac{\langle q|\sigma^\mu \not{p}_1|q\rangle P_{1,2}^2}{\langle q1\rangle\langle 12\rangle\langle 2q\rangle}.\end{aligned}$$

Therefore the whole term evaluates to

$$J^\mu(1^+, 2^+) = \frac{\langle q|\sigma^\mu \not{P}_{1,2}|q\rangle}{\sqrt{2}\langle q1\rangle\langle 12\rangle\langle 2q\rangle},$$

which equals equation (4.5) for $n = 2$.

Now, we perform the induction step. Assuming that the proposition holds for $k \leq n-1$, we want to find $J^\mu(1^+, 2^+, \dots, n^+)$. The V_4 term vanishes, because of

$$\begin{aligned}\eta^{\nu\sigma} J_\nu(1, \dots, i) J_\sigma(j+1, \dots, n) &\propto \eta^{\nu\sigma} \langle q|\sigma_\nu \not{P}_{1,i}|q\rangle \langle q|\sigma_\sigma \not{P}_{j+1,n}|q\rangle \\ &= \langle q|\sigma^\sigma \not{P}_{1,i}|q\rangle \langle q|\sigma_\sigma \not{P}_{j+1,n}|q\rangle \\ &\propto \langle qq\rangle \\ &= 0.\end{aligned}$$

The same argument also applies to $\eta^{\rho\sigma} J_\rho(i+1, \dots, j) J_\sigma(j+1, \dots, n) = 0$ and $\eta^{\nu\rho} J_\nu(1, \dots, i) J_\rho(i+1, \dots, j) = 0$. Thus the whole V_4 term evaluates to zero. The first term in V_3 also vanishes, because of the same argument, $\eta^{\nu\rho} J_\nu(1, \dots, i) J_\rho(i+1, \dots, n) = 0$. We are left with only the two remaining V_3 terms,

$$\begin{aligned}J^\mu(1^+, 2^+, \dots, n^+) &= \frac{-i}{P_{1,n}^2} \sum_{i=1}^{n-1} V_3^{\mu\nu\rho}(P_{1,i}, P_{i+1,n}) J_\nu(1, \dots, i) J_\rho(i+1, \dots, n) \\ &= \frac{\sqrt{2}}{P_{1,n}^2} \sum_{i=1}^{n-1} (\eta^{\rho\mu} P_{i+1,n}^\nu - \eta^{\mu\nu} P_{1,i}^\rho) J_\nu(1, \dots, i) J_\rho(i+1, \dots, n).\end{aligned}$$

By assumption, equation (4.5) holds for values smaller than or equal to $n-1$, hence we can substitute equation (4.5) for J_ν and J_ρ ,

$$\begin{aligned}
& \frac{\sqrt{2}}{P_{1,n}^2} \sum_{i=1}^{n-1} (\eta^{\rho\mu} P_{i+1,n}^\nu - \eta^{\mu\nu} P_{1,i}^\rho) \frac{\langle q | \sigma_\nu \mathcal{P}_{1,i} | q \rangle}{\sqrt{2} \langle q1 \rangle \langle 12 \rangle \cdots \langle i-1, i \rangle \langle iq \rangle} \frac{\langle q | \sigma_\rho \mathcal{P}_{i+1,n} | q \rangle}{\sqrt{2} \langle q, i+1 \rangle \cdots \langle n-1, n \rangle \langle nq \rangle} \\
&= \frac{1}{\sqrt{2} P_{1,n}^2 \langle q1 \rangle \langle 12 \rangle \cdots \langle n-1, n \rangle \langle nq \rangle} \sum_{i=1}^{n-1} \frac{\langle i, i+1 \rangle}{\langle iq \rangle \langle q, i+1 \rangle} \\
&\quad \times \left(\langle q | \mathcal{P}_{i+1,n} \mathcal{P}_{1,i} | q \rangle \langle q | \sigma^\mu \mathcal{P}_{i+1,n} | q \rangle - \langle q | \sigma^\mu \mathcal{P}_{1,i} | q \rangle \langle q | \mathcal{P}_{1,i} \mathcal{P}_{i+1,n} | q \rangle \right) \\
&= \frac{\langle q | \sigma^\mu \mathcal{P}_{1,n} | q \rangle}{\sqrt{2} P_{1,n}^2 \langle q1 \rangle \langle 12 \rangle \cdots \langle n-1, n \rangle \langle nq \rangle} \sum_{i=1}^{n-1} \frac{\langle i, i+1 \rangle}{\langle iq \rangle \langle q, i+1 \rangle} \\
&\quad \times \left(\langle q | \sigma^\mu \mathcal{P}_{i+1,n} | q \rangle \langle q | \mathcal{P}_{1,i} \mathcal{P}_{i+1,n} | q \rangle - \langle q | \mathcal{P}_{i+1,n} \mathcal{P}_{1,i} | q \rangle \langle q | \sigma^\mu \mathcal{P}_{1,i} | q \rangle \right)
\end{aligned}$$

We use that $\not{a}\not{b} = 2ab\mathbb{I} - \not{b}\not{a}$ and therefore $\langle q | \mathcal{P}_{1,i} \mathcal{P}_{i+1,n} | q \rangle = -\langle q | \mathcal{P}_{i+1,n} \mathcal{P}_{1,i} | q \rangle$. We obtain

$$\begin{aligned}
& \frac{1}{\sqrt{2} P_{1,n}^2 \langle q1 \rangle \langle 12 \rangle \cdots \langle n-1, n \rangle \langle nq \rangle} \sum_{i=1}^{n-1} \frac{\langle i, i+1 \rangle}{\langle iq \rangle \langle q, i+1 \rangle} \langle q | \mathcal{P}_{i+1,n} \mathcal{P}_{1,i} | q \rangle \\
& \times \underbrace{\left(\langle q | \sigma^\mu \mathcal{P}_{i+1,n} | q \rangle + \langle q | \sigma^\mu \mathcal{P}_{1,i} | q \rangle \right)}_{= \langle q | \sigma^\mu (\mathcal{P}_{i+1,n} + \mathcal{P}_{1,i}) | q \rangle = \langle q | \sigma^\mu \mathcal{P}_{1,n} | q \rangle}
\end{aligned}$$

