

---

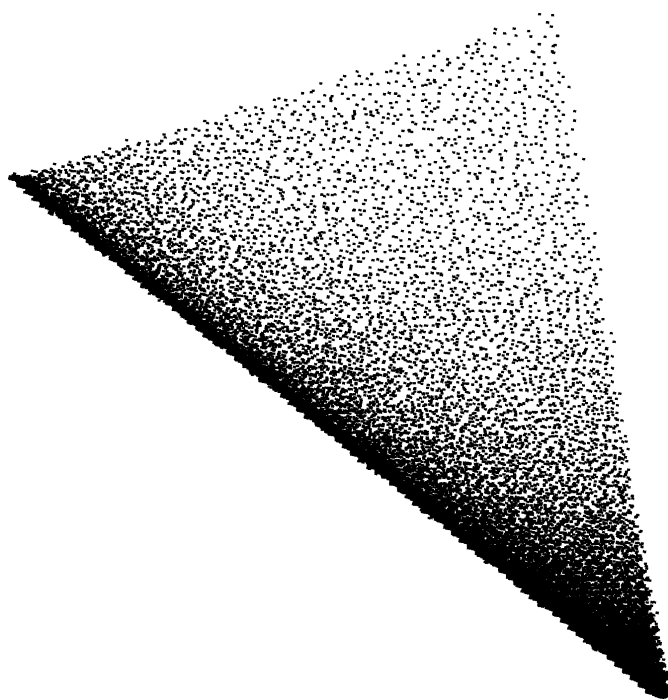
# Simulation de l'interaction

$$e^+e^- \rightarrow q\bar{q}g$$

---

PHYS-F477 : Physique des interactions fortes

Antoine Dierckx & Téo Bruffaerts



Supervisé par Max Vanden Bemden

IIHE

ULB, Belgique  
2024

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Correction radiative . . . . .	2
1.2	Calcul de la section efficace $e^+e^- \rightarrow q\bar{q}g$ . . . . .	2
1.3	Signification des divergences . . . . .	3
<b>2</b>	<b>Explication du code</b>	<b>3</b>
2.1	Changement de référentiel . . . . .	4
2.2	Variables générées et section efficace . . . . .	4
2.3	Méthode de réjection . . . . .	5
2.4	Niveau généré et reconstruit . . . . .	5
2.5	Angle de Ellis-Karliner $\theta_{EK}$ . . . . .	6
2.6	Algorithme JADE . . . . .	6
<b>3</b>	<b>Analyse des graphes</b>	<b>7</b>
3.1	Cosinus des angles $\theta_q$ et $\theta_{\bar{q}}$ dans le référentiel du centre de masse . . . . .	7
3.2	Les 3 angles d'Euler . . . . .	8
3.3	Fractions d'énergie $x_1$ et $x_2$ au niveau reconstruit . . . . .	9
3.4	Corrélations entre les $x_1$ , $x_2$ et $x_3$ . . . . .	11
3.5	$\theta_{EK}$ , l'angle d'Ellis-Karliner . . . . .	12
3.6	Corrélation entre la section efficace $\sigma$ en la fraction d'énergie $x_1^{\text{rec}}$ . . . . .	13
3.7	Algorithme de JADE . . . . .	14
<b>4</b>	<b>Interaction <math>q\bar{q} \rightarrow e^+e^-</math>, processus de Drell-Yan</b>	<b>15</b>
<b>5</b>	<b>Conclusion</b>	<b>15</b>
<b>6</b>	<b>Annexe</b>	<b>16</b>

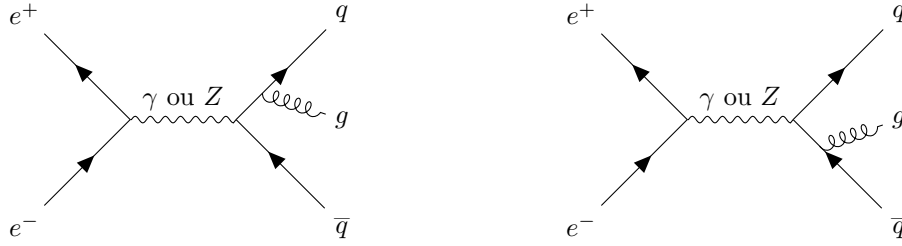
# 1 Introduction

Dans ce rapport, nous allons étudier le processus  $e^+e^- \rightarrow q\bar{q}g$ . Il s'agit d'une correction radiative QCD au processus  $e^+e^- \rightarrow q\bar{q}$ .

Après une brève introduction, nous rappellerons quelques résultats théoriques sur l'expression des différents éléments de matrices ainsi que sur le calcul de la section efficace. Nous préciserons les approximations nécessaires à la simplification des expressions obtenues. Ensuite, nous détaillerons les étapes permettant la simulation de ce processus, et analyserons les résultats de cette simulation. Enfin, nous terminerons avec un commentaire concernant le processus inverse et une courte conclusion.

## 1.1 Correction radiative

Le but de notre analyse est de simuler l'interaction  $e^+e^- \rightarrow q\bar{q}g$ . Au *tree level*, quatre diagrammes peuvent contribuer :



Nous allons effectuer un certain nombre d'approximations afin de simplifier l'élément de matrice et obtenir une simulation générée avec un nombre limité de variables aléatoires.

Remarquons que ce processus est une correction radiative QCD à l'interaction  $e^+e^- \rightarrow q\bar{q}$ . On considère un faisceau d'électrons ayant une énergie de  $E_{\text{beam}} = 10$  GeV dans le référentiel du laboratoire. Ainsi, l'énergie dans le centre de masse est donnée par :

$$\sqrt{s} \approx \sqrt{4E_{\text{beam}}^2} = 2E_{\text{beam}} = 20 \text{ GeV}$$

On peut alors négliger la contribution des diagrammes où un boson  $Z$  est échangé car nous sommes éloigné de son pic de masse ( $\sqrt{s} \ll m_Z \approx 91$  GeV). On peut également négliger la masse des électrons ( $\sqrt{s} \gg m_e \approx 0.511$  GeV). Il est possible de montrer [1] que l'élément de matrice au *tree level* sans correction peut s'écrire comme :

$$|\overline{\mathcal{M}}|^2 = 8 \frac{(4\pi)^2 \alpha^2}{s^2} N_C Q_q^2 \{ (p_q \cdot p_{e^+})(p_{\bar{q}} \cdot p_{e^-}) + (p_q \cdot p_{e^-})(p_{\bar{q}} \cdot p_{e^+}) + m_q^2(p_{e^+} \cdot p_{e^-}) \} \quad (1.1)$$

où  $\alpha = e^2/4\pi$  et  $N_C = 3$  est le nombre de couleurs.

Le calcul de la section efficace pour la correction radiative étant un problème à 3 corps, il nécessite un peu plus de travail.

## 1.2 Calcul de la section efficace $e^+e^- \rightarrow q\bar{q}g$

Afin de déterminer la section efficace différentielle de ce processus, nous effectuons un changement de coordonnées bien choisi pour nous placer dans un repère adapté.

### Paramétrisation de l'espace des phases

**Définition 1.1.** On définit la fraction d'impulsion  $x_i$  d'une particule  $i$  de l'état final selon :

$$x_i \triangleq \frac{2}{\sqrt{s}} E_i \quad (1.2)$$

Le facteur d'espace des phases s'exprime alors comme :

$$d\Phi_3 = (2\pi)^4 \delta^{(4)}(p_{e^+} + p_{e^-} - p_q - p_{\bar{q}} - p_g) \prod_{i=1}^3 \frac{s}{8(2\pi)^3} x_i dx_i d\cos\theta_i d\phi_i$$

**Changement de référentiel** Dans le référentiel du laboratoire  $(x, y, z)$ , le faisceau d'électrons arrive selon  $\vec{u}_z$  et le faisceau de positrons selon  $-\vec{u}_z$ . Afin de simplifier les calculs, nous nous plaçons un référentiel  $(X, Y, Z)$  dans lequel le gluon est émis selon  $u_{\vec{X}}$ .

**Définition 1.2.** On introduit les trois angles d'Euler suivants :

- $\alpha \in [0, 2\pi]$  le premier angle de rotation autour de l'axe  $z$
- $\beta \in [0, \pi]$  l'angle de rotation autour de l'axe  $x$
- $\gamma \in [0, 2\pi]$  le second angle de rotation autour de l'axe  $z$

Dans ce nouveau référentiel  $(X, Y, Z)$ , on introduit les angles suivants.

