# CWebGen

## A tool to study colour structure of scattering amplitudes in IR limit

Neelima Agarwal [a], Sourav Pal [b], Aditya Srivastav [c], Anurag Tripathi [c]

[a] *Department of Physics, Chaitanya Bharathi Institute of Technology, Gandipet, Hyderabad, Telangana State 500075, India*

[b] *School of Physical Sciences, National Institute of Science Education and Research, An OCC of Homi Bhaba National Institute, Jatni 752050, India*

[c] *Department of Physics, Indian Institute of Tehnology Hyderabad, Kandi, Sangareddy, Telenagana State 502284, India*

*E-mail:* dragarwalphysics@gmail.com, sourav@niser.ac.in, shrivastavadi333@gmail.com, tripathi@phy.iith.ac.in

Abstract: Infrared singularities in perturbative Quantum Chromodynamics (QCD) are captured by the Soft function, which can be calculated efficiently using Feynman diagrams known as webs. The starting point for calculating Soft function using webs is to compute the web mixing matrices using a well known replica trick algorithm. We present a package implemented in Mathematica to calculate these mixing matrices. Along with the package, we provide several state-of-the art computations.

**Program Summary**

*Manuscript Title:* `CWebGen`– A tool to study colour structure of scattering amplitudes in IR limit
*Authors:* Neelima Agarwal, Sourav Pal, Aditya Srivastav, Anurag Tripathi
*Program title:* `CWebGen`
*Licensing provisions: GNU General Public License 3 (GPL)*
*Programming language:* WOLFRAM MATHEMATICA version 11.3 or higher
*Computer(s) for which the program has been designed:* desktop PC, compute nodes/workstations
*Operating system(s) for which the program has been designed:* Linux 64bit, Mac Os, Windows
*RAM required to execute with typical data:* 8 GB and above depending on the complexity of the problem.
*Has the code been vectorized or parallelized?:* no
*Number of processors used:* any number of processors
*Supplementary material:* this article, examples
*Keywords:* Feynman diagrams, Multiparton scattering amplitudes, Soft function, Webs, Cwebs, Replica trick
*Nature of problem:* Calculation of Web mixing matrices that appear in the Soft function in the study of scattering amplitudes.
*Solution method:* Starting with the colour of one diagram the code generates all the diagrams of a chosen Cweb. It then proceeds through the implementation of the replica trick to obtain Web mixing matrices. These matrices are then diagonalized and exponentiated colour factors are obtained which can be further simplified by the user separately.

## Contents



# 1 Introduction

Perturbative corrections of scattering amplitudes with massless gauge bosons exhibit infrared (IR) and mass (collinear) divergences. Over nearly a century of studies [1–17], several important features of these singularities have been uncovered (see [18] for a recent review). These singularities are universal, *i.e.*, they do not depend on the structure of hard scattering processes. The IR divergences in a multiparton amplitude are captured by the Soft function which is given by the correlators of Wilson lines,

$$\mathcal{S}_n\left(\zeta_i\right) \equiv \langle 0|\prod_{k=1}^{n} \Phi\left(\zeta_k\right)|0\rangle \,, \tag{1}$$

where $\Phi\left(\zeta\right)$ is given by,

$$\Phi\left(\zeta\right) \equiv \mathcal{P}\exp\left[\mathrm{i}g\int_{\zeta} dx\cdot \mathbf{A}(x)\right] . \tag{2}$$

Here $\mathbf{A}^{\mu}(x) = A^{\mu}_{a}(x)\,\mathbf{T}^{a}$ is a non-abelian gauge field, and $\mathbf{T}^{a}$ are the generators of the gauge algebra, which can be taken to belong to any desired representation, and $\mathcal{P}$ denotes path ordering of the gauge fields.

The soft function can be written as an exponential of sum of sets of diagrams:

$$\mathcal{S}_n = \exp\left[\sum W_n\right], \tag{3}$$

here $W_n$ is known as a *web*. Webs were first defined as two-line irreducible diagrams for scattering involving two Wilson lines [19–21]. Webs in mulitparton scattering amplitudes in non abelian gauge theories [22, 23] are defined as the sets of diagrams that differ from each other by the order of gluon attachments on each Wilson line. A generalisation of a web called Cweb was introduced in [24, 25]. Cwebs are especially useful for enumeration and organisation of webs at higher orders in the perturbation theory, as already shown in [24–27] for four loops and in [28] for five loops.

A Cweb is defined as a set of diagrams closed under shuffles of gluon attachments on each Wilson line and are made up of gluon correlators instead of Feynman diagrams. The Soft function is then given by,

$$\mathcal{S} = \exp\left[\sum_W \sum_{d,d' \in W} \mathcal{K}(d)\, R(d,d')\, C(d')\right]. \tag{4}$$

Here $d$ denotes a diagram in a Cweb $W$, $\mathcal{K}(d)$ and $C(d)$ denote their kinematic and colour factors, and $R$ is called the web mixing matrix. Web mixing matrix acts on the colour of the diagrams to provide the exponentiated colour factor $\widetilde{C}(d)$ as,

$$\widetilde{C}(d) = \sum_{d' \in W} R(d,d')\, C(d'). \tag{5}$$

The starting point of calculating Soft function using diagrammatic exponentiation is to compute the web mixing matrices. The web mixing matrices have three general properties:

(i) Web mixing matrices are idempotent, that is, $R^2 = R$.

(ii) The entries of $R$ obey the zero row sum rule, $\sum_{d'} R(d,d') = 0$.

(iii) The mixing matrices also obey a conjectured column-sum rule of the form $\sum_d s(d)R(d,d') = 0$.

