

Mesure de la masse du muon

Romain COHET - Alexandre FAURE - Nicolas FOY

8 avril 2011

Résumé

Ce rapport est issu des mesures et travaux effectués à l'Université Montpellier II dans le cadre des travaux pratiques de physique des particules pour le diplôme de Master 2 Cosmos, Champs et Particules pendant les mois de janvier et février 2011. Dans une première partie, nous tenterons de comprendre de manière générale le fonctionnement d'un photomultiplicateur avant de poursuivre par une étude plus détaillée de la mesure indirecte de la masse du muon avec un système de blocs scintillateurs.

Table des matières

1	Les muons	4
1.1	Les rayons cosmiques	4
1.2	Caractéristiques des muons	4
1.2.1	Application	5
2	Matériel	6
2.1	Photomultiplicateur	6
2.2	Scintillateur	8
2.3	Raquettes	10
3	Caractérisation du photo-multiplicateur	11
3.1	Programmes utilisés	11
3.2	Résultats des mesures	11
3.2.1	Confirmation physique de la loi de Beer-Lambert	12
3.2.2	Evolution de l'amplitude du signal en fonction de la haute-tension HV	13
4	Protocole expérimental	14
4.1	Etude préliminaire	14
4.1.1	Détection des muons	14
4.2	Manipulation à quatre raquettes	16
4.2.1	Evènement "Vrai"	16
4.2.2	Mesures	17
4.2.2.1	Calcul des temps de détections	18
4.2.2.2	Calcul des vitesses	18
4.2.2.3	Energie perdue	20
4.2.3	Protocole expérimental	21
5	Résultats de l'étude	24
6	Conclusion	26
7	Annexes	27
7.1	Figures	27
7.2	Programme d'analyse des données du PM <code>etalonnage.C</code>	31
7.3	Programme d'analyse	39

Chapitre 1

Les muons

Nous détaillerons dans cette partie introductives quelques notions basiques en physique des particules afin de cerner le but du travail réalisé.

1.1 Les rayons cosmiques

Les *rayons cosmiques* sont des particules subatomiques chargées provenant de l'espace, entrant en collision avec la haute atmosphère. Elles produisent des particules dites *secondaires* qui arriveront, à leur tour, jusqu'au sol.

Leur particularité est également d'arriver au sol individuellement et non sous forme de *paquet* ou de *faisceau* comme c'est le cas dans les accélérateurs de particules par exemple.

La plupart des rayons cosmiques arrivant dans la haute-atmosphère sont les protons à 89%, 10% de noyaux d'Hélium et 1% d'éléments plus lourds. Ces particules entrent en collision avec les molécules de l'atmosphère et provoquent de multiples ionisation donnant naissance à une douche de particules secondaires appelées *gerbes*. De ces gerbes hadroniques va résulter la production des particules qui nous intéressent dans notre études, les *muons*.

En effet, il existe plusieurs types de gerbes, les gerbes *électromagnétique* qui résultent de la collision d'un photon primaire dans l'atmosphère, donnant naissance à des particules filles (des paires d'électrons-positrons) qui vont apparaître sous forme de véritables *cascades*. On distingue également les gerbes hadroniques qui, elles, sont provoquées par des protons primaires cosmiques qui vont à leur tour déclencher des *douches* de hadrons.

1.2 Caractéristiques des muons

Le *muon* est une particule élémentaire du modèle standard possédant une charge négative et un spin $\frac{1}{2}$ (fermion), elle appartient à la catégorie des leptons.

Les muons ont une masse de $105.7MeV.c^{-2}$ soit 200 fois la masse de l'électron. Ils possèdent également la propriété de pénétrer la matière de manière plus profonde que les électrons. Leur masse étant plus importante que celle des électrons, leur ralentissement par *brehmstrahlung* s'en trouve d'autant plus difficile. Le libre parcours moyen des muons est donc supérieur à celui des électrons.

1.2.1 Application

Dans notre cas pratique, nous voulons détecter des muons ayant effectivement traversé la haute atmosphère terrestre. Sachant que l'épaisseur de cette atmosphère est d'environ $1000g.cm^{-2}$, les muons vont perdre environ 2 GeV par ionisation après l'avoir traversée.

Il faut cependant noter que déjà à 15 000 mètres environ, nous notons une épaisseur de $175g.cm^{-2}$. Le muon perd donc déjà environ 350 MeV. C'est néanmoins dès cette étape que les muons secondaires sont générés¹.

Les muons arrivent au niveau de la mer avec un flux moyen typique de $1cm^{-2}.s^{-1}$ ce qui correspond à environ la moitié de la radiation totale naturelle de fond pour ces particules.

Le temps de vie moyen des muons mesuré dans leur référentiel propre est de 2.2 μs . Dès lors, à la vitesse de la lumière, cela donne une distance d'existence typique de 658.654 m².

¹N'oublions pas que nous effectuons nos mesures en laboratoire et qu'il faudrait en théorie prendre en compte les interactions avec la structure même du bâtiment.

²Distance et temps qui nous paraissent évidemment bien plus longs tels que nous les percevons en vertu de la théorie de la relativité restreinte.

Chapitre 2

Matériel

Dans cette partie du rapport, nous détaillerons le principe de fonctionnement des deux composants majeurs utilisés lors de ces travaux pratiques : le photomultiplicateur et les raquettes scintillantes auxquelles est rattaché un photomultiplicateur pour chacune d'entre elle.

2.1 Photomultiplicateur

Le Photomultiplicateur (ou *PM*), est un dispositif qui permet de produire un courant électrique lorsque celui-ci détecte des photons.

Le schéma de ce dispositif couplé à un scintillateur¹ est représenté sur la figure 2.1.

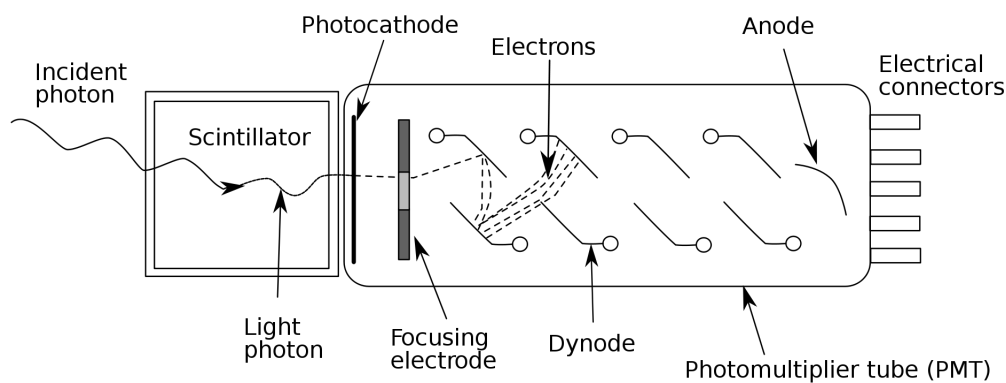


FIG. 2.1 – Schéma d'un photomultiplicateur couplé à un scintillateur.

Il est composé d'une tube principal dans lequel règne un vide poussé afin de perturber le moins possible la circulation des électrons dans le dispositif.

¹Nous aborderons la notion de milieu scintillant dans la partie suivante.

En entrée du dispositif se trouve une *photocathode*² qui va permettre la détection des photons incidents. En effet, le principe de fonctionnement est basé sur l'effet photoélectrique ; c'est à dire que lorsqu'un photon traverse le matériau de la photocathode, celui-ci éjecte un (ou plusieurs) électrons. Ces électrons vont ensuite être envoyés dans le tube à l'aide d'une électrode convergente.

La seconde partie cruciale du PM est ce que nous pourrions appeler, le multiplicateur d'électrons que l'on retrouve par un agencement de multiples électrodes que l'on appelle des *dynodes*. Chacune des dynodes sont soumises à un potentiel électrique positif, qui croît pour chaque dynode lorsque l'électron les rencontre. Ainsi, les électrons qui avaient à l'origine, une énergie³ $h\nu - \Phi$ en sortie de la photocathode, se retrouvent avec une énergie beaucoup plus importante en sortie. Le principe physique de l'effet photoélectrique est résumé par soucis de clarté, avec la figure 2.2

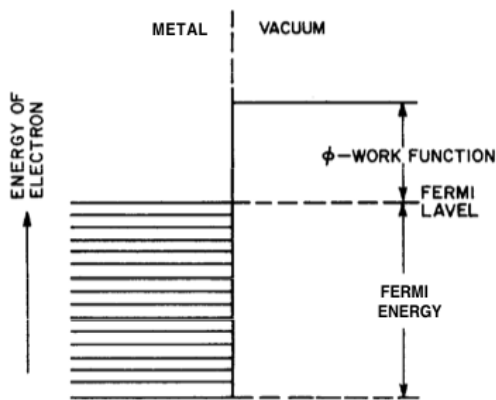


FIG. 2.2 – Principe physique de l'effet photoélectrique avec le niveau de Fermi et le travail de sortie Φ . Source : *Photomultiplier Handbook*

De plus, lorsque les électrons heurtent les dynodes, il se crée une "avalanche d'électrons" secondaires qui vont être arrachés de la dynode pour ensuite être accélérés par la dynode suivante. A chaque étape de rencontre de dynode, le nombre d'électrons secondaires en sortie est supérieur.

Typiquement, ces photomultiplicateurs sont sensibles de l'ultraviolet à l'infrarouge proche et le temps de réponse est de l'ordre de la nanoseconde. Ainsi, à partir de la détection d'un seul photon par le dispositif, le PM, alimenté par une source de haute tension⁴ est en mesure de générer un courant que l'on peut facilement mesurer, même si un seul photo-électron est détecté par l'appareil.

²Il existe plusieurs dizaines de matériaux différents dont l'AG-O-Cs ou encore le GaAs :Cs. Le matériau constituant les photocathodes de nos PM est le

³Avec la variable Φ qui correspond au potentiel à appliquer à l'électron pour lui permettre d'être arraché de la surface du métal, également appelé le *travail de sortie*

⁴La haute-tension sera notée HV pour *High-Voltage* dans la suite du rapport.

2.2 Scintillateur

Le *scintillateur* ou *milieu scintillant* désigne un instrument qui est composé d'un matériau émettant des photons suite à un dépôt d'énergie par interaction rayonnement-matière.

De façon plus concrète, une particule d'énergie donnée qui rencontre le scintillateur dépose une partie de son énergie incidente dans le matériau en excitant les molécules de celui-ci. Les électrons passent donc à un état d'énergie supérieur et ré-emettent cette énergie par desexcitation spontanée sous forme de rayonnement. Ce principe physique est résumé par la figure 2.3.

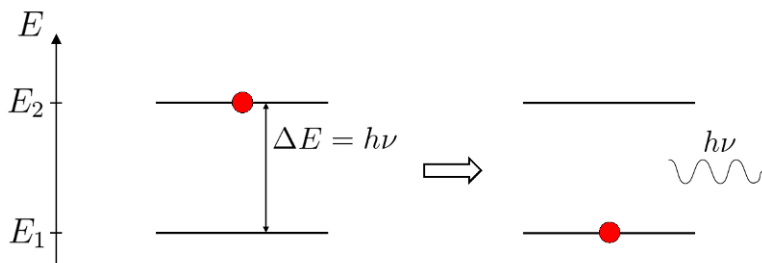


FIG. 2.3 – Principe physique de la desexcitation électronique spontanée. Source : *media4.obspm.fr*

Il existe deux grandes familles de scintillateurs :

- **Scintillateurs organiques** composé de molécules organiques que l'on retrouve sous forme de cristaux, de liquides ou de plastique.

En ce qui concerne la physique qui intervient lors de la propagation de l'énergie dans le matériau organique, celle-ci fait intervenir des transitions énergétiques via les électrons de valence des molécules.

- **Scintillateurs inorganiques** composé d'un matériau minéral, le plus souvent sous forme cristalline. Nous noterons aussi l'existence de scintillateur gazeux⁵ ou à base de verre.

Dans ce cas, la propagation d'énergie est expliquée par la théorie des bandes des solides cristallins proposée par Bloch. Cette physique entre alors en jeu et il faut considérer les perturbations qui entrent en jeu dues aux interactions atomes et ions qui forment notre cristal. La structure de conduction du cristal est donc ainsi modifiée (i.e. modification de la bande de valence et de l'énergie de *gap* de notre matériau scintillant).