**Définition 1.3.**

- $\theta_1 \in [0, \pi]$  l'angle entre le quark et le gluon
- $\theta_2 \in [0, \pi]$  l'angle entre l'antiquark et le gluon
- $\phi_2 \in [0, 2\pi]$  l'angle entre la paire quark-antiquark

Il est possible de montrer [2] par quelques calculs triviaux qu'on obtient les conditions suivantes :

$$\cos \theta_1 = -\frac{x_1^2 + x_3^2 - x_2^2}{2x_1x_3} \quad (1.3)$$

$$\cos \theta_2 = -\frac{x_2^2 + x_3^2 - x_1^2}{2x_2x_3} \quad (1.4)$$

$$x_1 + x_2 + x_3 = 2 \quad (1.5)$$

$$\phi_2 = \pi \quad (1.6)$$

où  $x_{1,2,3}$  sont respectivement les fractions d'énergie du quark, de l'antiquark et du gluon. On obtient également que la section efficace différentielle est donnée par :

$$\frac{d\sigma}{d\alpha dx_1 dx_2 d\cos\beta d\gamma} = \frac{C_F \alpha_{\text{EM}}^2 \alpha_s N_c Q_q^2}{4\pi^2} \left\{ \frac{x_1^2(1 + \sin^2(\beta) \sin^2(\gamma + \theta_1)) + x_2^2(1 + \sin^2(\beta) \sin^2(\gamma - \theta_2))}{s(1 - x_1)(1 - x_2)} \right\} \quad (1.7)$$

### 1.3 Signification des divergences

Dans l'expression (1.7), on observe des divergences en  $x_1 = 1$  et en  $x_2 = 1$ . Puisque les  $\{x_i\}$  répondent à la relation (1.5), deux types de divergences sont possibles :

- Lorsque soit  $x_1 \rightarrow 1$ , soit  $x_2 \rightarrow 1$ , il y a émission d'un *gluon colinéaire* à la particule dont la fraction d'impulsion tend vers 1.
- Lorsque les deux fractions d'impulsion  $x_1$  et  $x_2$  tendent vers 1 simultanément, on a  $x_3 \rightarrow 0$  : il y a émission d'un *gluon soft*.

## 2 Explication du code

Cette partie est dédiée aux détails de l'implémentation de la simulation de l'interaction  $e^+e^- \rightarrow q\bar{q}g$ . Le code peut être retrouvé dans son intégralité dans l'annexe 6.

*Remarque.* La simulation a été effectuée séparément pour des quarks de charges  $Q_{\uparrow} = 2/3$  (simulation *up*) et  $Q_{\downarrow} = -1/3$  (simulation *down*), puis sur un ensemble contenant une moitié de  $Q_{\uparrow}$  et une moitié de  $Q_{\downarrow}$  (simulation *mix*).

Lors de la création de la paire  $q - \bar{q}$  à partir d'un photon, la probabilité d'obtenir une paire  $u - \bar{u}$ ,  $d - \bar{d}$ ,  $c - \bar{c}$  et  $s - \bar{s}$  est la même. Les paires  $t - \bar{t}$  et  $b - \bar{b}$  ne peuvent pas être produites car l'énergie dans le centre de masse  $\sqrt{s}$  est insuffisante.

Le code commenté ci-dessous est issu de la simulation *down*.

## 2.1 Changement de référentiel

Le changement de référentiel s'effectue principalement dans cette partie :

```

1 // Cette fonction doit prendre les 3 angles en argument
2 // plus une reference au quadrivecteur a modifier.
3 void myRotate(double alpha, double beta, double gamma, TLorentzVector &particle)
4 {
5     // Pour acceder au trivecteur du quadrivecteur
6     // on utilise la methode Vect()
7     TVector3 myVect = particle.Vect();
8     TVector3 X (1.,0.,0.);
9     TVector3 Z (0.,0.,1.);
10    // La methode Rotate(alpha, V) effectue
11    // une rotation d un angle alpha autour de
12    // l axe defini par le vecteur V
13    myVect.Rotate(-gamma, Z); X.Rotate(-gamma, Z);
14
15    myVect.Rotate(-beta, X); Z.Rotate(-beta, X);
16
17    myVect.Rotate(-alpha, Z);
18
19    particle.SetVect(myVect);
20 }

```

Listing 1 – Fonction myRotate

La fonction *myRotate* effectue les rotations inverses à celles utilisées pour calculer la section efficace afin de revenir dans le référentiel du laboratoire. Dans la figure 1, on peut voir qu'il suffit de reproduire les 3 rotations comme suit : une rotation de  $-\gamma$  autour de  $Z$ , une rotation de  $-\beta$  autour de  $x''$  et finalement une rotation de  $-\alpha$  autour de  $z'$ .

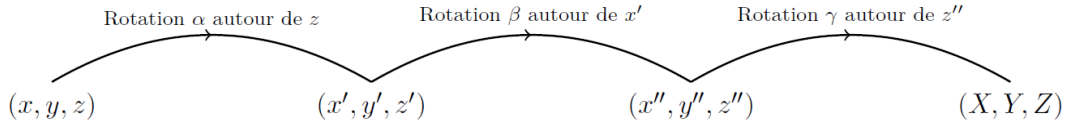


FIGURE 1 – Représentation des 3 rotations afin de passer du référentiel du laboratoire à celui du centre de masse du système [2].

## 2.2 Variables générées et section efficace

Les variables sont générées dans la section suivante :

```

1 // Générer les variables pour construire les 4-vecteurs de le CM
2 alpha = generateur.Rndm() * 2*pi; //alpha entre 0 et 2pi
3 cosBeta = (generateur.Rndm() * 2) -1; //beta entre 0 et pi
4 gamma = generateur.Rndm() * 2*pi; //gamma entre 0 et 2pi
5 //Changement de variable
6 y_min = -log(1-x_m); y_max = -log(x_m);
7 y1 = generateur.Rndm() * (y_max - y_min) + y_min;
8 y2 = generateur.Rndm() * (y_max - y_min) + y_min;
9 x1 = 1. - exp(-y1);
10 x2 = 1. - exp(-y2);
11 //x1 = generateur.Rndm() * (1 - 2* x_m) + x_m; // évite les divergences
12 //x2 = generateur.Rndm() * (1 - 2* x_m) + x_m; //évite les divergences
13 beta = acos(cosBeta);
14 fmax = sigmaMax * generateur.Rndm();
15 // Construisez toute la cinématique dans le referentiel alpha, beta, gamma
16 x3 = 2 - x1 - x2;

```

Listing 2 – Variables générées

On génère les fractions d'énergie du quark et de l'antiquark,  $x_1$  et  $x_2$ , ce qui donnera également la fraction d'énergie du gluon,  $x_3$ , par conservation d'énergie. Pour ce faire, on va générer uniformément des variables  $y_{1,2}$  puis faire un changement de variable ( $y_i = -\log(1 - x_i)$ ) afin de retrouver les  $x_{1,2}$ . Ce changement de variable sera décrit plus explicitement dans la section 3.3. Pour compenser la génération non-uniforme des fractions d'énergie, on multiplie simplement la section efficace par le déterminant de la Jacobienne du changement de variable. Cela va principalement permettre d'augmenter l'efficacité de notre méthode de réjection en passant de  $\sim 1 \times 10^{-4}\%$

à  $\sim 56\%$  d'efficacité.

On va également générer les angles d'Euler suivants :  $\alpha$ ,  $\cos \beta$  et  $\gamma$ .

*Remarque.* On génère  $\cos \beta$  plutôt que  $\beta$  car c'est bien  $\cos \beta$  qui apparaît dans la section efficace différentielle.

## 2.3 Méthode de réjection

On va ensuite utiliser une méthode de réjection afin de sélectionner nos événements. Dans notre code cela se traduit par :

```
1 fmax = sigmaMax * generateur.Rndm();
```

On commence par générer une valeur maximale  $fmax$  comme notre maximum de section efficace  $sigmaMax$ , pondérée par une variable aléatoire entre 0 et 1.

```
1 double Sigma = (1-x1)*(1-x2)*dsigma(x1,x2,beta,gamma, -1./3); //pour un quark
  Down
2 if (Sigma > sigmaMax) {
3   sigmaMax = Sigma;
4   cout << " !!!!!!! " << endl;
5 }
```

Afin d'obtenir ce maximum de section efficace, on compare chaque valeur de notre section efficace différentielle qui dépend des variables générées  $x_1$ ,  $x_2$ ,  $\beta$  et  $\gamma$  (et multipliée par un facteur  $(1-x_1)(1-x_2)$  dû au changement de variable) à ce maximum  $sigmaMax$  et on le remplace si l'on obtient des valeurs supérieures. Cela va permettre d'affiner notre valeur de  $sigmaMax$ .

```
1 if ( Sigma < fmax ) continue;
2 nAccepted++;
```

Listing 3 – Implémentation de  $fmax$  et Méthode de réjection

La méthode de réjection consiste alors à ce que l'on peut appeler du *Hit and Miss*. Pour toute valeur  $Sigma$  obtenue ci-dessus, si celle-ci est bien inférieure à  $fmax$ , l'événement est accepté et l'itération peut continuer. Dans le cas contraire, l'événement est rejeté et l'on perd en efficacité. Dans notre cas, en partant de  $5 \times 10^6$  événements, notre efficacité finale est de  $\sim 56\%$ .

## 2.4 Niveau généré et reconstruit

Nous allons classer les données issues des simulations dans deux types de « niveaux » :

### Définition 2.1.

- On désigne par « **niveau généré** » le niveau d'analyse avant l'interaction des particules avec le détecteur.
- On désigne par « **niveau reconstruit** » le niveau d'analyse après la simulation de l'interaction des particules avec le détecteur et le processus de reconstruction.