For a detailed discussion on these properties and their physical implications, see refs [23, 24, 26, 28–31].

Web mixing matrices were first introduced in [23] and a replica trick algorithm was developed to calculate them. Combinatorially, application of replica trick algorithm is cumbersome and thus several attempts have been made towards calculating the mixing matrices using other methods. Use of Posets [32–34], and diagrammatic idea of Fused Webs [27, 29] provide two alternate methods to calculate the web mixing matrices bypassing the replica trick algorithm. Both these methods can calculate the full mixing matrices for certain classes of Cwebs. However, for complete calculation of web mixing matrices at a given loop order, employing replica trick algorithm is indispensable. In this paper, we address the problem of computing web mixing matrices using the well-known replica trick algorithm.

We present a Mathematica package `CWebGen` that calculates the web mixing matrices by taking the expression of colour of a single diagram of a Cweb as an input. Our package operates independent of any external dependencies and is capable of calculating web mixing matrices at significantly higher perturbative orders.

The rest of the paper is structured as follows. In section (2), we explain replica trick algorithm and explain its application with an example of a two-loop Cweb. Next, in section (3), we provide the Mathematica package `CWebGen` which implements the replica trick algorithm to obtain web mixing matrices discussed above. In section (4) we provide the instructions for installation of the Mathematica package and explain the usage of the package in *integrated* and in *standalone* modes. Finally, in section (5), we summarise our package.

# 2 Preliminaries

In this section we briefly discuss the replica trick algorithm which was developed in [23, 35] and used in [23, 36] to calculate three-loop web mixing matrices. Recently, we calculated four- and five-loop web mixing matrices in [24–26] and [28] respectively. We outline the steps of the replica trick algorithm below and provide an example on how to calculate it for a two-loop Cweb.

## 2.1 Replica Trick algorithm

Replica trick is often used in statistical physics to calculate partition functions with very large number of particles [37]. It has been used for the soft gluon exponentiation [23] and recently explored for the study of soft quark exponentiation [38] as well. In this section, we explain the use of replica trick in the context of soft gluon exponentiation. To start with, consider the following path integral,

$$\mathcal{S}_n(\gamma_i) = \int \mathcal{D}A_\mu^a \, \exp(iS(A_\mu^a)) \prod_{k=1}^{n} \phi_k(\gamma_k) = \exp[\mathcal{W}_n(\gamma_i)] \,, \tag{6}$$

where $S(A_\mu^a)$ is the classical action of the gauge fields and $\phi_k$ are the Wilson lines. Note that, here $\mathcal{W}_n$ denotes the sum of webs as described in eq. (3). In order to implement the replica trick algorithm, $N_r$ non-interacting, identical copies of each gluon field $A_\mu$ are introduced, meaning that each $A_\mu$ is replaced by $A_\mu^i$ for $i = 1, \ldots, N_r$. For each replica, a corresponding copy of each Wilson line is also associated, effectively replacing each Wilson line with a product of $N_r$ Wilson lines. Consequently, the path integral in the replicated theory is expressed as follows:

$$\mathcal{S}_n^{\text{repl.}}(\gamma_i) = \Big[\mathcal{S}_n(\gamma_i)\Big]^{N_r} = \exp\Big[N_r \, \mathcal{W}_n(\gamma_i)\Big] = \mathbf{1} + N_r \, \mathcal{W}_n(\gamma_i) + \mathcal{O}(N_r^2) \,. \tag{7}$$

Now, using this equation, one can determine $\mathcal{W}_n$ by calculating $\mathcal{O}(N_r)$ terms of the Wilson line correlator in the replicated theory. The method of replicas involves five steps, which are summarized below.

- Associate a replica number to each gluon correlator in a Cweb. This number is associated to each of the colour generators of that correlator; for example a gluon attachment on the Wilson line $k$, is represented by the colour generator $\mathbf{T}_k^{(i)}$ which belongs to a correlator with replica number $i$.

- In the next step one needs to find the hierarchies between the replica numbers present in a Cweb. If a Cweb has $m$ connected pieces, the number of possible hierarchies, denoted by $h(m)$, corresponds to the Bell number or Fubini number [39]. The first few Fubini numbers are given by $h(m) = \{1, 1, 3, 13, 75, 541\}$ for $m = 0, 1, 2, 3, 4, 5$.

- Define a replica ordering operator $\mathbf{R}$ which acts on a product of colour matrices, say $\mathbf{T}_k^{(i)}\mathbf{T}_k^{(j)}$ along a given line $k$. The action of $\mathbf{R}$ for two color matrices acting on line $k$ is defined as

$$\mathcal{R}\left[\mathbf{T}_k^{(i)}\,\mathbf{T}_k^{(j)}\right] = \begin{cases} \mathbf{T}_k^{(i)}\,\mathbf{T}_k^{(j)} & i \leq j \\ \mathbf{T}_k^{(j)}\,\mathbf{T}_k^{(i)} & i > j \end{cases} . \tag{8}$$

  This ordering on each line gives the replica ordered colour factor for a given diagram.

- Now one calculates the multiplicity $M_{N_r}(h)$, which counts the number of appearances of a given hierarchy $h$ in the presence of $N_r$ replicas, as

$$M_{N_r}(h) \;=\; {}^{N_r}\mathrm{C}_{n_r(h)}\,, \tag{9}$$

  where $n_r(h)$ is the number of distinct replica number.

- For a diagram $D$, colour factor in the replicated theory is then given by,

$$C_{N_r}^{\text{repl.}}(D) \;=\; \sum_h M_{N_r}(h)\,\mathbf{R}\big[C(D)|h\big]\,, \tag{10}$$

  where $\mathbf{R}[C(D)|h]$ is the replica ordered colour factor of diagram $D$, for hierarchy $h$. Finally, the exponentiated colour factor (ECF) for diagram $D$ is computed by extracting the coefficient of $\mathcal{O}(N_r)$ terms of the above equation.