L'étape clé de la détection de la particule chargée par le matériau scintillant est le dépôt d'énergie de la particule incidente. En physique des particules, une particule chargée

⁵Les scintillateurs gazeux sont composés le plus souvent d'azote ou de gaz nobles (He, Ar, Kr, Xe)

qui traverse un matériau va ioniser les atomes ou les molécules qu'elle rencontre sur son parcours.

On définit alors le *pouvoir d'arrêt* d'un matériau qui définit la perte moyenne d'énergie de la particule par unité de longueur (et s'exprime en $MeV.cm^{-1}$) lorsqu'elle le traversera. Cette perte d'énergie est modélisée par la formule relativiste de Bethe et Bloch (1932) dont une expression possible est la suivante :

$$-\frac{dE}{dx} = \frac{2\pi N z^2 e^4 Z \rho}{m_0 c^2 A \beta^2} \left[\ln \left(\frac{2m_0 c^2 \beta^2 \left(\frac{2m_0 c^2 \beta^2}{1-\beta^2} \right)}{I^2 (1-\beta^2)} \right) - 2\beta^2 \right] \quad (2.1)$$

Avec les paramètres N : nombre d'Avogadro, z : charge de la particule incidente, Z : numéro atomique du matériau traversé, A : nombre de masse du matériau traversé, ρ : densité volumique du matériau traversé, m_0 masse de l'électron et I : potentiel d'excitation moyen du matériau cible.

Dans cette formule, nous pouvons remarquer que :

- La masse de la particule incidente n'intervient pas.
- $-\frac{dE}{dx}$ décroît comme $\beta^{-5/3}$ pour $\beta\gamma < 3$.
- $-\frac{dE}{dx}$ possède un minimum aux alentours de $\beta\gamma = 3$. Sa position dépend évidemment de I mais reste faible en pratique.
- Au delà du minimum, $\frac{dE}{dx}$ remonte en $\ln(\gamma)^6$. Peut s'interpréter comme provenant du fait que l'énergie de seuil a un effet sur des atomes de plus en plus éloignés de la trajectoire de la particule.
- A très haute énergie, la remontée relativiste est interrompue par l'effet de densité : la polarisation du milieu écran le champ électrique pour les atomes les plus lointains.

Dans la formule 4.1, le terme $\ln(E_M)$ est du aux collisions proches de la trajectoire. Alors que le terme $\ln\left(\frac{\beta^2}{1-\beta^2}\right)$ exprime l'influence du champs électrique sur les atomes éloignés. Quand beta croît, c'est le dernier terme en beta carré qui devient évidemment prépondérant et la particule ioniserait dans un cylindre de diamètre de plus en plus grand.

En fait, le champs électrique de polarisation des atomes voisins va réduire cet effet, et la perte d'énergie tend ainsi vers une constante appelée, le *plateau de Fermi*.

⁶C'est ce qu'on appelle la remontée relativiste.

2.3 Raquettes

Lors de cet étude, nous avons à dispositions quatre *raquettes*. Comme leur dénomination l'indique, celles-ci sont constituées d'une partie *scintillateur* qui est le détecteur primaire de l'instrument, dans lequel nous détectons nos particules chargées.

Le scintillateur est directement couplé à un photomultiplicateur qui permet de recueillir les photons émis comme nous l'avons détaillé précédemment.

Les dimensions des raquettes que nous avons utilisé sont les suivantes :

$$30cm \times 20cm \times 1cm$$

Chapitre 3

Caractérisation du photo-multiplicateur

3.1 Programmes utilisés

Le programme¹ *etalonnage.C* est le premier programme que nous avons utilisé pour les premiers traitements de données en sortie d'acquisition avec le logiciel *LabView*.

Le programme *etalonnage.C* est un programme conçu en C++, chargé d'analyser tous les fichiers en sortie du programme LabView utilisé en travaux pratiques, pour un numéro de run donné (et demandé à l'utilisateur au démarrage du programme). Le programme va ensuite répertorier la valeur minimum de chaque fichier du run, pour chacun des quatre canaux en sortie. On calcul également l'intégrale de chaque pic à partir d'une valeur seuil (i.e. *threshold*) spécifié également par l'utilisateur à l'entrée du programme. Enfin, le programme calcul le temps δT qui s'écoule entre les dépassements montant et descendant de la valeur du seuil.

Nous executons donc ce programme lors de la première phase de caractérisation du gain du photomultiplicateur.

3.2 Résultats des mesures

Dans un premier temps, pour la caractérisation du photomultiplicateur utilisé en travaux pratiques, nous avons effectué deux séries de mesures de calibration :

- Etude de la variation de l'amplitude et de l'intégrale du pic du signal en fonction de la distance.
- Etude de la variation de l'amplitude et de l'intégrale du pic du signal en fonction de la haute-tension² appliquée au PM.

¹Le programme *etalonnage.C* est fourni en annexes du rapport

²Notée *HV*.

3.2.1 Confirmation physique de la loi de Beer-Lambert

Après avoir utilisé notre programme *etalonnage.C*, nous avons pu avoir accès aux représentations graphiques énoncées plus haut et notamment celle de l'amplitude du signal en fonction de la distance de la DEL au photomultiplicateur. Nous obtenons alors le résultat présenté en figure 3.1.

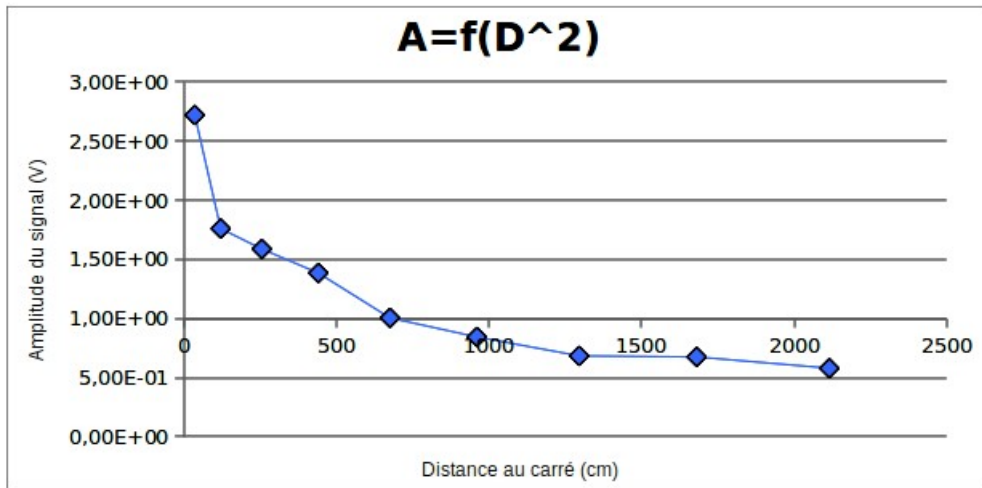


FIG. 3.1 – Amplitude du signal de sortie du PM en fonction de la distance d'éloignement de la DEL.

Nous constatons alors que la courbe possède une décroissance de type exponentiel. Ceci est conforme aux lois physiques connues. En effet, la loi de Beer-Lambert nous indique que l'absorption de la lumière est directement liée au milieu traversé grâce à la formule suivante :

$$I(\lambda, X) = I_0(\lambda)e^{-\alpha X} \quad (3.1)$$

C'est ce qui explique cette décroissance en forme exponentielle lorsque nous éloignons la DEL du PM. Nous pouvons voir ceci comme une confirmation empirique de la loi de Beer-Lambert.

3.2.2 Evolution de l'amplitude du signal en fonction de la haute-tension HV

Nous avons également tracé la variation de l'amplitude du signal reçu par le PM en fonction de la haute-tension appliquée à celui-ci. Nous obtenons le résultat de la figure 3.2.

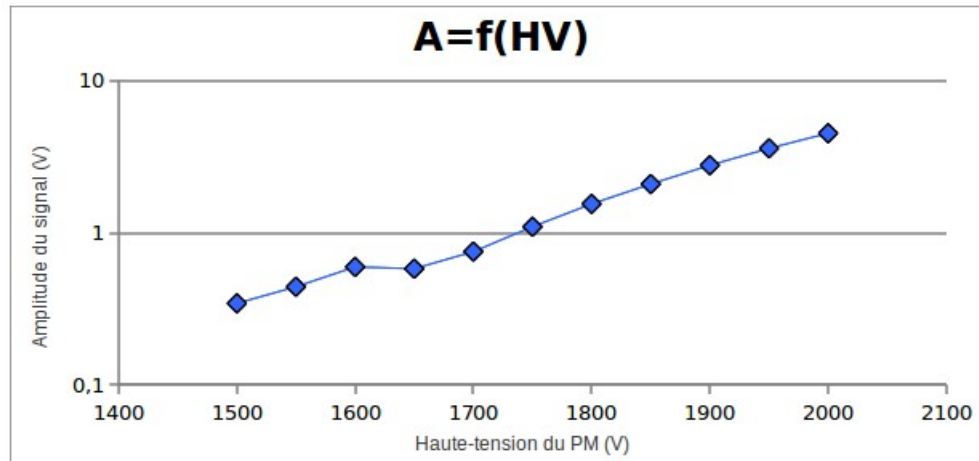


FIG. 3.2 – Amplitude du signal de sortie du PM en fonction de la haute-tension HV appliquée au PM.

On remarque que l'amplitude du signal acquis par le PM dépend de façon linéaire avec la haute-tension appliquée. Ceci est conforme à ce que l'on s'attendait à obtenir. En effet, les deux valeurs mesurées dans ce cas, proviennent de grandeurs issues du PM lui-même. On s'attendait donc à une relation bien linéaire entre amplitude et haute-tension³.

³C'est d'ailleurs bien également le cas pour l'intégrale des pics de signal en fonction de HV (que nous n'avons pas inclus dans le rapport car il n'y a aucun apport protocolaire supplémentaire).

Chapitre 4

Protocole expérimental

4.1 Etude préliminaire

Avant de commencer l'expérience sur les muons, nous étudions le fonctionnement d'un photo-multiplicateur dans une boîte noire, illuminé par une diode. Le PM est alimenté à un générateur haute tension (maximum de 2000V) et branché à un oscilloscope afin de voir sa réponse. La diode est alimentée par un générateur à impulsion qui est également relié à l'oscilloscope.

Les impulsions sur la diode sont fixées et on varie seulement l'amplitude de l'impulsion afin d'éliminer les rebonds du signal du PM observés sur l'oscilloscope. Le trigger est réglé sur -0.100 mV sur le signal du PM de manière à être au dessus du bruit de fond afin de ne pas déclencher de faux événements issus de ce bruit.

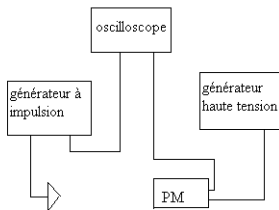


FIG. 4.1 – Schéma du protocole de caractérisation du gain du photomultiplicateur.

Une fois les réglages effectués on observe la réponse du PM en faisant varier 2 paramètres :

- La distance diode-PM par pas de 5 cm.
- La tension sur le PM par pas de 50 V.

4.1.1 Détection des muons

Pour détecter les muons, on utilise des raquettes dans lesquels se trouve un PM. Ces raquettes sont alimentées par un générateur en 24V et sont reliées à l'oscilloscope.

Elles sont également reliées à un *discriminant* qui permet d'effectuer des opérations logiques entre les différentes raquettes utilisées ainsi que de compter les événements.

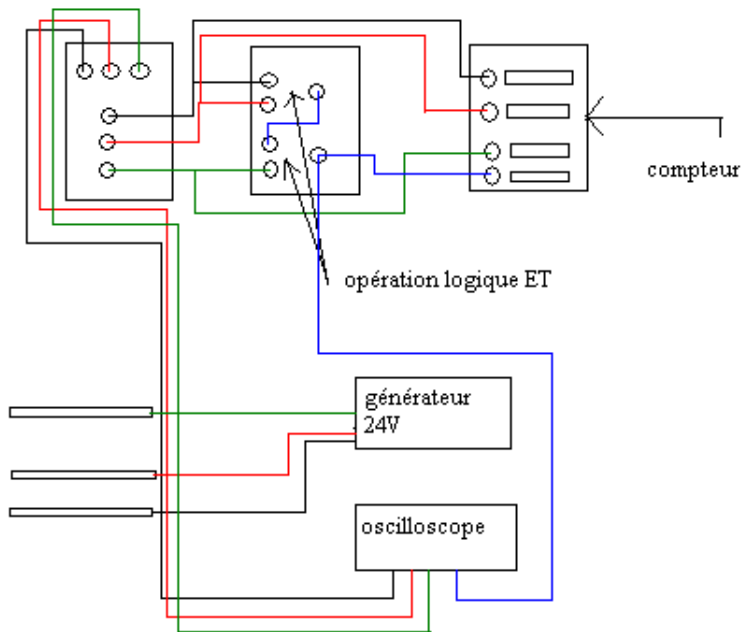


FIG. 4.2 – Schéma des câblages effectués sur les différents matériels pour la détection des muons.