Au niveau généré, nous savons quelle impulsion correspond à quelle particule. Cependant, au niveau reconstruit, seules les signatures peuvent nous indiquer la nature d'une particule. Ici, nous nous contenterons de d'attribuer les impulsions aux particules de la manière suivante :

```
1 //Variable reconstruites
2 x1_Rec = max( max(x1, x2), max(x1, x3) );
3 x3_Rec = min( min(x1, x2), min(x1, x3) );
4 x2_Rec = 2 - x1_Rec - x3_Rec; // De telle manière que x1_Rec > x2_Rec > x3_Rec.
```

Listing 4 – Variables reconstruites

Cela permet de labéliser les fractions d'énergie du niveau généré de telle sorte que  $x_1 > x_2 > x_3$  au niveau reconstruit.

## 2.5 Angle de Ellis-Karliner $\theta_{EK}$

La définition de l'angle de Ellis-Karliner  $\theta_{EK}$  peut être trouvée ici [3] et est la suivante :

*Définition 2.2.* L'angle de Ellis-Karliner  $\theta_{EK}$  est défini selon

$$\cos \theta_{EK} = \frac{x_2 - x_3}{x_1} \quad (2.1)$$

« L'angle de Ellis-Karliner  $\theta_{EK}$  représente l'angle entre les jets 1 et 2 dans le référentiel au repos du système constitué des jets 2 et 3. Cette variable permet de mettre en évidence le caractère vectoriel dans l'émission de gluon. » [3]

Dans le code, on applique simplement sa définition au niveau reconstruit :

```
1 cosThetaEK = (x2_Rec - x3_Rec)/x1_Rec;
```

L'interprétation physique de cette angle sera faite en même temps que l'analyse des résultats de la simulation dans la section 3.5.

## 2.6 Algorithme JADE

La définition de l'algorithme JADE peut être trouvée ici [3]<sup>1</sup>. Rappelons la ici. Soit  $p_i$  la quadri-impulsion d'une particule  $i$ , et  $y_{cut} \in [0, 1]$  un paramètre. Si la masse invariante de la paire de particule  $i - j$  est telle que

$$M_{ij}^2 < y_{cut} \cdot s$$

alors ces deux particules sont considérées comme appartenant au même jet, et on définit la pseudo-particule  $p_{ij} \hat{=} p_i + p_j$ . On réitère alors la procédure avec toutes paires possibles.

Le but de cet algorithme va donc être de déterminer le nombre de jets par évènement. Prenons un exemple :

Supposons que l'on ait les 3 particules de l'état final comme labellisées par 1, 2 et 3. On démarre donc avec 3 jets. L'algorithme commence par associer les particules deux par deux (par exemple 1-2 puis 2-3 puis 1-3). S'il obtient pour une valeur choisie de  $y_{cut}$  que le carré de leur masse invariante donné par  $M_{ij}^2 = (p_i + p_j)^2$  est inférieur à  $y_{cut}s$ , il va combiner ces deux particules en une seule. Pour notre exemple, supposons que ce soit le cas pour la paire de particules 2-3. On n'a alors plus que 2 jets. L'algorithme va donc ensuite regarder le carré de la masse invariante du système composé de notre nouvelle particule 2-3 et de la particules 1. Si cette masse invariante est telle que son carré est de nouveau inférieur à  $y_{cut}s$ , la troisième particule va être combinée avec les deux autres et il ne restera plus qu'un seul jet.

Dans le code, l'implémentation de l'algorithme va se faire de la manière suivant :

```
1  ///Algorithme JADE
2  //Masse invariantes des différentes combinaisons quark, antiq, gluon
3  Mqqb = (quark + antiq).M2();
4  Mqg = (quark + gluon).M2();
5  Mqbg = (antiq + gluon).M2();
6  Mqqbg = (quark + antiq + gluon).M2();
7  //Energie et cut
8  s = 4*Ebeam*Ebeam;
9  Njet1 = 3.; Njet2 = 3.; Njet3 = 3.; Njet4 = 3.;
10 ycut1 = 0.00001; ycut2 = 0.001; ycut3 = 0.01; ycut4 = 0.5;
11 if (Mqg < s * ycut1 or Mqqb < s * ycut1 or Mqbg < s * ycut1){
12     Njet1 = 2.;
13     if (Mqqbg < s * ycut1 ) Njet1 = 1.;
14 }
15 if (Mqg < s * ycut2 or Mqqb < s * ycut2 or Mqbg < s * ycut2) Njet2 +=-1;
16 if (Mqqbg < s * ycut2 ) Njet2 +=-1;
17 if (Mqg < s * ycut3 or Mqqb < s * ycut3 or Mqbg < s * ycut3) Njet3 +=-1;
18 if (Mqqbg < s * ycut3 ) Njet3 +=-1;
19 if (Mqg < s * ycut4 or Mqqb < s * ycut4 or Mqbg < s * ycut4) Njet4 +=-1;
20 if (Mqqbg < s * ycut4 ) Njet4 +=-1;
```

Listing 5 – Définitions des masses invariantes et algorithme JADE

1. Point 3.9.1, page 64

### 3 Analyse des graphes

Cette partie est dédiée à l'analyse des graphes issus de la simulation. Lorsque ce que ce sera pertinent, ils seront comparés avec ceux vu au cours [3].

*Remarque.* L'analyse présentée dans cette section prend pour exemple la simulation *up* générant des quarks de charge  $+2/3$ . Lorsque des différences se manifestent avec la simulation *down* générant des quarks de charges  $-1/3$ , ou la simulation *mix*, nous l'indiquerons.

#### 3.1 Cosinus des angles $\theta_q$ et $\theta_{\bar{q}}$ dans le référentiel du centre de masse

On peut voir dans les graphes des figures 2 et 3 les distributions des angles de diffusion entre le quark (respectivement l'antiquark) et le gluon. Pour rappel, ceux-ci sont donnés par les équations (1.3) et (1.4) et dépendent donc des fractions d'énergie des particules de l'état final générées. On retrouve bien ici une distribution en cosinus pour  $\theta_q$  et  $\theta_{\bar{q}}$  :

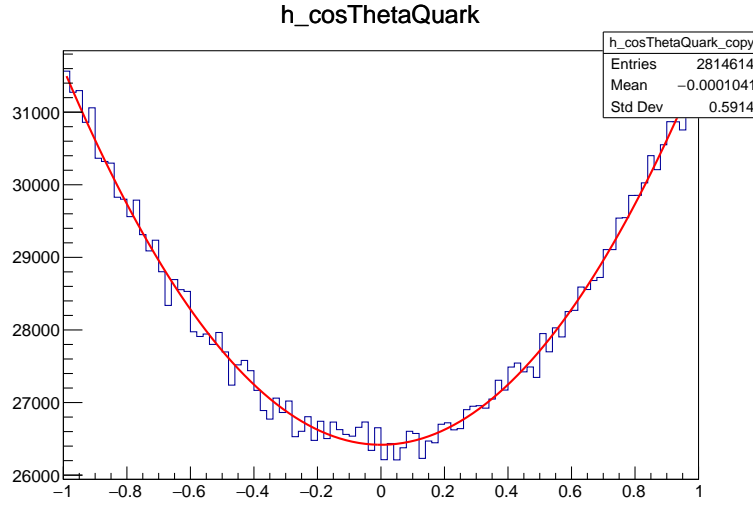


FIGURE 2 – Distribution de  $\cos\theta_q$  entre -1 et 1 défini par les fractions de masses des particules où  $\theta_q$  est l'angle entre le quark de charge  $2/3$  et le gluon dans le référentiel du centre de masse du système.

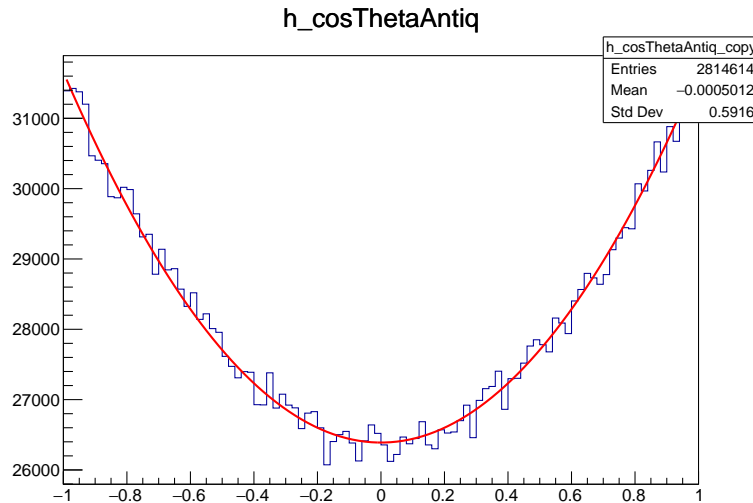


FIGURE 3 – Distribution de  $\cos\theta_{\bar{q}}$  entre -1 et 1 défini par les fractions de masses des particules où  $\theta_{\bar{q}}$  est l'angle entre l'antiquark de charge  $-2/3$  et le gluon dans le référentiel du centre de masse du système.



