



Propriétés cinématiques des désintégrations muoniques des quarkonia

F. Medard

► To cite this version:

F. Medard. Propriétés cinématiques des désintégrations muoniques des quarkonia. 2004, pp.1-23.
in2p3-00023567

HAL Id: in2p3-00023567

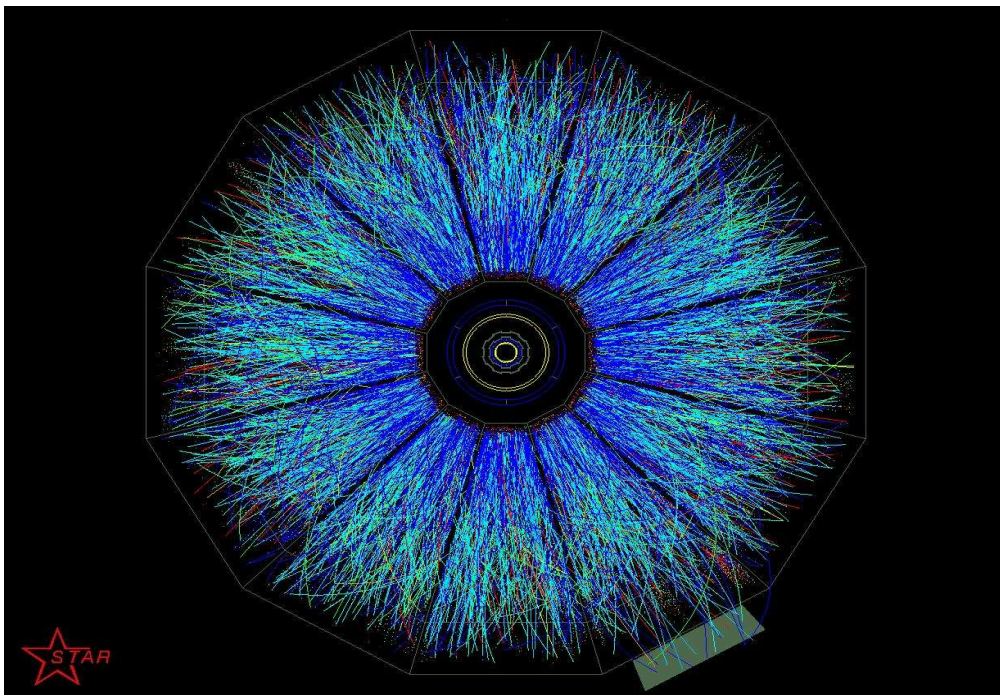
<https://hal.in2p3.fr/in2p3-00023567>

Submitted on 12 Jan 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Propriétés cinématiques
des désintégrations muoniques
des quarkonia.



Depuis plus de 20 ans, la physique des particules s'intéresse à un état de la matière obtenu pour des températures et des densités de particules très élevées. Lors du choc de noyaux lourds (or, plomb, uranium, ...) à des énergies très importantes, on obtient un déconfinement de la matière. Les particules composants les nucléons (quarks et gluons) se retrouvent libres et ne sont plus retenues par l'interaction forte. On parle de Plasma de Quarks et Gluons ou *QGP* pour Quark-Gluon Plasma. Lors du refroidissement de cet état, il y a formation de quarkonia, c'est-à-dire des particules composées d'un quark et de son anti-quark par exemple les ϕ , J/ψ , Υ . Ces quarkonia vont ensuite se désintégrer en une cascade de particules.

L'étude du *QGP* a d'abord été purement théorique. A partir des équations de la Chromodynamique Quantique (*QCD* en anglais, la théorie de l'interaction forte qui est présente à l'intérieur des nucléons), on a pu prédire un certain nombre de phénomènes qui se produiraient lorsque la matière possède une énergie importante et/ou une densité élevée. La mise en place des premier collisionneurs, accélérateurs permettant de réaliser des chocs frontaux entre particules, a permis de voir les confirmations de la théorie. A la fin des années 90, le *CERN* (Centre Européen pour la Recherche Nucléaire, Genève), a détecté les premières preuves directes de création du *QGP* lors de collisions entre noyaux lourds. L'arrêt du collisionneur au *CERN* a empêché d'apporter plus d'informations. Mais depuis 2000, d'autre expériences sont menées au *RHIC* (Relativistic Heavy Ion Collider, Brookhaven, USA) dans l'optique de fournir des éléments plus tangibles de la formation du *QGP*. A l'horizon 2007 se profile la mise en fonctionnement du *LHC* (Large Hadron Collider) au *CERN* qui permettra d'obtenir des énergies beaucoup plus importantes.

Parmi les signatures propres à la formation du *QGP* se trouve la diminution du nombre de quarkonia formés. Une des principales voie de désintégration de ces derniers est la désintégration muonique (par exemple $J/\psi \rightarrow \mu^+ \mu^-$). Il sera donc pratique de considérer la production des muons pour revenir aux quarkonia formés puis à l'état de *QGP* et donc avoir une preuve de son existence. Le but de mon stage est donc d'étudier la formation des quarkonia (notamment des J/ψ) puis leur désintégration en paires de muons, dans un choc entre nucléons (ici des protons). Pour cela, après avoir réaliser les calculs théoriques nécessaires, j'ai alors utilisé une simulation informatique avec le logiciel *ROOT* afin de connaître la forme des variables cinématiques des particules produites lors du choc dans des conditions d'énergies appropriées.

Je vais tout d'abord présenter les aspects importants du *QGP*, les grands accélérateurs utilisés pour sa détection et plus particulièrement la signature apportée par la formation des quarkonia. Dans un deuxième temps, j'aborderai les calculs de relativité restreinte que j'ai réalisé pour décrire physiquement les différentes étapes amenant à la création d'une paire de muons ainsi que les outils mathématiques permettant d'établir des simulations. Enfin je terminerai par une partie informatique qui montrera les fonctions utilisées dans le programme créant la simulation.

Introduction	page 1
Table des matières	page 2
1.Le Plasma de Quarks et Gluons	page 3
1.1.Formation du QGP	
1.1.1.Nucléons, quarks et gluons	
1.1.2.Equation d'état de la matière	
1.1.3.Les signatures du QGP	
1.2.Détection du QGP	
1.2.1.Les collisionneurs	
1.2.2.Les détecteurs	
2.Physique relativiste de la collision proton-proton	page 7
2.1.Changements de référentiels, quadrivecteurs des particules	
2.2.Choc entres protons, formation des quarkonia	
2.2.1.Impulsion des gluons	
2.2.2.Techniques de Monte-Carlo	
2.3.Conditions sur la formation des quarkonia	
3.Simulation informatique, résultats :	page 9
3.1.Interface graphique, utilisation de ROOT	
3.2.Simulation physique, algorithmes	
Conclusion	page 11
Annexes	page 12 ²

1. Le Plasma de Quark et Gluons :

1.1. Formation du QGP :

1.1.1. Nucléons, quarks et gluons :

Les noyaux des atomes sont formés de nucléons (protons et neutrons) qui ne sont pas des particules élémentaires mais des compositions de quarks et de gluons. Il existe 6 sortes de quarks classés par paires et dont la masse va croissante. On trouve en premier (et c'est eux qui composent la matière courante comme illustré sur la fig.1) les quarks Up et Down, puis notamment les quarks Strange, Charmed ... que l'on voit dans les quarkonia. Les quarks sont maintenus ensemble par la force forte dont les médiateurs sont les gluons. L'interaction forte se décline en 3 saveurs ou couleurs et chacune des 8 sortes de gluons "relie" entre elles des saveurs bien particulières.