Comme le montre la figure 4.2, on utilise 3 raquettes reliées à l'oscilloscope et reliées au discriminant. On effectue les opérations noir ET rouge ET vert et on relie la réponse à l'oscilloscope. Cette opération, si elle est vraie, envoie un signal porte. On trigge sur cette fonction. Ainsi, si l'opération logique donne la réponse vraie, l'événement d'un muon passant dans la raquette 1, 2 et 3 est enregistré.

Un trigger est également appliqué à l'entrée du discriminant pour sélectionner des événements en dehors du bruit de fond. On règle également la largeur de la fonction porte qui doit être suffisamment large pour laisser le temps au muons de traverser les raquettes, de traverser les fils et de réaliser les opérations logiques.

Au fil des séances de travaux pratiques, nous avons donc penché pour une méthode de détection à quatre raquettes, plus efficace et plus restrictive dans nos résultats comme nous le verrons dans la partie suivante.

4.2 Manipulation à quatre raquettes

Nous souhaitons obtenir la masse du muon à partir de l'expérience définie par la figure 4.3. Dans cette partie, on se propose d'expliquer la mesure des différentes grandeurs intervenant dans le calcul de la masse du muon. Nous expliquerons le protocole de mesure, calculerons les sources d'incertitudes, et établirons l'expression de la masse à partir de l'énergie perdue.

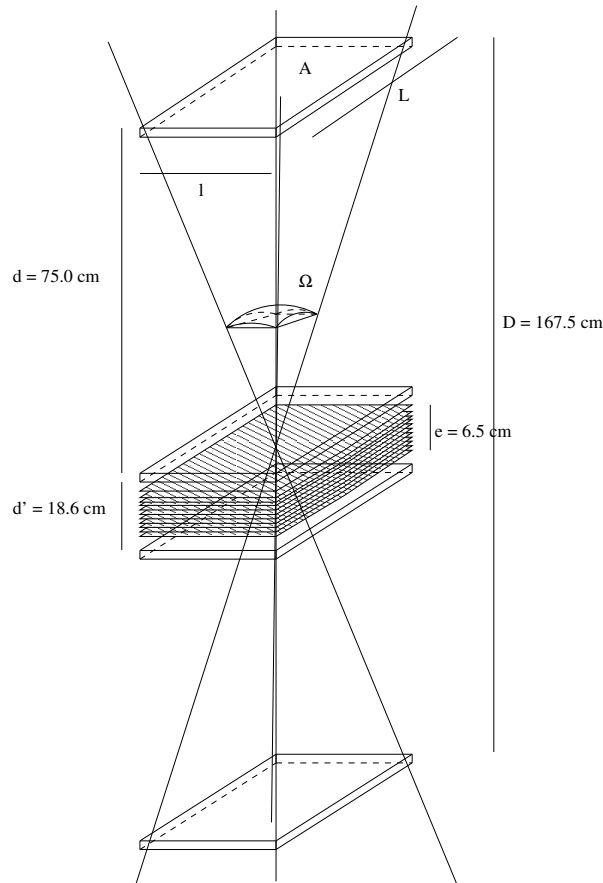


FIG. 4.3 – Dispositif expérimental.

4.2.1 Évènement “Vrai”

Nous qualifierons d'*évènement vrai* tout évènement qui correspond à un muon traversant les quatre raquettes (cf. figure 4.3). Pour fixer les idées, sur 1565 évènements détectés, seuls quelques uns sont des *évènements vrais*. Pour expliquer cet écart, il suffit de calculer *l'angle solide de détection* de notre système. En effet, toutes les raquettes, de surface $A = L \cdot l$, permettent de définir un parallélépipède constituant notre détecteur. Sa hauteur totale D en définit le centre, et moyennant l'hypothèse très grossière d'un flux de muon isotrope, on peut rapporter la géométrie de notre détecteur

à celle d'une demi-sphère de rayon

$$R = \frac{D}{2\cos\theta} \quad (4.1)$$

où

$$\theta = \arctan\left(\frac{2}{D}\sqrt{\frac{A}{\pi}}\right) \quad (4.2)$$

avec A est l'aire du cercle défini par l'intersection du plan d'équation $z = D/2$ avec la demi-sphère, qui est aussi l'aire d'une raquette.

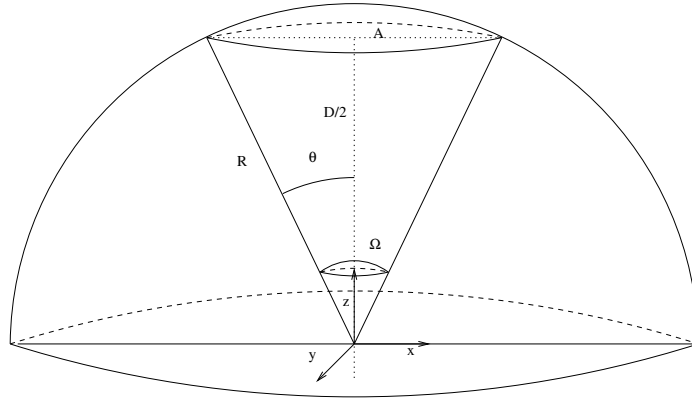


FIG. 4.4 – Géométrie équivalente pour l'angle solide de détection.

Ceci permet le calcul de l'angle solide de détection, en sphérique :

$$\Omega = \int_{\vartheta'=0}^{\vartheta} \int_{\phi=0}^{2\pi} d\Omega' = 2\pi (1 - \cos\vartheta) \quad (4.3)$$

Le rapport de l'angle solide de détection par l'angle solide total de $\Omega_{Tot} = 2\pi$ donne le rapport entre le nombre d'événements vrais et le nombre d'événements total :

$$\frac{\Omega}{\Omega_{Tot}} \approx 1.3 \cdot 10^{-2} \quad (4.4)$$

soit pour 1565 évènements, 20 évènements vrais seulement.

Le traitement des données de l'expérience nous a permis d'identifier *quatre* évènements vrais. Le moyen de sélection sera discuté plus loin.

4.2.2 Mesures

Dans cette partie, nous expliciterons les différentes mesures qui sont nécessaires pour calculer la masse du muon.

4.2.2.1 Calcul des temps de détections

On définit le *temps de détection* t_i comme le temps où un muon passe par une raquette. A chaque fois qu'un muon passe par une raquette, il dépose de l'énergie sous forme électromagnétique. Cette lumière est recueillie par le photomultiplicateur, générant un signal sur l'oscilloscope illustré par la figure 4.5.

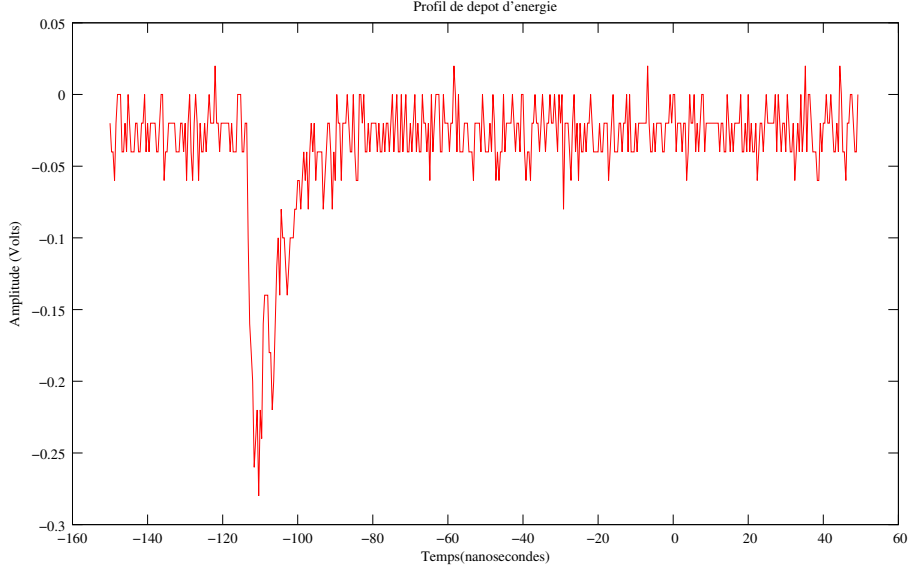


FIG. 4.5 – Exemple de signal laissé par un muon dans le bloque de scintillateur.

On estime tout d'abord le temps moyen \bar{t}_i , en considérant le signal comme une distribution statistique discrète $V(t_n)$:

$$\bar{t}_i = \frac{\sum_n t_n V(t_n)}{\sum_n V(t_n)} \quad (4.5)$$

la sommation sur n étant prise sur un interval symétrique par rapport au minimum du signal. On estime ensuite l'erreur Δt_{mes} , que l'on qualifera d'*erreur de mesure directe*, relative à \bar{t}_i . Nous avons choisit

$$\Delta t_{mes} = 7 \cdot 10^{-10} \cdot s \quad (4.6)$$

correspondant à cinq fois la periode d'échantillonnage de l'oscilloscope.

4.2.2.2 Calcul des vitesses

Les vitesses, notées β_i , se calculent comme

$$\beta_i = \frac{x_i}{c(t_{i+1} - t_i)} \quad (4.7)$$

où les x_i représentent la distance entre les raquettes i et $i + 1$. L'erreur associée à la vitesse se calcule comme

$$\Delta\beta_i = \beta_i \left(\frac{\Delta x_i}{x_i} + \frac{\Delta t_{i+1} + \Delta t_i}{t_{i+1} - t_i} \right) \quad (4.8)$$

Nous calculerons trois vitesses pour chaque évènement vrai pour des raisons que nous évoquerons par la suite.

x_i correspond à la distance entre deux raquettes. Ce n'est pas la longueur d'air traversée par la particule. En effet, comme il est impossible de connaître la trajectoire du muon incident, on considère que *toutes les particules ont une trajectoire verticale*, et on calcule la déviation maximum entre une trajectoire verticale et une trajectoire faisant un angle Θ avec l'axe des z . Sachant que la trajectoire limite détectable par notre système correspond à la diagonale du parallélépipède Λ de la figure 4.3, l'angle Θ est l'angle que fait Λ avec la verticale, calculé à partir de la figure 4.6.

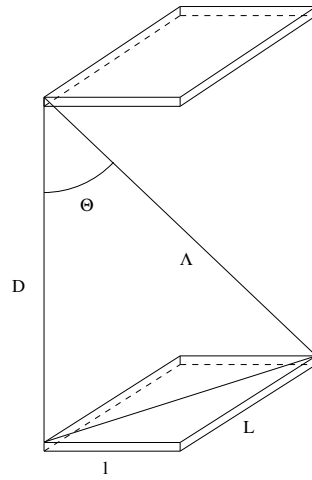


FIG. 4.6 – Définition de l'angle Θ . Le schéma n'est pas à l'échelle. Λ est la diagonale du parallélépipède.

Il est défini comme

$$\Theta = \arctan \frac{\sqrt{L^2 + l^2}}{D} \quad (4.9)$$

L'erreur Δx_i commise sur la distance entre deux raquettes aura donc pour expression

$$\Delta x_i = \Delta x_{dev} + \Delta x_{mes} \quad (4.10)$$

où Δx_{mes} est l'erreur lié à la mesure directe, et Δx_{dev} est le terme que l'on qualifera de *déviaton*, valant

$$\Delta x_{dev} = x_i \left(\frac{1 - \cos\Theta}{\cos\Theta} \right) \quad (4.11)$$

Les valeurs numériques donnent

$$\Theta = 0.212 \cdot rad \quad (4.12)$$

$$\Delta x_i = 2.29 \cdot 10^{-2} x_i + \Delta x_{mes} \quad (4.13)$$

La source d'erreur liée à la mesure directe est de l'ordre du demi centimètre, donc une distance $x_i = 21 \cdot cm$ suffit pour que le terme de déviaton soit égal au terme de mesure directe.

