Computational Physics Laboratory:
Tree-Level Gluon Scattering Amplitudes in Mathematica

Jiahao Miao
AY 2024/25

Abstract

This report explores the use of the Mathematica software package to compute tree-level gluon scattering amplitudes in perturbative quantum chromodynamics (QCD), leveraging its powerful algebraic capabilities. The main focus is on utilizing Mathematica's list manipulation functions to represent Feynman diagrams, applying Feynman rules to evaluate the corresponding scattering amplitudes. Emphasis is placed on the computational techniques used to automate and simplify these calculations, including the verification of key properties such as Ward identities and symmetry under the exchange of external legs. The report highlights the efficiency of Mathematica in handling complex algebraic expressions and streamlining the process of evaluating quantum field theory amplitudes.

Abstract

Contents

1 Introduction **1**

2 Implementation **1**

2.1 Generating Feynman Diagrams 1

2.2 Feynman Rules 2

2.3 Color Algebra and Lorentz 3

3 Benchmarks and Calculations **4**

3.1 Number of Feynman Diagrams and Color Structures 4

3.2 Amplitude Generation 5

3.3 Symmetry under Exchange of External Legs 5

3.4 Ward Identities 5

1 Introduction

Quantum Chromodynamics (QCD), the fundamental theory of the strong interaction, describes the behavior of quarks and gluons. Understanding high-energy collider phenomena, such as jet production, necessitates the precise calculation of scattering amplitudes involving these fundamental particles.

While the theoretical framework for computing these amplitudes using Feynman diagrams is well-established, the practical execution of these calculations quickly becomes an arduous and error-prone task due to the rapidly increasing number of diagrams, the complexity of algebraic expressions, and the intricate handling of momentum, color, and Lorentz indices.

For gluon scattering in QCD, the number of tree-level Feynman diagrams grows more than factorially with the number of external legs $n+1$, following an asymptotic growth pattern given by:

$$\mathcal{O}\left(\left(\frac{9\sqrt{3}+12}{11}\right)^{n}\frac{n!}{n^{3/2}}\right) \tag{1}$$

For instance, the number of tree-level diagrams for the first few $n+1$ gluons can be computed using a recurrence relation listing 1 or counted directly after generating the diagrams in Mathematica in the following section.

Listing 1: Number of tree-level diagrams for $n+1$ gluons

```
a[0] = 0; a[1] = 1; a[2] = 1;
a[n_] := a[n_]:=a[n]=(12(2n-3)a[n-1]+(3n-5)(3n-7)a[n-2])/11;
Table[a[n], {n, 3, 12}];
Output: {4, 25, 220, 2485, 34300, 559405, 10525900, 224449225, 5348843500, 140880765025}
```

This increasing complexity, particularly in perturbative QCD, makes a computational approach essential. For this project, Mathematica [1] was selected as the primary computational environment due to its unparalleled symbolic manipulation capabilities, which are uniquely well-suited for the challenges of calculating Feynman amplitudes. Unlike compiled languages such as C++ or Fortran, which excel in numerical computations and low-level control, Mathematica provides a high-level, interactive environment that natively understands and operates on symbolic expressions.

This report details the computational implementation of gluon amplitude calculations in QCD using Mathematica. We will demonstrate how Mathematica's unique strengths in symbolic computation, automated algebraic simplification, and high-level programming facilitate the systematic construction of Feynman diagrams, the automated assignment of momenta, the application of Feynman rules, and ultimately, the derivation of explicit amplitude expressions and modulus squared.

The subsequent sections will walk through the design of our data structures, the algorithms for momentum assignment and constraint solving, the implementation of Feynman rules as substitution rules, color algebra handling, verification of key properties such as Ward identities, and the final evaluation of scattering amplitudes.

2 Implementation

2.1 Generating Feynman Diagrams

The first step is to encode the Feynman diagrams in a way that Mathematica can manipulate them fig. 1. We represent each diagram as a list of pairs, for instance, the third diagram in fig. 1 can be represented as:

$$\{\{1,-1\},\{2,-1\},\{3,-2\},\{4,-2\},\{5,-2\},\{-1,-2\}\};$$

with the convention that the external legs are represented by positive integers and the internal vertices by negative integers, the pairs representing the connections between the vertices always ordered in decreasing order.