### 3.2 Les 3 angles d'Euler

À part les fractions d'énergie des particules de l'état final, les autres variables générées uniformément sont les angles d'Euler de rotations autour des axes du référentiel du laboratoire. En particulier, on génère ici entre 0 et  $2\pi$  les angles  $\alpha$  et  $\gamma$ , et pour  $\beta$  on génère son cosinus entre -1 et 1 (dont dépendra explicitement la section efficace différentielle de l'interaction). Les diagrammes des figures 4, 6 et 7 montrent respectivement les générations uniformes de  $\alpha$ ,  $\gamma$  et  $\cos\beta$  ce qui permet d'obtenir la distribution en  $\arccos x$  pour  $\beta$  dans le diagramme de la figure 5 :

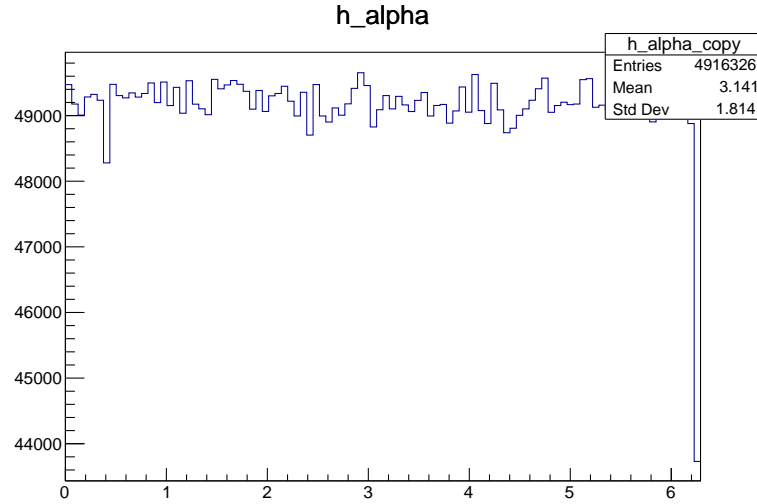


FIGURE 4 – Distribution de l'angle d'Euler  $\alpha$  généré uniformément entre 0 et  $2\pi$ .

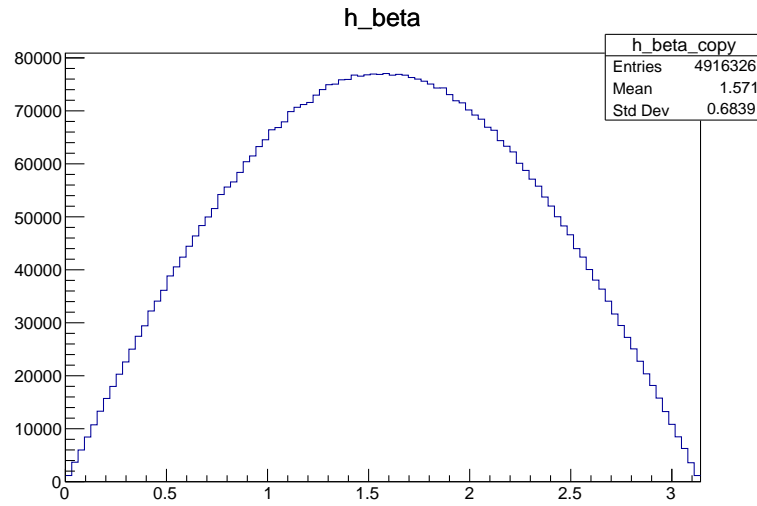


FIGURE 5 – Distribution de l'angle d'Euler  $\beta$  généré uniformément par  $\cos\beta$  entre -1 et 1.

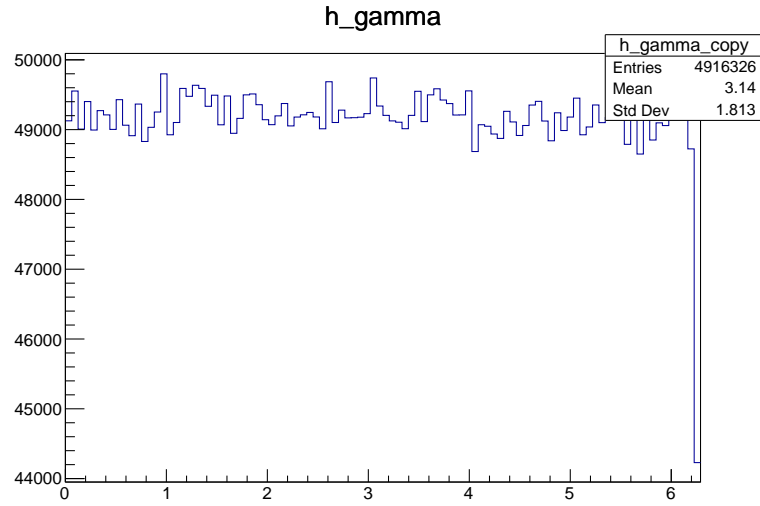


FIGURE 6 – Distribution de l'angle d'Euler  $\gamma$  généré uniformément entre 0 et  $2\pi$ .

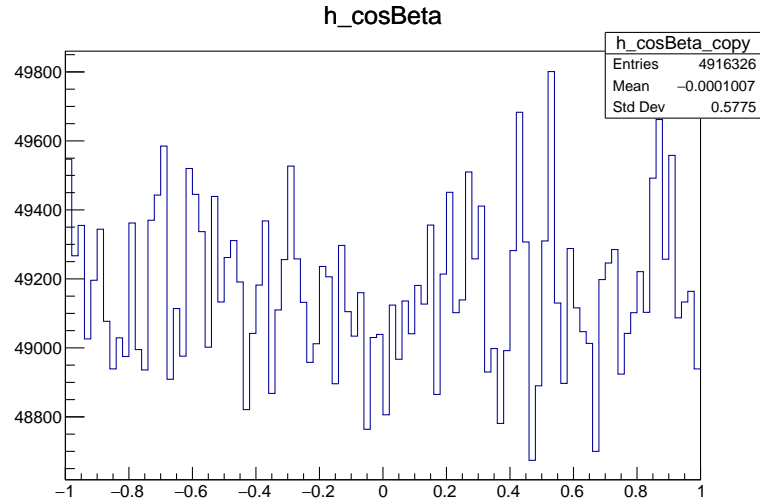


FIGURE 7 – Distribution de  $\cos \beta$  généré uniformément -1 et 1.

### 3.3 Fractions d'énergie $x_1$ et $x_2$ au niveau reconstruit

Pour rappel, au niveau reconstruit, les fractions d'énergies sont assignées aux particules de sorte que  $x_1^{\text{rec}} > x_2^{\text{rec}} > x_3^{\text{rec}}$ .

De plus, afin d'éviter les divergences évoquées précédemment, nous définissons une borne supérieure et inférieure pour  $x_1$  et  $x_2$  :

$$\epsilon_x = 5.10^{-5} \text{ et } x_{\min} \hat{=} \epsilon_x, x_{\max} \hat{=} 1 - \epsilon_x$$

Par ailleurs, les  $\{x_i\}$  remplissent la condition (1.5). On a alors les propriétés suivantes :

*Propriété 3.1.*

$$\begin{aligned} x_1^{\text{rec}} &\geq \frac{2}{3} \\ x_2^{\text{rec}} &\geq \frac{1}{2} \end{aligned}$$

**Preuve.**

- $x_1^{\text{rec}}$  est minimal lorsque  $x_1^{\text{rec}} = x_2^{\text{rec}} = x_3^{\text{rec}}$ . Or,  $x_1 + x_2 + x_3 = 2$  Ainsi,  $x_{1,\min}^{\text{rec}} = \frac{2}{3}$
- $x_2^{\text{rec}}$  est minimal lorsque  $x_1^{\text{rec}} = 1$  et  $x_2 = x_3$ . Ainsi,  $x_{2,\min}^{\text{rec}} = \frac{1}{2}$

□

Cependant, nous avons effectué un changement de variable qui affecte la distribution des  $\{x_i\}$ . Nous avons effectué

$$y_i = -\ln(x_i - 1) \iff x_i = 1 - e^{-y_i}$$

Les bornes minimales et maximales sont maintenant données par  $y_{i,\min} = -\ln(1 - x_{i,\min})$  et  $y_{i,\max} = -\ln(1 - x_{i,\max})$ .

On observe bien, dans la figure 8 (resp. la figure 9), une distribution piquée autour de 1, et nulle en deçà de  $2/3$  (resp. en deçà de  $1/2$ ).