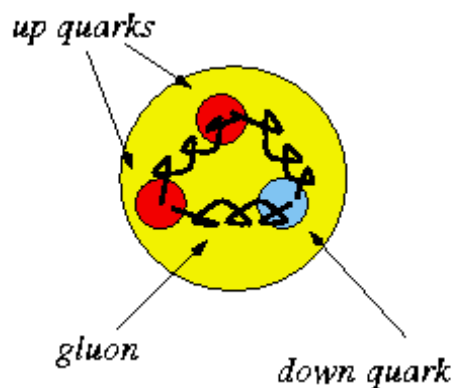


Fig.1 : représentation schématique d'un proton montrant les quarks et les gluons.

1.1.2. Equation d'état de la matière :

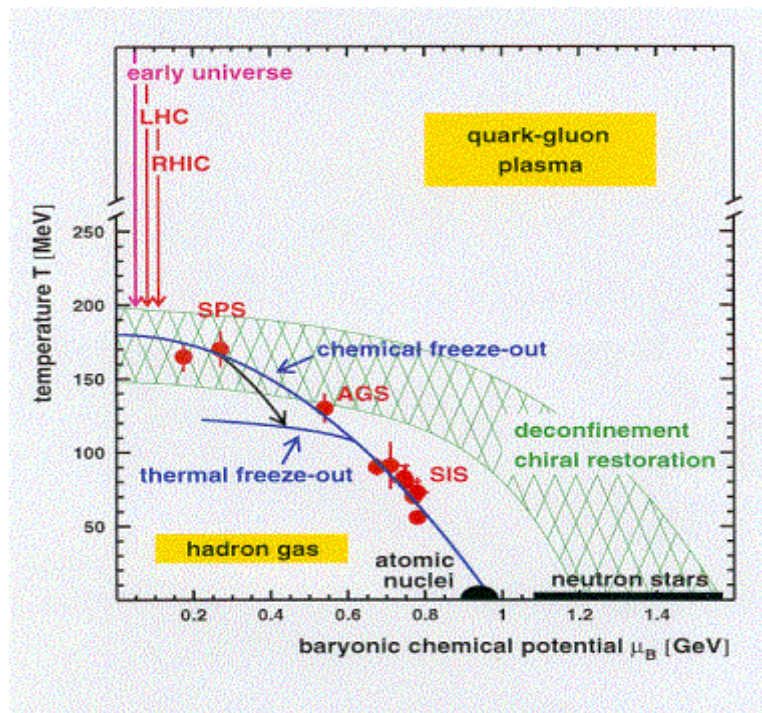


Fig.2 : Diagramme de phase de la matière nucléaire.³

Cette configuration de la matière nucléaire est caractéristique d'une température et d'une densité de matière bien précises. Cela correspond aux énergies présentes naturellement et à une densité "normale". Lorsque l'on fait subir à la matière un changement de densité et/ou d'énergie, il y a changement de phase. Notamment, lors d'une augmentation à la fois de l'énergie et de la densité nucléaire (conditions obtenues lors des chocs frontaux de noyaux lourds à hautes énergies), la matière se trouve sous la forme d'un Plasma de Quarks et de Gluons (*QGP*). Les quarks et les gluons ne sont plus confinés à l'intérieur des nucléons par l'action de l'interaction forte mais ils sont "libres" de la même façon qu'un plasma électrique, état d'un arc électrique (étincelle, foudre, ...) qui se caractérise par une dissociation des atomes en électrons et ions positifs. La matière se trouve sous la forme d'une "soupe" où quarks et gluons interagissent sans être contraints dans un espace de la taille du nucléon. Cet état est d'après la théorie du Big Bang, celui qui prévalait environ 10^{-33} secondes après la formation de notre univers.

1.1.3. Les signatures du QGP :

Lors des collisions pouvant amener à la formation du plasma, on cherche la présence de signatures significatives de la présence du QGP. On a envisagé de nombreux signes pouvant être étudiés mais certains sont plus révélateurs. Les expériences ont constaté des phénomènes tels que :

- la suppression des "jets" de particules (les particules émises avec une grande énergie forment des jets de particules très observables lors des collisions protons-protons)
- l'excès de particules étranges (les particules contenant des quarks Strange nécessitent une énergie très importante pour se former)
- la suppression des particules J/ψ (elles se forment à hautes énergies mais leur nombre diminue lors de la création de QGP)

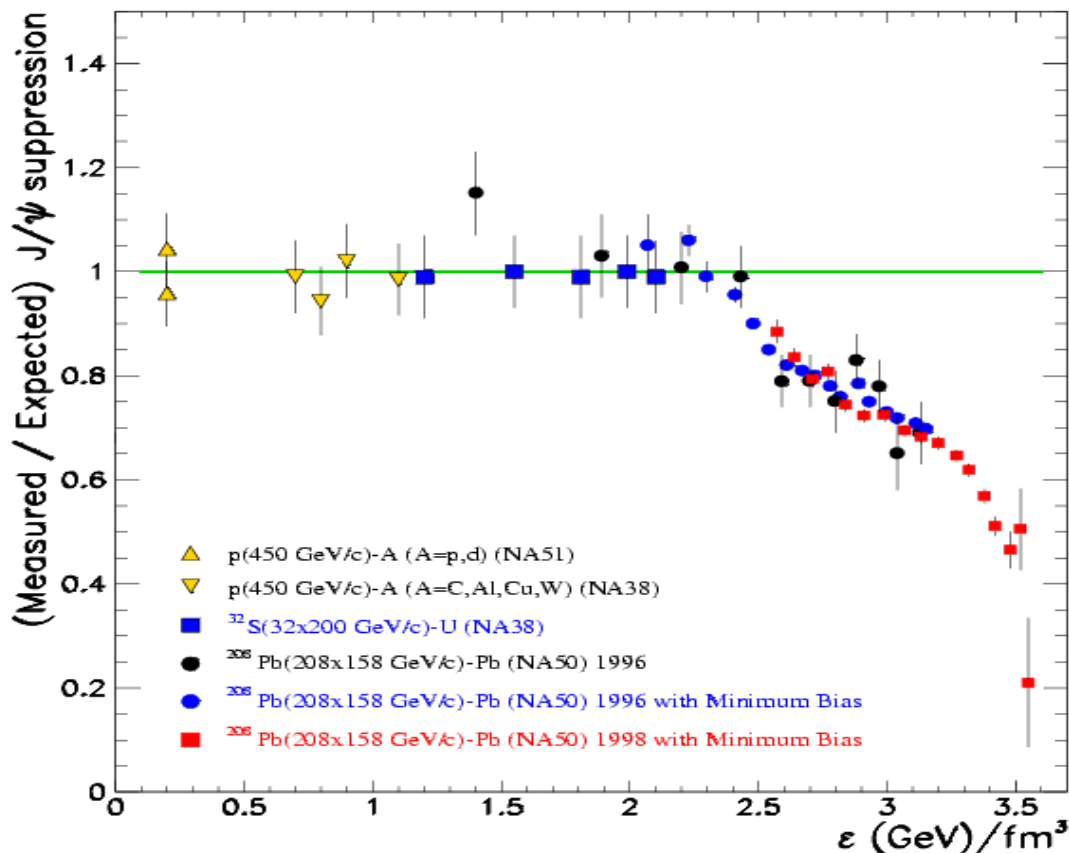


Fig.3 : suppression du J/ψ ? à hautes densités d'énergie.⁴

C'est sur cette dernière signature que va se concentrer cette étude. En effet, la voie de désintégration muonique de la particule J/ψ a un rapport d'embranchement de 5,88% ce qui permet une étude aisée en détectant les muons produits. La particule J/ψ est composée d'un quark Charmed et de son antiquark, sa masse est grande, de l'ordre de $3 \text{ GeV}/c^2$. Lors de la collision de protons à hautes énergies (telles que celles atteintes dans les grands collisionneurs), on observe l'apparition du méson J/ψ . On peut alors établir une corrélation entre le nombre de particules produites et la densité d'énergie qui varie selon le type de noyaux utilisés dans la collision et leur paramètre d'impact (défini comme la distance entre le centre des noyaux lors de la collision). On observe alors à partir d'une certaine énergie que la production ne suit plus l'extrapolation réalisée. Ce phénomène marque la présence du QGP selon certaines théories ; en effet le plasma provoque un effet d'écrantage entre quarks et antiquarks Charmed qui vont donc beaucoup moins interagir, d'où la suppression de la résonance.

