2015 年度　卒業論文

ミュー粒子の寿命と崩壊電子エネルギースペクトラムの研究

奈良女子大学　理学部
物理科学科　高エネルギー物理学研究室

池田侑加　坂本朋子

2016 年 3 月 10 日

目次

概要

本実験の目的は、シンチレーションカウンターを用いて、シンチレーションカウンター中に静止したμ粒子を観測し、そこからμ粒子の寿命の測定とμ粒子の崩壊により生成される電子のエネルギースペクトラムを計測することである。これにより、高エネルギー物理学における実験技術の基礎を学ぶ。

本実験には、静止したμ粒子の観測用の大型プラスチックシンチレーター(580 × 260 × 250[mm]) 1 個と、トリガーカウンター及び Veto カウンター 10 枚を用いた。ADC(Analog-to-Digital Converter) はμ粒子のエネルギーを測定する ADC(μ) と、電子のエネルギーを測定する ADC(e) の 2 台を用いた。また、TDC(Time-to-Digital Converter) はμ粒子の入射と崩壊の時間差の測定に使用した。

μ粒子の寿命測定の為に TDC でμ粒子の入射と崩壊の時間差を、電子のエネルギースペクトラム計測の為に ADC(e) でμ粒子の崩壊後放出される電子のエネルギーを測定した。約 20 日間かけてデータを収集した結果、μ粒子がメインシンチレーター内に入射した事象 19,098,430 イベントを観測し、そのうち、167,574 イベントがメインシンチレーター内で静止した。

実験の結果、μ粒子の平均寿命τは $\tau = 2.211 \pm 0.007[\mu\mathrm{s}]$ となった。この誤差は TDC 分布の指数関数 Fit と、TDC 時間較正の線形近似によるものである。Particle Data Book(Particle Data Group,2010,Review of Particle Physics,p27) に掲載されているμ粒子の平均寿命は $\tau_{\mathrm{PDG}} = 2.197034 \pm 0.000021[\mu\mathrm{s}]$ である。実験結果と比較すると以下のようになった。

$$\frac{\tau_{\text{観測}}}{\tau_{\mathrm{PDG}}} = \frac{2.211}{2.197} = 1.006$$

よって、文献値とのずれは 0.6 %であり、精度の高い実験結果を得ることができた。実験によって算出される寿命が TDC 分布の Fit 範囲によって異なることの原因究明は今後の課題である。

また、電子のエネルギースペクトラムは実験と理論でエネルギー分布の概形及びピークに違いが認められた。Geant4 を用いたコンピューターシミュレーションで実験を再現した結果、この違いは主に制動放射によるものと考えられる。今後の課題として、μ粒子の入射エネルギーのばらつきやその他の効果を考慮して、シミュレーションの再現性の向上を目指す。

第 1 章

宇宙線・μ粒子

1.1　宇宙線

宇宙線とは、宇宙空間を飛び交う高エネルギーの放射線のことである。一次宇宙線が地球に入射し大気と反応することで、二次宇宙線が発生する。一次宇宙線は主に高いエネルギーを持った陽子と α 線から成る。

一次宇宙線が大気中の窒素原子核や酸素原子核と相互作用し、高エネルギーの中間子などの 2 次粒子を生成する。生成された 2 次粒子がさらに別の粒子を生成する。このように粒子が大気中でシャワー状に生成される現象を、空気シャワー現象と呼ぶ。

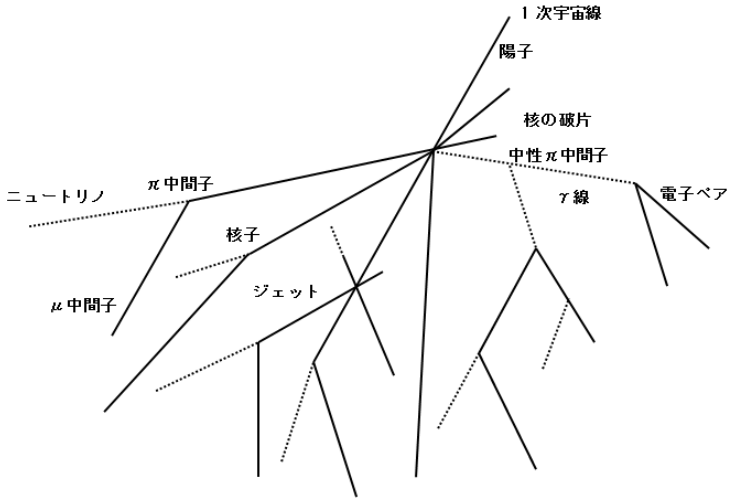


図 1.1.1　宇宙線シャワーの模式図

1.2　μ 粒子とは

1.2.1　μ 粒子の概要

μ 粒子は π 中間子が崩壊して生成される粒子で、二次宇宙線の一つである。π 中間子とは高速の陽子が原子核に衝突することによって生成される粒子である。π 中間子には電荷を持たない中性の π^0 と電荷を持つ π^+,π^- があり、π^0 の質量は $139.57\left(MeV/c^2\right)$、$\pi^{\pm}$ の質量は $134.97\left(MeV/c^2\right)$ である。荷電中間子 π^+,π^- は 100 %の確率で次のように崩壊する。

$$\pi^+ \rightarrow \mu^+ + \nu_\mu$$
$$\pi^- \rightarrow \mu^- + \nu_\mu$$

一方、中性 π 中間子 π^0 は電磁相互作用によりほぼ 100 %の確率で $\pi^0 \rightarrow \gamma\gamma$ に崩壊する。上に示した荷電中間子 π^+,π^- の崩壊によって生成された μ 粒子は質量が $105.7\mathrm{MeV}/c^2$、電荷が ± 1、スピンが $1/2$ であり、μ 粒子はほぼ 100 %の確率で次のように崩壊する。