On peut remarquer que si l'angle solide de collection génère une incertitude sur la distance parcourue entre les deux raquettes, il en est de même pour la différence de temps. En effet, pour deux raquettes séparées d'une distance x , cela signifie que le point de dépôt de l'énergie sur chaque raquette ne sera pas identique, mais décalé d'une distance $x \cdot \tan\Theta$ par rapport à la verticale, introduisant une incertitude Δt_{dev} sur le temps de passage

$$\Delta t_{dev} = \frac{n}{c} \cdot x \cdot \tan\Theta \quad (4.14)$$

où n est l'indice du scintillateur et c la vitesse de la lumière. Pour un indice optique de $n = 1,5$, l'erreur associée pour la première raquette a pour valeur

$$\Delta t_{dev} = 8.6 \cdot 10^{-2} \cdot ns \quad (4.15)$$

On pourrai se poser la question de la validité de prendre en compte Δt_{dev} dans le calcul des incertitudes, sachant que la longueur parcourue par la particule entre les deux raquettes fait également intervenir une incertitude Δx_{dev} , liée à l'angle solide. Mais comme les mesures de la différence de temps reste *indépendante* de la mesure de la distance parcourue par la particule, dans le sens où l'oscilloscope ne mesure pas de distance, alors Δt_{dev} est indépendant de Δx_{dev} . L'angle solide de détection intervient donc dans deux source d'erreur différentes.

4.2.2.3 Energie perdue

Pour un événement vrai, le muon passe par une épaisseur e de plombs, qui va considérablement ralentir la particule. On peut calculer l'énergie perdue via la formule de Bethe, qui traduit l'énergie perdue par unité de longueur dans la matière

$$-\frac{dE}{dx} = \frac{4\pi}{m_e c^2} \cdot \frac{Z}{A} \cdot \frac{\rho \mathcal{N}_a}{\beta^2} \cdot \left(\frac{e^2}{4\pi\epsilon_0} \right)^2 \cdot \left[\ln \left(\frac{2m_e c^2 \beta^2 \gamma^2}{I} \right) - \beta^2 \right] \quad (4.16)$$

où

- $[m_e]$ Masse de l'électron
- $[c]$ Vitesse de la lumière
- $[e]$ Charge électrique
- $[I]$ Potentiel moyen d'excitation
- $[Z]$ Nombre de charge de la cible
- $[A]$ Masse molaire de la cible
- $[\rho]$ Masse volumique de la cible
- $[\mathcal{N}_a]$ Nombre d'Avogadro
- $[\epsilon_0]$ Permittivité du vide
- $[\gamma = \frac{1}{\sqrt{1-\beta^2}}]$ Facteur de Lorentz

Si on trace la perte d'énergie en fonction de $\beta\gamma$, on obtient la courbe bien connue de la figure 4.7.

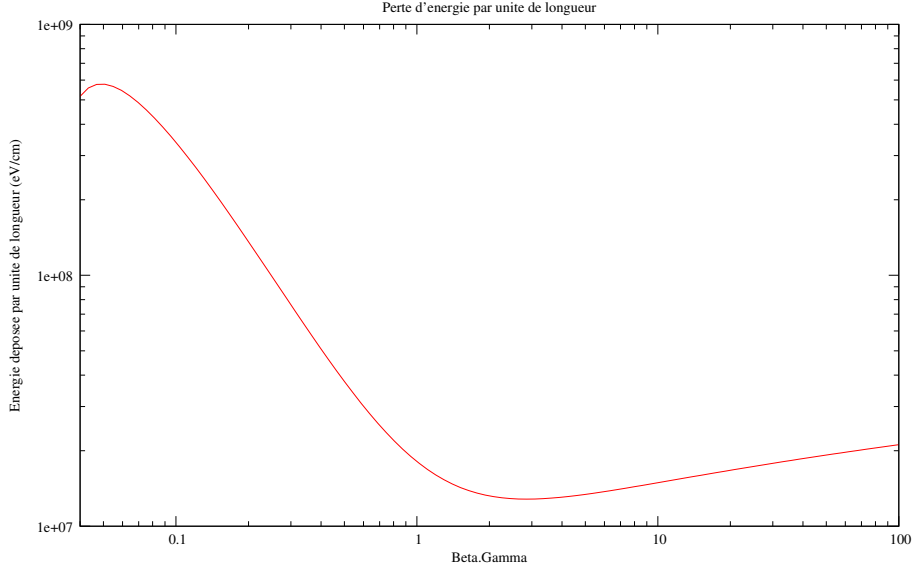


FIG. 4.7 – Energie en fonction de $\beta\gamma$.

4.2.3 Protocole expérimental

On considère un muon de vitesse β_1 issu de la haute atmosphère par un processus physique quelconque. On commence par mesurer la vitesse de la particule à l'aide des deux premières raquettes, séparées d'une distance d . Ce calcul fait intervenir d , ainsi que les temps de détection t_1 et t_2 de chacune des raquettes :

$$\beta_1 = \frac{d}{c(t_2 - t_1)} \quad (4.17)$$

On mesure ensuite la vitesse β_3 entre les deux dernières raquettes

$$\beta_3 = \frac{D - d - d'}{c(t_4 - t_3)} \quad (4.18)$$

On estime alors l'énergie perdue dE dans le plomb d'épaisseur e via la formule de Bethe

$$dE = -\frac{dE}{dx} \cdot e \quad (4.19)$$

L'erreur associée à dE , ΔdE , est calculée par

$$\Delta dE = dE \frac{\Delta e}{e} + 2 \frac{\Delta \beta_1}{\beta_1} \left(dE + \alpha e \left(\frac{\gamma_1^2}{\beta_1^2} - 1 \right) \right) \quad (4.20)$$

où

$$\alpha = \frac{4\pi}{m_e c^2} \cdot \frac{Z}{A} \cdot \rho \mathcal{N}_a \cdot \left(\frac{e^2}{4\pi\epsilon_0} \right)^2 \quad (4.21)$$

La masse de la particule peut alors être déterminée en égalant la variation de l'énergie cinétique de la particule avec l'énergie dissipée dans la matière

$$dE = (\gamma_1 - \gamma_2) m_\mu c^2 \quad (4.22)$$

soit

$$m_\mu c^2 = \frac{dE}{(\gamma_1 - \gamma_2)} \quad (4.23)$$

avec une erreur associée

$$\frac{\Delta m_\mu c^2}{m_\mu c^2} = \frac{\Delta dE}{dE} + \frac{\Delta \gamma_1 + \Delta \gamma_2}{(\gamma_1 - \gamma_2)} = \frac{\Delta dE}{dE} + \frac{\gamma_1^3 \beta_1 \Delta \beta_1 + \gamma_2^3 \beta_2 \Delta \beta_2}{(\gamma_1 - \gamma_2)} \quad (4.24)$$

Dans ce qui précède, on considère une perte d'énergie constante dans toute l'épaisseur du plomb. Ceci constitue une approximation valide si la vitesse de la particule incidente est telle que $\beta\gamma > 3$. Par contre, pour les événements qui ne satisfont pas cette condition, cette approximation n'est plus valable, car la particule perd de plus en plus d'énergie au fur et à mesure que sa vitesse diminue, comme le montre la figure 4.7. On doit donc implémenter dans le code une fonction qui prendra en compte la variation de vitesse. Pour se faire, on se propose de partir de l'énergie du muon

$$E = \gamma m_\mu c^2 = \gamma E_0 \quad (4.25)$$

avec E_0 l'énergie de masse. En *différenciant*, il vient

$$dE = E_0 d\gamma = E_0 \beta \gamma^3 d\beta \quad (4.26)$$

En injectant (4.26) dans (4.16), avec (4.21), on obtient

$$-\frac{d\beta}{dx} = \frac{\alpha}{E_0} \cdot \frac{\ln [2m_e c^2 \gamma^2 \beta^2 / I] - \beta^2}{\beta^3 \gamma^3} \quad (4.27)$$

Après intégration, on a

$$E_0 = \frac{\alpha e}{\int_{\beta_2}^{\beta_1} d\beta \cdot f(\beta)} \quad (4.28)$$

où

$$f(\beta) = \frac{\beta^3 \gamma^3}{\ln [2m_e c^2 \gamma^2 \beta^2 / I] - \beta^2} \quad (4.29)$$

Le dénominateur de (4.28) correspond à l'aire de la fonction f entre β_1 et β_2 . f est tracé sur la figure 4.2.3

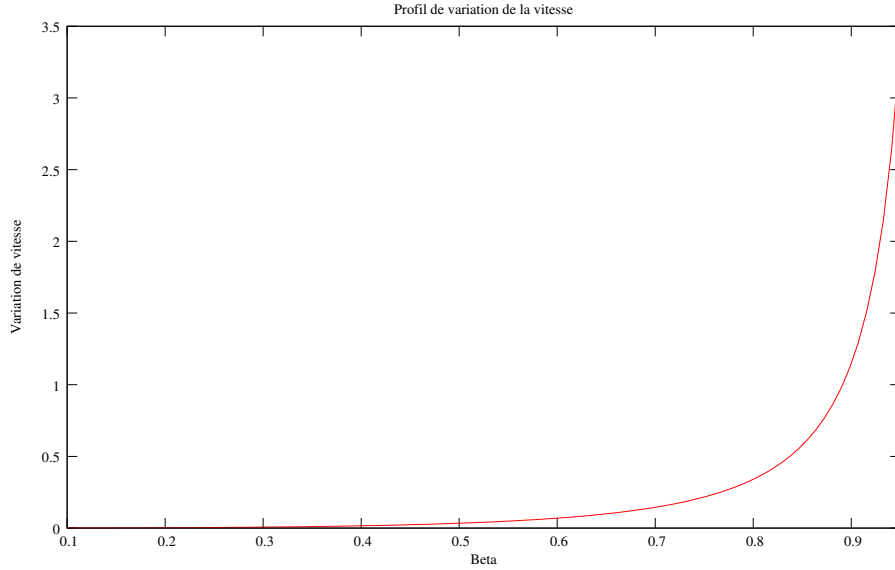


FIG. 4.8 – Variation de la vitesse en fonction de β .

Cependant, un des résultats sera que l'incertitude sur les vitesses $\Delta\beta$ sera de l'ordre de β . Ce qui signifie que l'aire peut varier énormément. A cause de la croissance exponentielle de (4.29) vers $\beta = 1$, on se limitera à une valeur de $\beta = 0.95$ pour le calcul de l'aire. L'incertitude sur l'aire peut donc être estimée à trois fois cette aire. Dans ce cas, l'incertitude sur la masse devient

$$\Delta E_0 = E_0 \left(\frac{\Delta e}{e} + \frac{\Delta F}{F} \right) \quad (4.30)$$

où F est le dénominateur de l'équation (4.28).

Pour sélectionner les événements vrais, on procède en suivant deux étapes :

- [1.] On commence par sélectionner les événements qui ont un signal mesurable sur les raquettes 2 et 3
- [2.] Puis on sélectionne les événements satisfaisant $0 < \beta_2 < \beta_{12} < \beta_1 < 1$

La vitesse β_{12} est la vitesse intermédiaire entre les raquettes 2 et 3. L'ensemble des résultats seront exposés dans le chapitre 6.

Chapitre 5

Résultats de l'étude

Après avoir appliqué nos règles de sélections, on réalise trois séries de données :

1. pour $t_i = \bar{t}_i$, défini par l'équation (4.5), on obtient au final seulement 4 évènements vrais sur les 1565 détections. Les résultats sont regroupés dans le tableau 5.1, et les signaux sont tracés dans la première partie de l'annexe, de la figure (7.1) à (7.4).

évènement	$m_\mu \cdot MeV$	$\Delta m \cdot (MeV)$	γ_1	$\Delta\gamma_1$	γ_2	$\Delta\gamma_2$
EVT1122	83.5805	453.491	2.03576	3.9223	1.06804	0.04581
EVT174	92.0537	564.101	2.04777	4.01406	1.17174	0.19124
EVT2	140.969	1678.54	2.11709	4.56840	1.55406	1.38167
EVT916	49.4624	426.759	2.66651	10.5944	1.17374	0.19480

Table 5.1: Tableau récapitulatif des résultats pour $t_i = \bar{t}_i$

2. pour $t_i = t_{min}$, on obtient 3 évènements vrais, regroupés dans le tableau 5.2, correspondant aux figures (7.5) à (7.7) de l'annexe

évènement	$m_\mu \cdot (MeV)$	$\Delta m \cdot (MeV)$	γ_1	$\Delta\gamma_1$	γ_2	$\Delta\gamma_2$
EVT654	71.7304	578.896	2.22661	6.79126	1.14308	0.17517
EVT683	67.8670	518.169	2.22661	6.79126	1.08140	0.07314
EVT954	230.976	1412.08	1.60374	1.70930	1.18176	0.25619

Table 5.2: Tableau récapitulatif des résultats pour $t_i = t_{min}$

3. enfin, en supposant tout les évènements précédent comme vrais, on calcul la masse en prenant en compte la variation de la vitesse dans le plomb, ce qui donne les valeurs répertoriées dans le tableau 5.3

évènement	$m_\mu \cdot (MeV)$	$\Delta m \cdot (MeV)$	γ_1	$\Delta\gamma_1$	γ_2	$\Delta\gamma_2$
EVT1122	116.692	351.79	2.03576	3.9223	1.06804	0.04581
EVT174	120.759	364.1398	2.04777	4.01406	1.17174	0.19124
EVT2	164.431	495.8182	2.11709	4.56840	1.55406	1.38167
EVT916	64.2382	193.7011	2.66651	10.5944	1.17374	0.19480
EVT654	95.3010	287.3665	2.22661	6.79126	1.14308	0.17517
EVT683	93.1389	280.8470	2.22661	6.79126	1.08140	0.07314
EVT954	291.720	879.6401	1.60374	1.70930	1.18176	0.25619

Table 5.3: Tableau récapitulatif des résultats provenant des 2 méthodes de calcul de temps en calculant la perte d'énergie en tenant compte de la variation de la vitesse dans le plomb.