Mon étude va donc porter sur une simulation de la création des J/ψ lors des collisions entre protons, permettant d'établir une référence du nombre de créations qui pourra ensuite être comparée aux chocs entre protons.

1.2.Détection du QGP :

1.2.1.Les collisionneurs :

Les premières expériences apportant des preuves de la formation du QGP ont été réalisées à la fin des années 90 au CERN. Le collisionneur SPS (Super Proton Synchrotron) a permis d'atteindre une énergie suffisante et les premières signatures ont pu être avancées en février 2000. Un nouveau stade a été atteint à partir de 2000 avec la mise en place au Brookhaven National Laboratory (BNL) d'un accélérateur permettant d'atteindre des conditions plus extrêmes pour la formation du QGP.



Fig.4 : le tunnel du RHIC à Brookhaven.⁵

Le RHIC, acronyme pour Relativistic Heavy Ion Collider (collisionneur d'ions lourds relativistes) est principalement dédié à l'étude du QGP au travers de quatre expériences correspondant aux quatre détecteurs situés le long de l'anneau. Il réalise des collisions entre noyaux d'or avec une énergie de 200 GeV dans le centre de masse. Les deux principales collaborations travaillant au RHIC (les expériences STAR et PHENIX) ont apporté des preuves plus concluantes qu'au SPS. On a notamment observé une diminution de l'émission de particules à haute impulsion transverse interprété comme une perte d'énergie des particules dans le plasma.

De plus amples informations et une nouvelle confirmation de la formation du plasma seront sûrement apportées à l'horizon 2007 avec la mise en fonctionnement du LHC (Large Hadron Collider) au CERN à Genève. Ce collisionneur pourra atteindre des énergies de 5500 GeV dans le centre de masse pour des chocs entre noyaux de plomb. L'expérience ALICE (A Large Ion Collider Experiment) consacrée à l'étude du QGP est menée par une collaboration d'environ 1000 ingénieurs et physiciens à laquelle participe une équipe clermontoise du LPC.

1.2.2. Les détecteurs :

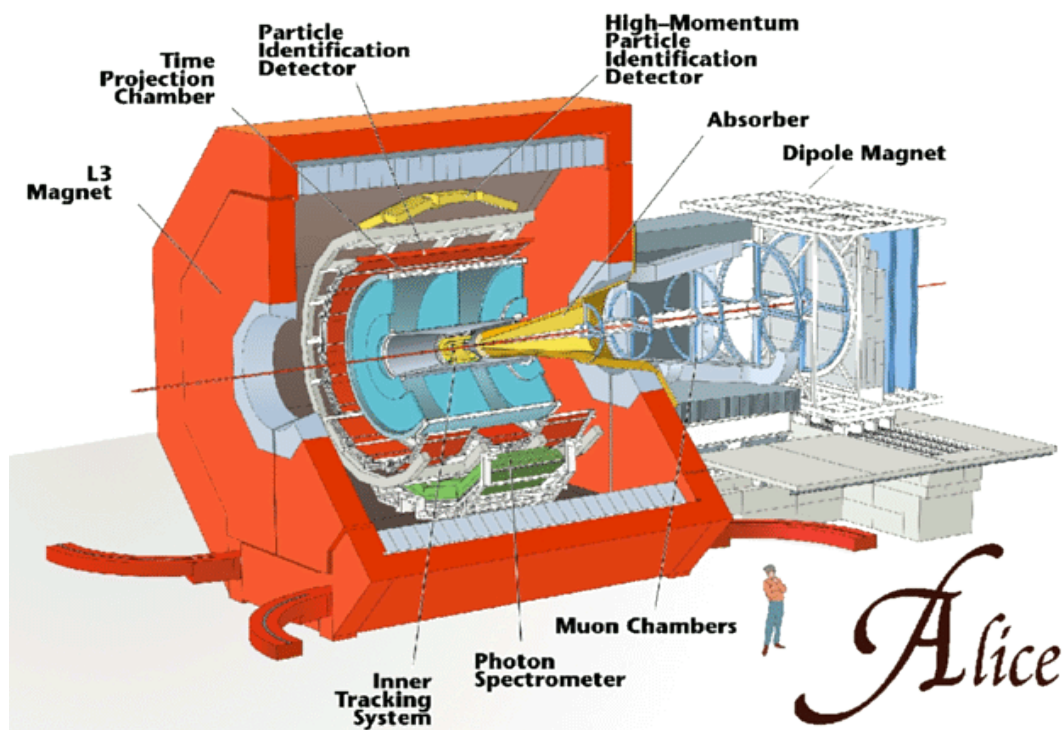


Fig. 5 : le détecteur ALICE dans le futur collisionneur LHC.⁶

Les détecteurs utilisés dans l'étude des collisions entre ions lourds sont principalement de grands ensembles contenant de nombreux sous-détecteurs. L'exemple du détecteur ALICE est représentatif de cette classe de par ses dimensions impressionnantes mais aussi son assemblage complexe de détecteurs.

Il apparaît donc intéressant de décrire plus précisément les sous-détecteurs d'ALICE pour comprendre le fonctionnement de la structure :

- autour du point de collision se trouve un ensemble de trois détecteurs cylindriques permettant de suivre avec grande précision et rapidité les particules chargées ; on a notamment l'ITS (Inner Tracking System) constitué de plaques de silicium qui va détecter le vertex et la TPC (Time Projection Chamber), cylindre rempli de gaz parcouru par un champ électrique qui décrit la trajectoire des particules.

- ensuite est placé un détecteur de photons (PHOS , PHOton Spectrometer) constitué de matériaux extrêmement dense (tungstate de plomb), il va stopper la plus grande partie des photons émis lors de la collision ; un ensemble de photomultiplicateurs permet de recueillir la lumière produite et d'obtenir l'énergie des photons (et donc la température de collision).
- le détecteur possède aussi un spectromètre à muons situé sur l'avant du site de collision qui permet de détecter les muons, signes probables de la désintégration de particules comme le J/ψ .

L'ensemble des impulsions électriques envoyées par tous les détecteurs doit être transformé en données puis ensuite stockées (chaque année il sera produit l'équivalent d'une pile de CD de 4km de haut). Ces informations vont pouvoir être traitées par un ensemble de milliers d'ordinateurs reliés à travers le monde grâce à internet. Viendra alors une phase où les données traitées pourront être utilisées par la collaboration scientifique qui travaille sur l'expérience ALICE.