Also

$$\begin{aligned}
\langle q | \mathcal{P}_{i+1,n} \mathcal{P}_{1,i} | q \rangle &= \langle q | \mathcal{P}_{i+1,n} (\mathcal{P}_{1,n} - \mathcal{P}_{i+1,n}) | q \rangle \\
&= \langle q | \mathcal{P}_{i+1,n} \mathcal{P}_{1,n} | q \rangle - \langle q | \mathcal{P}_{i+1,n}^2 | q \rangle \\
&= \langle q | \mathcal{P}_{i+1,n} \mathcal{P}_{1,n} | q \rangle - \mathcal{P}_{i+1,n}^2 \langle qq \rangle \\
&= \langle q | \mathcal{P}_{i+1,n} \mathcal{P}_{1,n} | q \rangle,
\end{aligned}$$

where we have used $\not{p}\not{p} = p^2\mathbb{I}$. The term then simplifies to

$$\frac{\langle q | \sigma^\mu \mathcal{P}_{1,n} | q \rangle}{\sqrt{2} P_{1,n}^2 \langle q1 \rangle \langle 12 \rangle \cdots \langle n-1, n \rangle \langle nq \rangle} \left[\sum_{i=1}^{n-1} \frac{\langle i, i+1 \rangle}{\langle iq \rangle \langle q, i+1 \rangle} \langle q | \mathcal{P}_{i+1,n} \right] \mathcal{P}_{1,n} | q \rangle. \quad (4.6)$$

The term in the square brackets can be simplified by the Shouten identity (2.5):

$$\sum_{i=1}^{n-1} \frac{\langle i, i+1 \rangle}{\langle iq \rangle \langle q, i+1 \rangle} \langle q | \mathcal{P}_{i+1,n} = \sum_{i=1}^{n-1} \left[\frac{-\langle i+1, q \rangle \langle i | \mathcal{P}_{i+1,n}}{\langle iq \rangle \langle q, i+1 \rangle} + \frac{-\langle qi \rangle \langle i+1 | \mathcal{P}_{i+1,n}}{\langle iq \rangle \langle q, i+1 \rangle} \right] \quad (4.7)$$

$$= \sum_{i=1}^{n-1} \left[\frac{\langle i | \mathcal{P}_{i,n}}{\langle iq \rangle} - \frac{\langle i+1 | \mathcal{P}_{i+1,n}}{\langle i+1, q \rangle} \right] \quad (4.8)$$

$$= \frac{\langle 1 | \mathcal{P}_{1,n}}{\langle 1q \rangle} - \frac{\langle n | \mathcal{P}_n}{\langle nq \rangle} \quad (4.9)$$

$$= \frac{\langle 1 | \mathcal{P}_{1,n}}{\langle 1q \rangle}. \quad (4.10)$$

To remove the sum, we concluded that the second term in the sum always cancels with the first term of next summand. Thus, leaving only the first term with $i = 1$ and the second term with $i = n - 1$. Using this on equation (4.6), we get the desired formula (4.5) for arbitrary n ,

$$\begin{aligned}
J^\mu(1^+, 2^+, \dots, n^+) &= \frac{\langle q | \sigma^\mu \mathcal{P}_{1,n} | q \rangle}{\sqrt{2} P_{1,n}^2 \langle q1 \rangle \langle 12 \rangle \cdots \langle n-1, n \rangle \langle nq \rangle} \frac{\langle 1 | \mathcal{P}_{1,n}^2 | q \rangle}{\langle 1q \rangle} \\
&= \frac{\langle q | \sigma^\mu \mathcal{P}_{1,n} | q \rangle}{\sqrt{2} \langle q1 \rangle \langle 12 \rangle \cdots \langle n-1, n \rangle \langle nq \rangle}.
\end{aligned}$$

□

Proposition 4.2. If only the helicity of the first gluon is negative and the rest are positive, then equation (4.1) reduces to the following form [Berends and Giele, 1988]:

$$J^\mu(1^-, 2^+, \dots, n^+) = \frac{\langle 1 | \sigma^\mu \not{P}_{2,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle n1 \rangle} \sum_{m=3}^n \frac{\langle 1 | \not{P}_m \not{P}_{1,m} | 1 \rangle}{P_{1,m-1}^2 P_{1,m}^2}, \quad (4.11)$$

where the reference momenta are $q_1 = p_2$ and $q_2 = \cdots = q_n = p_1$. So the reference momenta for the gluons with same helicities are equal.

Proof. The proof is given by induction. We begin with the base case $n = 2$:

$$\begin{aligned} J^\mu(1^-, 2^+) &= \frac{-i}{P_{1,2}^2} V_3^{\mu\nu\rho}(P_{1,1}, P_{2,2}) J_\nu(1^-) J_\rho(2^+) \\ &= \frac{-i}{P_{1,2}^2} \frac{i}{\sqrt{2}} (\eta^{\nu\rho} (p_1 - p_2)^\mu + 2\eta^{\rho\mu} p_2^\nu - 2\eta^{\mu\nu} p_1^\rho) \varepsilon_{-\nu}(p_1, p_2) \varepsilon_{+\rho}(p_2, p_1). \end{aligned}$$

The first term evaluates to zero after applying the Fierz identity,

$$\begin{aligned} \varepsilon_-^\rho(p_1, p_2) \varepsilon_{+\rho}(p_2, p_1) &= \frac{1}{\sqrt{2}} \frac{[2|\bar{\sigma}^\rho|1]}{[21]} \frac{1}{\sqrt{2}} \frac{\langle 1|\sigma_\rho|2]}{\langle 12 \rangle} \\ &= \frac{1}{2} \frac{\langle 1|\sigma^\rho|2] \langle 1|\sigma_\rho|2]}{[21] \langle 12 \rangle} \\ &= \frac{\langle 11 \rangle [22]}{[21] \langle 12 \rangle} = 0. \end{aligned}$$

The second and the third term also vanish,

$$\begin{aligned} p_2^\nu \varepsilon_{-\nu}(p_1, p_2) &= -\frac{1}{2} \langle 2|\sigma^\nu|2] \frac{1}{\sqrt{2}} \frac{[2|\bar{\sigma}_\nu|1]}{[21]} \\ &= -\frac{\langle 21 \rangle [22]}{\sqrt{2} [21]} = 0. \end{aligned}$$

Therefore, we get

$$J^\mu(1^-, 2^+) = 0. \quad (4.12)$$

Now, we perform the induction step:

$$\begin{aligned} J^\mu(1^-, 2^+, \dots, n^+) &= \frac{-i}{P_{1,n}^2} \left[\sum_{i=1}^{n-1} V_3^{\mu\nu\rho}(P_{1,i}, P_{i+1,n}) J_\nu(1^-, \dots, i^+) J_\rho((i+1)^+, \dots, n^+) \right. \\ &\quad \left. + \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} V_4^{\mu\nu\rho\sigma} J_\nu(1^-, \dots, i^+) J_\rho((i+1)^+, \dots, j^+) J_\sigma((j+1)^+, \dots, n^+) \right] \end{aligned}$$