Dans le premier cas, on obtient une masse moyenne pour le muon de

$$\langle m_\mu \rangle = 92 \pm 780 \cdot MeV$$

où l'incertitude associée a été prise comme la moyenne des incertitudes du tableau (6.1).

Si maintenant nous prenons simplement $t_i = t_{min}$, la masse moyenne pour le muon est de

$$\langle m_\mu \rangle = 124 \pm 836 \cdot MeV$$

L'hypothèse faite sur le temps de l'évènement est beaucoup plus discutable que précédemment.

On admet maintenant que tous les évènements sélectionnés par les deux méthodes précédentes sont des évènements vrais. La masse moyenne pour le muon devient

$$\langle m_\mu \rangle = 105 \pm 804 \cdot MeV$$

En considérant maintenant tout les évènements comme vrai, on fait la mesure de la masse par la méthode de variation de la vitesse, et on obtient une masse moyenne pour le muon de

$$\langle m_\mu \rangle = 135.18 \pm 408 \cdot MeV$$

La source d'incertitude sur la dernière valeur est plus faible car la propagation de l'incertitude exprimée par l'équation (4.24) fait intervenir les facteur de Lorentz au cube, ce qui fait grandir énormément l'incertitude. Nous avons aussi tronquée la fonction f définie par (4.29) pour éviter que l'intégrale ne diverge, et que l'erreur sur l'aire ne deviennent infinie.

Ces valeurs semblent être en accord avec les mesure de la littérature, qui fixe la valeur de la masse du muon à $m_\mu c^2 = 105 \cdot MeV$.

Chapitre 6

Conclusion

La première partie du travail nous a permis de travailler avec les photomultiplicateurs, et de comprendre leur fonctionnement. Nous avons donc pu mettre en évidence le domaine de linéarité du photomultiplicateur avec la haute tension, et avons retrouvé la loi de Beer-Lamber, traduisant l'absorption du rayonnement électromagnétique dans une épaisseur d'aire.

Dans la seconde partie du tp, nous avons obtenu des résultats satisfaisants, malgré les incertitudes très élevées.

Les principaux problèmes, relatifs à la faible statistique, sont nos critères de sélection. En effet, aux vues de la faible qualité des signaux, les différentes mesures de temps restent très sélective quand à la nature de l'évènement. Ceci nous amène sûrement à écarter des évènements vrais parmi les 1500 détections de la semaine.

Pour augmenter le nombre d'évènements, nous aurions eu besoin de plus de temps pour l'expérience, car en une semaine d'acquisition, nous n'avons eu que sept évènements vrais au total. Augmenter le temps d'acquisition nous aurai permis de répartir les mesures sur un graphique, pour étudier la forme de la distribution en masse.

Nous pourrions modifier le protocole de mesure en optimisant le flux de muons. En diminuant l'écart entre les raquettes, nous augmenterions l'incertitude liée à l'angle solide de collection. Mais en superposant les raquettes de façon à diminuer le temps de vol tout en conservant l'angle solide constant, nous pourrions augmenter le nombre d'évènements vrais tout en conservant des incertitudes raisonnables. Un système où les raquettes sont disposées en croix les unes au dessus des autres pourrait satisfaire ces conditions.

Enfin, le plomb constitue une source très efficace d'atténuation. En diminuant son épaisseur, le nombre de muons s'arrêtant dans l'épaisseur devrait diminuer, augmentant ainsi la statistique.

Chapitre 7

Annexes

7.1 Figures

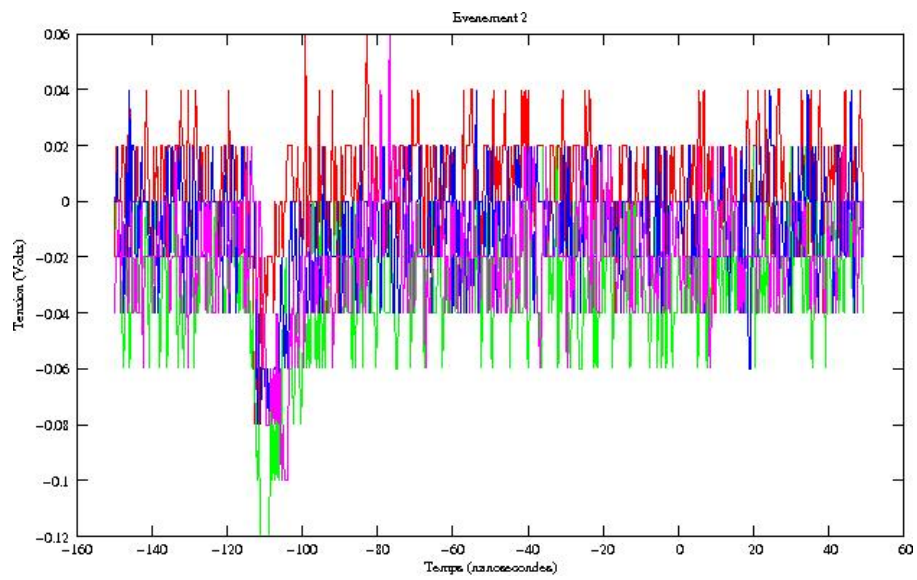


Figure 7.1: Second évènement, sélectionné par le calcul du temps moyen \bar{t}_i

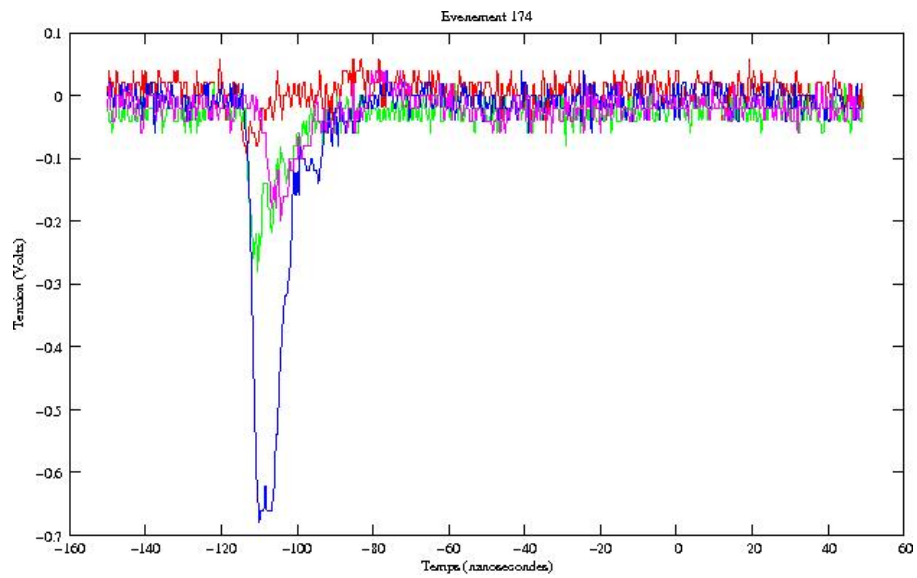


Figure 7.2: Evènement 174, sélectionné par le calcul du temps moyen \bar{t}_i

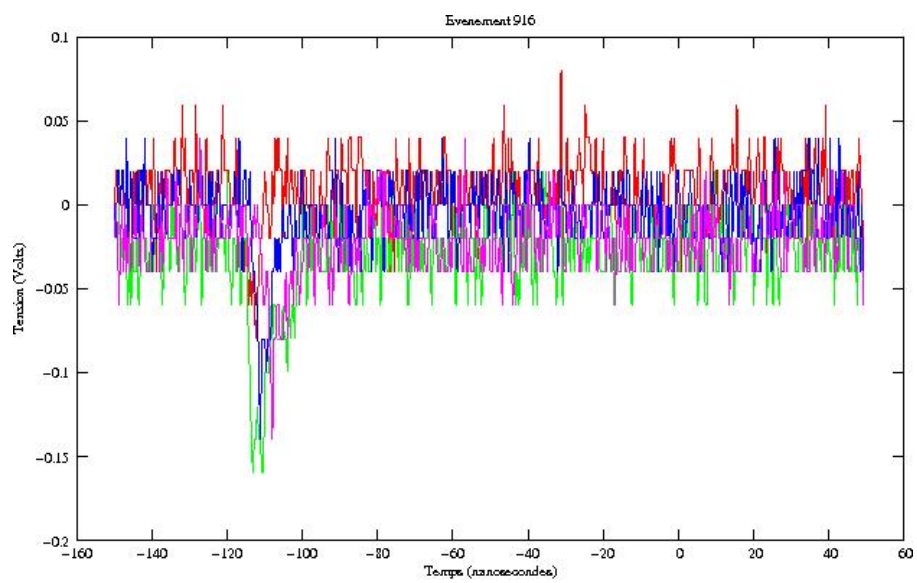


Figure 7.3: Evènement 916, sélectionné par le calcul du temps moyen \bar{t}_i

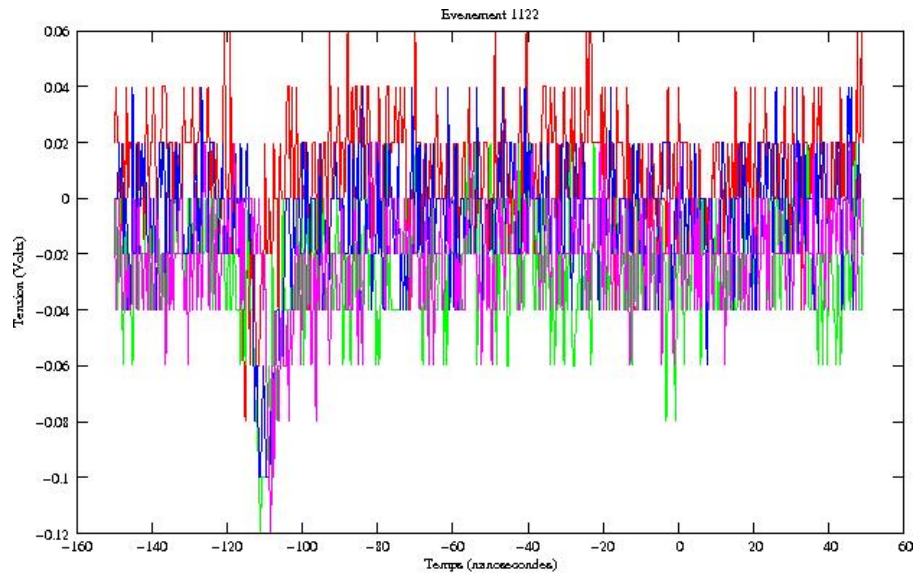


Figure 7.4: Evènement 1122, sélectionné par le calcul du temps moyen \bar{t}_i

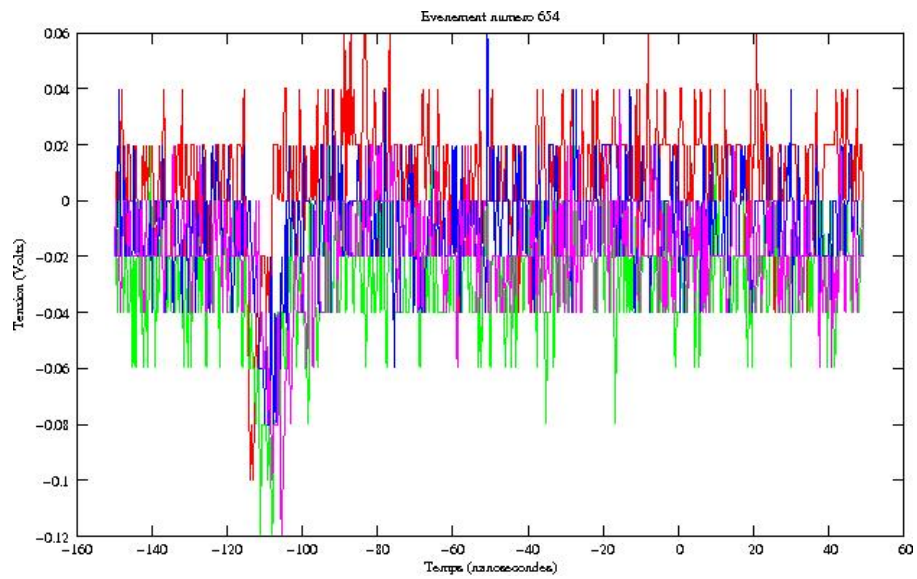


Figure 7.5: Evènement 654, sélectionné par la détection du minimum de chaque signal t_{min}