### 2.2 A two-loop example

In this section, we illustrate the calculation of the mixing matrix for a two-loop web. Later, in section (4) we will show how to obtain the same mixing matrix using our package. Throughout this paper, we will follow the notation of Cwebs given refs. [24–26, 28, 29].

We choose the Cweb $\mathrm{W}_3^{(2)}(1, 2, 1)$ to illustrate the application of the replica trick in calculating the mixing matrices. The first step in the calculation is to generate the set of diagrams that remain closed under shuffle of the attachments on the Wilson lines. For this specific Cweb, the closed set contains two diagrams, as shown in Fig. (1). In the original (unreplicated) theory, the amplitudes of these two diagrams are given by

$$A_1 = \mathcal{K}_1 C_1, \qquad A_2 = \mathcal{K}_2 C_2, \tag{11}$$

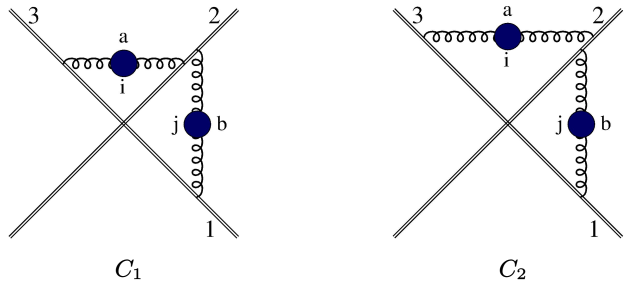


Figure 1: Diagrams of Cweb $\mathrm{W}_3^{(2)}(1,2,1)$

where $\mathcal{K}_i$ and $C_i$ represent the kinematic and color factors of diagram $i$, respectively. In the replicated theory, the color factors of these diagrams are modified to $\widetilde{C}_i$. Our goal is to express the color factors in the replicated theory in terms of the color factors of the diagrams in the original (unreplicated) theory.

As mentioned in the previous section, we assign replica numbers to each of the two correlators. For the case in hand, we assign replica $i$ and $j$ to the two two-point gluon correlators. The action of replica ordering operator $\mathbf{R}$ only depends on whether the replica numbers are greater, equal or smaller than each other.

The orderings of the replica numbers are hierarchies, which in this case are given by: $i = j, i > j, i < j$. For diagram $C_1$, consider the case where $i = j$, the action of $\mathbf{R}$ does not change the colour of the diagram and the colour in the replicated theory remains $C_1$. For $i > j$, the gluons are ordered according to their replica numbers, thus $\mathbf{R}(C_1|i > j) = C_2$, similarly for $i < j$, $\mathbf{R}(C_1|i < j) = C_1$. Similar analysis can be performed for $C_2$ as shown in table (1).

The next object $M_{N_r}(h)$ can be obtained simply by summing over number of appearances of a particular hierarchy in the presence of $N_r$ replicas. For $i = j$, this is simply given by $N_r$, and for $i \neq j$, this number is given by ${}^{N_r}\mathrm{C}_2$. Then following eq. (10), the exponentiated colour factors are obtained by multiplying the $\mathcal{O}(N_r)$ coefficient with the replica ordered colour factor. These steps are summarised in table (1). The ECFs for this Cweb are then given by,

$$\begin{aligned} \widetilde{C}_1 &= \frac{1}{2}\,(C_1 - C_2) \\ \widetilde{C}_2 &= \frac{1}{2}\,(C_2 - C_1)\ . \end{aligned} \tag{12}$$

The corresponding mixing matrix is then given by,

$$R = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\ . \tag{13}$$

In our package `CWebGen`, we automate the process described above and generate table (1) for each web. The next section provides a detailed explanation of the implementation of these steps in various subroutines of `CWebGen`.

| Diagrams | hierarchy | $\mathbf{R}(C_i\|h)$ | $M_{N_r}$ | $\mathcal{O}(N_r)$ |
|---|---|---|---|---|
| $\mathbf{C_1}$ | $i=j$ | $\mathbf{C_1}$ | ${}^{N_r}\mathrm{C}_1$ | 1 |
| | $i>j$ | $\mathbf{C_2}$ | ${}^{N_r}\mathrm{C}_2$ | $-\frac{1}{2}$ |
| | $i<j$ | $\mathbf{C_1}$ | ${}^{N_r}\mathrm{C}_2$ | $-\frac{1}{2}$ |
| $\mathbf{C_2}$ | $i=j$ | $\mathbf{C_2}$ | ${}^{N_r}\mathrm{C}_1$ | 1 |
| | $i>j$ | $\mathbf{C_2}$ | ${}^{N_r}\mathrm{C}_2$ | $-\frac{1}{2}$ |
| | $i<j$ | $\mathbf{C_1}$ | ${}^{N_r}\mathrm{C}_2$ | $-\frac{1}{2}$ |

Table 1: Replica trick analysis of $\mathrm{W}_3^{(2)}(1,2,1)$.

# 3 Replica Trick – Implementation

The web mixing matrices are combinatorial objects and obtained using replica trick, often used to solve complicated problems in statistical mechanics. The steps to obtain the mixing matrices are mentioned in the previous section. In this section we describe the subroutines in our code that perform the required jobs to calculate the web mixing matrices.

1. To initiate the package, one needs to provide the colour factor of one of the diagrams of a web tagged with replica indices. An example of this input colour factor is presented in the next section, and in the example files.