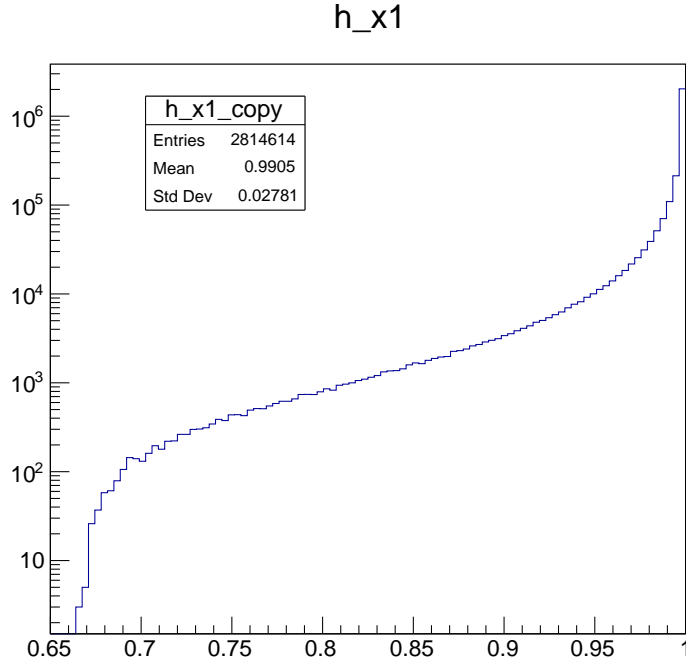


FIGURE 8 – Distribution de la fraction d'énergie  $x_1^{\text{rec}}$

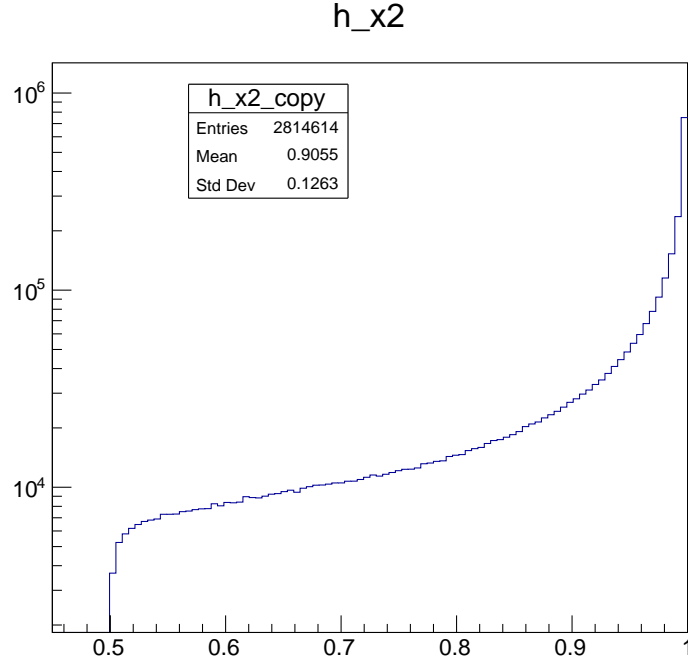


FIGURE 9 – Distribution de la fraction d'énergie  $x_2^{\text{rec}}$

### 3.4 Corrélations entre les $x_1$ , $x_2$ et $x_3$

Afin d'observer l'impact des conditions imposées par le passage au niveau reconstruit et par la conservation de l'énergie, on peut analyser les diagrammes des figures 10 et 11.

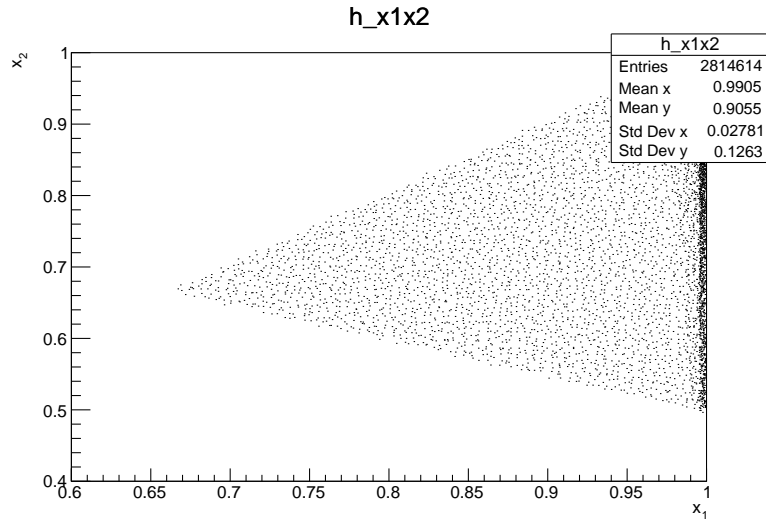


FIGURE 10 – Diagramme de corrélation entre  $x_1^{\text{rec}}$  et  $x_2^{\text{rec}}$

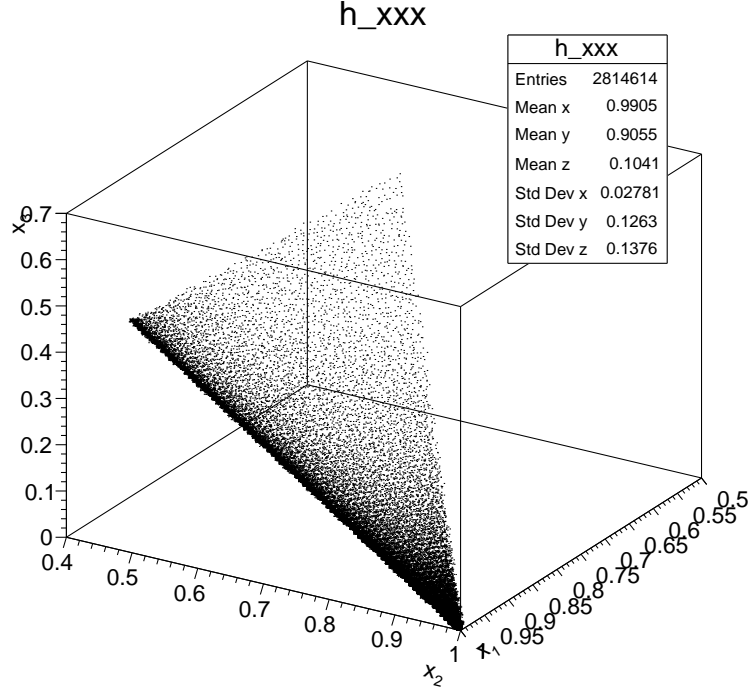


FIGURE 11 – Diagramme de corrélation entre  $x_1^{\text{rec}}$ ,  $x_2^{\text{rec}}$  et  $x_3^{\text{rec}}$

Pour résumer, nous avons 3 conditions sur les fractions d'énergie, données par :

$$x_1 + x_2 + x_3 = 2 \text{ (conservation de l'énergie)} \quad (3.1)$$

$$x_1 > x_2 > x_3 \quad (3.2)$$

$$x_{1,2,3} \in [x_{\min}, 1 - x_{\min}] \quad (3.3)$$

Graphiquement, cela nous mets des droites limites excluant des parties des diagrammes. De plus, la concentration importante d'évènements générés ayant  $x_1^{\text{rec}}$  proche de 1 et  $x_2^{\text{rec}}$  maximal reflète le changement de variable effectué.

### 3.5 $\theta_{EK}$ , l'angle d'Ellis-Karliner

L'angle de Ellis-Karliner est défini par l'équation (2.1) et dépend donc des fractions d'énergie. Au niveau reconstruit, cette angle représente l'angle entre les deux jets (de plus haute énergie s'il y a 3 jets) dans le référentiel de centre de masse du système. On observe dans le diagramme de la figure 12 que la distribution de son cosinus est fort piquée en  $\cos \theta_{EK} = 1$ .

Puis que  $x_2^{\text{rec}} < x_1^{\text{rec}}$ , la seule manière d'obtenir  $\frac{x_2^{\text{rec}} - x_3^{\text{rec}}}{x_1^{\text{rec}}} \rightarrow 1$  est le cas où  $x_1^{\text{rec}} = x_2^{\text{rec}} = 1$  et  $x_3^{\text{rec}} = 0$ . En comparant avec les résultats basés sur la théorie [3], on peut voir que la distribution obtenue ici confirme le caractère vectoriel des gluons.

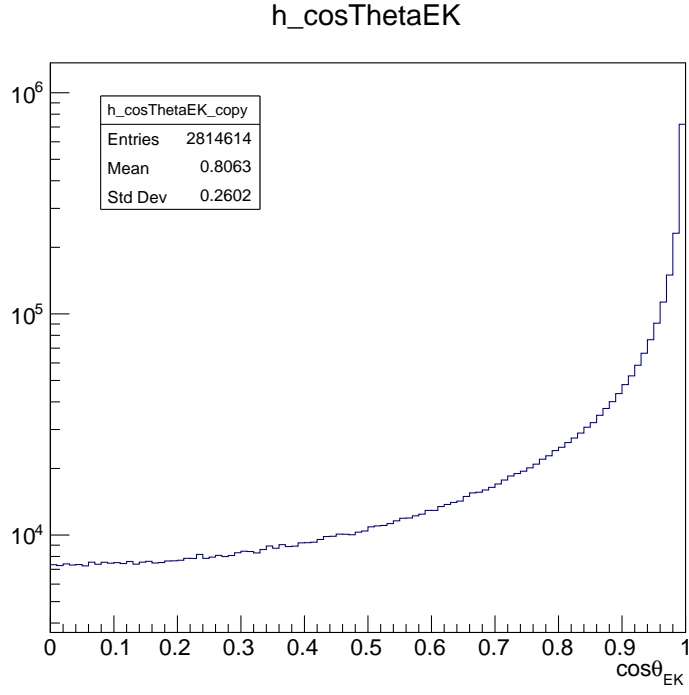


FIGURE 12 – Distribution de l’angle d’Ellis-Karliner,  $\theta_{EK}$ , entre les deux jets de plus haute énergie dans le centre de masse du système au niveau reconstruit.