To generate the Feynman diagrams, we use a recursive function that starts with an initial diagram and progressively adds vertices. First, 3-point vertices are introduced at each internal line, and then 4-point vertices are added at each 3-point vertex. The

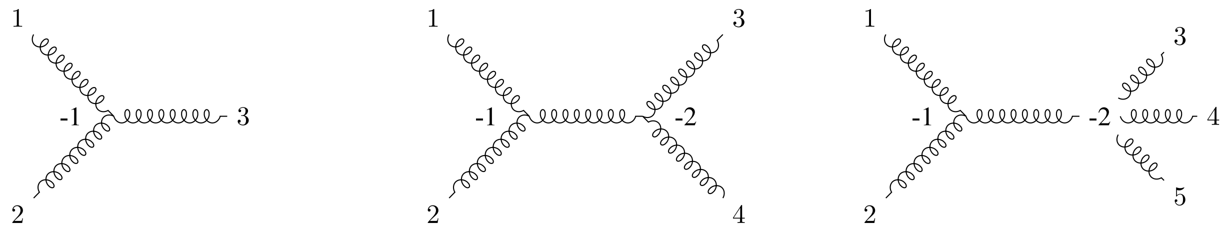

Figure 1: Example of a Feynman diagram for gluon scattering with 3, 4, and 5 external legs. On the left, the diagram with 3 external legs is the skeleton from which the other diagrams are built.

number of 3-point vertices is determined by tallying how many times a negative number appears in the list, with each negative number appearing at most 4 times.

The next step is to assign momenta to the external and internal lines. For a generic diagram with N external lines, P internal lines , L loops and V_3, V_4 the number of 3 and 4 point vertices respectively, the number of loops is equal to the number of independent momenta:

$$L = P - V_3 - V_4 + 1 \tag{2}$$

in tree-level diagrams $L = 0$, so that we have $V_3 + V_4 = P + 1$. Imposing momentum conservation at each vertex provides $P + 1$ equations. However, there are only P unknowns, meaning the system is underdetermined. To solve this, we must include the constraint of global momentum conservation, which removes one of the equations, since momentum conservation in each vertex imply the global momentum consevation. All the previous steps are implemented in the function `GenerateDiagrams` in listing 2.

Listing 2: Generating Feynman diagrams

```
Diagrams[graph_] (* Generates diagrams from a graph, calculating internal momenta and Lorentz and color indices for the vertices and
    propagators *)
generatediagrams[n_] (* Generates diagrams with n external points and only 3 and 4−point vertices *)
generatediagrams3[n_] (* Generates diagrams with n external points and only 3−point vertices *)
```

To complete the Feynman diagrams, we need to include propagators and vertices, each with momenta, color, and Lorentz indices fig. 2. We can achieve this by using Mathematica's list manipulation and pattern matching capabilities, constructing vertices and propagators as functions that take momentum, Lorentz indices, and color indices as arguments listing 3. These functions will be used later to apply the Feynman rules.

Listing 3: Feynman propagator and vertex

```
Prop[p_, {mu1_, mu2_}, {c1_, c2_}]
V[i_][{p1_, p2_, p3_}, {mu1_, mu2_, mu3_}, {c1_, c2_, c3_}]
V[i_][{p1_, p2_, p3_, p4_}, {mu1_, mu2_, mu3_, mu4_}, {c1_, c2_, c3_, c4_}]
```

where p is the momentum, μ_i are the Lorentz indices and c_i are the color indices with the convention fig. 2. We later use these functions to apply the Feynman rules by substituting these functions with the corresponding expressions.

2.2 Feynman Rules

The QCD Feynman rules are implemented in Mathematica as follows:

Listing 4: Feynman rules for QCD

```
gluon = prop[p_, {mu1_, mu2_}, {c1_,c2_}] :> −I myDelta[c1, c2] gprop[p, mu1, mu2];
propagator[p_, mu1_, mu2_] := SP[{mu1}, {mu2}]/SP[p, p]
V3QCD = V[i_][{p1_, p2_, p3_}, {mu1_, mu2_, mu3_}, {c1_, c2_, c3_}] :> −g f[c1, c2, c3] V3Lorentz[{p1, p2, p3}, {mu1, mu2,
    mu3}]
```