2.Physique relativiste de la collision proton-proton :

2.1.Changement de référentiels, quadrivecteurs des particules :

L'énergie apportée aux protons dans les collisionneurs tels que RHIC et LHC leurs confèrent une vitesse très proche de celle de la lumière. Il est donc nécessaire lorsque l'on aborde une partie calculatoire, d'utiliser les formules de la relativité restreinte. Les données qui devront être connues grâce à la simulation sont exprimées dans un référentiel associé au détecteur nommé référentiel du laboratoire. Cependant pour étudier la désintégration des quarkonia en paires de muons (notamment la désintégration $J/\psi \rightarrow \mu^+ \mu^-$), il est nécessaire de posséder les paramètres du quarkonia dans le référentiel constitué par son centre de masse (référentiel CM). Pour passer d'un référentiel à l'autre, il faut utiliser la transformation de Lorentz dont la formulation est rappelée ci-dessous :

$$p = \begin{bmatrix} \gamma & 0 & 0 & \gamma\beta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \gamma\beta & 0 & 0 & \gamma \end{bmatrix} * p' \quad (matrice de Lorentz)$$

On obtient après calculs les paramètres des muons exprimés dans le référentiel du laboratoire en fonction des paramètres des muons dans le CMS, des facteurs de Lorentz γ et β , des angles θ^* et ϕ^* entre le CMS et le référentiel du laboratoire qui sont tirés de façon aléatoire et des paramètres p et θ de la collision.

$$\begin{aligned} E_{\mu} &= \gamma (E_{\mu}^* \pm \beta p_{\mu}^* \cos \theta^*) \\ p_{x\mu} &= \pm p_{\mu}^* \sin \theta^* \cos \phi^* \\ p_{y\mu} &= \pm p_{\mu}^* \sin \theta^* \sin \phi^* \cos \theta + \gamma (\beta E_{\mu}^* \pm p_{\mu}^* \cos \theta^*) \sin \theta \\ p_{z\mu} &= \pm p_{\mu}^* \sin \theta^* \sin \phi^* \sin \theta + \gamma (\beta E_{\mu}^* \pm p_{\mu}^* \cos \theta^*) \cos \theta \end{aligned}$$

Une fois le quadrivecteur des muons obtenu, il va être possible de reconstituer la collision.

2.2.Choc entres protons, formation des quarkonia :

2.2.1.Impulsion des gluons :

La formation des quarkonia (par exemple les J/ψ) ne peut pas être directement

considérée lors du choc de protons mais plutôt comme collision de deux gluons. Le paramètre fondamental dans le choc des gluons est leur impulsion qui va pouvoir être décomposée en deux parties :

- une impulsion longitudinale $p_{//}$ fortement prépondérante et dont la valeur va pouvoir être obtenue par un tirage aléatoire d'une distribution appelée fonction de structure gluonique des protons :

$$x g(x) = 35,8 x^{2,3} (1-x)^{4,0}$$

- une impulsion transverse k_T faible mais néanmoins présente puisque les deux gluons ne se déplacent pas selon le même axe qui sera tirée aléatoirement selon une distribution gaussienne.

Une fois ces paramètres connus, il est possible d'obtenir deux des éléments nécessaires à la connaissance du quadrivecteur énergie/impulsion des muons : l'impulsion p et l'angle θ d'émission du quarkonia. Pour cette détermination on a les formules suivantes :

$$p = \sqrt{(k_{T1} - k_{T2})^2 + (x_1 - x_2)^2 \cdot p_{//}^2}$$

$$\tan \theta = \frac{(k_{T1} - k_{T2})}{((x_1 - x_2) \cdot p_{//})} \quad 8$$

2.2.2. Techniques de Monte-Carlo :

Pour obtenir une simulation, il est nécessaire de tirer aléatoirement les impulsions des gluons en respectant cette fonction de structure. On utilise donc les techniques de Monte-Carlo.

Comme la distribution est d'une forme complexe, il va être plus simple de se servir de la méthode mise au point par Von Neumann. On considère que pour tout x on connaît la fonction de densité de probabilité correspondante $f(x)$ et qu'elle peut être traitée informatiquement. Puis qu'il est possible de l'inclure dans un contour qui soit une fonction $h(x)$ de forme simple. Un générateur de nombre aléatoires tire alors un x correspondant qui se trouve à l'intérieur de la fonction $h(x)$. On cherche alors $f(x)$ pour savoir si le point est compris dans l'intégrale formée par la courbe de $f(x)$, si oui le tirage est validée. On obtient donc un tirage aléatoire respectant la fonction de distribution.

2.3. Conditions sur la formation des quarkonia :

La formation d'un quarkonia dans la collision de deux gluons est seulement possible si l'énergie cinétique apportée par les gluons est supérieure à l'énergie de masse du quarkonia lorsqu'on se place dans le référentiel du centre de masse du quarkonia. Plus la masse des quarkonia sera importante, plus cette condition va enlever un grand nombre d'événements (par exemple pour le J/ψ on a une masse de 3096,87 MeV donc il sera beaucoup moins contraint que le Υ à 9460,30 MeV). De même plus les gluons auront une énergie élevée, moins la condition aura d'importance.

On réalise une autre série de calculs relativistes en passant par des changements de référentiels qui nous permettent d'obtenir une formulation de cette condition :

(soit E_g l'énergie de la paire de gluons, nécessairement supérieure à la masse du quarkonia).

on a : $E_g = \sqrt{2 \cdot E_{g1} \cdot E_{g2} + 2 \cdot k_{T1} \cdot k_{T2} + 2 \cdot x_1 \cdot x_2 \cdot p_{//}^2}$

avec $E_{g1} = \sqrt{k_{T1}^2 + x_1^2 \cdot p_{//}^2}$ et $E_{g2} = \sqrt{k_{T2}^2 + x_2^2 \cdot p_{//}^2}$

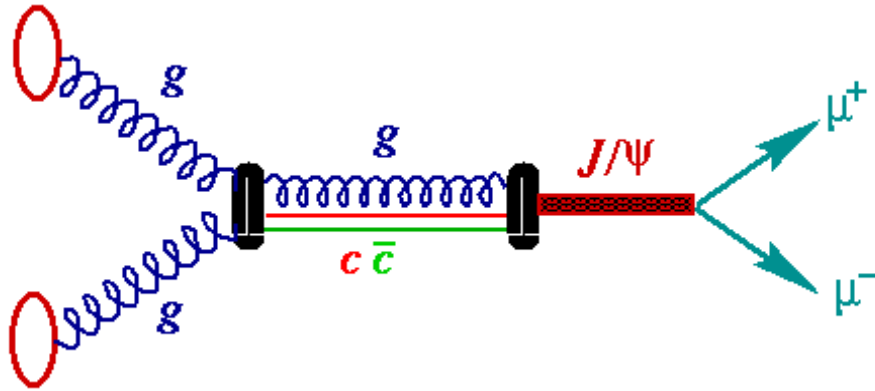


Fig. 6 : formation du J/ψ puis sa désintégration muonique.⁹

3.Simulation informatique, résultats :

Le but de ces calculs relativistes est de me permettre d'obtenir les équations nécessaires à une modélisation sur ordinateur de la collision et des quarkonia émis. Le langage de programmation choisi est le C/C++ qui permet une certaine souplesse d'utilisation mais aussi un nombre étendu de possibilités. Basiquement, il est suffisant de coder le programme puis de le compiler sans chercher des outils plus compliqués. Cependant, il m'a paru plus intéressant de me servir d'un logiciel développé par le CERN qui permet la création d'une interface graphique de très bonne qualité. Il s'agit de ROOT qui se trouve actuellement dans sa version 4.00 et possède une multitude de fonctionnalités.

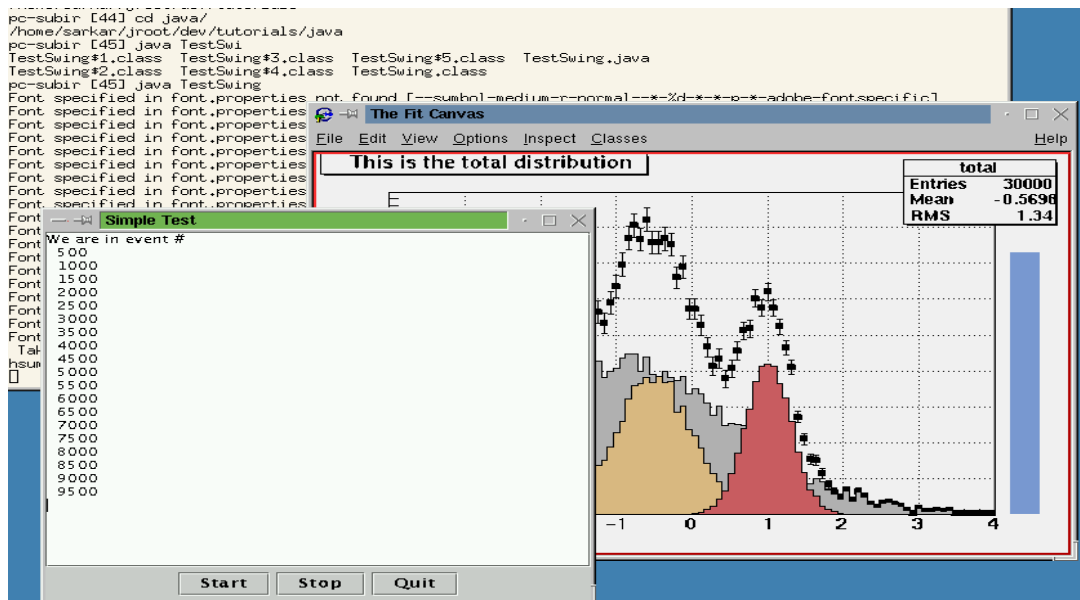
3.1.Interface graphique, utilisation de ROOT :