### 3.6 Corrélation entre la section efficace $\sigma$ en la fraction d’énergie $x_1^{\text{rec}}$

Contrairement aux sections précédentes, la charge du quark influence la section efficace, et donc à fortiori sa corrélation avec  $x_1^{\text{rec}}$ . On retrouve les diagrammes de corrélation issus de la simulation dans les figures 13 et 14.

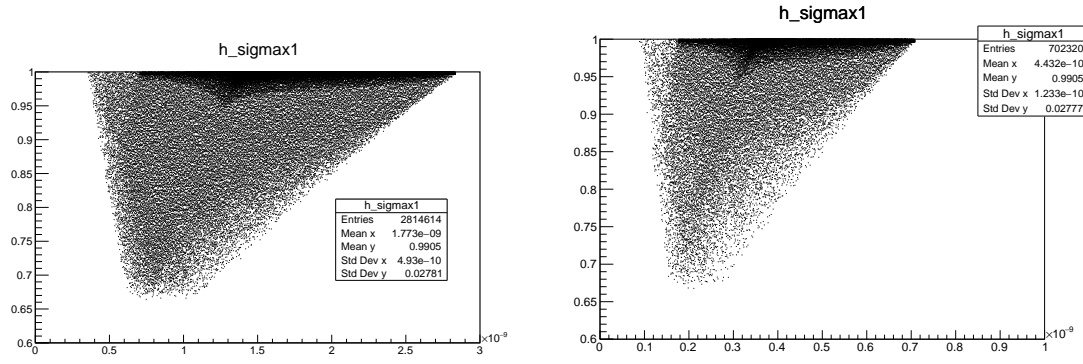


FIGURE 13 – Diagrammes de la corrélation entre section efficace totale  $\sigma$  et la fraction d'énergie  $x_1^{\text{rec}}$ , dans la simulation *up* à gauche et *down* à droite.

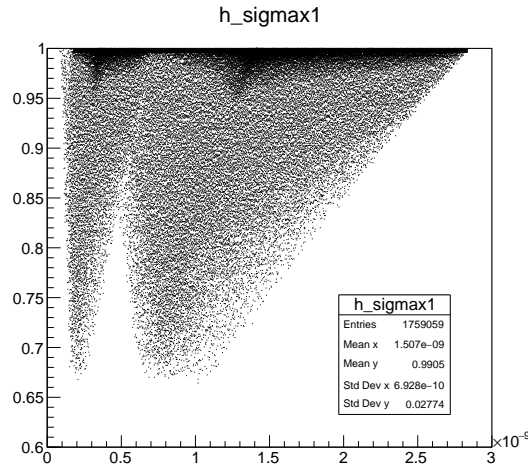


FIGURE 14 – Diagramme de la corrélation entre section efficace totale  $\sigma$  et la fraction d'énergie  $x_1^{\text{rec}}$  dans la simulation *mix*

Dans les 3 cas, on observe une densité d'évènement plus élevée pour de grande valeurs de  $x_1^{\text{rec}}$ , ce qui est cohérent avec le changement de variable visible sur la figure 8. On observe également une relation linéaire entre  $x_1^{\text{rec}}$  et  $\sigma$ . De plus, une zone d'exclusion apparaît pour de faibles valeurs de  $\sigma$ . Enfin, on observe un décalage du diagramme de corrélation vers la gauche pour une quark de type *down*, ce qui est en accord avec l'expression de la section efficace, celle-ci étant multipliée par  $Q_q^2$ .

### 3.7 Algorithme de JADE

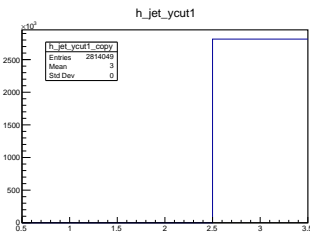


FIGURE 15 – Nombre de jets pour  $y_{\text{cut}} = 0.00001$

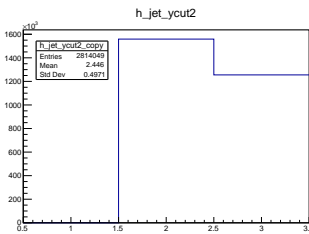


FIGURE 16 – Nombre de jets pour  $y_{\text{cut}} = 0.001$

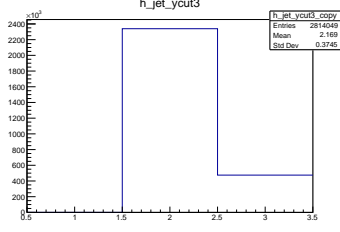


FIGURE 17 – Nombre de jets pour  $y_{\text{cut}} = 0.01$

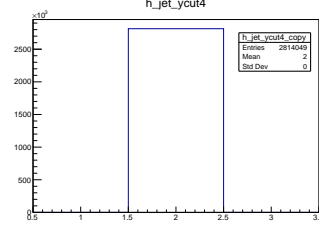


FIGURE 18 – Nombre de jets pour  $y_{\text{cut}} = 0.5$

On peut observer que le nombre d'événements avec un nombre de jets important diminue lorsque on augmente la valeur de  $y_{\text{cut}}$ .

Nous avons illustré plusieurs cas correspondant à plusieurs valeurs distinctes de  $y_{\text{cut}}$ . Par exemple, pour un  $y_{\text{cut}} \rightarrow 1$ , on constate que chaque particule est identifiée comme une jet à part entière. Au contraire, dans le cas de  $y_{\text{cut},4}$  (fig. 18), on observe une disparition totale des événements à 3 jets. Cela peut s'interpréter comme l'impossibilité pour le gluon (au niveau reconstruit) d'emporter plus de la moitié de l'énergie disponible dans centre de masse.

Enfin, notons qu'il existe des algorithmes plus complexes (par exemple l'algorithme de  $k_T$ ) permettant d'obtenir de meilleurs résultats.

## 4 Interaction $q\bar{q} \rightarrow e^+e^-$ , processus de Drell-Yan

Lorsque l'on considère le processus  $q\bar{q} \rightarrow e^+e^-$  plutôt que  $e^+e^- \rightarrow q\bar{q}$ , la principale différence est le rapport entre la masse des particules de l'état final et initial, qui est inversé. Pour que cela soit significatif, plaçons nous dans la limite des basses énergies. On a alors  $s = (p_1 + p_2)^2 \approx (m_q + m_{\bar{q}})^2$ . Considérons le cas le plus léger, soit l'annihilation d'une paire  $u - \bar{u}$ . Alors  $\sqrt{s} \approx 2.2$  MeV, là où la masse d'une paire électron - positron est  $m_{e^+e^-} \approx 1.2$  MeV.

De plus, contrairement au cas considéré ici, les particules finales étant des leptons, elles ne peuvent radier de gluon. Elles pourront néanmoins radier des photons, ce qui constituerait une correction QED et non QCD. S'ajoute à cela le besoin de retirer le facteur du nombre de couleur présent dans l'expression de la section efficace (1.7).

Dans l'hypothèse d'un collisionneur proton-proton, au moins un des deux quarks (l'antiquark) doit venir de la mer. Les quarks n'emportant qu'une fraction de l'impulsion du proton, on ne connaît pas exactement les impulsions des quarks qui interagissent. On doit alors considérer les fonctions des distributions des partons (PDF, Parton Distribution Function). Pour obtenir une masse invariante de l'ordre de  $\sqrt{20}$  GeV, il faut ainsi une énergie du faisceau plus importante. Négliger la contribution d'un échange de boson  $Z$  n'est alors plus pertinent.

## 5 Conclusion

Dans ce rapport, nous avons étudié la correction radiative QCD de l'interaction  $e^+e^- \rightarrow q\bar{q}$  avec émission d'un gluon dans l'état final.

Pour cela, nous avons dans un premier temps travaillé l'expression de la section efficace  $\sigma$  afin d'y faire apparaître les variables aléatoires et indépendantes.

Dans un deuxième temps, nous avons simulé des événements  $e^+e^- \rightarrow q\bar{q}g$  à l'aide de ces variables aléatoires grâce à une méthode de type *de Monte-Carlo*. L'efficacité étant particulièrement basse, nous avons appliqué un changement de variable bien choisi. L'efficacité finale est d'environ 56%. Nous nous sommes également intéressé au niveau reconstruit de la simulation.

Nous avons alors analysé les différents graphes obtenus et pu retrouver les corrélations et résultats attendus. En particulier, nous pouvons citer l'exemple de l'angle d'Ellis-Karliner  $\theta_{EK}$ , et la corrélation entre la section efficace et la fraction d'énergie du quark.

Enfin, nous avons tenté d'implémenter une version simple de l'algorithme *JADE* afin de reconstituer le nombre de jets par événement et de comparer ces résultats avec ceux présents dans la littérature.

Pour finir, nous avons commenté le processus inverse  $q\bar{q} \rightarrow e^+e^-$ , ses éventuelles radiations ainsi que sa réalisation dans un collisionneur  $p - p$ .