The first V_3 term and all of the V_4 terms vanish, due to the choice of reference momenta and Fierzing. In each term in the double sum of the V_4 's and in the first V_3 term, there appears a term proportional to

$$\eta^{\nu\sigma} \langle 1 | \sigma_\nu \not{P}_{i,j} | 1 \rangle \langle 1 | \sigma_\sigma \not{P}_{k,l} | 1 \rangle = -2 \langle 11 \rangle \langle 1 | \not{P}_{k,l} \not{P}_{i,j} | 1 \rangle = 0,$$

where $i \leq j, k \leq l$ are some indices. The sum over i can actually be written as starting from 3 rather than 1, because the two cases $i = 1$ and $i = 2$, we will treat separately. Let's first examine $i = 1$,

$$\begin{aligned} I_1 &= \frac{-i}{P_{1,n}^2} V_3^{\mu\nu\rho}(p_1, P_{2,n}) J_\nu(1^-) J_\rho(2^+, \dots, n^+) \\ &= -\frac{1}{\sqrt{2} P_{1,n}^2} (2\eta^{\rho\mu} P_{2,n}^\nu - 2\eta^{\mu\nu} p_1^\rho) \frac{1}{\sqrt{2}} \frac{[q_1|\bar{\sigma}_\nu|1]}{[q_1, 1]} \frac{\langle 1 | \sigma_\rho \not{P}_{2,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \langle 23 \rangle \cdots \langle n-1, n \rangle \langle n1 \rangle} \end{aligned}$$

$$\begin{aligned}
&= - \frac{\langle 1 | \not{P}_{2,n} | q_1 \rangle \langle 1 | \sigma^\mu \not{P}_{2,n} | 1 \rangle - \langle 1 | \sigma^\mu | q_1 \rangle \langle 1 | \not{p}_1 \not{P}_{2,n} | 1 \rangle}{P_{1,n}^2 \sqrt{2} \langle 12 \rangle \cdots \langle n1 \rangle} \\
&= - \frac{\langle 1 | \not{P}_{2,n} | q_1 \rangle}{[q_1, 1] P_{1,n}^2} \frac{\langle 1 | \sigma^\mu \not{P}_{2,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle n1 \rangle}.
\end{aligned}$$

Using the spinor product (2.6) and the slash notation (2.7),

$$\begin{aligned}
- \frac{\langle 1 | \not{P}_{2,n} | q_1 \rangle}{[q_1, 1] P_{1,n}^2} &= \frac{\langle 1 | \not{P}_{2,n} | q_1 \rangle \langle q_1, 1 \rangle}{(q_1 + p_1)^2 P_{1,n}^2} \\
&= \frac{\langle 1 | \not{P}_{2,n} \not{q}_1 | 1 \rangle}{(q_1 + p_1)^2 P_{1,n}^2}.
\end{aligned}$$

The reference momentum q_1 was not chosen until now. By staring at this, we see that, if the reference momentum choice is $q_1 = P_{2,n-1}$, then this looks very similar to the $m = n$ term in the sum of equation (4.11),

$$\begin{aligned}
\frac{\langle 1 | \not{P}_{2,n} \not{q}_1 | 1 \rangle}{(q_1 + p_1)^2 P_{1,n}^2} &= \frac{\langle 1 | \not{P}_{2,n} \not{P}_{2,n-1} | 1 \rangle}{P_{1,n-1}^2 P_{1,n}^2} \\
&= \frac{\langle 1 | \not{P}_{1,n} \not{P}_{1,n-1} | 1 \rangle}{P_{1,n-1}^2 P_{1,n}^2} \\
&= - \frac{\langle 1 | \not{P}_{1,n} \not{p}_n | 1 \rangle}{P_{1,n-1}^2 P_{1,n}^2} \\
&= \frac{\langle 1 | \not{p}_n \not{P}_{1,n} | 1 \rangle}{P_{1,n-1}^2 P_{1,n}^2}.
\end{aligned}$$

To summarize, the $i = 1$ case:

$$I_1 = \frac{\langle 1 | \sigma^\mu \not{P}_{1,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle n1 \rangle} \frac{\langle 1 | \not{p}_n \not{P}_{1,n} | 1 \rangle}{P_{1,n-1}^2 P_{1,n}^2}. \quad (4.13)$$

The case for $i = 2$ vanishes, because of equation (4.12),

$$I_2 = V_3^{\mu\nu\rho}(P_{1,2}, P_{3,n}) J_\nu(1^-, 2^+) J_\rho(3^+, \dots, n^+) = 0.$$

Back to the full term, we are left with the two terms in V_3 , where the reference momentum for J_σ is p_1 and the sum over i omitting the $i = 2$ case (the $i = 1$ case is taken out in front of the sum),

$$\begin{aligned}
J^\mu(1^-, 2^+, \dots, n^+) &= I_1 + \frac{-i}{P_{1,n}^2} \left[\sum_{i=3}^{n-1} \frac{i}{\sqrt{2}} (2\eta^{\rho\mu} P_{i+1,n}^\nu - 2\eta^{\mu\nu} P_{1,i}^\rho) \right. \\
&\quad \times \left(\frac{\langle 1 | \sigma_\nu \not{P}_{2,i} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle i1 \rangle} \sum_{m=3}^i \frac{\langle 1 | \not{p}_m \not{P}_{1,m} | 1 \rangle}{P_{1,m-1}^2 P_{1,m}^2} \right) \left(\frac{\langle 1 | \sigma_\rho \not{P}_{i+1,n} | 1 \rangle}{\sqrt{2} \langle 1, i+1 \rangle \cdots \langle n1 \rangle} \right) \Big] \\
&= I_1 + \frac{1}{P_{1,n}^2} \left[\sum_{i=3}^{n-1} \sum_{m=3}^i \frac{\langle 1 | \not{p}_m \not{P}_{1,m} | 1 \rangle}{P_{1,m-1}^2 P_{1,m}^2} \right. \\
&\quad \times \left. \frac{\langle 1 | \not{P}_{i+1,n} \not{P}_{1,i} | 1 \rangle \langle 1 | \sigma^\mu \not{P}_{i+1,n} | 1 \rangle - \langle 1 | \sigma^\mu \not{P}_{1,i} | 1 \rangle \langle 1 | \not{P}_{1,i} \not{P}_{i+1,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle i1 \rangle \langle 1, i+1 \rangle \cdots \langle n1 \rangle} \right].
\end{aligned}$$