2. The next step is to generate the color factors for all the diagrams present in a Cweb. We perform this step in a subroutine `diagramsOfWeb`. This subroutine generates shuffles among the gluon attachments on each Wilson lines and provides the list of colour of diagrams of a web.

3. In the next step we generate hierarchies among the replica numbers. Note that hierarchies are the ordering among replica numbers associated with gluon correlators. This step is performed in a subroutine named `hierarchies`. To perform this step we associate replica numbers to positive integers then use tuples to generate orderings among these integers.

4. In the next step we construct the replica ordering operator $\mathbf{R}$ and apply it on each diagram of a web to calculate the replica-ordered colour factor. We perform this step via subroutine named `repOrdColour`. Once the replica numbers are associated with the positive integers, for a given hierarchy it is then straightforward to arrange the colour on each leg of a diagram in ascending order of replica numbers. This effectively performs the action of operation of $\mathbf{R}$.

5. The next step is to calculate $M_{N_r}(h)$ which can be obtained simply by summing over number of appearances of a particular hierarchy in the presence of $N_r$ replicas. We need to extract $\mathcal{O}(N_r)$ terms of this quantity as mentioned in eq. (10).

For example, in a hierarchy $i = j, i \geq k$, $n_r = 2$. For each hierarchy, we obtain the number of distinct replica variables and obtain their $\mathcal{O}(N_r)$ terms by a subroutine named `orderNCoeff`.

6. The above five steps complete the five rows of table (1). A final function `mixingMatrix` combines all the above subroutines to calculate the mixing matrices. Once we have the mixing matrix, we diagonalise them by obtaining diagonalising matrices constructed out of the right eigenvectors of the mixing matrices. The independent ECFs are then obtained by multiplying the diagonalising matrix to the column vector constructed out of the colours of the diagrams of a web.

## 4 CWebGen - Installation and Usage

The package can be downloaded using the link,

https://github.com/souravhep/CwebGen

Alternatively, it can also be obtained by using the command

```
git clone https://github.com/souravhep/CwebGen.git
```

The package `CWebGen` does not need an installation and can be directly called in a Mathematica session. In the following subsections, we provide the details of usage and illustrate several examples.

### 4.1 Inputs

The package can be operated in two modes: `Integrated` and `Standalone`. As the name suggests, the integrated mode provides the mixing matrices of webs directly, whereas the Standalone mode provides step-by-step results to the user. In both of these modes the input format is the same. The colour of a diagram which is provided to the modules is written with the following instructions.

In Catani-Seymour colour notation [40], to write down the colour of gluon one needs two indices: (i) the leg index and (ii) the SU(3) index of the gluon. Though, this notation is very useful in simplifying certain colour algebra, however for applying replica trick algorithm, we need slight modification from this notation. In `CWebGen`, the colour of a gluon attachment has three indices (i) leg number, (ii) SU(3) colour index, and (iii) a replica index. For example, in a case where a gluon is attached to Wilson line 3 with SU(3) index $a$ and replica index $i$, the colour factor for this attachment is written as,

$$\mathbf{T}_3^a \quad \longrightarrow \quad \texttt{T[3,i,a]}\,. \tag{14}$$

Here LHS shows usual notation, and RHS shows notation that is required in the package. For more than one attachment to a Wilson line, the colour factor for that line is given by Noncommutative multiplication of several such factors. The colour for a diagram is a non-commutating product of such colour generators on each line. Along with these, one needs to provide the structure constants for colour of the diagrams in a web. Note that the structure constants are commuting in nature, and thus can be provided using `Times` function of Mathematica. For example,

$$f^{abc} f^{cde} \quad \longrightarrow \quad \texttt{f[a,b,c]*f[c,d,e]}\,, \tag{15}$$

where the LHS again denotes colour factors in $SU(3)$, whereas the RHS denotes the format required as input to run the package.

The replica trick algorithm only requires the information about attachments on the Wilson lines and does not require the information happening away from the Wilson lines. Therefore in our package, we consider the attachments denoted by head `T` and structure constant denoted by head `f` as two different objects. Though the structure constants do not take part in the replica trick algorithm, however not providing them as an input to the package will not generate the correct expressions for exponentiated colour factors once all the operations are done.

Except the `mixingMatrix[ ]` module all other Mathematica commands of `CWebGen` does not need the structure constants. Therefore for the package – we denote the input for each web, these two parts of the colour factors as,

$$\begin{aligned} \texttt{webcolour} &= \texttt{T[1,i1,a1]**T[1,i2,a2]**...**T[n,in,an]} \\ \texttt{ff} &= \texttt{f[a1,b1,c1]*f[a2,b2,c2]*...*f[an,bn,cn]} \end{aligned} \tag{16}$$

For the two-loop web shown in fig. (2), one can write the colour factor as

$$C_1 = \mathbf{T}_1^b \mathbf{T}_2^b \mathbf{T}_2^a \mathbf{T}_3^a \,. \tag{17}$$

which can be translated to the input for `CWebGen` as

$$\begin{aligned} \texttt{webcolour} &= \texttt{T[1,j,b]**T[2,j,b]**T[2,i,a]**T[3,i,a]} \\ \texttt{ff} &= 1 \end{aligned} \tag{18}$$

## 4.2 Operating modes

The Mathematica package `CWebGen` offers two operational modes, `Integrated` and `Standalone`. Both these modes take inputs in the same format as described in the previous section.