Dans l'utilisation que j'ai eu de ROOT, j'ai surtout tiré profit des nombreuses fonctions graphiques permettant de créer une interface utilisateur plus attractive. Il me semble donc important de présenter les divers fonctions graphiques que j'ai utilisé. Toutes ces fonctions sont réunies dans une classe nommée GUI (Graphical User Interface) dont je vais détailler quelques utilisations.

Fig. 7 : interface graphique simple créée avec ROOT.

- **TGMainFrame** : création d'une fenêtre principale avec comme paramètres principaux l'endroit de création, la hauteur et la largeur.
- **TGHorizontalFrame** et **TGVerticalFrame** : permet de diviser la fenêtre principale en sous espaces réalisé soit dans le sens horizontal, soit dans le sens vertical.
- **TGGroupFrame** : pour regrouper des fenêtres en un seul groupe mais aussi éventuellement pour donner un nom au groupe ou une forme (couleur, bordure...).
- **TGLayoutHints** : mise en forme d'une fenêtre à passer en paramètre lors de la création ; permet de l'aligner et de préciser sa position par des coordonnées relatives.

- **TGNumberEntryField** et **TGTextEntryField** : création d'un champ dans lequel l'utilisateur pourra entrer des nombres ou des caractères puis les retourner au programme.
- **TGListBox** et **TGComboBox** : ouverture de boites contenant des listes d'options permettant le choix de l'utilisateur (défilantes ou non).
- **TGTextButton** : pour définir des boutons renvoyant une information lorsque l'utilisateur clique dessus (on peut aussi utiliser **TGPictureButton** pour avoir un bouton à image).



10

Ces quelques commandes très utiles dans la construction de l'interface graphique doivent être utilisées de façon très précises et avec certains paramètres tel que le montre le code source placé en annexe.

3.2.Simulation physique, algorithmes :

La deuxième partie du programme est constitué de l'algorithme permettant la construction d'une simulation du choc entre deux protons ammenant la création de quarkonia puis leur désintégration muonique. L'interface permet de demander à l'utilisateur un nombre important de paramètres à savoir : les particules entrant dans la simulation (J/ψ , Υ , χ), l'énergie du faisceau, le nombres de particules simulées et le type de collisionneur pour connaître l'angle de détection.

Une fois ces données intégrées, il est nécessaire de réaliser le tirage aléatoire des impulsions des gluons. A cet effet, il est pratique de créer un graphe de la fonction de structure puis de prendre aléatoirement un nombre à l'intérieur selon les méthodes de Monte-Carlo en utilisant les fonctions **GetRandom**.

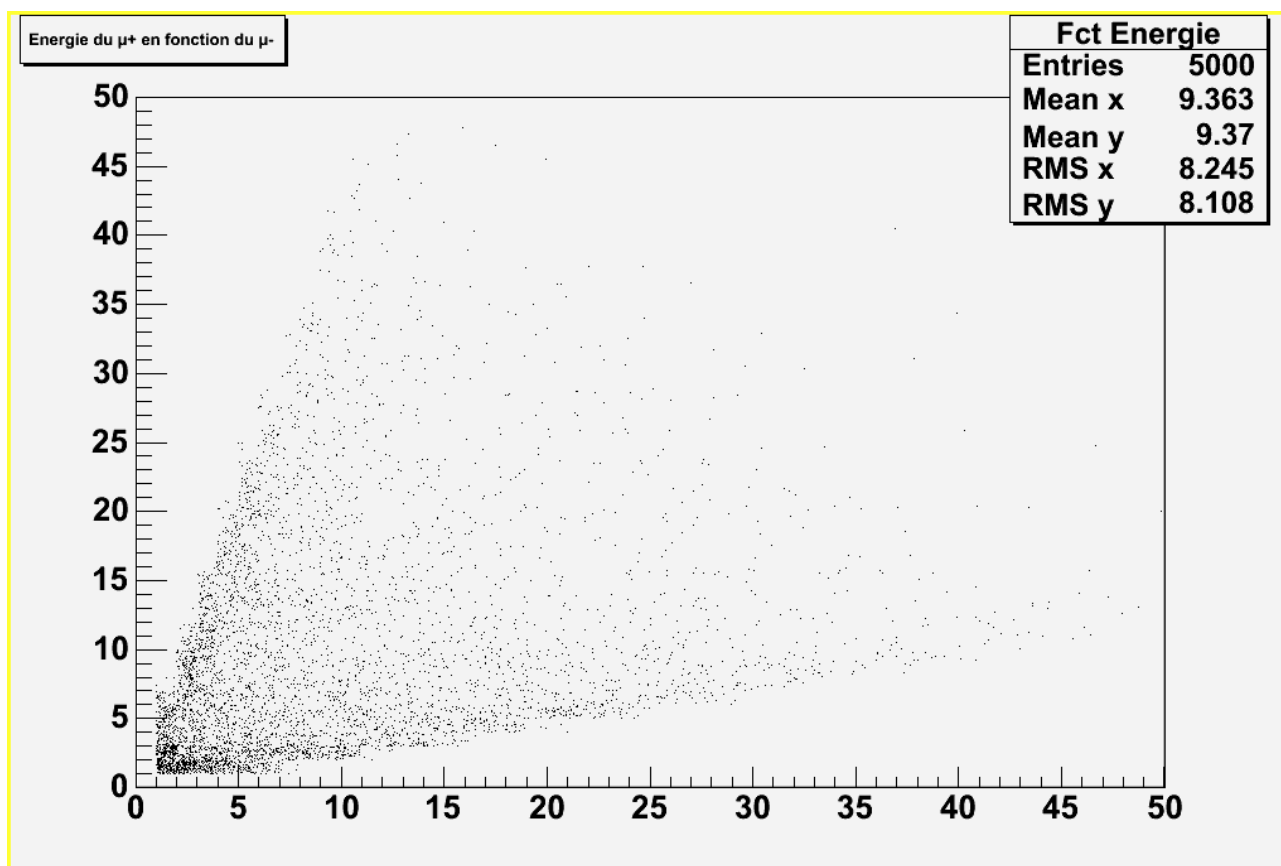
Le programme, une fois les calculs réalisés, propose un menu permettant de connaître les paramètres calculés et de les représenter sous un format de type histogramme. On peut présenter ainsi les paramètres énergétiques ou angulaires des muons, informations nécessaires au développement des détecteurs adaptés. L'application affiche aussi le pourcentage de muons détectés selon l'acceptance des détecteurs.

Les résultats sont présentés en annexe. On retrouve plusieurs paramètres importants qui vont permettre de caractériser les muons provenant de la désintégration des quarkonia.