The two sums can be interchanged in the way illustrated by figure 2. We can therefore switch the two sums, according to

$$\sum_{i=3}^{n-1} \sum_{m=3}^i = \sum_{m=3}^{n-1} \sum_{i=m}^{n-1}.$$

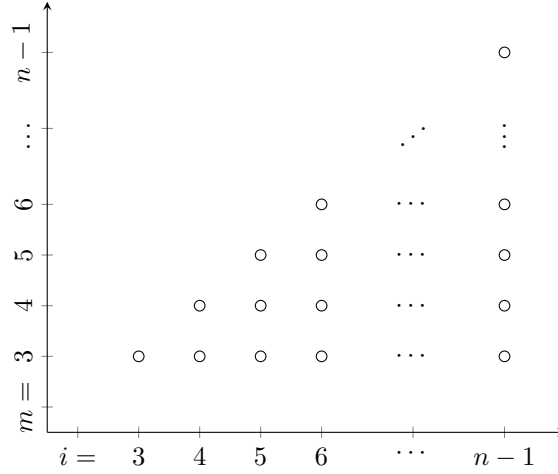


Figure 2: From the two sums, we can see that if $i = 3$, then m takes values from 3 to i , so m can only be 3. When i increments to $i = 4$, then m can take values 3 and 4 and so on. For $i = n - 1$, m can take all values between 3 to $n - 1$. If we sum over m before summing over i , we can read off the limits from the plot above. Namely m starts at 3 and goes up to $n - 1$, whereas i always starts from m and goes up to $n - 1$.

The numerator can be simplified by using $\langle 1 | \not{P}_{1,i} \not{P}_{i+1,n} | 1 \rangle = -\langle 1 | \not{P}_{i+1,n} \not{P}_{1,i} | 1 \rangle$ and $\not{P}_{1,i} = \not{P}_{1,n} - \not{P}_{i+1,n}$, therefore $\langle 1 | \not{P}_{i+1,n} \not{P}_{1,i} | 1 \rangle = \langle 1 | \not{P}_{i+1,n} \not{P}_{1,n} | 1 \rangle$, where the right half is independent of i . The denominator can also be written as one part independent and one dependent of i ,

$$\frac{1}{\langle 12 \rangle \cdots \langle i1 \rangle \langle 1, i+1 \rangle \cdots \langle n1 \rangle} = \frac{1}{\langle 12 \rangle \cdots \langle n1 \rangle} \frac{\langle i, i+1 \rangle}{\langle i1 \rangle \langle 1, i+1 \rangle}.$$

Pulling all terms independent on i together out of the sum on i and switching the sums, results in

$$J^\mu(1^-, 2^+, \dots, n^+) = I_1 + \frac{1}{P_{1,n}^2} \left[\frac{\langle 1 | \sigma^\mu \not{P}_{1,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle n1 \rangle} \sum_{m=3}^{n-1} \frac{\langle 1 | \not{P}_m \not{P}_{1,m} | 1 \rangle}{P_{1,m-1}^2 P_{1,m}^2} \times \sum_{i=m}^{n-1} \frac{\langle i, i+1 \rangle}{\langle i1 \rangle \langle 1, i+1 \rangle} \langle 1 | \not{P}_{i+1,n} \not{P}_{1,n} | 1 \rangle \right].$$

We can evaluate the term in the second line that is dependent on i , which has appeared before in (4.10), and inserting equation (4.13) for I_1 , concluding the proof,

$$\begin{aligned} J^\mu(1^-, 2^+, \dots, n^+) &= I_1 + \frac{1}{P_{1,n}^2} \left[\frac{\langle 1 | \sigma^\mu \not{P}_{1,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle n1 \rangle} \sum_{m=3}^{n-1} \frac{\langle 1 | \not{P}_m \not{P}_{1,m} | 1 \rangle}{P_{1,m-1}^2 P_{1,m}^2} \right. \\ &\quad \left. \times \frac{\langle m | \not{P}_{1,n} | 1 \rangle}{\langle m1 \rangle} \not{P}_{1,n} | 1 \rangle \right] \\ &= I_1 + \frac{\langle 1 | \sigma^\mu \not{P}_{1,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle n1 \rangle} \sum_{m=3}^{n-1} \frac{\langle 1 | \not{P}_m \not{P}_{1,m} | 1 \rangle}{P_{1,m-1}^2 P_{1,m}^2} \\ &= \frac{\langle 1 | \sigma^\mu \not{P}_{1,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle n1 \rangle} \frac{\langle 1 | \not{P}_n \not{P}_{1,n} | 1 \rangle}{P_{1,n-1}^2 P_{1,n}^2} + \frac{\langle 1 | \sigma^\mu \not{P}_{1,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle n1 \rangle} \sum_{m=3}^{n-1} \frac{\langle 1 | \not{P}_m \not{P}_{1,m} | 1 \rangle}{P_{1,m-1}^2 P_{1,m}^2} \\ &= \frac{\langle 1 | \sigma^\mu \not{P}_{1,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \cdots \langle n1 \rangle} \sum_{m=3}^n \frac{\langle 1 | \not{P}_m \not{P}_{1,m} | 1 \rangle}{P_{1,m-1}^2 P_{1,m}^2}. \end{aligned}$$

□

5 Parke Taylor formula

Vector particles such as gluons do not have a conserved helicity. The most helicity violating amplitudes are zero at tree level. The first non-zero amplitudes are when the helicities of two gluons have the opposite sign of the helicities of the other $n - 2$ gluons. These amplitudes are called MHV amplitudes (maximally helicity violating).

Theorem 5.1 (Parke Taylor [Parke and Taylor, 1986]). The tree-level amplitude for the first two gluons with negative helicity and the rest positive helicity can be written as

$$A_n^{tree}(1^-, 2^-, 3^+, \dots, n^+) = i \frac{\langle 12 \rangle^4}{\langle 12 \rangle \dots \langle n1 \rangle}. \quad (5.1)$$

This formula was first speculated by Parke and Taylor and also numerically proven up to $n = 6$. It was first proofed by Berends and Giele [Berends and Giele, 1988].