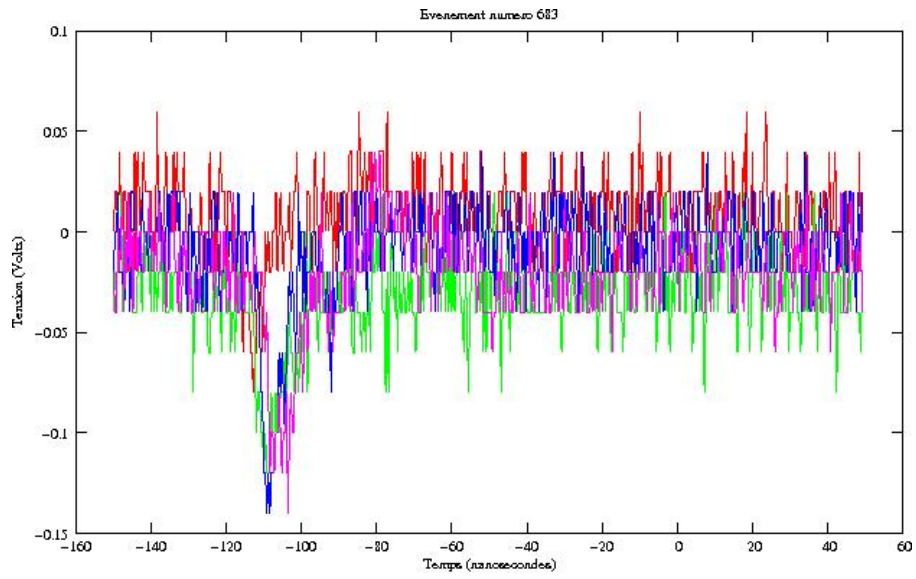


Figure 7.6: Evènement 683, sélectionné par la détection du minimum de chaque signal t_{min}

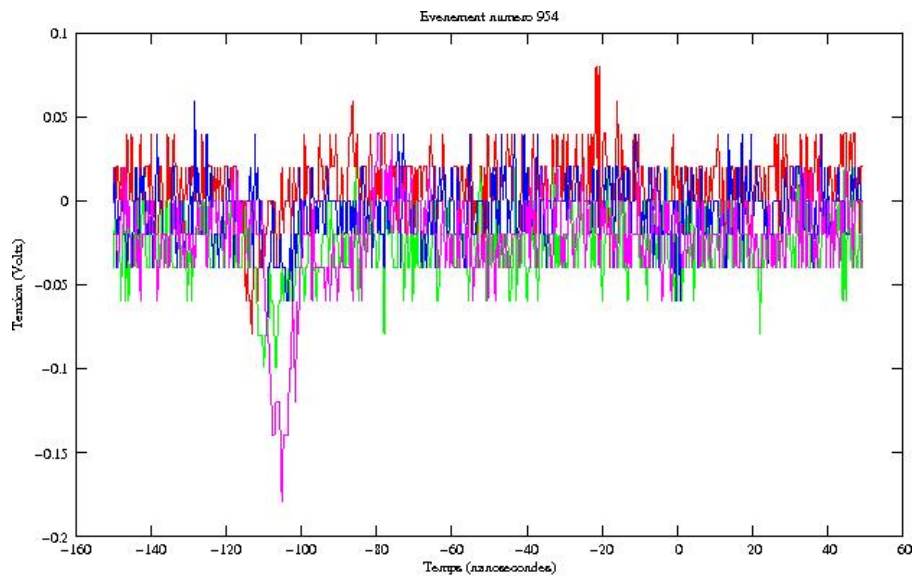


Figure 7.7: Evènement 954, sélectionné par la détection du minimum de chaque signal t_{min}

7.2 Programme d'analyse des données du PM etalonnage.C


```
////////////////////////////////////  
PROGRAMME DE TRAITEMENT DE DONNEES: TP DETECTION DES MUONS////////////////////////////////////  
ALEXANDRE FAURE - JANVIER 2011////////////////////////////////////  
////////////////////////////////////
```

```
// Bibliothèques de bases
```

```
#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>  
#include <iostream>  
#include <string.h>  
#include <errno.h>  
#include <dirent.h>
```

```
int main(int argc, char *argv[])  
{
```

```
    int const max_length=100;  
    int const ECH = 500; // Echantillonnage des programmes  
    int line; // Chaîne vide de taille max_length  
    int line2; // Chaîne vide de taille max_length  
    char str0[max_length];  
    char str0b[max_length];  
    char str0c[max_length];  
    char str0d[max_length];  
    char str0e[max_length];  
    char str12[max_length];  
    char str23[max_length];  
    char str34[max_length];  
    char str1[max_length];  
    char str2[max_length];  
    char str3[max_length];  
    char str4[max_length];  
    char strp[1];  
    char fl[max_length];
```

```
    system("clear"); // On efface l'écran à chaque lancement du programme
```

```
    // POUR UN RUN : indiquer le nombre de fichiers de sorties
```

```
    char filename[100];  
    char filedir[100];
```

```
    char numerof[100];  
    int max_files;  
    int num_file;  
    int num_run;  
    int n;
```

```
    float min1=0;  
    float min2=0;  
    float min3=0;  
    float min4=0;
```

```
// Déclaration des tableaux
```

```
float chan1[ECH];  
float chan2[ECH];  
float chan3[ECH];  
float chan4[ECH];
```

```
printf("TP MUONS M2R CCP DATA ANALYSIS \n v.1.0 - 2011 \n Alexandre Faure\n\n\n");
```

```
printf("Numéro du Run : ");  
scanf("%d", &num_run);
```

```
// On liste automatiquement le nombre de fichiers contenus dans le repertoire
```

```
struct dirent **namelist;  
sprintf(filedir,"datas/run_%d",num_run);  
n=scandir(filedir,&namelist, 0, alphasort)-2;
```

```
    if(n<=0)
```

```

{
    printf("Pas de fichiers detectes. Arret du programme.\n\n");
    return 0;
}

printf("%d fichiers dans run %d.\n\n", n, num_run);

float tresh=0;
printf("Entrez la valeur du seuil signe pour Chan1(V) : ");
scanf("%f",&tresh);

/////////Initialisation tableaux pour un run (stat) dynamique
float *min_chan1;
float *min_chan2;
float *min_chan3;
float *min_chan4;
min_chan1 = (float*) malloc(n * sizeof(float)); // La fonction malloc permet l'allocation dynamique
min_chan2 = (float*) malloc(n * sizeof(float));
min_chan3 = (float*) malloc(n * sizeof(float));
min_chan4 = (float*) malloc(n * sizeof(float));

float *int_tab_chan1;
int_tab_chan1 = (float*) malloc(n * sizeof(float));

float *deltat_tab_chan1;
deltat_tab_chan1 = (float*) malloc(n * sizeof(float));

printf("//////////PARAMETRES MESURES \n\n");

FILE *file;
sprintf(filename,"datas/run_%d/%dEVT.txt",num_run, 1); // Numérote les numéros de fichiers .txt

//////////REPLACEMENT VIRGULE//////////
file = fopen(filename, "r+");
// On s'assure que le fichier n'est pas vide
if (file == NULL)
{
    printf("Impossible d'ouvrir le fichier");
}

// Remplacement des virgules par des points
int virg;
while ((virg=fgetc(file))!=EOF)
{
    if (virg == ',')
    {
        virg='.';
        fseek(file, -1, SEEK_CUR); // On remet le curseur au niveau de la virgule
        fputc(virg, file);
    }
}
fclose(file);
//////////REPLACEMENT VIRGULE FIN

////////// Lecture des donnees de chaque fichiers de RUN
file = fopen(filename, "r+");
for(line=1;line<=1; line++)
{
    fgets(str0, max_length, file);
}

for(line=2;line<=2; line++)
{
    fgets(str0b, max_length, file);
    printf("Mesure le %s\n",str0b); // Donne la date et heure de premiere mesure
}

for(line=3;line<=3; line++)
{

```

```

    fgets(str0c, max_length, file);
    printf("Trigger : %s\n",str0c); // Donne les valeurs du trigger H et V
}

for(line=4;line<=11; line++)
{
    fgets(str0d, max_length, file);
}

double deltat;
for(line=12;line<=12; line++)
{
    fgets(str0e, max_length, file);
    printf("Écart entre deux mesures : %s\n\n",str0e);
    deltat=strtold(str0e,NULL);
}
fclose(file);

//printf("Ecart mesures : %E\n\n",deltat); /// DELTAT = b-a

//////////BOUCLE SUR TOUS LES FICHIERS D'UN RUN//////////
for (num_file=1; num_file<=n;num_file++)
{
    sprintf(filename,"datas/run_%d/%dEVT.txt",num_run, num_file); // Numérote les numéros de
fichiers .txt

    ////////////OUVERTURE D'UN FICHIER .TXT//////////

    //FILE *file;
    ////////////REPLACEMENT VIRGULE
    file = fopen(filename, "r+");

    // On s'assure que le fichier n'est pas vide
    if (file == NULL)
    {
        printf("Impossible d'ouvrir le fichier");
    }

    // Remplacement des virgules par des points
    int virg;
    while ((virg=fgetc(file))!=EOF)
    {
        if (virg == ',')
        {
            virg='.';
            fseek(file, -1, SEEK_CUR); // On remet le curseur au niveau de la virgule
            fputc(virg, file);
        }
    }
    fclose(file);
    ////////////REPLACEMENT VIRGULE FIN

    line=0;
    line2=0;
    int c1=0;
    int c2=0;
    int c3=0;
    int c4=0;
    int c5=0;
    int c6=0;
    int c7=0;

    file = fopen(filename, "r+");

    for(line=1;line<=15; line++) //On ne prend pas en compte les 15 premières lignes du txt
    {
        fgets(str0, max_length, file);
    }
    int count1=0;

```

```

for(line=16;line<=(ECH-1)+16; line++) // On affiche les V pour le CHANNEL 1
{
    count1++;
    fgets(str1, max_length, file);
    chan1[count1]=strtof(str1,NULL); // Conversion String to num
}

c1=line;

for(line=c1;line<=c1+1; line++) // On ne prend pas en compte les 2 lignes de separation
{
    fgets(str12, max_length, file);
}

c2=line;

    int count2=0;
for(line=c2;line<=(ECH-1)+c2; line++) // On affiche les V pour le CHANNEL 2
{
    count2++;
    fgets(str2, max_length, file);
    chan2[count2]=strtof(str2,NULL); //Conversion String to num
}

c3=line;

for(line=c3;line<=c3+1; line++) // On ne prend pas en compte les 2 lignes de separation
{
    fgets(str23, max_length, file);
}

c4=line;

int count3=0;
for(line=c4;line<=(ECH-1)+c4; line++) //On affiche les V pour le CHANNEL 3

{
    count3++;
    fgets(str3, max_length, file);
    chan3[count3]=strtof(str3,NULL); //Conversion String to num
}

c5=line;

for(line=c5;line<=c5+1; line++) // On ne prend pas en compte les 2 lignes de separation
{
    fgets(str34, max_length, file);
}

c6=line;

//printf("//////////CHANNEL 4 A LIGNE %d\n",line);
int count4=0;

for(line=c6;line<=(ECH-1)+c6; line++) // On affiche les V pour le CHANNEL 4
{
    count4++;
    fgets(str4, max_length, file);
    chan4[count4]=strtof(str4,NULL); //Conversion String to num
}

fclose(file); // On ferme notre fichier
////////////////////////////////////

// 4 tableaux chan1,chan2,chan3,chan4
//////////////////////////////////// VALEUR MINIMUM CHANNELS////////////////////////////////////

//////////CHANNEL 1
// On balaye tous les éléments du tableau

min1=chan1[0]; // On initialise une valeur de départ
for (int i=1; i<=ECH;i++)
{
    if(chan1[i]<min1) // On compare les valeurs du tableau par rapport à min1