$$\mu^- \rightarrow e^- + \overline{\nu}_e + \nu_\mu \tag{1.1}$$

ここで、$\overline{\nu}_e$ は反電子ニュートリノ、ν_μ はミューニュートリノである。

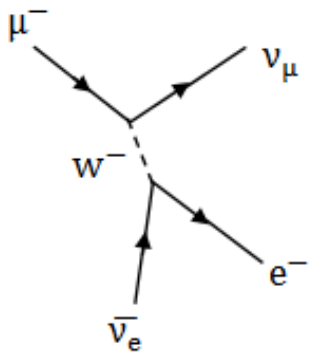


図 1.2.1　μ^- 崩壊のファインマン図

上の図は μ^- 崩壊のファインマン図である。ここで W^- は W ボソンである。

1.2.2　μ 粒子の寿命

μ 粒子の寿命の逆数 τ_μ^{-1} は

$$\tau_\mu^{-1} = \frac{G_F^2 m_\mu^5}{192\pi^3} F(x) \left(1 + \frac{3}{5}\frac{m_\mu^2}{M_W^2}\right) \left[1 + H_1(x)\frac{\hat{\alpha}(m_\mu)}{\pi} + H_2(x)\frac{\hat{\alpha}^2(m_\mu)}{\pi^2}\right] \tag{1.2}$$

と表せる。ここで、

$$x = \frac{m_e^2}{m_\mu^2} \tag{1.3}$$

$$F(x) = 1 - 8x + 8x^3 - x^4 - 12x^2\ln(x) = 0.999813 \tag{1.4}$$

$$H_1(x) = \frac{25}{8} - \frac{\pi^2}{2} - (9 + 4\pi + 12\ln(x))\,x + 16\pi^2 x^{3/2} + \mathcal{O}\left(x^2\right) = -1.8079 \tag{1.5}$$

$$H_2(x) = \frac{156815}{5184} - \frac{518}{81}\pi^2 - \frac{895}{36}\zeta(3) + \frac{67}{720}\pi^4 + \frac{53}{6}\pi^2\ln 2 - \frac{5}{4}\pi^2\sqrt{x} + \mathcal{O}(x) = 6.7 \tag{1.6}$$

$$\hat{\alpha}(m_\mu)^{-1} = \alpha^{-1} - \frac{2}{3\pi}\ln\left(\frac{m_\mu}{m_e}\right) + \frac{1}{6\pi} = 135.9 \tag{1.7}$$

$$G_F = 1.16637 \times 10^{-5}\left(\mathrm{GeV}^{-2}\right) \tag{1.8}$$

である。また、m_e は電子の質量、m_μ は μ 粒子の質量、M_W はボソンの質量である。既知の値を代入すると、

$$(\tau_\mu)_{\text{理論値}} = 2.197 \times 10^{-6}(\mathrm{sec}) \tag{1.9}$$

である。

本実験では、式 1.1 のように崩壊する μ 粒子の寿命と電子のエネルギースペクトラムを測定する。以下の章で実験の原理を説明する。

第 2 章

シンチレーターによる放射線測定の原理

この章では、シンチレーターを用いた放射線測定の原理について説明する。まず、シンチレーターとは、放射線（μ粒子や電子のような荷電粒子）が入射するとシンチレーション光を発する物質のことを指す。シンチレーターに荷電粒子が入射すると、電離損失によりシンチレーターの原子の電子が励起、もしくは電離することによる発光現象が起こる。私たちは普段放射線を感じることはできないが、本実験ではこの発光現象から放射線を検出する。シンチレーション光は微弱なので光電子増倍管を用いて増幅し、ADC にてアナログ信号をデジタル信号に変換する。

以下、本実験におけるμ粒子の寿命の測定と、μ粒子崩壊後に放出される電子のエネルギースペクトラムの測定の原理について解説する。

2.1 μ粒子の寿命測定の原理

μ粒子崩壊の時間的振る舞いは、放射性崩壊の指数関数法則に従う。本実験では、この法則を用いて実験からμ粒子の寿命を測定する。

1 個の粒子が単位時間あたりに崩壊する確率をλとする（崩壊定数λ）。今、 ある時間 t に存在する $N(t)$ 個の粒子が、独立した粒子の集合とすると、微小時間 dt の間に崩壊する粒子の個数 dN は次式で表される。

$$dN = -\lambda N(t)dt \tag{2.1}$$

式 2.1 を積分すると、

$$\begin{aligned}
\int_{N_0}^{N} \frac{dN}{N} &= -\lambda \int_{t_0}^{t} dt \\
[\ln N]_{N_0}^{N} &= -\lambda \, [t]_{t_0}^{t} \\
\ln \frac{N}{N_0} &= -\lambda (t - t_0)
\end{aligned} \tag{2.2}$$

$t = 0$ の時、式 2.2 は

$$N(t) = N_0 e^{-\lambda t} \tag{2.3}$$

となる。 $N(t)$ は時刻 t で崩壊せずに残っている粒子の個数、 N_0 は初期時刻 t_0 での粒子の個数である。ここで、時刻 $t \to t + dt$ の間の微小時間 dt で崩壊する粒子の個数は式 2.1 で表されるので、 N_0 個すべての粒子の生存時間を足し上げたものは次式となる。

$$\begin{aligned} L &= \int_0^\infty tN(t)\lambda dt \\ &= \int_0^\infty tN_0 e^{-\lambda t}\lambda dt \\ &= \left[-\frac{1}{\lambda}N_0 e^{-\lambda t}\right]_0^\infty \\ &= \frac{N_0}{\lambda} \end{aligned} \tag{2.4}$$