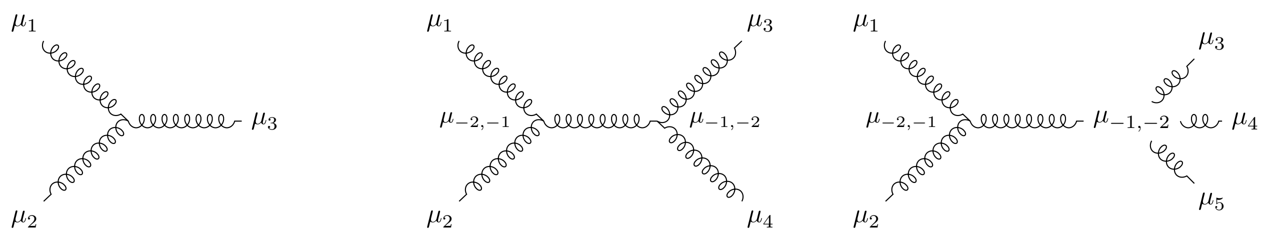

Figure 2: Each external leg is assigned a Lorentz index μ_i and each vertex j a Lorentz indices $\mu_{i,j}$. The same logic is applied for the color indices.

.

```
V3p[{p1_, p2_, p3_}, {mu1_, mu2_, mu3_}] := SP[p1 − p2, {mu3}] SP[{mu1}, {mu2}] + SP[p2 − p3, {mu1}] SP[{mu2}, {mu3}] +
    SP[p3 − p1, {mu2}] SP[{mu3}, {mu1}]
V4QCD = V[i_][{p1_, p2_, p3_, p4_}, {mu1_, mu2_, mu3_, mu4_}, {c1_, c2_, c3_, c4_}] :> −I g^2
        (f[c1, c2, c[x, i]] f[c3, c4, c[x, i]] (SP[{mu1}, {mu3}] SP[{mu2}, {mu4}] − SP[{mu1}, {mu4}] SP[{mu2}, {mu3}])
        − f[c1, c3, c[x, i]] f[c2, c4, c[x, i]] (SP[{mu1}, {mu4}] SP[{mu2}, {mu3}] − SP[{mu1}, {mu2}] SP[{mu3}, {mu4}])
        + f[c1, c4, c[x, i]] f[c2, c3, c[x, i]] (SP[{mu1}, {mu2}] SP[{mu3}, {mu4}] − SP[{mu1}, {mu3}] SP[{mu2}, {mu4}]))
SetAttributes[myDelta, Orderless]
ColorDelta = # //. {myDelta[a_, b_] myDelta[a_, b_] :> (Nc^2 − 1) ,
   myDelta[a_, a_] :> (Nc^2 − 1),
   myDelta[a_, b_]*expr_ :> (expr /. b −> a),
   c[i_, j_] :> a[i,j]} &
FeynmanRules = (# /. gluon /. V3QCD /. V4QCD /.gprop −> propagator /. V3Lorentz −> V3p // ColorDelta // antisymf // Expand)
    &
```

where Lorentz structure and color structure are factorized when possible so we can manipulate them separately. This property is especially useful for generating all the distinct products of structure constants f^{abc} for a given number of external legs. The Feynman rules are applied to the Feynman diagrams using the `Replace` function, which substitutes the propagators and vertices with their corresponding expressions.

2.3 Color Algebra and Lorentz

The color algebra is handled using the following functions:

- **antisymf**: This functions applies the antisymmetry property of the structure constants f^{abc}, and brings the color indices in decreasing alfanumeric order.
- **frule**: This functions acts on products of structure constants, it counts the number of repeated indices and substitute them with $a[i]$. For $n \geq 5$ there are more than one repeated indices and the degeneracy associated with the permutation of the dummy indices must be taken into account.
- **Color**: This function acts on fully contracted products of structure constants and it calculates the results in terms of N_c, the dimensions of the fundamental representation of the $SU(3)$ group. This function converts the products of structure constants into its fundamental representation, using the following identity:

$$f^{abc} = -2iTr\left([T^a, T^b]T^c\right) \tag{3}$$

where T^a are the generators of the $SU(3)$ group in the fundamental representation. The trace is expressed as a product of matrices with cyclic indices, and the the Fierz identity is utilized to convert the product of T^a into a product of Kronecker deltas:

$$T^a_{i,j}T^b_{k,l} = \frac{1}{2}\left(\delta_{i,l}\delta_{j,k} - \frac{1}{N_c}\delta_{i,j}\delta_{k,l}\right) \tag{4}$$

This conversion allows us to express the color algebra in terms of Kronecker deltas, which can be manipulated in Mathematica.