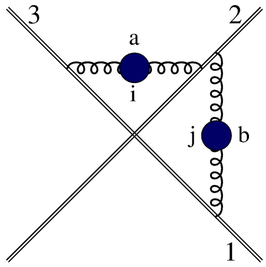


Figure 2: One diagram of a two loop Cweb

### 4.2.1 Integrated Mode

In this mode of the package, there is one Mathematica module,

`mixingMatrix[webcolour,ff]` ,

which takes two inputs derived for a web as mentioned in the eq. (16). The output of this module creates a directory **`results/`** which contain four storing Mathematica files

| | |
|---|---|
| **`diagrams.m`** | contains a list of diagrams (colours) of the web |
| `R.m` | contains the rank and mixing matrix for the web |
| | default basis for `R` is list of diagrams in provided in **`diagrams.m`** |
| `Y.m` | contains a diagonalizing matrix for $R$ obtained from right eigenvectors |
| `ECFs.m` | contains a list of exponentiated colour factors ($YC$) of the web |

Along with these files, it provides the check of row-sum rule for $R$ and idempotence in the current session as

$$\texttt{idempotence} = \texttt{True}$$
$$\texttt{Row Sum} = \{0, 0, \ldots, 0\}$$

**Two-loop Web:** The inputs for two loop web shown in fig. (2) are provided in eq. (16) is fed to `mixingMatrix[]` as

`mixingMatrix[T[1,j,b]**T[2,j,b]**T[2,i,a]**T[3,i,a],1]` ,

and it creates a **`results/`** directory with the following files therein,

**`diagrams.m`**

{ `T[1,j,b]**T[2,j,b]**T[2,i,a]**T[3,i,a]`,
`T[1,j,b]**T[2,i,a]**T[2,j,b]**T[3,i,a]` }

We can see two diagrams are generated for the web.

**R.m**

$$\{1,\ \{\{1/2,\ -1/2\},\ \{-1/2,\ 1/2\}\}\}$$

The first entry of list is the rank of the mixing matrix, and the second is an array which is mixing matrix $R$.

**Y.m**

$$\{\{-1,\ 1\},\ \{1,\ 1\}\}$$

It is an array representing the diagonalizing matrix $Y$.

**ECFs.m**

```
{T[1,j,b]**T[2,i,a]**T[2,j,b]**T[3,i,a]**1
-T[1,j,b]**T[2,j,b]**T[2,i,a]**T[3,i,a]**1}
```

Note that this file provides the ECFs which is obtained using $Y$ not using $R$ (see the difference [24, 25]). Evidently we can see only one ECF in this case which is consistent with the rank of the mixing matrix.

### 4.2.2 Standalone mode

In this mode of the package there are modules for each subroutine and they provide the associated outputs. One can obtain the intermediate results of the Integrated mode – e.g. list of diagrams, hierarchies, replica ordered colour factors – without running the `mixingMatrix[]`. We provide below the the list of functions available and their functionality.

1. `diagramsOfWeb[webcolour]`: This function takes input the colour factor of one diagram and do the shuffle to generate all the diagrams of the web. The output of this function is list of the diagrams (colour) of web. For example, for the two loop case mentioned in previous section, running

   `diagramsOfWeb[T[1,j,b]**T[2,j,b]**T[2,i,a]**T[3,i,a]]` (19)

   will give the following list

   ```
   { T[1,j,b]**T[2,j,b]**T[2,i,a]**T[3,i,a],
     T[1,j,b]**T[2,i,a]**T[2,j,b]**T[3,i,a] }
   ```

2. `hierarchies[webcolour]`: The input to this function is again the same – colour of one diagram. For a Cweb with $n$-correlators it provides all the possible orderings among the correlator indices (replica indices). For the same two-loop web:

   `hierarchies[T[1,j,b]**T[2,j,b]**T[2,i,a]**T[3,i,a]]` $= \{\{1,1\},\{1,2\},\{2,1\}\}$.

In this case we have two correlators assigned with replica indices $\{j,i\}$. Each sublist of the output above defines an integer ordering which is understood as the ordering of replica indices

$$\{\{1,1\},\{1,2\},\{2,1\}\} \;=\; \{\{j=i\},\{j<i\},\{j>i\}\}$$

Therefore this function provides the ordering among replica indices.

3. `repOrdColour[colour,hierarchy]`: This function is used to calculate the replica ordered colour factor for each diagram of the web subjected a given hierarchy. For example if we want to calculate replica ordered colour factor for second diagram of the two loop web with the condition that $\{j>i\}$, the inputs are,

   `repOrdColour[T[1,j,b]**T[2,i,a]**T[2,j,b]**T[3,i,a],{2,1}]` ,

   we get `T[1,j,b]**T[2,j,b]**T[2,i,a]**T[3,i,a]`, which is the colour structure of first diagram.

4. `orderNCoeff[hierarchy]`: It provides $\mathcal{O}(N_r)$ coefficient that is multiplied with the replica ordered colour factor generated for a given hierarchy ordering. Input to this function is the hierarchy ordering for which the replica ordered colour factor is being calculated. For the case illustrated above, i.e. $\{j>i\}$ we have,

$$\texttt{orderNCoeff}[\{2,1\}] \;=\; -\frac{1}{2}$$

## 4.3 Examples

Along with the package we provide a directory of examples for Cwebs appearing upto six-loop. In this section we provide details of three Cwebs of different kinds. The examples can be downloaded from,

https://github.com/souravhep/CWebGen/tree/main/examples.