## 6 Annexe

Le code de la simulation combinée *mix* peut être trouvée [ici](#). Par souci de concision, seul le code pour la simulation *down* est affiché ci-dessous.

```
1 #include <TF1.h>
2 #include <TH1.h>
3 #include <TH2.h>
4 #include <TH3.h>
5 #include <TFile.h>
6 #include <TTree.h>
7 #include <TCanvas.h>
8 #include <TVector3.h>
9 #include <TLorentzVector.h>
10 #include <iostream>
11 #include <TRandom3.h>
12 #include <time.h>
13 #include <cmath>
14
15 // On definit des constantes valables
16 // dans tout le code.
17 #define pi 3.141592
18 #define Ebeam 10.
19 #define alphaEM 0.0072973525
20 #define alphaS 0.1184
21 #define Nc 3.
22 #define CF 4./3
23
24 using namespace std;
25
26 // Codez ici une formule permettant de faire le passage
27 // du referentiel alpha, beta, gamma au referentiel du
28 // laboratoire.
29 // Cette fonction doit prendre les 3 angles en argument
30 // plus une reference au quadrivecteur a modifier.
31 void myRotate(double alpha, double beta, double gamma, TLorentzVector &particle)
32 {
33     // Pour acceder au trivecteur du quadrivecteur
34     // on utilise la methode Vect()
35     TVector3 myVect = particle.Vect();
36     TVector3 X (1.,0.,0.);
37     TVector3 Z (0.,0.,1.);
38     // La methode Rotate(alpha, V) effectue
39     // une rotation d un angle alpha autour de
40     // l axe defini par le vecteur V
41     myVect.Rotate(-gamma, Z); X.Rotate(-gamma, Z);
42
43     myVect.Rotate(-beta, X); Z.Rotate(-beta, X);
44
45     myVect.Rotate(-alpha, Z);
46
47     particle.SetVect(myVect);
48 }
49 // Codez ici l expression de la section efficace
50 // differentielle en termes des variables x1, x2, beta, gamma
51 double dsigma(double x1, double x2, double beta, double gamma, double Qq)
52 {
53     double x3 = 2. - x1 - x2;
54     double cosTheta1 = -((x1*x1) + (x3*x3) - (x2*x2))/(2*x1*x3);
55     double cosTheta2 = -((x2*x2) + (x3*x3) - (x1*x1))/(2*x2*x3);
56     double theta1= acos( cosTheta1);
57     double theta2= acos(cosTheta2);
58     double factor = (CF*alphaEM*alphaEM*alphaS*Nc*(Qq)*(Qq))/(4*pi*pi); //Qq: charge
        du quark
59     double bracket = (x1*x1*(1+sin(beta)*sin(beta)*sin(gamma+theta1)*sin(gamma+theta1
        ))) + x2*x2*(1+sin(beta)*sin(beta)*sin(gamma-theta2)*sin(gamma-theta2));
60     double deno = (2*Ebeam)*(2*Ebeam)*(1-x1)*(1-x2);
61     double result = factor*bracket/deno;
62     return result;
63 }
64 // Codez ici la fonction principale qui genere les
65 // evenements et qui applique la methode de rejection.
66 void rejMethode()
67 {
68     // Determinez le sigmaMax en faisant une premiere
69     // boucle et en sauvant la valeur maximale de la
```

```

70 // section efficace
71 double sigmaMax = 7.2e-10;
72 // Declarez toutes les variables necessaires pour
73 // votre programme
74 int npoints = 5000000, nAccepted = 0;
75 double alpha, cosBeta, beta, gamma, x1, x2, x3, y1, y2, y_min, y_max, cosTheta1,
    cosTheta2, theta1, theta2, fmax, cosThetaEK;
76 //Variables pour Jade
77 double Mqqb, Mqg, Mqbg, Mqqbg, s, ycut1, ycut2, ycut3, ycut4, Njet1, Njet2, Njet3, Njet4
    ;
78 double x_m = 0.00005; //Xmin et 1-Xmin, évite les divergences
79 // Declarez les 5 quadri vecteurs ainsi que les
80 // variables a sauver
81 TLorentzVector ele, pos, quark, antiq, gluon;
82 ele.SetPxPyPzE((2*Ebeam/2), 0., 0., (2*Ebeam/2));
83 pos.SetPxPyPzE(-(2*Ebeam/2), 0., 0., (2*Ebeam/2));
84 double x1_Rec, x2_Rec, x3_Rec;
85 // Declarez une serie d histogrammes vous permettant
86 // de controler vos resultats
87 TH1F *h_cosThetaQuark = new TH1F("h_cosThetaQuark", "h_cosThetaQuark", 100, -1.,
    1.);
88 TH1F *h_cosThetaAntiq = new TH1F("h_cosThetaAntiq", "h_cosThetaAntiq", 100, -1.,
    1.);
89 TH1F *h_alpha = new TH1F("h_alpha", "h_alpha", 100, 0., 6.29);
90 TH1F *h_gamma = new TH1F("h_gamma", "h_gamma", 100, 0., 6.29);
91 TH1F *h_cosBeta = new TH1F("h_cosBeta", "h_cosBeta", 100, -1., 1.);
92 TH1F *h_beta = new TH1F("h_beta", "h_beta", 100, 0., 3.142);
93 TH3F *h_xxx = new TH3F("h_xxx", "h_xxx;x_{1};x_{2};x_{3}", 100, 0.5, 1, 100, 0.4,
    1, 100, 0, 0.7);
94 TH1F *h_x1 = new TH1F("h_x1", "h_x1", 100, 0.65, 1);
95 TH1F *h_x2 = new TH1F("h_x2", "h_x2", 100, 0.45, 1);
96 TH1F *h_x3 = new TH1F("h_x3", "h_x3", 100, 0, 1);
97 TH2F *h_x1x2 = new TH2F("h_x1x2", "h_x1x2;x_{1};x_{2}", 100, 0.6, 1, 100, 0.4, 1);
98 TH1F *h_cosThetaEK = new TH1F("h_cosThetaEK", "h_cosThetaEK;cos#theta_{EK}"
    , 100, 0, 1);
99 TH2F *h_sigmax1 = new TH2F("h_sigmax1", "h_sigmax1", 1000, 0, 3e-09, 100, 0.6, 1);
100 //Algo JADE
101 TH1F *h_jet_ycut1 = new TH1F("h_jet_ycut1", "h_jet_ycut1", 3, 0.5, 3.5);
102 TH1F *h_jet_ycut2 = new TH1F("h_jet_ycut2", "h_jet_ycut2", 3, 0.5, 3.5);
103 TH1F *h_jet_ycut3 = new TH1F("h_jet_ycut3", "h_jet_ycut3", 3, 0.5, 3.5);
104 TH1F *h_jet_ycut4 = new TH1F("h_jet_ycut4", "h_jet_ycut4", 3, 0.5, 3.5);
105 // Initialisez le generateur de nombres aleatoires
106 TRandom3 generateur;
107 generateur.SetSeed(123456);
108 // Ici commence la boucle sur les evenements a generer
109 for (int i=0; i<npoints; i++)
110 {
111     // Générer les variables pour construire les 4-vecteurs de le CM
112     alpha = generateur.Rndm() * 2*pi; //alpha entre 0 et 2pi
113     cosBeta = (generateur.Rndm() * 2) - 1; //beta entre 0 et pi
114     gamma = generateur.Rndm() * 2*pi; //gamma entre 0 et 2pi
115     //Changement de variable
116     y_min = -log(1.-x_m); y_max = -log(x_m);
117     y1 = generateur.Rndm() * (y_max - y_min) + y_min;
118     y2 = generateur.Rndm() * (y_max - y_min) + y_min;
119     x1 = 1. - exp(-y1);
120     x2 = 1. - exp(-y2);
121     //x1 = generateur.Rndm() * (1 - 2* x_m) + x_m; // évite les divergences
122     //x2 = generateur.Rndm() * (1 - 2* x_m) + x_m; //évite les divergences
123     beta = acos(cosBeta);
124     fmax = sigmaMax * generateur.Rndm();
125     // Construisez toute la cinematique dans le referentiel alpha, beta, gamma
126     x3 = 2 - x1 - x2;
127
128     if (x3>=1-x_m) continue;
129
130     cosTheta1 = -((x1*x1) + (x3*x3) - (x2*x2))/(2*x1*x3);
131     cosTheta2 = -((x2*x2) + (x3*x3) - (x1*x1))/(2*x2*x3);
132     theta1 = acos( cosTheta1);
133     theta2 = acos(cosTheta2);
134     //Vecteurs des particules sortantes
135     quark.SetPxPyPzE(Ebeam*x1*cos(theta1), Ebeam*x1*sin(theta1), 0., Ebeam*x1);
136     antiq.SetPxPyPzE(Ebeam*x2*cos(theta2), Ebeam*x2*cos(pi)*sin(theta2), 0., Ebeam*x2
    );
137     gluon.SetPxPyPzE(Ebeam*x3, 0., 0., Ebeam*x3);
138