よって、粒子の平均生存時間 L/N_0 、すなわち粒子の平均寿命 τ は

$$\begin{aligned} \tau &= \frac{L}{N_0} = \frac{1}{\lambda} \\ \lambda &= \frac{1}{\tau} \end{aligned} \tag{2.5}$$

である。これを式 2.3 に代入すると、

$$N(t) = N_0 e^{-\frac{t}{\tau}} \tag{2.6}$$

が得られる。式 2.6 を図 2.1.1 に示す。

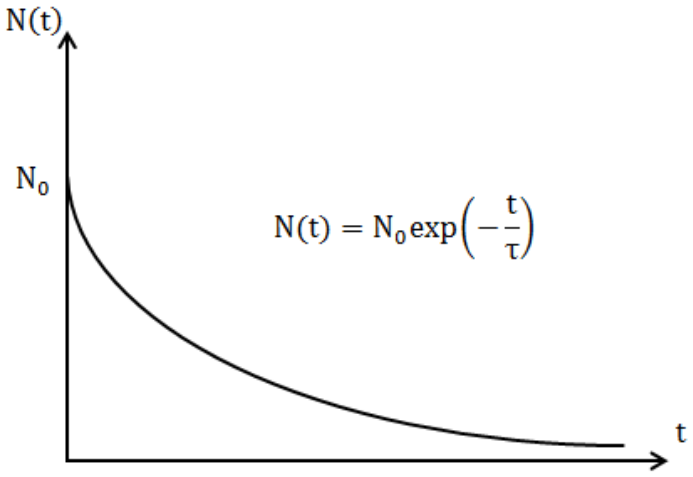


図 2.1.1　放射性崩壊の図

また、式 2.6 を t で微分すると、単位時間あたりの崩壊数を表す式が得られる。

$$\frac{dN}{dt} = -\frac{N_0}{\tau}\exp\left(-\frac{t}{\tau}\right) \tag{2.7}$$

dN/dt は $t \to t+dt$ の間に N 個の粒子が $N+dN$ 個に減少したことを表すので、dN は負である。$t \to t+dt$ の間に崩壊した粒子の個数を $dN_{\text{崩壊}}$ とすると、$dN_{\text{崩壊}} = -dN$ なので、式 2.7 は

$$\frac{dN_{\text{崩壊}}}{dt} = \frac{N_0}{\tau}\exp\left(-\frac{t}{\tau}\right) \tag{2.8}$$

となる。μ 粒子がシンチレーター内に入射しシンチレーター内で静止、その後崩壊してから電子を放出するまでの時間を TDC(Time-to-Digital Converter) で測定することで得られる TDC 分布は式 2.8 に従う。実験で得た TDC 分布を実験式

$$f(t) = A\exp\left(-\frac{t}{\tau}\right) \tag{2.9}$$

でフィットすることにより平均寿命 τ が算出できる（A は定数)。図 2.1.2 は、式 2.8 及び TDC 分布の概形である。

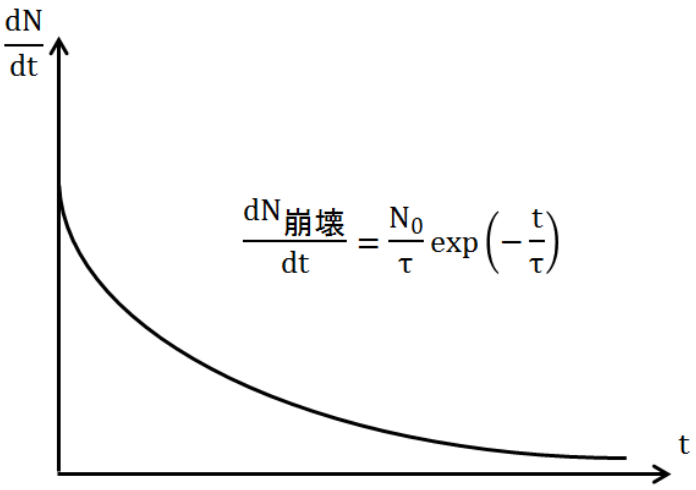


図 2.1.2　TDC 分布の概形図

2.2　電子のエネルギースペクトラムの原理

μ 粒子の崩壊によって生成される電子のエネルギースペクトラムについて調べる。μ 粒子崩壊のファインマン図と崩壊式を以下に示す。

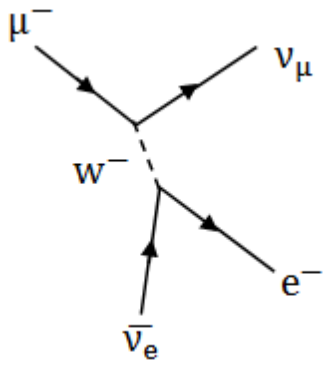


図 2.2.1　μ^- 崩壊のファインマン図

$$\mu^-(P_1) \to e^-(P_2) + \overline{\nu}_e(P_3) + \nu_\mu(P_4)$$

P_1, P_2, P_3, P_4 はそれぞれ μ 粒子、電子、ミューニュートリノ、反電子ニュートリノの 4 元運動量である。

μ 粒子の静止系における、μ 粒子崩壊時に生成される電子のエネルギー分布は図 2.2.2 のようになる。図 2.2.2 から、電子は大きなエネルギーを持って生成される割合が高いことがわかる。また、ここでは電子の質量を無視している。