Proof. To obtain amplitudes out of off-shell currents J_μ , we need to multiply the current by $iP_{1,n}^2$ and then contract it with an appropriate polarisation vector ε^μ which is on the mass-shell. Multiplication by $iP_{1,n}^2$ is needed, because the off-shell current is defined to include the propagator of external off-shell leg μ . However, we then have $p_{n+1}^2 = P_{1,n}^2 = 0$. Therefore, we begin with

$$A_n^{tree}(1^-, 2^+, \dots, n^+, (n+1)^-) \quad (5.2)$$

$$= iP_{1,n}^2 \varepsilon_-^\mu(p_{n+1}, q_{n+1}) J_\mu(1^-, 2^+, \dots, n^+) \quad (5.3)$$

$$= -iP_{1,n}^2 \frac{1}{\sqrt{2}} \frac{[q_{n+1} | \bar{\sigma}^\mu | p_{n+1}]}{[q_{n+1}, p_{n+1}]} \frac{\langle 1 | \sigma_\mu \not{P}_{2,n} | 1 \rangle}{\sqrt{2} \langle 12 \rangle \dots \langle n1 \rangle} \sum_{m=3}^n \frac{\langle 1 | \not{p}_m \not{P}_{1,m} | 1 \rangle}{P_{1,m-1}^2 P_{1,m}^2} \quad (5.4)$$

We used proposition 4.2 to express J_μ . The $\not{P}_{2,n} | 1 \rangle$ in the middle term, can actually be written as $\not{P}_{1,n} | 1 \rangle$ because the $|i\rangle$ are solutions to the massless Dirac equation, $\not{p}|p\rangle = 0$. We can also use (2.4) to switch q and p in the numerator of the polarisation vector. Since $P_{1,n}^2 = 0$, all terms in the sum vanish except when $m = n$. In this case $P_{1,n}^2$ is cancelled from the denominator.

Next, we evaluate

$$\not{P}_{1,n} | 1 \rangle = -\not{p}_{n+1} | 1 \rangle \quad (5.5)$$

$$= -(|n+1\rangle[n+1] + |n+1\rangle\langle n+1|) | 1 \rangle \quad (5.6)$$

$$= -|n+1\rangle\langle n+1, 1| \quad (5.7)$$

$$= |n+1\rangle\langle 1, n+1|, \quad (5.8)$$

where we have used momentum conservation in the first step (5.5). From (5.5) to (5.6) definition (2.7) was inserted. In the last step, we used the antisymmetry of the spinor product. In the sum we have ($m = n$),

$$\not{p}_n \not{P}_{1,n} | 1 \rangle = \not{p}_n |n+1\rangle\langle 1, n+1| \quad (5.9)$$

$$= (|n\rangle[n] + |n\rangle\langle n|) |n+1\rangle\langle 1, n+1| \quad (5.10)$$

$$= |n\rangle[n, n+1]\langle 1, n+1|, \quad (5.11)$$

in an analogous way as before. Using these equations, we obtain (writing $|p_i\rangle$ as $|i\rangle$)

$$A_n^{tree}(1^-, 2^+, \dots, n^+, (n+1)^-) \quad (5.12)$$

$$= -i \frac{1}{2} \frac{\langle n+1 | \sigma^\mu | q_{n+1} \rangle}{[q_{n+1}, n+1]} \frac{\langle 1 | \sigma_\mu | n+1 \rangle \langle 1, n+1|}{\langle 12 \rangle \dots \langle n1 \rangle} \frac{\langle 1n \rangle [n, n+1] \langle 1, n+1|}{P_{1,n-1}^2}. \quad (5.13)$$

We have momentum conservation, therefore

$$P_{1,n-1} = -P_{n,n+1} \implies P_{1,n-1}^2 = P_{n,n+1}^2 \quad (5.14)$$

$$= (p_n + p_{n+1})^2 \quad (5.15)$$

$$= s_{n,n+1} \quad (5.16)$$

$$= \langle n, n+1 \rangle [n+1, n], \quad (5.17)$$

where we have used (2.6) in the last step. Applying the Fierz identity (2.3) to (5.13), we finally get

$$A_n^{tree}(1^-, 2^+, \dots, n^+, (n+1)^-) \quad (5.18)$$

$$= -i \frac{\langle n+1, 1 \rangle [n+1, q_{n+1}] \langle 1, n+1 \rangle \langle 1n \rangle [n, n+1] \langle 1, n+1 \rangle}{[q_{n+1}, n+1] \langle 12 \rangle \dots \langle n1 \rangle \langle n, n+1 \rangle [n+1, n]} \quad (5.19)$$

$$= -i \frac{-\langle 1, n+1 \rangle [n+1, q_{n+1}] \langle 1, n+1 \rangle - \langle n1 \rangle [n, n+1] \langle 1, n+1 \rangle}{-[n+1, q_{n+1}] \langle 12 \rangle \dots \langle n1 \rangle - \langle n, n+1 \rangle [n, n+1]} \quad (5.20)$$

$$= -i \frac{\langle 1, n+1 \rangle^3}{\langle 12 \rangle \dots \langle n, n+1 \rangle} \frac{-\langle 1, n+1 \rangle}{\langle n+1, 1 \rangle} \quad (5.21)$$

$$= i \frac{\langle 1, n+1 \rangle^4}{\langle 12 \rangle \dots \langle n+1, 1 \rangle}. \quad (5.22)$$

In the last step we multiplied by 1 to get the desired structure of the Parke Taylor formula. Notice that gluons with negative helicities - gluon 1 and gluon $n+1$ - are adjacent. The formula for n gluons can therefore also be written as

$$A_n^{tree}(1^-, 2^-, 3^+, \dots, n^+) = i \frac{\langle 12 \rangle^4}{\langle 12 \rangle \dots \langle n1 \rangle}, \quad (5.23)$$

by relabelling the momenta. □

Notice that, the general Parke Taylor formula reads

$$A_n^{tree}(1^+, \dots, j^-, \dots, k^-, \dots, n^+) = i \frac{\langle jk \rangle^4}{\langle 12 \rangle \dots \langle n1 \rangle}, \quad (5.24)$$

where j and k may not be adjacent.

6 Implementation

The implementation was done in Python using various helper classes from the libraries NumPy [Oliphant, 06] and SciPy [Virtanen et al., 2019]. For the full code, please refer to the GitHub repository [Gruber, 2019].

6.1 Conventions

I defined 4-vectors with their index up v^μ as row vectors. This corresponds to two dimensional arrays of the form $[[v^0, v^1, v^2, v^3]]$. On the other hand, 4-vectors with their index down w_μ are defined as column vectors and two dimensional arrays of the form $[[w^0], [w^1], [w^2], [w^3]]$. Using this convention, I can simply perform a dot product between these vectors to obtain a contraction of two indices.

Since function and variable names in Python can only consist of alphanumeric characters and underscores `_`, I used for variables with their index down v_μ names as `v_mu` and with index up w^μ names of the form `w_mu` (two underscores), such that $v^\mu w_\mu$ equals `v_mu.dot(w_mu)`.