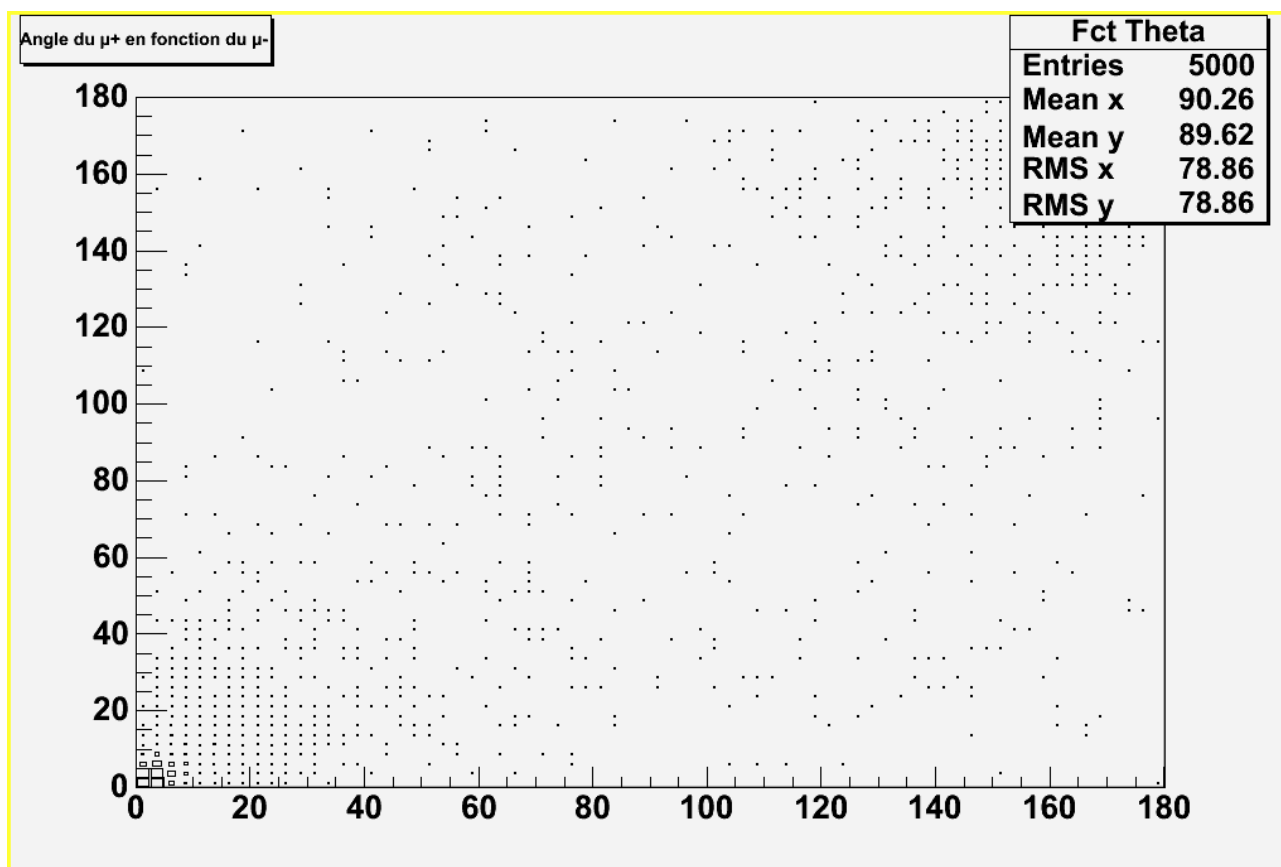
¹¹La réalisation de cette simulation informatique nous permet d'obtenir deux paramètres importants dans la détection des muons provenant de la désintégration de quarkonia : la corrélation des énergies de la paire de muons et la corrélation des angles de départ des muons. On constate que le rapport des énergies est nettement contenu dans une zone bien définie, il permet de caractériser assez précisément l'origine des muons détectés (ici l'origine est la désintégration du quarkonia). De même la corrélation des angles d'émission permet de dégager une tendance : les muons sont émis dos à dos. Une troisième information va permettre de reconstituer l'ensemble des particules émises à partir de la zone de détection, c'est le pourcentage de muons détectées dans la zone d'acceptance du détecteur.

Toutes ces informations conduisent à la mise en place des applications qui vont permettre la reconstitution de la gerbe de particules. A partir de la mesure de différentes variables dans les sous-ensembles du détecteur, il va être possible de remonter le trajet des particules et de reconstituer le trajet qu'elles ont suivies.

Une fois ce travail de « dépouillage » des données réalisés, il sera possible d'étudier les phénomènes supposés liés à la formation du Plasma de Quarks et Gluons, par exemple la diminution du nombre de quarkonia émis. Ces nombreuses acquisitions de données permettent d'établir une base statistique du comportement des nucléons à haute densité d'énergie qui une fois confrontée aux hypothèses théoriques pourra répondre de l'existence du QGP avec une probabilité importante (c'est-à-dire une quasi certitude scientifique).



Annexe 1 : corrélation des énergies des muons.



Annexe 2 : corrélation des angles d'émission des muons.¹²

```

//*****
/** Simulation de la création de quarkonia **
/** dans un choc entre protons. **
//*****
/** version 1.0 12/07/2007 **
//*****

```

```

#include <TGClient.h>
#include <TCanvas.h>
#include <TF1.h>
#include <TGText.h>
#include <TGWindow.h>
#include <TRandom.h>
#include <TGButton.h>
#include <TGFrame.h>
#include <TRootEmbeddedCanvas.h>
#include <RQ_OBJECT.h>
#include <TGWidget.h>
#include <TGNumberEntry.h>

```

```

class MyMainFrame {

```

```

    RQ_OBJECT("MyMainFrame")

```

```

private :

```

```

    TGMMainFrame *fmain;
    TGMMainFrame *main;
    TGComboBox *fCombo;
    TGListBox *MList;
    TGListBox *AList;
    TGNumberEntryField *energyBox;
    TGNumberEntryField *eveBox;
    TH1F *F_xg;
    TH1F *F_p;
    TH1F *F_theta;
    TH1F *F_thetas;
    TH1F *F_E_mu;
    TH1F *F_p_mu;
    TH1F *F_theta_mu;
    TH1F *F_phi_mu;
    TH2F *F_E;
    TH2F *F_th;

```

```

    Float_t percent;

```

```

public :

```

```

    MyMainFrame(const TGWindow *p, Int_t w, Int_t h);
    virtual ~MyMainFrame();
    void Detector();
    void Parameters();

```

```

void DoDraw();
void DrawQuarkonia();
void DrawMuons();
char tmp[3];

};

MyMainFrame::MyMainFrame(const TGWindow *p, Int_t w, Int_t h)
{
    /*** Fenetre principale ***/

    fMain = new TGMainFrame(p,w,h);

    /*** Frames horizontaux ***/

    TGHorizontalFrame *hframe_up = new TGHorizontalFrame(fMain,400,150);
    TGHorizontalFrame *hframe_down = new TGHorizontalFrame(fMain,400,50);
    fMain->AddFrame(hframe_up, new TGLayoutHints(kLHintsExpandX |
kLHintsExpandY,5,5,5,5));
    fMain->AddFrame(hframe_down, new TGLayoutHints(kLHintsExpandX |
kLHintsExpandY,5,5,5,5));

    /*** Energie du faisceau ***/

    groupFrame1 = new TGGroupFrame(hframe_up,"Beam Energy");
    groupFrame1->SetTitlePos(TGGroupFrame::kLeft);
    energyBox = new TGNumberEntryField(groupFrame1,-1,100);
    energyBox->Resize(100,20);
    groupFrame1->AddFrame(energyBox);
    hframe_up->AddFrame(groupFrame1, new TGLayoutHints(kLHintsTop |
kLHintsLeft,5,5,10,10));

    /*** Nombre d'evenements ***/

    groupFrame2 = new TGGroupFrame(hframe_up,"Number of Events");
    groupFrame2->SetTitlePos(TGGroupFrame::kLeft);
    eveBox = new TGNumberEntryField(groupFrame2,1,100);
    eveBox->Resize(100,20);
    groupFrame2->AddFrame(eveBox);
    hframe_up->AddFrame(groupFrame2, new TGLayoutHints(kLHintsTop |
kLHintsLeft,5,5,10,10));

    /*** Choix des quarkonia ***/

    groupFrame3 = new TGGroupFrame(hframe_up,"Mass of Quarkonia");
    groupFrame3->SetTitlePos(TGGroupFrame::kLeft);
    fCombo = new TGComboBox(groupFrame3,100);
    sprintf(tmp,"J/Psi(1s) (3096.87 MeV)",0);