```

```

    {
        min1=chan1[i];
    }
} // boucle for
min_chan1[num_file]=min1; // On écrit la valeur dans le tableau stat.
////////// ATTENTION A L'ORDRE D'AFFECTATION
// printf("Valeur minimum pour Channel 1 : %f\n\n",min1);

//////////CHANNEL 2
// On balaye tous les éléments du tableau

min2=chan2[0]; // On initialise une valeur de départ
for (int i=1; i<=ECH;i++)
{
    if(chan2[i]<min2) // On compare les valeurs du tableau par rapport à min1
    {
        min2=chan2[i];
    }
} // boucle for
min_chan2[num_file]=min2; // On écrit la valeur dans le tableau stat.
// printf("Valeur minimum pour Channel 2 : %f\n\n",min2);

//////////CHANNEL 3
// On balaye tous les éléments du tableau

min3=chan3[0]; // On initialise une valeur de départ
for (int i=1; i<=ECH;i++)
{
    if(chan3[i]<min3) // On compare les valeurs du tableau par rapport à min1
    {
        min3=chan3[i];
    }
} // boucle for
min_chan3[num_file]=min3; // On écrit la valeur dans le tableau stat.
// printf("Valeur minimum pour Channel 3 : %f\n\n",min3);

//////////CHANNEL 4
// On balaye tous les éléments du tableau

min4=chan4[0]; // On initialise une valeur de départ
for (int i=1; i<=ECH;i++)
{
    if(chan4[i]<min4) // On compare les valeurs du tableau par rapport à min1
    {
        min4=chan4[i];
    }
} // boucle for
min_chan4[num_file]=min4; // On écrit la valeur dans le tableau stat.
// printf("Valeur minimum pour Channel 4 : %f\n\n",min4);

////////// VALEURS MINIMUM CHAQUE FILE
FIN//////////

////////// CALCUL INTEGRAL DEBUT//////////
///// Intervalle entre deux points = deltat en début de programme
///// PRECISION : variable en format double pour les nombres petits
/////CHANNEL 1 (valeurs négatives pour PM)

double int_chan1=0;
int m=0;
for (int i=1;i<=ECH;i++) // On balaye les valeurs du tableau
{
    if (chan1[i]<=tresh) // Calcul d'integrale pour un certain seuil (SOUS LE
SEUIL CAR NEGATIF)
    {
        int_chan1=int_chan1+deltat*fabs(chan1[i]); // Bien prendre la valeur
absolue (negatifs)
    }
}
//printf("\nIntegrale du pic Chan 1 : %E\n\n", int_chan1);
int_tab_chan1[num_file]=int_chan1;
////////// CALCUL INTEGRAL CHAQUE FICHER
FIN//////////

```

```

////////// CALCUL DELTAT POUR CHAQUE FICHER DEBUT //////////
// CHANNEL 1 : RENOMMEZ LES VARIABLES
double deltaT=0;
double deltaT0=0;
double deltaT1=0;
double deltaT2=0;
int c=0;
for (int i=1;i<=ECH;i++) // On balaye les valeurs du tableau
{
    deltaT0=deltaT0+deltat;
    c++;
    if (chan1[i]>=tresh) // Calcul d'integrale pour un certain seuil
    {
        deltaT1=deltaT0;
        //printf("Int1 %E\n\n", deltaT1);
        break;
    }
}

min
for (int i=c;i<=ECH;i++) // On balaye les valeurs du tableau après le c de la valeur
{
    deltaT0=deltaT0+deltat;

    if (chan1[i]<=tresh) // Calcul d'integrale pour un certain seuil
    {
        deltaT2=deltaT0;
        //printf("Int2 %E\n\n", deltaT2);
        break;
    }
}
deltaT=deltaT2-deltaT1; // Calcul de l'intervalle deltaT
//printf("Intervalle deltat %E\n\n", deltaT);
////////// CALCUL DELTAT POUR CHAQUE FICHER
FIN//////////

deltat_tab_chan1[num_file]=deltaT;

} //boucle sur num_file

// Sortie actuelle : min_chan1,min_chan2,min_chan3,min_chan4 = tableau de tous les minimums, pour
chaque fichier, associé à un channel donné.

// Verification des tableaux
// int i=0;
// for(i=1;i<=n;i++)
// {
//     printf("%f\n\n",chan4[i]);
// }

////////// TRAITEMENT STATISTIQUE MOYEN//////////

///// 1) MOYENNE DES MINIMUMS => Valeur moyenne de l'amplitude pour chaque Channel

printf("//////////CALCUL STAT DES EXTREMA\n\n");

/////CHANNEL 1
float sum_min_chan1=0;
float moy_min_chan1=0;
for (int i=1;i<=n-1;i++)
{
    sum_min_chan1=sum_min_chan1+min_chan1[i];
}
moy_min_chan1=sum_min_chan1/n;
printf("Moyenne statistique de l'amplitude minimum Chan 1 : %f\n\n", moy_min_chan1);

// /////CHANNEL 2
float sum_min_chan2=0;
float moy_min_chan2=0;
for (int i=1;i<=n-1;i++)
{

```

```

    sum_min_chan2=sum_min_chan2+min_chan2[i];
}
moy_min_chan2=sum_min_chan2/n;
printf("Moyenne statistique de l'amplitude minimum Chan 2 : %f\n\n", moy_min_chan2);

//      /////CHANNEL 3
float sum_min_chan3=0;
float moy_min_chan3=0;
for (int i=1;i<=n-1;i++)
{
    sum_min_chan3=sum_min_chan3+min_chan3[i];
}
moy_min_chan3=sum_min_chan3/n;
printf("Moyenne statistique de l'amplitude minimum Chan 3 : %f\n\n", moy_min_chan3);

//      /////CHANNEL 4
float sum_min_chan4=0;
float moy_min_chan4=0;
for (int i=1;i<=n-1;i++)
{
    sum_min_chan4=sum_min_chan4+min_chan4[i];
}
moy_min_chan4=sum_min_chan4/n;
printf("Moyenne statistique de l'amplitude minimum Chan 4 : %e\n\n", moy_min_chan4);

        ///// INCLURE LES PARTIES 2,3 et 4 DANS BOUCLE FILE NUM

///// 2) CALCUL DES INTEGRALES DE PIC --> Pour chaque channel à partir d'une valeur seuil
printf("///// CALCUL STAT DES INTEGRALES DE PIC\n\n");
/////CHANNEL 1
double sum_int_tab_chan1=0;
double moy_int_tab_chan1=0;
for (int i=1;i<=n-1;i++)
{
    sum_int_tab_chan1=sum_int_tab_chan1+int_tab_chan1[i];
}
moy_int_tab_chan1=sum_int_tab_chan1/n;
printf("Moyenne statistique de l'intégrale de pic Chan 1 : %e\n\n", moy_int_tab_chan1);

///// 3) CALCUL DU DELTA T : temps écoulé entre les depassements de seuil

printf("///// CALCUL STAT DES DELTAT DE PIC\n\n");
/////CHANNEL 1
double sum_deltat_tab_chan1=0;
double moy_deltat_tab_chan1=0;
for (int i=1;i<=n-1;i++)
{
    sum_deltat_tab_chan1=sum_deltat_tab_chan1+deltat_tab_chan1[i];
}
moy_deltat_tab_chan1=sum_deltat_tab_chan1/n;
printf("Moyenne statistique du deltat de pic Chan 1 : %e\n\n", moy_deltat_tab_chan1);

///// 4) CALCUL DES SIGMAS = largeur a mi-hauteur ???
printf("/////CALCUL DU SIGMA\n\n");

// Algorithme
// Prendre valeur max du pic
// Se placer à deltaT et prendre la valeur du tableau
// Diviser la différence des deux par 2 pour trouver la valeur a mi hauteur theorique
// if (après max), valeur tabl depasse mi hauteur theorique, retourner indicem
// Sigma = indicem-indice_pic max

return 0; // Fin du programme
}

```

7.3 Programme d'analyse

Dans cette partie, nous exposons le programme qui a permis de traiter les données de l'oscilloscope. Les sous-routines de fin sont les actrices du calcul de la masse. La première calcule la masse avec une vitesse constante, tandis que la seconde prend en compte le profil de vitesse.


```

    k=index(Char,"")
    Char(k:k)="."
    write(Ifp_out,'(a)') Char
end do
close(ifp_out1)
close(ifp_out)

! lecture dans le fichier texte transposé des différents channel

open(ifp_out,file='Trans.txt', status='unknown', position='REWIND',action='read')

ReadFirstLoop : do
    read(ifp_out, '(a)') Char
    if (index(Char,"Temps_lere_mesure(s):") == 1) exit ReadFirstLoop
end do ReadFirstLoop

read(ifp_out,'(E14.6)') Temps_lere_mesure

ReadSecondLoop : do
    read(ifp_out, '(a)') Char
    if (index(Char,"dt_entre_2_mesures(s):") == 1) exit ReadSecondLoop
end do ReadSecondLoop

read(ifp_out,'(E14.6)') dt_entre_2_mesures

ReadThirdLoop : do
    read(ifp_out, '(a)') Char
    if (index(Char,"CHANNEL 1") == 1) exit ReadThirdLoop
end do ReadThirdLoop

do j=1, NumPoints
    read(ifp_out,'(E14.6)') Channel(j,1)
end do

ReadfourthLoop : do
    read(ifp_out, '(a)') Char
    if (index(Char,"CHANNEL 2") == 1) exit ReadfourthLoop
end do ReadfourthLoop

do j=1, NumPoints
    read(ifp_out,'(E14.6)') Channel(j,2)
end do

ReadfifthLoop : do
    read(ifp_out, '(a)') Char
    if (index(Char,"CHANNEL 3") == 1) exit ReadfifthLoop
end do ReadfifthLoop

do j=1, NumPoints
    read(ifp_out,'(E14.6)') Channel(j,3)
end do

ReadsixthLoop : do
    read(ifp_out, '(a)') Char
    if (index(Char,"CHANNEL 4") == 1) exit ReadsixthLoop
end do ReadsixthLoop

do j=1, NumPoints
    read(ifp_out,'(E14.6)') Channel(j,4)
end do

close(ifp_out)

!module de rejet de l'évènement :
!si le channel 2 et/ou 3 n'ont pas de signaux
!rejet de l'évènement
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! calcul de statistique
DeltaTime=dt_entre_2_mesures

do k=1,NChannel

    !chargement des point du channel k dans le tableau "trans"
    do j=1,NumPoints

```

```

    Trans(j)=Channel(j,k)
end do

! Localisation de l'argument min du vecteur Trans
Location(k)=minval(minloc(Trans))

sum5=0.d0
sum4=0.d0
sum3=0.d0
sum2=0.d0
sum1=0.d0

! calcul de l'offset sur le channel k
do j=1,Location(k)-40
    sum2=sum2+Trans(j)
end do
Offset=sum2/(real(Location(k), kind=8)-40.d0)

! enleve l'offset à Trans(j)
do j=1,NumPoints
    Trans(j)=Trans(j)-Offset
end do

! calcul de statistique
do j=Location(k)-20, Location(k)+40
    sum4=sum4+Trans(j)
    sum3=sum3+Trans(j)*((real(j,kind=8)-1.d0)*DeltaTime+Temps_lere_mesure)
    sum5=sum5+Trans(j)*((real(j,kind=8)-1.d0)*DeltaTime+Temps_lere_mesure)**2
end do

!calcul des différentes grandeurs statistiques :

ImpTime(k)    = Temps_lere_mesure+DeltaTime*(real(Location(k), kind=8)-1.d0)
MeanImpTime(k)= sum3/sum4
Extremum(k)   = Trans(Location(k))
Err_Time(k)   = 2.d0*DeltaTime
! Err_Time(k)  = sqrt(abs((sum5/sum4)-(sum3/sum4)**2))

end do

! module de selection de l'évènement :

if (channel(Location(2),2)<=seuil.AND.channel(Location(3),3)<=seuil) then
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Calcul des vitesses !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!+++++
beta1=75.d0/(SpeedLight*(MeanImpTime(2)-MeanImpTime(1))) !+
beta2=63.9d0/(SpeedLight*(MeanImpTime(4)-MeanImpTime(3))) !+
beta3=18.6d0/(SpeedLight*(MeanImpTime(3)-MeanImpTime(2))) !+
!+++++

if(beta2>0.d0.AND.beta3<beta1.AND.beta2<beta3.AND.beta1<1.d0)then
    write(*,*)Err_Time
    !Calcul des erreurs sur les Betas
    !+++++
    Err_beta1=beta1*(((abs(Err_Time(2))+abs(Err_Time(1))+1.d-12*75.d0)/(MeanImpTime(2)-
    & MeanImpTime(1)))+ 2.29d-2+0.5d0/75.d0)
    Err_beta2=beta2*(((abs(Err_Time(3))+abs(Err_Time(4))+1.d-12*63.9d0)/(MeanImpTime(4)-
    & MeanImpTime(3)))+ 2.29d-2+0.5d0/63.9d0)
    !+++++

    !Création du fichier de sortie pour le tracé des plots
    !+++++
    filenamermk(1:11)=Filename(i)
    filenamermk(12:15)='_rmk'
    !ouverture
    open(ifp_out1,file=filenamermk, status='unknown', action='write')
    !écriture
    Time=Temps_lere_mesure
    do j=1,NumPoints
        write (ifp_out1,'(5E15.6)') Time,Channel(j,1),Channel(j,2),Channel(j,3),Channel(j,4)
        Time=Time+DeltaTime
    end do