6.2 Particle class

The first thing I've implemented was a class called `Particle(h, p, q)` with a constructor expecting arguments `h` for the helicity, `p` for an array with 4 entries representing the four momentum of the particle and `q` also an array with 4 entries to specify the reference momentum.

```
1 class Particle:
2     # Class representing a single particle.
3     #
4     # @param Particle self      reference to the class instance
5     # @param int h             the helicity, defaults to +1
```

```

6      # @param array p          the 4-momentum
7      # @param array q          the 4-momentum reference vector
8      # @return Particle        class instance
9      #
10     def __init__(self,
11                   # default values
12                   h = +1,
13                   p = masslessMomentum(),
14                   q = masslessMomentum()
15     ):

```

Listing 1: Particle class definition

The class contains the following properties:

- `eps_a` returning the polarisation vector ε_μ as a column vector,
- `eps__a` returning the polarisation vector ε^μ as a row vector,
- `lambda_pa` returning the spinor $\lambda_{p\alpha}$ as a column vector,
- `lambda_p__a` returning the spinor λ_p^α as a row vector,
- `lambda_tilda_pa_dot` returning the spinor $\tilde{\lambda}_{p\dot{\alpha}}$ as a row vector,
- `lambda_tilda_p__a_dot` returning the spinor $\tilde{\lambda}_p^{\dot{\alpha}}$ as a column vector.

6.3 Berends-Giele current

My first implementation as of the date of my presentation was a pure Python function that called itself exactly as equation (4.1). Also I had used for loops for contractions of the Lorentz-indices μ, ν, ρ and σ , which ended up in a nesting of up to 6 for loops for the double sum in equation (4.1). This slowed the whole calculation down. After refactoring the code, I was able to gain a performance increase of a factor of about 3 by using the `einsum()` method of NumPy for the contractions. This function loops over given indices in C such that array elements are accessed in C rather than Python. For this I had to implement the 3- and 4-vertex as a 3- and a 4-dimensional array respectively (g is the metric and i, j, k and l are the Lorentz indices):

```

1 def v3(P, Q):
2     # Feynman rule for the 3-gluon vertex
3     #
4     # @param array P          4-momentum of gluon 1
5     # @param array Q          4-momentum of gluon 2
6     # @return ndarray        the vertex factor
7     #
8     #      P,i
9     #      |
10    #      / \
11    # -(P+Q),k   Q,j
12    #
13    return (1j/np.sqrt(2))*(
14            np.einsum('jk,i', g, P-Q)
15            - 2*np.einsum('ij,k', g, P)
16            + 2*np.einsum('ki,j', g, Q)
17    )
18
19 def v4():
20     # Feynman rule for the 4-gluon vertex
21     #
22     # @return ndarray        the vertex factor
23     #
24     #      i       j
25     #      \ /
26     #      /\
27     #      l       k
28     #
29     return 0.5j*(
30         2*np.einsum('ik,jl', g, g)
31         - np.einsum('ij,kl', g, g)
32         - np.einsum('il,jk', g, g)
33     )

```

In further optimization, I decided to create an object for the off-shell current, such that I was able to store intermediate results, to not calculate recursive steps multiple times. Therefore,

the off-shell current is now defined as a class whose constructor expects an array of `Particle` objects:

```

1 class BGCurrent:
2     #
3     # @param BGCurrent self      reference to the class instance
4     # @param Particle[] gluons   array of gluonic Particle objects
5     # @return BGCurrent          class instance
6     #
7     def __init__(self, gluons):
8         self.gluons = gluons
9         self.n = gluons.size
10        self.regJ = np.zeros((self.n, self.n, 4), dtype=complex)
11        self.memoryJ = np.zeros((self.n, self.n), dtype=bool)
12        self.vertex4 = v4()
13        self.P = np.array([g.p for g in self.gluons])

```

Listing 2: Berends-Giele current object

When instantiating the class, an array called `regJ` of dimensions $n \times n \times 4$ is initialized to store values of $J^\mu(a^\pm, \dots, b^\pm)$ with starting index a and ending index b ($0 \leq a < b \leq n$) as soon as they are calculated. Later when the same current is needed in another recursion step, the register `regJ` is probed if the value was already calculated before. In such a case the stored result is returned instead of a recalculation. What this means is that each J^μ appearing in the recursive formula is calculated exactly once and reused in later steps. This made a significant performance improvement of a factor of about 500 in the $n = 10$ case although in exchange of some system memory. The allocated memory for the `regJ` array is [SciPy, 2019]

$$128 + 64n^2 \text{ bytes}$$

which is about 6.4 MB in the $n = 10$ case and scales quadratically in n - the number of gluons.

To calculate the current J^μ for n gluons one has to create an array with n entries of `Particle` objects and instantiate the object with that array as argument. After that one has to call `J__a()` with no argument.

```

1 def J__a(self, a = 0, b = None, rec = 1):
2     #
3     # @param BGCurrent self      reference to the class instance
4     # @param int a               starting index
5     # @param int|None b          ending index, defaults to n
6     # @param int rec             recursion depth
7     # @return array              the current as a 4-vector
8     #

```

Listing 3: Recursion step method

The `rec` argument is just there to keep track of the recursion depth and to append the propagator only in steps that are deeper in the recursion than 1.

If the starting and the ending index are equal, the function returns the polarisation vector of the a -th `Particle` object:

```

1 # recursion base
2 if a == b:
3     return self.gluons[a].eps__a

```

This is the end of the recursion as given in equations (4.4a) and (4.4b). The 3-vertex sum and the 4-vertex double sum are written as:

```

1 # 1st recursion step from i=0 to n-2
2 for i in range(a, b):
3     J_nu = self.J__a(a, i, rec+1)
4     J_rho = self.J__a(i+1, b, rec+1)
5     v = v3(self.P_ij(a, i), self.P_ij(i+1, b))
6     self.regJ[a,b] += np.einsum('mnr,n,r->m', v, J_nu, J_rho)
7
8 # 2nd recursion step from i=0 to n-3 (including n-3) ...
9 for i in range(a, b-1):
10    J_nu = self.J__a(a, i, rec+1)
11    # ... and j=i+1 to n-2 (including n-2)
12    for j in range(i+1, b):

```

```

13     J_rho    = self.J_a(i+1, j, rec+1)
14     J_sigma = self.J_a(j+1, b, rec+1)
15     self.regJ[a,b] += np.einsum('mnr{s,n,r,s->m}', self.vertex4, J_nu,
    J_rho, J_sigma)

```

Listing 4: Sums of the interaction vertices

These 3 loops are still implemented using native for loops in Python. The 3-vertex has to be calculated in each step, since it depends on the momenta $P_{0,i}$ and $P_{i+1,n}$. The 4-vertex on the other hand is a constant and is calculated once for the whole process in the constructor (see listing 2).