```

```

fCombo->AddEntry(tmp,0);
sprintf(tmp,"Upsilon(1s) (9460.30 MeV)",1);
fCombo->AddEntry(tmp,1);
sprintf(tmp,"Psi (2s) (3685.96 MeV)",2);
fCombo->AddEntry(tmp,2);
sprintf(tmp,"Upsilon(2s) (10023.26 MeV)",3);
fCombo->AddEntry(tmp,3);
sprintf(tmp,"Upsilon(3s) (10355.20 MeV)",4);
fCombo->AddEntry(tmp,4);
sprintf(tmp,"Phi (1s) (1000.00 MeV)",5);
fCombo->AddEntry(tmp,5);
fCombo->Resize(180,20);
fCombo->Select(0);
groupFrame3->AddFrame(fCombo);
hframe_up->AddFrame(groupFrame3, new TGLayoutHints(kLHintsTop |
kLHintsLeft,5,5,10,10));

/**** Creation des boutons ****/

TGTextButton *detect = new TGTextButton(hframe_down,"&Detector");
detect->Connect("Clicked()", "MyMainFrame",this,"Detector()");
TGTextButton *calcul = new TGTextButton(hframe_down,"&Calculate");
calcul->Connect("Clicked()", "MyMainFrame",this,"Parameters()");
TGTextButton *quarkonia = new TGTextButton(hframe_down,"&Quarkonia Parameters");
quarkonia->Connect("Clicked()", "MyMainFrame",this,"DrawQuarkonia()");
TGTextButton *muons = new TGTextButton(hframe_down,"&Muons Parameters");
muons->Connect("Clicked()", "MyMainFrame",this,"DrawMuons()");
TGTextButton *exit = new TGTextButton(hframe_down,"&Exit", "gApplication-
>Terminate(0)");
hframe_down->AddFrame(detect, new TGLayoutHints(kLHintsLeft,15,10,0,0));
hframe_down->AddFrame(calcul, new TGLayoutHints(kLHintsLeft,10,10,0,0));
hframe_down->AddFrame(quarkonia, new TGLayoutHints(kLHintsLeft,10,10,0,0));
hframe_down->AddFrame(muons, new TGLayoutHints(kLHintsLeft,10,10,0,0));
hframe_down->AddFrame(exit, new TGLayoutHints(kLHintsLeft,10,10,0,0));

/**** Affichage des fenetres ****/

fMain->SetWindowName("Simulation");
fMain->MapSubwindows();
fMain->Resize(fMain->GetDefaultSize());
fMain->MapWindow();

}

void MyMainFrame::Detector()

{

    Int_t c=0;

```

```

/**** Acceptance des collisionneurs ****/

main = new TGMainFrame(gClient->GetRoot(),150,150);

AList = new TGListBox(main);
sprintf(tmp,"PHENIX @ 100 GeV",0);
AList->AddEntry(tmp,0);
sprintf(tmp,"PHENIX @ 250 GeV",1);
AList->AddEntry(tmp,1);
sprintf(tmp,"ALICE @ 2250 GeV",2);
AList->AddEntry(tmp,2);
sprintf(tmp,"ALICE @ 7000 GeV",3);
AList->AddEntry(tmp,3);
AList->Resize(130,75);
AList->Select(0);
main->AddFrame(AList, new TGLayoutHints(kLHintsTop | kLHintsLeft,5,5,10,10));

main->SetWindowName("Detector");
main->MapSubwindows();
main->Resize(main->GetDefaultSize());
main->MapWindow();

}

void MyMainFrame::Parameters()

{

    Int_t ni;
    Float_t kt1, kt2, xg1, xg2, Eg1, Eg2, Egg;
    Float_t p, theta, cos;
    Float_t Ejp, Beta, Gamma, thetas, phis;
    Float_t Esm, psm, E_mu1, E_mu2, px_mu1, px_mu2, py_mu1, py_mu2, pz_mu1, pz_mu2,
p_mu1, p_mu2;
    Float_t pxy_mu1, pxy_mu2, theta_mu1, theta_mu2, phi_mu1, phi_mu2;
    Float_t Eb, Nevt, Mass, Choice;

    Int_t d = 0;
    Int_t c = 0;
    Int_t compt = 0;

/**** Lecture des variables ****/

Eb = energyBox->GetNumber();
Nevt = eveBox->GetNumber();
Choice = fCombo->GetSelected();

/**** Determination du detecteur ****/

c = AList->GetSelected();

```

```

if (c==0) {
    energyBox->SetNumber(100);
    d=0;
}
if (c==1) {
    energyBox->SetNumber(250);
    d=0;
}
if (c==2) {
    energyBox->SetNumber(2250);
    d=1;
}
if (c==3) {
    energyBox->SetNumber(7000);
    d=1;
}

```

/**/ Determination de la masse des quarkonia **/

```

if (Choice==0) Mass=3.09687;
if (Choice==1) Mass=9.46030;
if (Choice==2) Mass=3.68596;
if (Choice==3) Mass=10.02326;
if (Choice==4) Mass=10.35520;
if (Choice==5) Mass=10.00000;

```

/**/ Declaration des graphes **/

```

F_xg = new TH1F("Fct Gluons","Fonction de structure des gluons",100,0,1);
F_p = new TH1F("Fct Impulsion","Impulsion des quarkonia",50,0,50);
F_theta = new TH1F("Fct Theta","Angle des quarkonia",72,0,180);
F_thetas = new TH1F("Fct Theta Star","Angle des muons",90,0,360);
F_E_mu = new TH1F("Fct Energie μ","Energie des muons",50,0,50);
F_p_mu = new TH1F("Fct Impulsion μ","Impulsion des muons",60,0,30);
F_theta_mu = new TH1F("Fct Angle Theta μ","Angle polaire des muons",72,0,180);
F_phi_mu = new TH1F("Fct Angle Phi μ","Angle azymutal des muons",72,0,180);
F_E = new TH2F("Fct Energie","Energie du μ+ en fonction du μ-",50,0,50,50,0,50);
F_th = new TH2F("Fct Theta","Angle du μ+ en fonction du μ-",72,0,180,72,0,180);

```

/**/ Creation des evenements **/

```

for (ni=0; ni<Nevt; ni++)

```

```

{

```

/**/ Tirage aléatoire des impulsions **/

```

TF1 * fun_kt = new TF1("fun_kt","(1./sqrt(6.28))) * exp(-x**2/2)",-10,10);
kt1 = fun_kt->GetRandom();

```

```

kt2 = fun_kt->GetRandom();
TF1 * fun_xg = new TF1("fun_xg","(35.8*x**2.3 * (1-x)**4) / x",0.0,1.0);
xg1 = fun_xg->GetRandom();
xg2 = fun_xg->GetRandom();
F_xg->Fill(xg1);
F_xg->Fill(xg2);

```

```

/**** Condition de création des quarkonia ****/

```

```

Eg1 = sqrt((kt1)**2+xg1**2*Eb**2);
Eg2 = sqrt((kt2)**2+xg2**2*Eb**2);
Egg = sqrt(2*Eg1*Eg2-2*kt1*kt2+2*xg1*xg2*Eb**2);

```

```

if (Egg > Mass)