### Cwebs $\mathrm{W}_{\mathrm{n+1}}^{(\mathrm{n})}(1,1,1,\ldots,\mathrm{n})$

First category contains the Cwebs of type $\mathrm{W}_{\mathrm{n+1}}^{(\mathrm{n})}(1,1,1,\ldots,\mathrm{n})$. We have provided the examples of this category from two-loop upto six-loop. These Cwebs are made out of two-point gluon correlators connected with the Wilson lines. For such a Cweb at $n$ loops there will be $n$ two-point correlators connecting $(n+1)$ Wilson lines as shown in fig. (3). For the illustration of code, we describe the details of the Cweb of this class present at six-loop. We choose to determine the mixing matrix and ECFs for $\mathrm{W}_{7}^{(6)}(1,1,1,\ldots,6)$. It has six two-point correlators connecting seven Wilson lines. The colour of one of the diagram is given as

$$C_1 = \mathbf{T}_1^a\mathbf{T}_2^b\mathbf{T}_3^c\mathbf{T}_4^d\mathbf{T}_5^e\mathbf{T}_6^g\mathbf{T}_7^g\mathbf{T}_7^e\mathbf{T}_7^d\mathbf{T}_7^c\mathbf{T}_7^b\mathbf{T}_7^a\,.$$

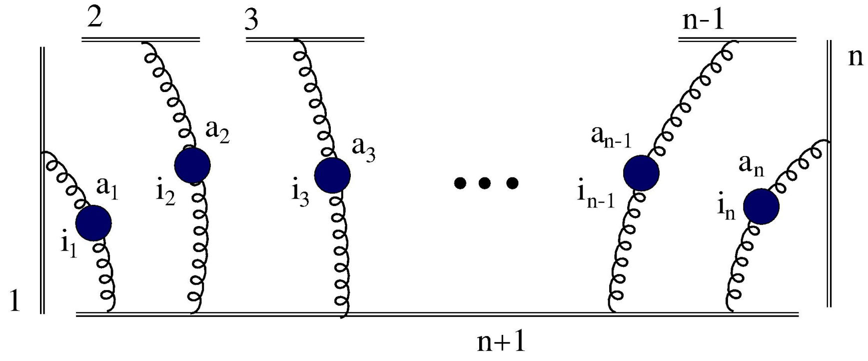


Figure 3: A diagram of a Cweb $\mathrm{W}^{(\mathrm{n})}_{\mathrm{n+1}}(1,1,1,\ldots,\mathrm{n})$

As there is no three or higher point gluon correlator, there is no structure constant in the colour of the diagrams thus we have the following input for the Mathematica function `mixingMatrix[]`,

```
webcolour = T[1,i,a]**T[2,j,b]**T[3,k,c]**T[4,l,d]**T[5,m,e]**T[6,n,g]
            **T[7,n,g]**T[7,m,e]**T[7,l,d]**T[7,k,c]**T[7,j,b]**T[7,i,a]
     ff = 1
```

With these inputs the mathemtica function `mixingMatrix[webcolour,ff]`, creates a directory `/results` to keep the results in separate files as mentioned in section (4.2.1). Along with the results, It provides the checks of idempotence and zero row sum on mixing matrix in the Mathematica notebook session as.

$$\texttt{idempotence} = \texttt{True}$$
$$\texttt{Row Sum} = \{0, 0, \ldots, 0\}$$

### Cwebs with higher-point correlators

In this class we provide two examples for the users. These two four loops Cwebs contain three and four point gluon correlators respectively, along with the two-point correlators. We describe the details of a Cweb $\mathrm{W}_4^{(2,1)}(2,1,1,3)$, made up of one three-point and two two-point correlators connected to four Wilson lines, as shown in fig. (4). Color of this diagram is written as

$$C_1 = i\, f^{abc}\, \mathbf{T}_1^c \mathbf{T}_1^e \mathbf{T}_2^e \mathbf{T}_3^d \mathbf{T}_4^d \mathbf{T}_4^a \mathbf{T}_4^b \,.$$

The `mixingMatrix[webcolour,ff]` with the input,

```
webcolour = T[1,i,c]**T[1,j,e]**T[2,j,e]**T[3,k,d]**T[4,k,d]**T[4,i,a]**T[4,i,b]
     ff = I*f[a,b,c]
```

provides the results in output files in a newly created `results/` directory as described in previous sections. Also, the mixing matrix for this case satisfies the idempotence and zero row sum property.

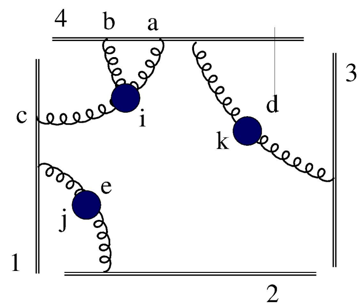


Figure 4: Diagram of a Cweb $\mathrm{W}_4^{(2,1)}(2,1,1,3)$

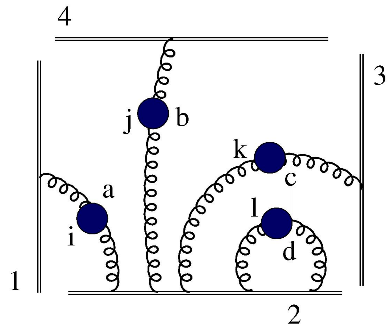


Figure 5: A boomerang Cweb $W_4^{(4)}(1,1,1,5)$.