```

```

!fermeture
close(ifp_out1)
!+++++

!Appel de la subroutine qui calcul la masse et son erreur associée
!+++++

call MassComputation(Mass,beta1,beta2,Err_Mass,Err_beta1,Err_beta2)
write(*,*)'masse '
write(*, '(E14.6)') Mass
write(*,*)'Erreur associée à la masse'
write(*, '(E14.6)') Err_Mass
!+++++
gamma1=1.d0/sqrt(1.d0-beta1**2)
gamma2=1.d0/sqrt(1.d0-beta2**2)
Err_gamma1=Err_beta1*beta1*(gamma1**3)
Err_gamma2=Err_beta2*beta2*(gamma2**3)
write(ifp_log, '(a,10E14.6)') filename(i),Mass,Err_Mass,beta1,Err_beta1,beta2,Err_beta2,&
& gamma1,Err_gamma1,gamma2,Err_gamma2

!+++++
!Appel de la subroutine integrant la fonction en beta +
!+++++
call Betafunction(Mass,beta1,beta2,Err_Mass,Err_beta1,Err_beta2)

!Ecriture dans le fichier data concerné (data_integration)

write(ifp_log_int, '(a,10E14.6)') filename(i),Mass,Err_Mass,beta1,Err_beta1,beta2,Err_beta2,&
& gamma1,Err_gamma1,gamma2,Err_gamma2
!+++++*

write(ifp_logfile,*)'unset key'
write(ifp_logfile,*)'plot "',filenamermk,'" using ($1*1.e9):2 w l, "'&
& ,filenamermk,'" using ($1*1.e9):3 w l, "'&
& ,filenamermk,'" using ($1*1.e9):4 w l, "'&
& ,filenamermk,'" using ($1*1.e9):5 w l'
write(ifp_logfile,*)'pause -1'
write(*,*)' '
end if

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Calcul des vitesses sur les minimums des signaux !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!+++++
beta1=75.d0 / (SpeedLight*(ImpTime(2)-ImpTime(1))) !+
beta2=63.9d0/(SpeedLight*(ImpTime(4)-ImpTime(3))) !+
beta3=18.6d0/(SpeedLight*(ImpTime(3)-ImpTime(2))) !+
!+++++

if(beta2>0.d0.AND.beta3<beta1.AND.beta2<beta3.AND.beta1<1.d0)then

!Calcul des erreurs sur les Betas
!+++++
Err_beta1 = beta1*(((5.d0*DeltaTime+1.d-12*75.d0)/(ImpTime(2)-ImpTime(1)))+2.29d-2+ &
& 0.5d0/75.d0)
Err_beta2 = beta2*(((5.d0*DeltaTime+1.d-12*63.9d0)/(ImpTime(4)-ImpTime(3)))+2.29d-2+ &
& 0.5d0/63.9d0)
!+++++

!Création du fichier de sortie pour le tracé des plots
!+++++
filenamermk(1:11)=Filename(i)
filenamermk(12:15)='2rmk'
!ouverture
open(ifp_out1,file=filenamermk, status='unknown', action='write')
!écriture
Time=Temps_lere_mesure
do j=1,NumPoints
write (ifp_out1, '(5E15.6)') Time,Channel(j,1),Channel(j,2),Channel(j,3),Channel(j,4)
Time=Time+DeltaTime
end do
!fermeture
close(ifp_out1)
!+++++

```

```

!Appel de la subroutine qui calcul la masse et son erreur associée
!+++++

call MassComputation(Mass,beta1,beta2,Err_Mass,Err_beta1,Err_beta2)
write(*,*)'masse '
write(*,'(E14.6)') Mass
write(*,*)'Erreur associée à la masse'
write(*,'(E14.6)') Err_Mass
!+++++
gamma1=1.d0/sqrt(1.d0-beta1**2)
gamma2=1.d0/sqrt(1.d0-beta2**2)
Err_gamma1=Err_beta1*beta1*(gamma1**3)
Err_gamma2=Err_beta2*beta2*(gamma2**3)
write(ifp_log_edit,'(a,10E14.6)')filename(i),Mass,Err_Mass,beta1,Err_beta1,beta2,Err_beta2,&
& gamma1,Err_gamma1,gamma2,Err_gamma2

!+++++
!Appel de la subroutine integrant la fonction en beta +
!+++++
call Betafunction(Mass,beta1,beta2,Err_Mass,Err_beta1,Err_beta2)

!Ecriture dans le fichier data concerné (data_integration)

write(ifp_log_int,'(a,10E14.6)')filename(i),Mass,Err_Mass,beta1,Err_beta1,beta2,Err_beta2,&
& gamma1,Err_gamma1,gamma2,Err_gamma2
!+++++*

write(ifp_logfile,*)'unset key'
write(ifp_logfile,*)'plot "',filenamermk,'" using ($1*1.e9):2 w l, "'&
& ,filenamermk,'" using ($1*1.e9):3 w l, "'&
& ,filenamermk,'" using ($1*1.e9):4 w l, "'&
& ,filenamermk,'" using ($1*1.e9):5 w l'
write(ifp_logfile,*)'pause -1'
write(*,*)' '

end if
end if

end do
close(ifp_logfile)
close(ifp_log)
close(ifp_log_edit)
close(ifp_log_int)

write(commande,*)"rm listfile.txt"
call system(commande)

write(commande,*)"rm Trans.txt"
call system(commande)

end program Analyse
!+++++
subroutine MassComputation(Masse,beta1,beta2,Emasse,Ebeta1,Ebeta2)
implicit none
real(kind=8), parameter :: c= 2.99792456d10
real(kind=8), parameter :: e=1.6022d-19
real(kind=8), parameter :: pi=3.141592654d0
real(kind=8), parameter :: Na=6.0221d23
real(kind=8), parameter :: MPb=207.2d0
real(kind=8), parameter :: IPb=932.56d0
real(kind=8), parameter :: rhoPb=11.35d0
real(kind=8), parameter :: Z_Pb=82.d0
real(kind=8), parameter :: epsilon=8.834187817e-21
real(kind=8), parameter :: me=9.1094e-28,meeV=5.11e5
real(kind=8), parameter :: jouletoeV=6.24150636309d11
real(kind=8), parameter :: Alpha=4.d0*pi*Na*rhoPb*Z_Pb*((e**2.d0)/(4.d0*pi*epsilon))**2.d0)/(MPb*me*
(c**2.d0))
real(kind=8), parameter :: Pb_size=6.5d0

real(kind=8), intent(out) :: Masse, Emasse
real(kind=8), intent(in) :: Ebeta1,Ebeta2,beta1,beta2

```

```

real(kind=8) :: energy=0.d0
real(kind=8) :: gammal=0.d0, gambeta=0.d0, gamma2=0.d0

energy=0.d0
gammal=0.d0
gambeta=0.d0
gamma2=0.d0

gammal =1.d0/sqrt(1.d0-beta1**2)
gamma2 =1.d0/sqrt(1.d0-beta2**2)
gambeta=gammal*beta1
if(gambeta>=3.d0) then
    masse=rhoPb*2.d6*Pb_size/(gammal-gamma2)
    Emasse=masse*((2.29d-2+(0.5d0/Pb_size))+(((gammal**3)*beta1*Ebetal&
        & +(gamma2**3)*beta2*Ebeta2)/abs(gamma2-gammal)))
else if(gambeta<=3.d0) then
    energy=(Alpha*(log(2.d0*meeV*gambeta/IPb)-beta1**2.d0)/(beta1**2.d0))*Pb_size*jouletoeV
    Emasse=energy*(2.29d-2+(0.5d0/Pb_size))+2.d0*(Ebetal/beta1)*(energy+Alpha*Pb_size*
        & abs(((gammal/beta1)**2)-1.d0))
    masse=energy/abs(gammal-gamma2)
    Emasse=masse*((Emasse/energy)+(((gammal**3)*beta1*Ebetal+(gamma2**3)*beta2*Ebeta2)/
        & abs(gamma2-gammal)))
end if

end subroutine MassComputation
=====
!+++++
subroutine Betafunction(Masse,beta1,beta2,Emasse,Ebetal,Ebeta2)
    implicit none
    real(kind=8), parameter :: c= 2.99792456d10
    real(kind=8), parameter :: e=1.6022d-19
    real(kind=8), parameter :: pi=3.141592654d0
    real(kind=8), parameter :: Na=6.0221d23
    real(kind=8), parameter :: MPb=207.2d0
    real(kind=8), parameter :: IPb=932.56d0
    real(kind=8), parameter :: rhoPb=11.35d0
    real(kind=8), parameter :: Z_Pb=82.d0
    real(kind=8), parameter :: epsilon=8.834187817e-21
    real(kind=8), parameter :: me=9.1094e-28,meeV=5.11e5
    real(kind=8), parameter :: jouletoeV=6.24150636309d11
    real(kind=8), parameter :: Alpha=4.d0*jouletoeV*pi*Na*rhoPb*Z_Pb*((e**2.d0)/(4.d0*pi*epsilon))
**2.d0)/(MPb*me*(c**2.d0))
    real(kind=8), parameter :: K=2.d0*meeV/IPb
    real(kind=8), parameter :: Pb_size=6.5d0

    real(kind=8), intent(out) :: Masse, Emasse
    real(kind=8), intent(in) :: Ebetal,Ebeta2,beta1,beta2
    real(kind=8) :: Delta_beta=0.d0,Beta_old=0.d0,Beta_new=0.d0
    real(kind=8) :: gamma_old=0.d0,gamma_new=0.d0,BF_new=0.d0,BF_old=0.d0,area=0.d0

    Delta_beta=(beta1-beta2)/1.d6
    Beta_old=beta2
    Beta_new=Beta2+Delta_beta
    area=0.d0
    contentloop:do
        if (Beta_old>=beta1)exit contentloop
        if (Beta_new>=beta1)then
            Beta_new=beta1
        end if

        gamma_old=1.d0/sqrt(1.d0-Beta_old**2)
        gamma_new=1.d0/sqrt(1.d0-Beta_new**2)

        BF_new=((gamma_new*Beta_new)**3)/(log(K*((gamma_new*Beta_new)**2))-Beta_new**2)
        BF_old=((gamma_old*Beta_old)**3)/(log(K*((gamma_old*Beta_old)**2))-Beta_old**2)
        area=area+Delta_beta*(BF_new+BF_old)/2.d0
        Beta_old=Beta_old+Delta_beta
        Beta_new=Beta_new+Delta_beta

    end do contentloop
    Masse=Alpha*Pb_size/area
    EMasse=Alpha*(2.29d-2+0.5d0/Pb_size)/area
end subroutine Betafunction

```

Chapitre 8

Bibliographie

<http://en.wikipedia.org/wiki/Photomultiplier>
A Photomultiplier Handbook
http://en.wikipedia.org/wiki/Scintillator#Organic_crystals
[http://en.wikipedia.org/wiki/Stopping_power_\(particle_radiation\)](http://en.wikipedia.org/wiki/Stopping_power_(particle_radiation))
http://en.wikipedia.org/wiki/Cosmic_ray
E.J. Weinberg, et.al., Physical Review D54, Particles and Fields, 1996
Particle Physics Booklet, 1994

<http://en.wikipedia.org/wiki/Muon>