6.4 Parke Taylor formula

The implementation of the Parke-Taylor formula was straight forward. I've implemented it also as a function of an array of **Particle** objects. Since the Parke-Taylor formula only gives a result for MHV amplitudes, I check the array for exactly 2 negative helicities, else the function returns **NaN** (not a number). In the correct case, the two negative helicities are extracted and the spinor products are taken. These products were implemented as in definition 2.2 as functions of two **Particle** objects. Since for the denominator there is a for loop, the complexity scales linearly in n .

6.5 Comparison

In figure 3 a performance comparison of the two algorithms can be found. The two formulas are called with the same set of **Particle** objects (with the same momenta and helicities). The number of gluons n is iterated from 4 to 110 with increasing steps and plotted against the time needed. For $n = 100$ gluons it takes about 90.7 seconds with the recursion, whereas the Parke-Taylor calculation needs only 0.0019 seconds.

An observation I've encountered, was that with increasing number of gluons n the results of the two formulae deviate sometimes by a small amount. For example in the $n = 105$ case the results where

$$\begin{aligned}
 A_{BG} &= (2.5090994313359261 - 4.4065818656235933i) * 10^{-6}, \\
 A_{PT} &= (2.4904251975961678 - 4.3896288844922791i) * 10^{-6},
 \end{aligned}$$

where A_{BG} and A_{PT} are the amplitudes recieved from the Berends-Giele and the Parke-Taylor formula respectively. These deviations come from rounding errors, when adding a small float number to a float number bigger in many magnitudes. The implementation of the float datatype in most programming languages exhibit that problem [Goldberg, 1991]. My implementation does not handle such problems, since the deviations are in the 10^{-7} scale for $n = 105$.

6.6 Further improvements

There are multiple things that can further improve the performance of the Berends-Giele current.

6.6.1 Parallel processing

Since the main loops in listing 4 are not dependent of each other, a parallel processing could be taken into account. Each step adds an additional amount to the total current. So if only considering one iteration the inner loops can be processed in any order - ideal for a parallel approach. The improvement would approximately be a factor of the number of CPU cores.

6.6.2 C loops

To utilize parallel processing the loops need to be rewritten in C first, because Python uses a GIL (global interpreter lock) mechanism, that prevents multiple parallel threads to run concurrently.

6.6.3 C arrays

Each iterative step in the loops do access multiple arrays in reading as well as writing mode. Python arrays as well as NumPy arrays are slow (but convenient) to access single elements at a time. Some NumPy functions are nonetheless as fast as C in iterating over arrays (for example

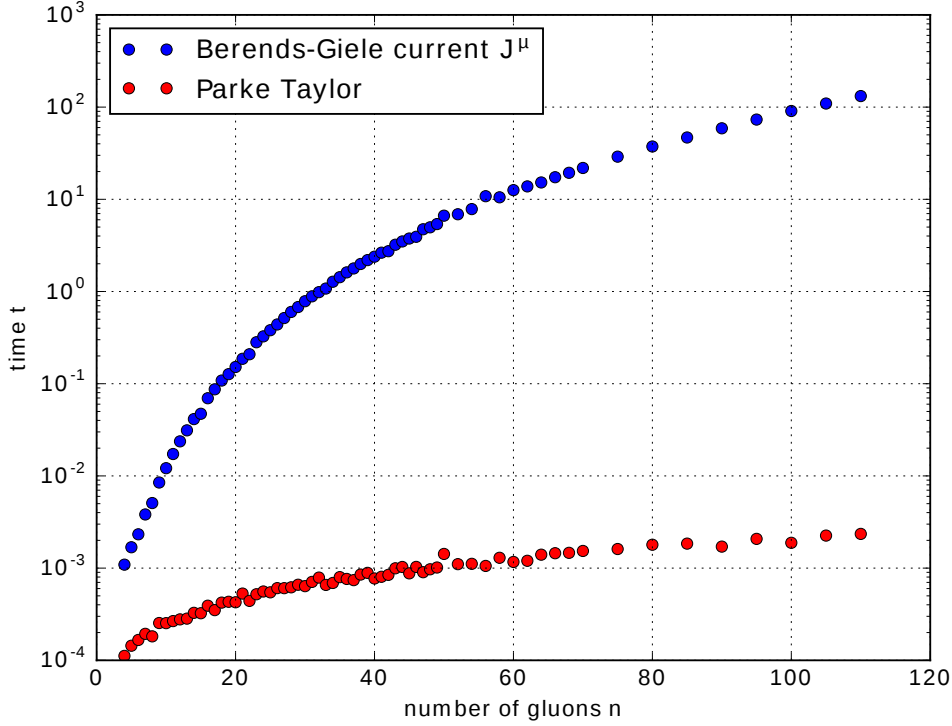


Figure 3: Comparison of the two formulas to calculate tree level partial amplitudes with n gluons. The x axis represents the number of scattering gluons involved in the calculation, whereas the y axis shows the time needed for the calculation on my laptop (an Intel(R) Core(TM) i5-6200U CPU at 2.30GHz with 4 cores. As described in section 6.6.1 only one of the 4 cores was used in the calculation).

the `einsum()` method), but not in the case of how the Berends-Giele current is implemented above. Still there are 3 remaining loops in listing 4, where optimisation can be achieved when utilizing `C` arrays.

7 Summary

The Berends-Giele current is a powerful recursive method to calculate general tree level scattering amplitudes in perturbative QCD. On the other hand, the spinor helicity formalism produces very compact results for specific problems such as MHV amplitudes in the form of the Parke-Taylor formula. These formulas were achieved by decomposing the full amplitude into color independent parts, considering only the color component of the Feynman rules to treat color separately. We were able to split the full problem into smaller problems with simpler kinematic properties. These could then be handled recursively. The implementation was mainly governed by improving the performance which is a heavily language-specific task. A professional implementation would at least incorporate the improvements addressed in section 6.6. The Berends-Giele current can also be generalised to all Feynman rules - not only the two gluon vertices - to calculate more general amplitudes including quarks in a recursive way (for example $n - 2$ gluons and 2 quarks) [Badger et al., 2013].