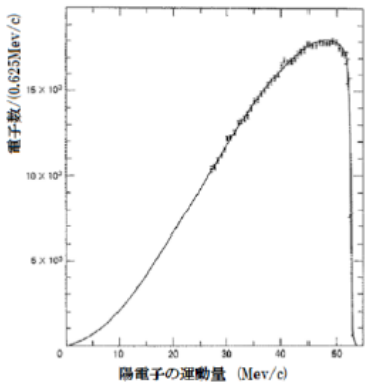


図 2.2.2　μ 粒子崩壊時の電子のエネルギー分布

電子のエネルギーが最小になるのは、μ 粒子の静止系で電子が静止した状態で生成され

るときである。一方、放出される電子のエネルギーが最大になるのは、同時に生成されるミューニュートリノと反電子ニュートリノの運動量がともに電子の運動量に対して 180° の方向を向き、電子に最大の反跳を与えるときである。（図 2.2.3）

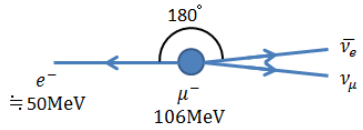


図 2.2.3　電子のエネルギーが最大になるときの各粒子の運動量ベクトル

μ 粒子崩壊時に生成される電子のエネルギー分布の理論式は以下のようにして求められる。μ 粒子の崩壊振幅を M とすると、式 2.10 となる。

$$M = \frac{g_w^2}{8(M_w c)^2}[u(3)\gamma^\mu(1-\gamma^5)u(1)][u(4)\gamma_\mu(1-\gamma^5)u(2)] \tag{2.10}$$

ここで、g_w は結合定数、$\gamma^\mu \equiv i\gamma^0\gamma^1\gamma^2\gamma^3$ 、$u(i)(i=1,2,3,4)$ はスピノール（ディラック方程式の解）である。よって、

$$< [M^2] >= 2\left(\frac{g_w}{M_w c}\right)^4 (P_1 \cdot P_2)(P_3 \cdot P_4) \tag{2.11}$$

式 2.2 と同様に、P_1, P_2, P_3, P_4 はそれぞれ μ 粒子、電子、ミューニュートリノ、反電子ニュートリノの 4 元運動量である。μ 粒子の静止系では、$P_1 = (m_\mu c, \overrightarrow{0})$ となり、P_1 と P_2 の内積 $P_1 \cdot P_2$ は

$$P_1 \cdot P_2 = m_\mu E_2 \tag{2.12}$$

となる。また、エネルギー運動量保存則より

$$(P_3 + P_4)^2 = M_\mu^2 c^2 - 2P_1 \cdot P_2 \tag{2.13}$$

よって、

$$P_3 \cdot P_4 = \frac{(m_\mu^2 - m_e^2)^2 c^2}{2} - m_\mu E_2 \tag{2.14}$$

となる。式 2.11 に式 2.12,2.14 を代入すると、振幅 M は

$$M = \left(\frac{g_w}{M_w c}\right)^2 m_\mu^2 E_2(m_\mu c^2 - 2E_2) \tag{2.15}$$

となる。式 2.15 より、崩壊幅は次のように与えられる。

$$d\Gamma = \frac{< [M^2] > c}{(4\pi)^4 h m_\mu} dE_2 \frac{d^3\overrightarrow{p_4}}{E_4^2} \tag{2.16}$$

式 2.16 に式 2.15 を代入し、微分すると以下のようになる。

$$d\Gamma = \left(\frac{g_w}{4\pi M_w c}\right)^4 \frac{m_\mu c}{h}\left(\frac{m_\mu c^2}{2} - \frac{2}{3}E_4\right) d^3\overrightarrow{p_4} \tag{2.17}$$

また、

$$d^3\overrightarrow{p_4} = 4\pi\left(\frac{E_4}{c}\right)^2 \frac{dE_4}{c} \tag{2.18}$$

なので、式 2.17 に式 2.18 を代入すると、

$$\frac{d\Gamma}{dE} = \left(\frac{g_w}{M_w c}\right)^4 \frac{{m_\mu}^2 E^2}{2h(4\pi)^3}\left(1 - \frac{4E}{3m_\mu c^3}\right) \tag{2.19}$$

を得られる。式 2.19 は、μ 粒子崩壊時の電子のエネルギー分布を表す。

実際に観測される電子のエネルギーは、式 2.19 で期待される分布より様々な理由で変形を受ける。その 1 つが高次の輻射補正で、もう 1 つが制動放射である。

2.3 物質中の荷電粒子（電子）のふるまい

2.3.1 制動放射

制動放射とは、荷電粒子が電場の中で力を受けて加速度運動をすることにより電磁波を放出する現象のことをいう。他の荷電粒子と衝突したとき原子の励起やイオン化を起こさない弾性衝突であっても、電子は質量が小さいため物質中の原子核が作る電場による力を受けて加速度運動をする。このときに、電磁波（光子）を放出する。これが制動放射である。この加速度の大きさは、物質中の荷電粒子の電荷 Z に比例し、入射電子の質量に反比例する。また、加速度運動によって放出される電磁波のエネルギーは加速度の二乗に比例する。よって、放出されるエネルギーは $(Z/m_e)^2$ に比例する。従って、最も制動放射に影響を与えるのは、物質内で大きな電荷 Z を持つ原子核である。制動放射によるエネルギー損失は入射粒子のエネルギーに比例するので、このエネルギー損失は粒子が十分高速のとき有効となる。単位長さあたりに制動放射で放出される電子のエネルギーは次式で与えられる。

$$\frac{dE}{dx} = \frac{NEZ^2{r_e}^2}{137}\left(4\ln\frac{183}{z^{\frac{1}{3}}} + \frac{2}{9}\right) \tag{2.20}$$