```

```

{

```

```

/**** Parametres des quarkonia dans le ref du labo ****/

```

```

p = sqrt(((kt1-kt2)**2+((xg1-xg2)**2)*Eb**2));
cos = ((xg1-xg2)*Eb)/p;
theta = (TMath::ACos(cos))*(180/3.14);
F_p->Fill(p);
F_theta->Fill(theta);
theta = theta*(3.14/180);

```

```

}

```

```

/**** Parametres du CMS des quarkonia ****/

```

```

Ejp = sqrt(Mass**2+p**2);
Beta = (p/Ejp);
Gamma = 1/sqrt(1-Beta**2);

```

```

/**** Tirage aleatoire des angles de production des muons ****/

```

```

TF1 * fun_theta_star = new TF1("fun_theta_star","1",0,360);
thetas = fun_theta_star->GetRandom();
TF1 * fun_phi_star = new TF1("fun_phi_star","1",0,360);
phis = fun_phi_star->GetRandom();
F_thetas->Fill(thetas);

```

```

/**** Parametres des muons dans le CMS ****/

```

```

Esm = Mass/2;
psm = sqrt(Esm**2-0.10566**2);

```

```

/**** Parametres energetiques des muons dans le ref du labo ****/

```

```

E_mu1 = Gamma*(Esm+Beta*psm*cos(thetas));

```



```

E_mu2 = Gamma*(Esm-Beta*psm*cos(thetas));
px_mu1 = psm*sin(thetas)*cos(phis);
px_mu2 = -(psm*sin(thetas)*cos(phis));
py_mu1 =
psm*sin(thetas)*sin(phis)*cos(theta)+Gamma*(Beta*Esm+psm*cos(thetas))*sin(theta);
py_mu2 = -(psm*sin(thetas)*sin(phis)*cos(theta))+Gamma*(Beta*Esm-
psm*cos(thetas))*sin(theta);
pz_mu1 = -
(psm*sin(thetas)*sin(phis)*sin(theta))+Gamma*(Beta*Esm+psm*cos(thetas))*cos(theta);
pz_mu2 = psm*sin(thetas)*sin(phis)*sin(theta)+Gamma*(Beta*Esm-
psm*cos(thetas))*cos(theta);
p_mu1 = sqrt(px_mu1**2+py_mu1**2+pz_mu1**2);
p_mu2 = sqrt(px_mu2**2+py_mu2**2+pz_mu2**2);
pxy_mu1 = sqrt(p_mu1**2-pz_mu1**2);
pxy_mu2 = sqrt(p_mu2**2-pz_mu2**2);
F_E_mu->Fill(E_mu1);
F_E_mu->Fill(E_mu2);
F_p_mu->Fill(p_mu1);
F_p_mu->Fill(p_mu2);

/**** Parametres angulaires des muons dans le ref du labo ****/

theta_mu1 = (TMath::ACos(pz_mu1/p_mu1))*(180/3.14);
theta_mu2 = (TMath::ACos(pz_mu2/p_mu2))*(180/3.14);
F_theta_mu->Fill(theta_mu1);
F_theta_mu->Fill(theta_mu2);
phi_mu1 = (TMath::ACos(px_mu1/pxy_mu1))*(180/3.14);
phi_mu2 = (TMath::ACos(px_mu2/pxy_mu2))*(180/3.14);
F_phi_mu->Fill(phi_mu1);
F_phi_mu->Fill(phi_mu2);
F_E->Fill(E_mu1,E_mu2);
F_th->Fill(theta_mu1,theta_mu2);

/**** Acceptance des detecteurs ****/

if (d==0) {
    if (theta_mu1>12 && theta_mu1<35 || theta_mu1>145 && theta_mu1<170) compt++;
    if (theta_mu2>12 && theta_mu2<35 || theta_mu2>145 && theta_mu2<170) compt++;
}

if (d==1) {
    if (theta_mu1>2 && theta_mu1<9) compt++;
    if (theta_mu2>2 && theta_mu2<9) compt++;
}

}

percent = (compt/Nevt)*100;

}

```

```

void MyMainFrame::DrawMuons()

{

    /*** Affichage des histogrammes : quarkonia ***/

    main = new TGMainFrame(gClient->GetRoot(),200,200);

    MList = new TGListBox(main);
    sprintf(tmp,"Energie des muons produits",0);
    MList->AddEntry(tmp,0);
    sprintf(tmp,"Correlation des energies",1);
    MList->AddEntry(tmp,1);
    sprintf(tmp,"Difference des angles (yz)",2);
    MList->AddEntry(tmp,2);
    sprintf(tmp,"Difference des angles (xy)",3);
    MList->AddEntry(tmp,3);
    sprintf(tmp,"Correlation des angles",4);
    MList->AddEntry(tmp,4);
    sprintf(tmp,"Pourcentage de muons detectes",5);
    MList->AddEntry(tmp,5);
    MList->Resize(180,90);
    MList->Select(0);
    main->AddFrame(MList, new TGLayoutHints(kLHintsTop | kLHintsLeft,5,5,10,10));

    TGTextButton *MButton = new TGTextButton(main,"&Draw");
    MButton->Connect("Clicked()", "MyMainFrame",this,"DoDraw()");
    main->AddFrame(MButton, new TGLayoutHints(kLHintsLeft,100,10,5,5));

    main->SetWindowName("Menu muons");
    main->MapSubwindows();
    main->Resize(main->GetDefaultSize());
    main->MapWindow();

}

void MyMainFrame::DrawQuarkonia()

{

    /*** Affichage des histogrammes : quarkonia ***/

    main = new TGMainFrame(gClient->GetRoot(),200,200);

    MList = new TGListBox(main);
    sprintf(tmp,"Fonction de structure (gluons)",10);
    MList->AddEntry(tmp,10);
    sprintf(tmp,"Impulsion des quarkonia",11);
    MList->AddEntry(tmp,11);

```

```

sprintf(tmp,"Angle des quarkonia",12);
MList->AddEntry(tmp,12);
MList->Resize(180,60);
MList->Select(10);
main->AddFrame(MList, new TGLayoutHints(kLHintsTop | kLHintsLeft,5,5,10,10));

```

```

TGTextButton *QButton = new TGTextButton(main,"&Draw");
QButton->Connect("Clicked()", "MyMainFrame",this,"DoDraw()");
main->AddFrame(QButton, new TGLayoutHints(kLHintsLeft,100,10,5,5));

```

```

main->SetWindowName("Menu quarkonia");
main->MapSubwindows();
main->Resize(main->GetDefaultSize());
main->MapWindow();

```

```

}

```

```

void MyMainFrame::DoDraw()

```

```

{

```

```

    /*** Dessin des histogrammes ***/

```

```

    Int_t c;

```

```

    c = MList->GetSelected();

```

```

    TCanvas *cx = new TCanvas("cx","canvas",0,0,800,250);

```

```

    if (c==0) {
        cx->cd(); F_E_mu->Draw();
    }

```

```

    if (c==1) {
        cx->cd(); F_E->Draw();
    }

```

```

    if (c==2) {
        cx->cd(); F_theta_mu->Draw();
    }

```

```

    if (c==3) {
        cx->cd(); F_phi_mu->Draw();
    }

```

```

    if (c==4) {
        cx->cd(); F_th->Draw("BOX");
    }

```

```

    if (c==5) {
        cx->cd(); F_theta_mu->Draw();

```

```

        cout << percent << endl;

```

```

    }

```

```

    if (c==10) {
        cx->cd(); F_xg->Draw();
    }
    if (c==11) {
        cx->cd(); F_p->Draw();
    }
    if (c==12) {
        cx->cd(); F_theta->Draw();
    }

    cx->Update();

}

MyMainFrame::~MyMainFrame()

{

    /*** Suppression de la Class MyMainFrame ***/

    fMain->Cleanup();
    delete fMain;

}

void simulb()

{

    /*** Programme principal ***/

    gROOT->Reset();
    new MyMainFrame(gClient->GetRoot(),400,400);

}

```