```

```

139 // Passage au referentiel du labo
140 myRotate(alpha, beta, gamma, quark);
141 myRotate(alpha, beta, gamma, antiq);
142 myRotate(alpha, beta, gamma, gluon);
143
144 h_alpha->Fill(alpha); h_gamma->Fill(gamma); h_beta->Fill(beta); h_cosBeta->Fill(
cosBeta);
145
146 // Inserez la condition de la methode de rejection
147 double Sigma = (1-x1)*(1-x2)*dsigma(x1,x2,beta,gamma, -1./3); //pour un quark
Down
148 if (Sigma > sigmaMax) {
149     sigmaMax = Sigma;
150     cout << " !!!!! " << endl;
151 }
152 if ( Sigma < fmax ) continue;
153 nAccepted++;
154 //Variable reconstruites
155 x1_Rec = max( max(x1, x2), max(x1, x3) );
156 x3_Rec = min( min(x1, x2), min(x1, x3) );
157 x2_Rec = 2 - x1_Rec - x3_Rec; // De telle maniere que x1_Rec > x2_Rec > x3_Rec.
158 cosThetaEK = (x2_Rec - x3_Rec)/x1_Rec;
159 // Remplissez vos Histogrammes
160 h_cosThetaQuark->Fill(cos(quark.Theta())); h_cosThetaAntiq->Fill(cos(antiq.
Theta()));
161 h_xxx->Fill(x1_Rec, x2_Rec, x3_Rec);
162 h_x1->Fill(x1_Rec);
163 h_x2->Fill(x2_Rec); h_x3->Fill(x3_Rec);
164 h_x1x2->Fill(x1_Rec,x2_Rec);
165 h_cosThetaEK->Fill(cosThetaEK);
166 h_sigmax1->Fill(Sigma,x1_Rec);
167 ///Algorithme JADE
168 //Masse invariantes des differentes combinaisons quark, antiq, gluon
169 Mqqb = (quark + antiq).M2();
170 Mqg = (quark + gluon).M2();
171 Mqbg = (antiq + gluon).M2();
172 Mqqbg = (quark + antiq + gluon).M2();
173 //Energie et cut
174 s = 4*Ebeam*Ebeam;
175 Njet1 = 3.; Njet2 = 3.; Njet3 = 3.; Njet4 = 3.;
176 ycut1 = 0.00001; ycut2 = 0.001; ycut3 = 0.01; ycut4 = 0.5;
177 if (Mqg < s * ycut1 or Mqqb < s * ycut1 or Mqbg < s * ycut1){
178     Njet1 = 2.;
179     if (Mqqbg < s * ycut1 ) Njet1 = 1.;
180 }
181 if (Mqg < s * ycut2 or Mqqb < s * ycut2 or Mqbg < s * ycut2) Njet2 +=-1;
182 if (Mqqbg < s * ycut2 ) Njet2 +=-1;
183 if (Mqg < s * ycut3 or Mqqb < s * ycut3 or Mqbg < s * ycut3) Njet3 +=-1;
184 if (Mqqbg < s * ycut3 ) Njet3 +=-1;
185 if (Mqg < s * ycut4 or Mqqb < s * ycut4 or Mqbg < s * ycut4) Njet4 +=-1;
186 if (Mqqbg < s * ycut4 ) Njet4 +=-1;
187
188 h_jet_ycut1->Fill(Njet1);
189 h_jet_ycut2->Fill(Njet2);
190 h_jet_ycut3->Fill(Njet3);
191 h_jet_ycut4->Fill(Njet4);
192
193
194 }//fin de la boucle de generation d evenements
195
196 // Hors de la boucle, dessinez vos histogrammes dans
197 // des canevas. Faites des fit sur certaines distributions,
198 // sauvez les histogrammes.
199
200 TF1 *f = new TF1("f", "[0]*(1.+[1]*x*x)", -0.9, 0.9);
201 f->SetParameters(100, 1);
202
203 cout << "Sigma Max = " << sigmaMax << endl;
204 cout << "Eff = " << nAccepted*1./npoints << endl;
205 // Sauvez le canevas en format pdf
206 TCanvas *can_temp11 = new TCanvas(); h_cosThetaQuark->Fit("f"); h_cosThetaQuark->
DrawCopy(); can_temp11->SaveAs("/home/antoine/Interaction Forte/PDF/
h_cosThetaQuark.pdf"); //can_temp11->Close();
207 TCanvas *can_temp12 = new TCanvas(); h_cosThetaAntiq->Fit("f"); h_cosThetaAntiq->
DrawCopy(); can_temp12->SaveAs("/home/antoine/Interaction Forte/PDF/
h_cosThetaAntiq.pdf"); can_temp12->Close();

```

```

208 TCanvas *can_temp13 = new TCanvas(); h_alpha->DrawCopy(); can_temp13->SaveAs("/home
    /antoine/Interaction Forte/PDF/h_alpha.pdf"); can_temp13->Close();
209 TCanvas *can_temp14 = new TCanvas(); h_beta->DrawCopy(); can_temp14->SaveAs("/home/
    antoine/Interaction Forte/PDF/h_beta.pdf"); can_temp14->Close();
210 TCanvas *can_temp15 = new TCanvas(); h_gamma->DrawCopy(); can_temp15->SaveAs("/home
    /antoine/Interaction Forte/PDF/h_gamma.pdf"); can_temp15->Close();
211 TCanvas *can_temp16 = new TCanvas(); h_cosBeta->DrawCopy(); can_temp16->SaveAs("/
    home/antoine/Interaction Forte/PDF/h_cosBeta.pdf"); can_temp16->Close();
212
213 TCanvas *can_temp21 = new TCanvas(); h_xxx->SetFillColor(kRed+1); h_xxx->Draw();
    can_temp21->SaveAs("/home/antoine/Interaction Forte/PDF/h_xxx.pdf"); //
    can_temp21->Close();
214 TCanvas *can_temp22 = new TCanvas(); h_x1->DrawCopy(); can_temp22->SaveAs("/home/
    antoine/Interaction Forte/PDF/h_x1.pdf"); // can_temp22->Close();
215 TCanvas *can_temp23 = new TCanvas(); h_x2->DrawCopy(); can_temp23->SaveAs("/home/
    antoine/Interaction Forte/PDF/h_x2.pdf"); // can_temp23->Close();
216 TCanvas *can_temp24 = new TCanvas(); h_sigmax1->SetLineColor(kRed+1); h_sigmax1->
    Draw(); can_temp24->SaveAs("/home/antoine/Interaction Forte/PDF/h_sigmax1.pdf")
    ; // can_temp24->Close();
217 TCanvas *can_temp25 = new TCanvas(); h_x1x2->SetLineColor(kRed+1); h_x1x2->Draw();
    can_temp25->SaveAs("/home/antoine/Interaction Forte/PDF/h_x1x2.pdf");
    can_temp25->Close();
218 TCanvas *can_temp26 = new TCanvas(); h_cosThetaEK->DrawCopy(); can_temp26->SaveAs("/
    home/antoine/Interaction Forte/PDF/h_cosThetaEK.pdf"); // can_temp26->Close();
219
220 TCanvas *can_temp31 = new TCanvas(); h_jet_ycut1->DrawCopy(); can_temp31->SaveAs("/
    home/antoine/Interaction Forte/PDF/h_jet_ycut1.pdf"); // can_temp31->Close();
221 TCanvas *can_temp32 = new TCanvas(); h_jet_ycut2->DrawCopy(); can_temp32->SaveAs("/
    home/antoine/Interaction Forte/PDF/h_jet_ycut2.pdf"); // can_temp32->Close();
222 TCanvas *can_temp33 = new TCanvas(); h_jet_ycut3->DrawCopy(); can_temp33->SaveAs("/
    home/antoine/Interaction Forte/PDF/h_jet_ycut3.pdf"); // can_temp33->Close();
223 TCanvas *can_temp34 = new TCanvas(); h_jet_ycut4->DrawCopy(); can_temp34->SaveAs("/
    home/antoine/Interaction Forte/PDF/h_jet_ycut4.pdf"); // can_temp34->Close();
224
225
226 }

```

Listing 6 – code utilisé pour la simulation

## Références

- [1] Max Vanden BEMDEN. *Annexe au correctif de la séance 3 : Démonstration de  $|M|^2$  pour l'interaction  $e^+e^- \rightarrow q\bar{q}$* . URL : [https://homepage.iihe.ac.be/~vandenbe/Teaching/PHYS-F-477/newFormat/Seance3/Annexe\\_seance3.pdf](https://homepage.iihe.ac.be/~vandenbe/Teaching/PHYS-F-477/newFormat/Seance3/Annexe_seance3.pdf). (accessed : 02.05.2024).
- [2] Max Vanden BEMDEN. *Correctif de la séance 4 : Simulation de l'interaction  $e^+e^- \rightarrow q\bar{q}g$* . URL : [https://homepage.iihe.ac.be/~vandenbe/Teaching/PHYS-F-477/newFormat/Seance4/Solution\\_seance4.pdf](https://homepage.iihe.ac.be/~vandenbe/Teaching/PHYS-F-477/newFormat/Seance4/Solution_seance4.pdf). (accessed : 02.05.2024).
- [3] Laurent FAVART. *Physique des interactions fortes*. URL : <https://www.iihe.ac.be/~lfavart/cours/Notes-PHYS-F477-202223.pdf>. (accessed : 02.05.2024).