ここで、N は物質の単位体積あたりの原子数 $(N = \frac{N_0\rho}{A})$ 、E は入射電子のエネルギー、r_e は電子の古典半径 $(r_e = \frac{{e_0}^2}{m_e c^2} = 2.81 \times 10^{-13}(\mathrm{cm}))$ 、 e_0 は電気素量、A は物質の原子量、 N_0 はアボガドロ数 $N_0 = 6.0210223 \times 10^{23}$ 、ρ は物質の密度を表す。また、制動放射でエネルギーを失うことによってエネルギーが初期エネルギーの $\frac{1}{e}$ になるまで物質中を走る長さを放射長という。放射長 x_0 は式 2.20 より

$$-\frac{dE}{E} = \frac{dx}{x_0} \tag{2.21}$$

と求めることができる。

放射長は、電子が制動放射でγ線を 1 回放出する際に走る距離である。プラスチックシンチレーターにおける放射長は、41.31(cm) である。この値は本実験で電子のエネルギースペクトラムを測定する際、電子が測定用プラスチックシンチレーター内を走るとどれだけのエネルギーを制動放射によって失うかを評価をするときに使用する。

2.3.2　電離損失

荷電粒子が物質中を通過すると、その物質を構成する原子と荷電粒子との相互作用によって、原子が電子と陽イオンに分離される。これを電離という。また電離せずに原子や分子がエネルギーの高い状態になることもある。これを励起といい、その状態を励起状態と呼ぶ。荷電粒子が物質中を通過するときに、物質中の電子と衝突し、電離や励起をくり返しながらエネルギーをの一部を失う。これを電離損失といい、電離損失によって荷電粒子が失うエネルギーは Bethe-Broch の式で表される。

$$\frac{dE}{dx} = 4\pi N_a r_e^2 m_e c^2 \rho \frac{Z}{A}\frac{z^2}{\beta^2}[ln(\frac{2m_e\gamma^2 v^2 W_{max}}{I^2}) - 2\beta^2 - \delta - 2\frac{C}{Z}] \left(\mathrm{MeV}\cdot\mathrm{cm}^2/\mathrm{g}\right) \quad (2.22)$$

ここで、

r_e :電子古典半径　　N_a:アボガドロ数

I :電離ポテンシャル　　Z :物質の原子番号

A :物質の原子量　　v :入射粒子の速度

$\beta = \frac{v}{c}$　　$\gamma = \frac{1}{\sqrt{1-\beta^2}}$

z :入射粒子の電荷　　m_e:電子の質量

W_{max}:入射粒子が 1 回の衝突で物質に与えることのできる最大エネルギー

ρ :物質の密度　　δ :密度補正

C :殻補正

である。

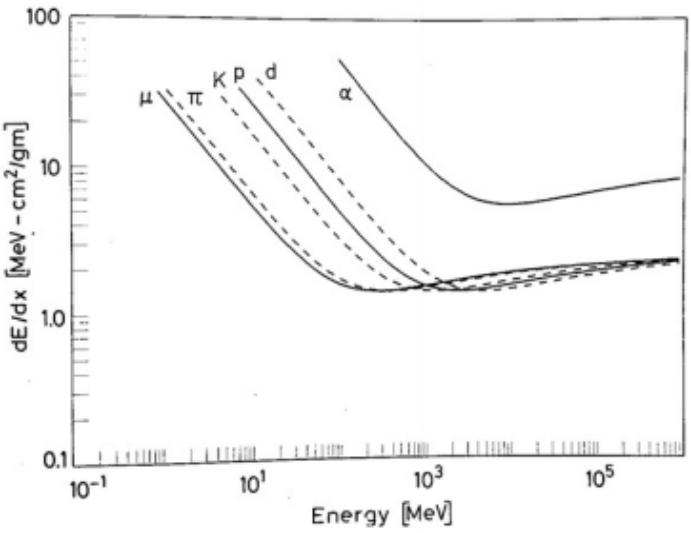


図 2.3.1 電離損失

電離損失 $\frac{dE}{dx}(\beta = \frac{v}{c})$ は物質に入射した荷電粒子の質量には依存しないが、速度 $v = \beta c$ には依存する。β が小さいとき、すなわち入射荷電粒子のエネルギーが小さいとき Bethe-Bloch の式は

$$\frac{dE}{dx} \propto \frac{1}{\beta^2} \tag{2.23}$$

となる。入射荷電粒子のエネルギーが大きくなると電離損失は $\frac{1}{\beta^2}$ に従い減少し、最小値に達する。この領域での電離および最小値のことを Minimum Ionization と呼ぶ。（宇宙線の場合は十分エネルギーが大きいので、電離損失はほとんど Minimum Ionization 領域で起きている。）入射荷電粒子のエネルギーが大きくなると、$\beta^2 \simeq 1$ となりの log の中の項が効くので、電離損失は logγ でエネルギーが増加すると上昇する。

$$\frac{dE}{dx} \propto log[\frac{1}{1-\beta^2}] \tag{2.24}$$

$$\gamma = \frac{1}{1-\beta^2} = \frac{E}{m} \tag{2.25}$$

荷電粒子がシンチレーター内に入射したとき、シンチレータ内の分子を電離および励起するために荷電粒子が失ったエネルギーを光として放出する。これをシンチレーションという。今回の実験では、放出されたシンチレーション光を用いて入射荷電粒子の電離損失によるエネルギー損失を測定する。