However this algorithm is not highly efficient. Before all the Kronecker deltas are contracted, the number of terms grows exponentially. For 6 fully contracted structure constants, using the fundamental representation, we have 2^6 products of 6 traces each, writing the traces as products of matrices we get $3 * 6$ matrices, and using Fierz it becomes 2^{6*3} terms, in total we have 2^{6*3+6} Kronecker deltas to contract. This whole process takes half a second to be computed.

while Lorentz algebra using the `SP` function, which is a wrapper for the `Dot` function. This function is used to contract Lorentz indices and it is defined as follows:

Listing 5: Lorentz algebra

```
SetAttributes[SP, Orderless]
Contract = (# //. {
  SP[a_, {mu_}] SP[b_, {mu_}] :> SP[a, b], (*Contraction*)
  SP[p_, {mu_}] SP[p_, {mu_}] :> SP[p, p],(*Square*)
  SP[a_Integer *p_, {mu_}] :> a SP[p, {mu}],(*Homogeneity*)
  SP[a_Integer *p_, q_] :> a SP[p, q], (*Homogeneity'*)
  SP[p_ + q_, {mu_}] :> SP[p, {mu}] + SP[q, {mu}] ,(*Linearity*)
  SP[a_, b_ + c_] :> (SP[a, b] + SP[a, c]),(*Distributive property'*)
  SP[{mu_}, {mu_}] :> 4 (*Trace*),
  SP[n, n] :> 0 (*Light Cone Gauge*),
   SP[p[i_], p[i_]] :> 0 (*On Shell Condition*)
  }) &;
```

3 Benchmarks and Calculations

Armed with the framework for constructing Feynman diagrams and applying Feynman rules, we can now proceed to compute the scattering amplitudes for gluon interactions in (QCD) and verify their properties.

Before diving into the calculations, it's important to understand the expected complexities arising from the generated amplitudes.

3.1 Number of Feynman Diagrams and Color Structures

The total number of tree-level Feynman diagrams for n gluons with 3 and 4 point vertices can be computed using a recurrence relation[2] and shown in listing 1. Among these, the number of diagrams featuring only 3-point vertices is given by $(2n-5)!!$ (double factorial) [3].Each of these diagrams corresponds to a unique color substructure. The remaining diagrams, which include at least one 4-point vertex, are generated by adding a 4-point vertex to existing diagrams with only 3-point vertices and they do not introduce any new color substructures.

In general, for n gluons, there are $(2n-5)!!$ color substructures.These takes the form of contractions of $n-2$ structure constants f^{abc} with $n-3$ dummy(summed over) indices a_i and n distinct indices c_i, e.g $f^{c_2c_1a_1}f^{c_4c_3a_1}$ for $n=4$, $f^{c_1a_2a_1}f^{c_3c_3a_1}f^{c_5c_4a_2}$ for $n=5$, and so on.

Table 1: Number of Tree-level Feynman Diagrams for n gluons. The first column indicates the number of external gluons, the second column shows the total number of diagrams, the third column counts the diagrams with only 3-point vertices, and the fourth column counts those with at least one 4-point vertex.

n	Number of Diagrams	3-point Vertices only	with 4-point Vertices
4	4	3	1
5	25	15	10
6	220	105	115
7	2485	945	1540
8	34300	10395	23905

3.2 Amplitude Generation

The process of generating the scattering amplitude is straightforward. We generate the Feynman diagrams using the ‘generatediagrams‘ function (listing 2), sum the contributions from each diagram, and then apply the Feynman rules to obtain the amplitude.

The computational resources required for generating the diagrams only, specifically the time and memory, are summarized in Table 2, which shows that time and memory usage grows exponentially with the number of gluons involved in the scattering process. Applying the Feynman rules further increases the computational cost, as shown in the same table.

Table 2: Computational Resources for Feynman Diagram Generation and Amplitude. The Amplitude is generated by applying the operations: Total, FeynmanRules and Expand. The result for 8 gluons is not available (TBD) due to the high computational cost.

Number of Gluons		Generated Diagrams		Amplitude	
n	**Number of Diagrams**	**Time (s)**	**Memory (MB)**	**Time (s)**	**Memory (MB)**
4	4	0.0011	0.01	0.004	0.096
5	25	0.0100	0.12	0.051	2.87
6	220	0.1153	1.51	6.503	104.52
7	2485	1.5998	22.99	1112.738	4430.25
8	34300	27.5766	404.70	TBD	TBD

After generating the amplitude, there are several important properties that we can verify to ensure the correctness of our calculations.