References

- [Badger et al., 2013] Badger, S., Biedermann, B., Hackl, L., Plefka, J., Schuster, T., and Uwer, P. (2013). Comparing efficient computation methods for massless qcd tree amplitudes: Closed analytic formulas versus berends-giele recursion. *Physical Review D*, 87(3):034011.
- [Berends and Giele, 1987] Berends, F. A. and Giele, W. (1987). The six-gluon process as an example of weyl-van der waerden spinor calculus. *Nuclear Physics B*, 294:700–732.
- [Berends and Giele, 1988] Berends, F. A. and Giele, W. (1988). Recursive calculations for processes with n gluons. *Nuclear Physics B*, 306(4):759–808.
- [Berends et al., 1981] Berends, F. A., Kleiss, R., De Causmaecker, P., Gastmans, R., and Wu, T. T. (1981). Single bremsstrahlung processes in gauge theories. *Physics Letters B*, 103(2):124–128.
- [Deur et al., 2016] Deur, A., Brodsky, S. J., and de Téramond, G. F. (2016). The qcd running coupling. *Progress in Particle and Nuclear Physics*, 90:1–74.
- [Dixon, 1996] Dixon, L. (1996). Calculating scattering amplitudes efficiently. *arXiv preprint hep-ph/9601359*.
- [Goldberg, 1991] Goldberg, D. (1991). What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys (CSUR)*, 23(1):5–48.
- [Gruber, 2019] Gruber, R. (2019). Github repository: Source code of the implementation. <http://github.com/chaos/report-proseminar-hs2019>. Accessed: 2019-12-03.
- [Lehmann et al., 1955] Lehmann, H., Symanzik, K., and Zimmermann, W. (1955). Zur formulierung quantisierter feldtheorien. *Il Nuovo Cimento (1955-1965)*, 1(1):205–225.
- [Mangano, 1999] Mangano, M. L. (1999). Introduction to qcd. Technical report, CERN.
- [Oliphant, 06] Oliphant, T. (2006–). NumPy: A guide to NumPy. USA: Trelgol Publishing. [Online; accessed 2019-12-03].
- [Parke and Taylor, 1986] Parke, S. J. and Taylor, T. R. (1986). Amplitude for n -gluon scattering. *Phys. Rev. Lett.*, 56:2459–2460.
- [SciPy, 2019] SciPy (2008-2019). Internal memory layout of an ndarray. <https://docs.scipy.org/doc/numpy/reference/arrays.ndarray.html#internal-memory-layout-of-an-ndarray>. Accessed: 2019-12-05.
- [Virtanen et al., 2019] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E. W., Vand erPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and Contributors, S. . . (2019). SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. *arXiv e-prints*, page arXiv:1907.10121.
- [Xu et al., 1987] Xu, Z., Zhang, D.-H., and Chang, L. (1987). Helicity Amplitudes for Multiple Bremsstrahlung in Massless Nonabelian Gauge Theories. *Nucl. Phys.*, B291:392–428.

Appendices

A QCD Feynman rules

The Feynman rules for QCD in Feynman gauge are [Mangano, 1999]

$$\begin{aligned}
 & \text{Diagram 1: } k, a, \mu \text{ (incoming), } p, b, \nu \text{ (outgoing), } q, c, \rho \text{ (outgoing)} \\
 & = \frac{ig}{\sqrt{2}} f^{abc} [\eta^{\nu\rho}(p-q)^\mu + \eta^{\rho\mu}(q-k)^\nu + \eta^{\mu\nu}(k-p)^\rho], \\
 & \text{Diagram 2: } a, \mu \text{ (incoming), } b, \nu \text{ (incoming), } d, \sigma \text{ (incoming), } c, \rho \text{ (outgoing)} \\
 & = -ig^2 \left[\begin{aligned} & f^{abe} f^{cde} (\eta^{\mu\rho} \eta^{\nu\sigma} - \eta^{\mu\sigma} \eta^{\nu\rho}) \\ & + f^{ace} f^{bde} (\eta^{\mu\nu} \eta^{\rho\sigma} - \eta^{\mu\sigma} \eta^{\rho\nu}) \\ & + f^{ade} f^{bce} (\eta^{\mu\nu} \eta^{\sigma\rho} - \eta^{\mu\rho} \eta^{\sigma\nu}) \end{aligned} \right], \\
 & \text{Diagram 3: } a, \mu \text{ (incoming), } f, i \text{ (incoming), } f', \bar{j} \text{ (outgoing)} \\
 & = -\frac{ig\gamma^\mu \delta_f^{f'}}{\sqrt{2}} (T^a)_{i\bar{j}}, \quad a, \mu \text{ (incoming), } b, \nu \text{ (incoming), } p \text{ (outgoing)} = -\frac{i\delta^{ab}\eta^{\mu\nu}}{p^2}, \\
 & \text{Diagram 4: } f, i \text{ (incoming), } f', \bar{j} \text{ (incoming), } a, \mu \text{ (outgoing)} \\
 & = \frac{ig\gamma^\mu \delta_f^{f'}}{\sqrt{2}} (T^a)_{i\bar{j}}, \quad f, i \text{ (incoming), } f', j \text{ (incoming), } p \text{ (outgoing)} = \frac{i\delta_f^{f'} \delta_j^i}{p^2}.
 \end{aligned}$$

B Color ordered QCD Feynman rules

The color ordered Feynman rules for QCD in Feynman gauge are [Dixon, 1996]

$$\begin{aligned}
 & \text{Diagram 1: } k, \mu \text{ (incoming), } p, \nu \text{ (outgoing), } q, \rho \text{ (outgoing)} \\
 & = \frac{i}{\sqrt{2}} [\eta^{\nu\rho}(p-q)^\mu + \eta^{\rho\mu}(q-k)^\nu + \eta^{\mu\nu}(k-p)^\rho], \\
 & \text{Diagram 2: } \mu \text{ (incoming), } \nu \text{ (incoming), } \sigma \text{ (incoming), } \rho \text{ (outgoing)} \\
 & = \frac{i}{2} [2\eta^{\mu\rho} \eta^{\nu\sigma} - \eta^{\mu\nu} \eta^{\rho\sigma} - \eta^{\mu\sigma} \eta^{\nu\rho}], \\
 & \text{Diagram 3: } \mu \text{ (incoming), } f \text{ (incoming), } f' \text{ (outgoing)} \\
 & = -\frac{i\gamma^\mu \delta_f^{f'}}{\sqrt{2}}, \quad \mu \text{ (incoming), } \nu \text{ (incoming), } p \text{ (outgoing)} = -\frac{i\eta^{\mu\nu}}{p^2}, \\
 & \text{Diagram 4: } f \text{ (incoming), } f' \text{ (incoming), } \mu \text{ (outgoing)} \\
 & = \frac{i\gamma^\mu \delta_f^{f'}}{\sqrt{2}}, \quad f \text{ (incoming), } f' \text{ (incoming), } p \text{ (outgoing)} = \frac{i\delta_f^{f'}}{p^2}.
 \end{aligned}$$

C Code

All code used in this report is open source and can be found in the GitHub repository [Gruber, 2019]