2.4　シンチレーションカウンターによる荷電粒子の測定の原理

シンチレーションカウンターは、シンチレーターと光電子増倍管を組み合わせた放射線検出器の一種である。下の図 2.4.1 は実験で使用したシンチレーションカウンターである。以下でシンチレーターと光電子増倍管について述べる。

図 2.4.1　シンチレーションカウンター

2.4.1　シンチレーター

シンチレーターとは、荷電粒子が通過する際にその粒子が電離損失によって失うエネルギーを光エネルギーに変換し、シンチレーション光を放出する物質である。シンチレーターには大きく分けて有機シンチレーターと無機シンチレーターがあり、有機シンチレーターは無機シンチレーターよりも光の減衰が約 100 倍早いので時間分解能が高いが、シンチレーション光が弱いため、高いエネルギー分解能が得られない。本実験では有機シンチレーターの中のプラスチックシンチレーターを用いる。

2.4.2　光電子増倍管

光電子増倍管とは、光電効果を利用して光エネルギーを電気エネルギーに変換し、電子の数を増倍させるものである。図 2.4.2 で光電子増倍管の仕組みの模式図を示す。シンチレーターから放出された光が光電面に入射して光電効果により電子が飛び出し、電子は強い電場によって加速されてダイノードに衝突する。ダイノードに衝突した電子はダイノード内の電子を飛び出させ、それを繰り返すことにより電子の数を増幅する。増幅された電子は陽極に集まり、電流として外部に読みだされる。図 2.4.3 は実験で使用したものと同じ型の光電子増倍管のダイノード部である。

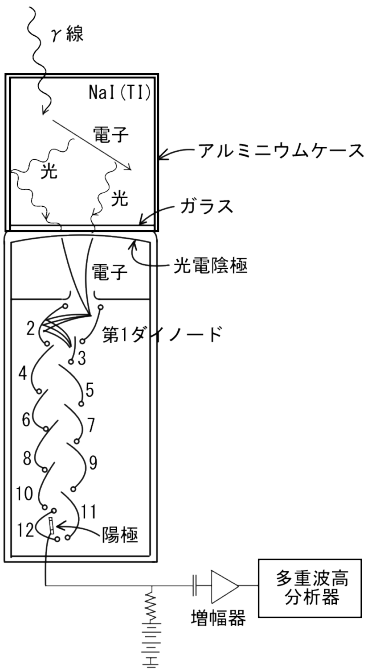


図 2.4.2　光電子増倍管

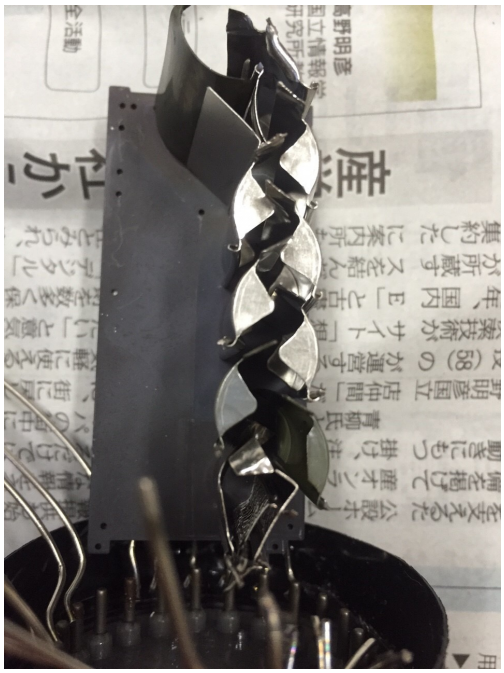


図 2.4.3　実際の光電子増倍管のダイノード

2.5　プラスチックシンチレーターのエネルギー損失

プラスチックシンチレーターのエネルギー損失は、

$$\triangle E\,(\mathrm{MeV}) = \frac{dE}{dx}\left(\mathrm{MeV}\cdot\mathrm{cm}^2/\mathrm{g}\right) \times \rho\,(\mathrm{g/cm}) \times t\,(\mathrm{cm}) \tag{2.26}$$

で求められる。ここで、$\frac{dE}{dx}|_{min}$ は minimam ionization の粒子に対する厚さ $1\,(\mathrm{g/cm})$ あたりのエネルギー損失であり、ρ は物質の密度、t は物質の厚さであり、値は $\frac{dE}{dx}|_{min}$=1.936 $\left(\mathrm{MeV}\cdot\mathrm{cm}^2/\mathrm{g}\right)$、$\rho$=1.06 $(\mathrm{g/cm})$ とする。
これより、プラスチックシンチレーターの 1 cm あたりのエネルギー損失は

$$\triangle E = 1.936 \times 1.06 \times 1 = 2.05 \quad (\mathrm{MeV}) \tag{2.27}$$