3.3 Symmetry under Exchange of External Legs

The scattering amplitude for gluon interactions should be symmetric under the exchange of external legs.

To veriry this property, the following function are defined:

- **swapTwoParticles[amp_,i_,j_]**: This function takes an amplitude and swaps the i-th and j-th external legs, by replacing the corresponding Lorentz, color and momentum labels in the amplitude expression.
- **pairmap[list1_, list2_]** : This function takes two lists and returns a list of pairs, where each pair consists of an element from the first list and the corresponding element from the second list.

In the two to two scattering case ($n = 4$), there are only 3 color substructures and after exchangin external legs, these color substructures are permuted among themselves, their Lorentz coefficients also changes accordingly, but the overall amplitude remains unchanged.

For $n \geq 5$, the permutation are more complex and the function **pairmap** is needed to keep track of which term goes where. Then verify that the amplitude remains unchanged by subtracting the original color substructure Lorentz coefficients from the swapped ones and checking if the result is zero.

From this we can conclude that the amplitude is symmetric under the exchange of external legs and that all the color substructures are permuted among themselves, so only a single color substructure is needed and rest can be generated by substitution rules, which is significatly more efficient than all the diagrams, substituting Feynman rules, summing over all diagrams and then collecting the color substructures.

3.4 Ward Identities

The Ward identities are a set of relations that must be satisfied by the scattering amplitudes in gauge theories. In the case of gluon scattering, the Ward identity states that the amplitude must vanish when any external on-shell gluon polarization is substituted with its momentum.

While the overall scattering amplitude for gluon interactions satisfies the Ward identity, the presence of distinct color substructures within the amplitude poses a unique challenge. Since these color substructures prevent a direct summation of the various Lorentz structures, each individual color substructure must inherently be gauge invariant for the full amplitude to satisfy the identity.

However, the $(2n-5)!!$ color substructures are not all linearly independent, as they are related by the Jacobi identity as shown in Equation eq. (5) for the structure constants f^{abc}, so they are not gauge invariant. These non gauge invariant dependent substructures still vanish when all external polarization vectors are simultaneously substituted with their respective momentum vectors.

$$f^{aeb}f^{ecd} + f^{ade}f^{ecb} - f^{ace}f^{edb} = 0 \tag{5}$$

For n gluons, each of these color substructures has $n-3$ dummy indices, so each term can generate $n-3$ jacobi identities, though these may not all be unique. These color substructures can be mapped to variables using the Mathematica function **MapIndexed** to variables $v[i]$ so that Mathematica can work with them. The Jacobi identities can then be transformed into a system of equations and solved using **Solve**.

In conclusion, the initial set of $(2n-5)!!$ color substructures reduces to $(n-2)!$ independent color substructures [4]. These independent color substructures are each accompanied by their respective Lorentz structures, which are inherently gauge invariant and collectively satisfy the Ward identity.

Thanks to the symmetry under exchange, it is sufficient to verify the Ward identity for just one of these independent color substructures.

When a Lorentz substructure is entirely contracted with all external momenta, it vanishes with a **Simplify** after contraction. However, when contracted with the external polarization vectors, it also needs momentum conservation and transverse polarization $p_i \cdot \epsilon_i = 0$.

References

[1] W. R. Inc., *Mathematica, Version 14.2*, Champaign, IL, 2024. [Online]. Available: https://www.wolfram.com/mathematica.

[2] N. J. A. Sloane, *A268163: Number of non-isomorphic 3-regular graphs on n nodes*, https://oeis.org/A268163, The On-Line Encyclopedia of Integer Sequences, 2015.

[3] N. J. A. Sloane, *A001147: Number of permutations of length n with no 1-cycles*, https://oeis.org/A001147, The On-Line Encyclopedia of Integer Sequences, 2015.

[4] V. Del Duca, L. Dixon, and F. Maltoni, "New color decompositions for gauge amplitudes at tree and loop level," *Nuclear Physics B*, vol. 571, no. 1, pp. 51–70, 2000, ISSN: 0550-3213. DOI: https://doi.org/10.1016/S0550-3213(99)00809-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0550321399008093.