### Boomerang Cweb

A Cweb containing one two-point correlator whose both ends attach to the same Wilson line, is defined as Boomerang Cwebs [26]. Our code works equally well for this class of Cwebs as well. The input and output format are same as previously. For this class we consider the Boomerang Cweb, $\mathrm{W}_4^{(4)}(1,1,1,5)$ present at four-loop, shown in fig. (5). The colour of one diagram of this Cweb is given as

$$C_1 = \mathbf{T}_1^a \mathbf{T}_2^b \mathbf{T}_3^c \mathbf{T}_4^d \mathbf{T}_4^d \mathbf{T}_4^c \mathbf{T}_4^b \mathbf{T}_4^a \,.$$

Looking closely the above expression, one can see a product of generators $\mathbf{T}_4^d \mathbf{T}_4^d$ corresponding to attachments of Boomerang correlator. Note that we do not put this product to Casimir in the beginning of the calculation. That simplification is performed in the results manually. Calculation of mixing matrix and ECFs for this case is no different than the previous ones. Executing the function `mixingMatrix[webcolour,ff]` with the input,

```
webcolour = T[1,i,a]**T[2,j,b]**T[3, k,c]**T[4,l,d]**T[4,l, d]
            **T[4,k,c]**T[4,j,b] **T[4,i,a]
ff = 1
```

saves the results in output files in the `results/` directory.

### 4.4 Benchmarks

To test the efficiency of our code, we choose a class of Cwebs $\mathrm{W}_{n+1}^{(n)}(1, 1, \ldots n)$, shown in fig. (3) whose mixing matrices have dimension $n!$, where $n$ is the perturbative order. Further, the Cwebs of this class have the largest dimension of mixing matrices at a given perturbative order connecting maximum number of lines. We run a benchmark test on the computation time for the two loop Cweb that belongs to this category. We also provide the computation times for the Cwebs of this category for three and four loops, which are already known previously in the literature. To test the capability of our code `CWebGen` in handling higher orders, we have computed mixing matrices of this category appearing at five and six loops. The computation times of all these mixing matrices are provided in table (2). It is evident that `CWebGen` is well-suited for handling higher loop computations.

| loops | Dimension of $R$ | Computation time |
|---|---|---|
| 2 | 2 | 0.021 s |
| 3 | 6 | 0.035 s |
| 4 | 24 | 0.407 s |
| 5 | 120 | 24.074 s |
| 6 | 720 | 4.64 h |

Table 2: Benchmark tests of `CWebGen`. All these tests are performed in a computer with Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz and 40GB RAM.

### 4.5 Limitations

While `CWebGen` is capable of calculating the mixing matrices for the majority of webs at a given perturbative order, it does have certain limitations. The mixing matrices for a particular class of Cwebs cannot be obtained form the package by providing colour of a single diagram of a web. In this class of webs, identical copies of two or more correlators are attached to the same set of Wilson lines in an identical fashion. The shuffle of these correlators on each Wilson line leads to multiple duplicates of each diagram within a Cweb, which must be discarded before applying the replica trick algorithm. Since the replica trick for these Cwebs is applied only to a subset of the shuffles, our package cannot be used to generate mixing matrices for them. The complexities of applying the replica trick to this class of Cwebs are discussed in more detail in section 3 of [25]. However, one can obtain the mixing matrices for these Cwebs by identifying and removing the duplicate diagrams and then applying the replica trick by using several functions of the package in the Standalone mode.

## 5 Summary

The Soft function in perturbative QCD can be computed using Feynman diagrams known as webs. A Web is a set of diagrams whose kinematic and colour factors mix via the web mixing matrix. A replica trick-based algorithm is crucial for calculating these mixing matrices at a given perturbative order. In this article, we introduce and describe the Mathematica package `CWebGen`, designed to efficiently compute web mixing matrices. Along with instructions for using the package, we provide several examples to demonstrate its application in future calculations. Except for a specific class of webs, `CWebGen` can be employed to compute web mixing matrices at any loop order. We have tested the package for webs up to six loops and included the benchmark points. We anticipate this package will be valuable for computing web mixing matrices at higher loop orders. The future goal of the package will be to integrate it with diagram generators such as QGRAF [41], FeynArts [42] to generate all the Cwebs at a given perturbative order and provide the colour factors of the required form.

## Acknowledgements

NA, SP, AT would like to thank Lorenzo Magnea for collaborating on the projects that necessitated the construction of the first working version of this package. The research of SP is supported by the SERB Grant SRG/2023/000591. AS would like to thank CSIR, Govt. of India, for an SRF fellowship (09/1001(0075)/2020-EMR-I).

## References

[1] F. Bloch and A. Nordsieck, *Note on the Radiation Field of the electron*, Phys. Rev. **52** (1937) 54.

[2] V. V. Sudakov, *Vertex parts at very high-energies in quantum electrodynamics*, *Sov. Phys. JETP* **3** (1956) 65.

[3] D. R. Yennie, S. C. Frautschi and H. Suura, *The infrared divergence phenomena and high-energy processes*, Annals Phys. **13** (1961) 379.

[4] T. Kinoshita, *Mass singularities of Feynman amplitudes*, J. Math. Phys. **3** (1962) 650.

[5] T. D. Lee and M. Nauenberg, *Degenerate Systems and Mass Singularities*, Phys. Rev. **133** (1964) B1549.

[6] G. Grammer, Jr. and D. R. Yennie, *Improved treatment for the infrared divergence problem in quantum electrodynamics*, Phys. Rev. **D8** (1973) 4332.

[7] A. H. Mueller, *On the Asymptotic Behavior of the Sudakov Form-factor*, *Phys. Rev.* **D20** (1979) 2037.

[8] J. C. Collins, *Algorithm to Compute Corrections to the Sudakov Form-factor*, *Phys. Rev.* **D22** (1980) 1478.

[9] A. Sen, *Asymptotic Behavior of the Sudakov Form-Factor in QCD*, *Phys. Rev.* **D24** (1981) 3281.

[10] A. Sen, *Asymptotic Behavior of the Wide Angle On-Shell Quark Scattering Amplitudes in Nonabelian Gauge Theories*, *Phys. Rev.* **D28** (1983) 860.