のように求められる。本実験で用いるメインシンチレーターの厚さは 26cm なので、メインシンチレーターを通過した際のエネルギー損失 $\triangle E$ は

$$\triangle E = 2.05 \times 26 = 53.3 \quad (\mathrm{MeV}) \tag{2.28}$$

となる。この値は ADC のエネルギー較正を行う際に使用する。

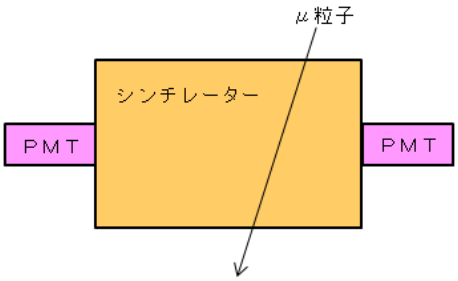


図 2.5.1　μ 粒子がシンチレーターを通過する様子

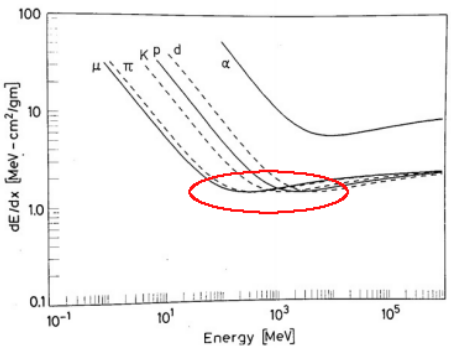


図 2.5.2　電離損失の minimun ionization 部

第 3 章

測定方法とセットアップ

3.1　計測全体の流れ

本実験で放射線計測に用いる装置の概要を図 3.1.1 に示す。

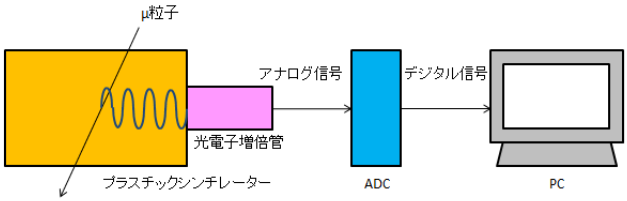


図 3.1.1　放射線計測装置の概要

まず、荷電粒子（本実験では μ 粒子）がシンチレーター内を通過することでシンチレーション光が発生する。シンチレーション光が光電子増倍管の光電面にあたると光電効果で電子が発生する。発生した電子を光電子増倍管内部のダイノードで増幅させ、アナログ信号として出力する。光電子増倍管が出力したアナログ信号を ADC(Analog-to-Digital Converter) によりデジタル信号に変換し、デジタル信号を PC で解析することで放射線を計測することができる。

3.2　通過 μ 粒子と静止後崩壊する μ 粒子の信号の見分け方

シンチレーターに μ 粒子が入射した際、μ 粒子がシンチレーターを通過したときと、μ 粒子がシンチレーター内で静止した後崩壊したときでは、観測される信号に以下で述べるように大きな違いがある。

- μ 粒子がシンチレーターを通過する場合

観測される信号は図 3.2.2 に示すように、1 つの山の信号が見られる。

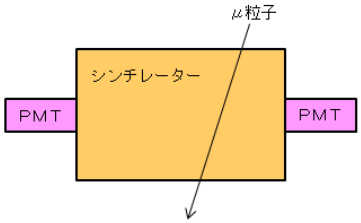


図 3.2.1　μ 粒子とシンチレーターの様子

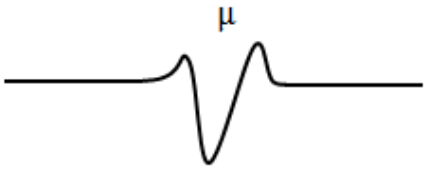


図 3.2.2　入射 μ 粒子の信号

- μ 粒子がシンチレーター内で静止した後崩壊する場合

崩壊するとその時電子が放出されるので、観測される信号は図 3.2.4 に示されるように入射 μ 粒子によるものと崩壊電子によるものの 2 つの信号の山が見られる。以下、前者を μ 粒子の波形、後者を電子の波形と呼ぶ。

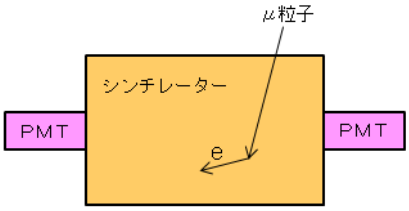


図 3.2.3　μ 粒子と電子とシンチレーターの様子

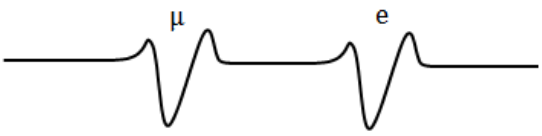


図 3.2.4　入射 μ 粒子と崩壊電子の信号

3.3　シンチレーションカウンター

3.3.1　シンチレーションカウンターの配置

本実験では 11 個のシンチレーションカウンターを用いた。用いたシンチレーションカウンターの配置図を図 3.3.1 から図 3.3.3 に示す。各シンチレーションカウンターの名称とサイズを表 3.3.1 に示す。

表 3.3.1　各シンチレーションカウンターの仕様

	名称	横幅 (mm)	縦幅 (mm)	奥行き (mm)	個数 (個)
大型プラスチックシンチレーター	S1,S2	580	260	250	1
プラスチックシンチレーター	T1,T2	582	10	250	2
プラスチックシンチレーター	V1,V2,V5,V6	131	250	10	4
プラスチックシンチレーター	V3,V4,V7,V8	280	261	10	4
ライトガイド					10
光電子増倍管					12

図 3.3.1 はシンチレーションカウンターの全体の配置図である。また、図 3.3.2 と図 3.3.3 はシンチレーションカウンターの分解図である。中心にある最も大きなシンチレーターをメインシンチレーターとし、メインシンチレータにはライトガイドを通さず直接 2 つの光電子増倍管 S1、S2 を取り付けている。図中ではわかりやすくするために S1、S2 以外の光電子増倍管は省略している。