[11] G. P. Korchemsky and A. V. Radyushkin, *Renormalization of the Wilson Loops Beyond the Leading Order*, *Nucl. Phys.* **B283** (1987) 342.

[12] G. P. Korchemsky, *Sudakov Form-factor in QCD*, *Phys. Lett.* **B220** (1989) 629.

[13] L. Magnea and G. F. Sterman, *Analytic continuation of the Sudakov form-factor in QCD*, *Phys. Rev.* **D42** (1990) 4222.

[14] L. J. Dixon, L. Magnea and G. F. Sterman, *Universal structure of subleading infrared poles in gauge theory amplitudes*, *JHEP* **08** (2008) 022 [0805.3515].

[15] E. Gardi and L. Magnea, *Factorization constraints for soft anomalous dimensions in QCD scattering amplitudes*, *JHEP* **03** (2009) 079 [0901.1091].

[16] T. Becher and M. Neubert, *On the Structure of Infrared Singularities of Gauge-Theory Amplitudes*, *JHEP* **06** (2009) 081 [0903.1126].

[17] I. Feige and M. D. Schwartz, *Hard-Soft-Collinear Factorization to All Orders*, *Phys. Rev.* **D90** (2014) 105020 [1403.6472].

[18] N. Agarwal, L. Magnea, C. Signorile-Signorile and A. Tripathi, *The infrared structure of perturbative gauge theories*, *Phys. Rept.* **994** (2023) 1 [2112.07099].

[19] G. F. Sterman, *Infrared divergences in perturbative QCD*, *AIP Conf. Proc.* **74** (1981) 22.

[20] J. G. M. Gatheral, *Exponentiation of Eikonal Cross-sections in Nonabelian Gauge Theories*, *Phys. Lett.* **133B** (1983) 90.

[21] J. Frenkel and J. C. Taylor, *Nonabelian eikonal exponentiation*, *Nucl. Phys.* **B246** (1984) 231.

[22] A. Mitov, G. Sterman and I. Sung, *Diagrammatic Exponentiation for Products of Wilson Lines*, *Phys. Rev.* **D82** (2010) 096010 [1008.0099].

[23] E. Gardi, E. Laenen, G. Stavenga and C. D. White, *Webs in multiparton scattering using the replica trick*, *JHEP* **11** (2010) 155 [1008.0098].

[24] N. Agarwal, A. Danish, L. Magnea, S. Pal and A. Tripathi, *Multiparton webs beyond three loops*, *JHEP* **05** (2020) 128 [2003.09714].

[25] N. Agarwal, L. Magnea, S. Pal and A. Tripathi, *Cwebs beyond three loops in multiparton amplitudes*, *JHEP* **03** (2021) 188 [2102.03598].

[26] N. Agarwal, S. Pal, A. Srivastav and A. Tripathi, *Deciphering colour building blocks of massive multiparton amplitudes at 4-loops and beyond*, *JHEP* **02** (2023) 258 [2212.06610].

[27] N. Agarwal, S. Pal, A. Srivastav and A. Tripathi, *Correlator webs of massive multiparton amplitudes at four loops: A study of boomerang webs*, *Phys. Rev. D* **109** (2024) 094038 [2307.15924].

[28] S. Mishra, S. Pal, A. Srivastav and A. Tripathi, *Multiparton Cwebs at five loops*, *JHEP* **07** (2024) 078 [2305.17452].

[29] N. Agarwal, S. Pal, A. Srivastav and A. Tripathi, *Building blocks of Cwebs in multiparton scattering amplitudes*, *JHEP* **06** (2022) 020 [2204.05936].

[30] E. Gardi and C. D. White, *General properties of multiparton webs: Proofs from combinatorics*, *JHEP* **03** (2011) 079 [1102.0756].

[31] C. D. White, *An Introduction to Webs*, *J. Phys. G* **43** (2016) 033002 [1507.02167].

[32] M. Dukes, E. Gardi, E. Steingrimsson and C. D. White, *Web worlds, web-colouring matrices, and web-mixing matrices*, *J. Comb. Theory Ser.* **A120** (2013) 1012 [1301.6576].

[33] M. Dukes, E. Gardi, H. McAslan, D. J. Scott and C. D. White, *Webs and Posets*, *JHEP* **01** (2014) 024 [1310.3127].

[34] M. Dukes and C. D. White, *Web matrices: structural properties and generating combinatorial identities*, 1603.01589.

[35] E. Laenen, G. Stavenga and C. D. White, *Path integral approach to eikonal and next-to-eikonal exponentiation*, *JHEP* **03** (2009) 054 [0811.2067].

[36] E. Gardi, J. M. Smillie and C. D. White, *The Non-Abelian Exponentiation theorem for multiple Wilson lines*, *JHEP* **06** (2013) 088 [1304.7040].

[37] M. Mézard, G. Parisi and M. Virasoro, *Spin glass theory and beyond*, *World Scientific Lecture Notes in Physics* **9** (1987) .

[38] M. van Beekveld, L. Vernazza and C. D. White, *Exponentiation of soft quark effects from the replica trick*, *JHEP* **07** (2024) 109 [2312.11606].

[39] https://oeis.org/A000670 .

[40] S. Catani and M. H. Seymour, *A General algorithm for calculating jet cross-sections in NLO QCD*, *Nucl. Phys. B* **485** (1997) 291 [hep-ph/9605323].

[41] P. Nogueira, *Automatic Feynman graph generation*, *J. Comput. Phys.* **105** (1993) 279.

[42] T. Hahn, *Generating Feynman diagrams and amplitudes with FeynArts 3*, *Comput. Phys. Commun.* **140** (2001) 418 [hep-ph/0012260].