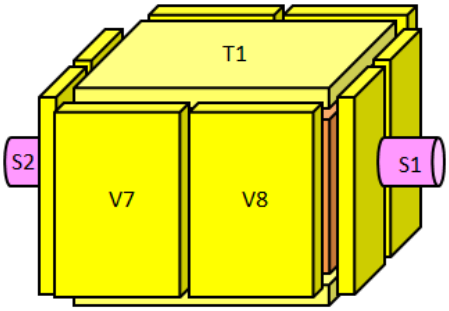


図 3.3.1　全体図

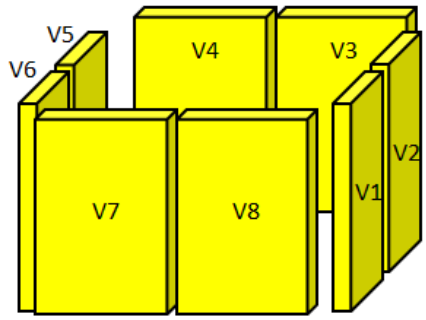


図 3.3.2　分解図 1

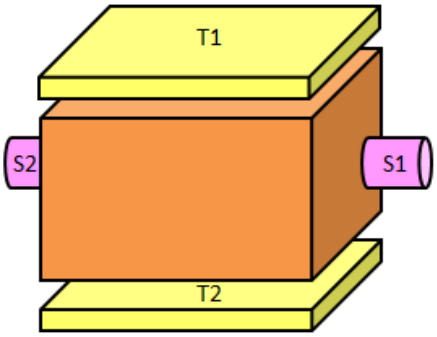


図 3.3.3　分解図 2

3.3.2　トリガーカウンター

宇宙線はあらゆる方向から絶え間なく降り注いでおり、本実験で使用するμ粒子もまた、あらゆる方向からシンチレーターに絶え間なく入射している。また、シンチレーターは本来捉えたいμ粒子以外にも、電子等の荷電粒子や電気ノイズに対しても信号を発生する。そこで、シンチレーターの発する信号の中からμ粒子の信号だけを取り出すために、いくつかのカウンターが同時に反応しているときの信号のみを採用する。このようにいくつかのカウンターを用いて信号が来たことを同定することをトリガーといい、トリガーに使用するシンチレーションカウンターをトリガーカウンターと呼ぶ。本実験ではメインシンチレータ中の S1 と、メインシンチレータの上部に設置した T1 をトリガーカウンターとし、図 3.3.4 の S1 と T1 が同時に信号を発しているとき（S1 と T1 のコインシデンスが ON のとき）、μ粒子が入射したとする。すなわち、S1、T1 のコインシデンスをトリガー信号として用いている。

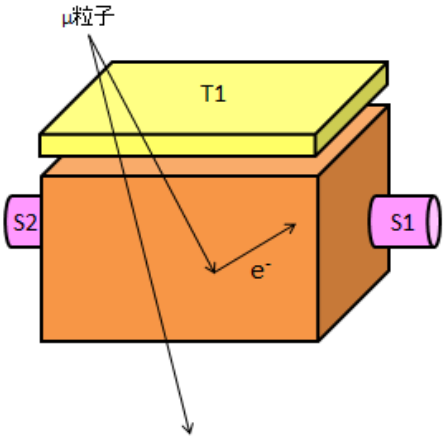


図 3.3.4　トリガーカウンターの概念図

3.3.3　ベトーカウンター

本実験では、メインシンチレーター内でμ粒子が静止した後電子を放出し、かつ生成された電子がメインシンチレーター内に留まったイベントについて測定を行う。μ粒子崩壊後、電子はシンチレーター内をエネルギーを放出しながら通過する。このとき、いくつかの電子は高いエネルギーを持っているためメインシンチレーター内で止まらず外に飛び出す。本実験において電子のエネルギースペクトラムを測定するために必要なイベントは、電子がメインシンチレーター内で全エネルギーを放出したイベントのみである。そのため、メインシンチレーターを飛び出した電子を計測し、電子がメインシンチレーターを飛び出したと判断されたイベントについてはデータ解析の段階で取り除く。この判断をするためのデータを与えるのがベトーカウンターである。

メインシンチレーター周囲に配置した T1、T2、V1〜V8 の 10 個のカウンターをベトーカウンターとし、μ粒子崩壊後にこれらのうち 1 つでもが信号を出力すれば、電子がメインシンチレーター外に飛び出したと判断した。図 3.3.5 はメインシンチレーター内で放出された電子が V1 を通過して外に飛び出したときの概念図である。また、メインシンチレーターの下部に設置した T2 は、メインシンチレーター内を貫通したμ粒子を識別する目的にも使用した。

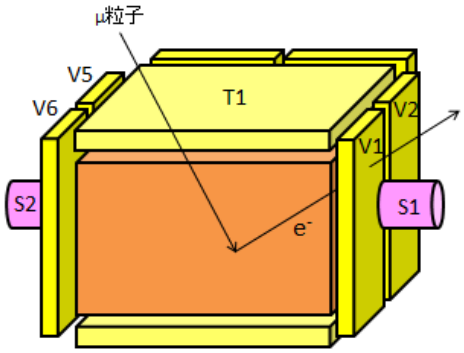


図 3.3.5　ベトーカウンターの概念図

3.4　エレクトロニクス

3.4.1　エレクトロニクスのセットアップ

本実験で使用したエレクトロニクスの全体図を図 3.4.1 に示す。

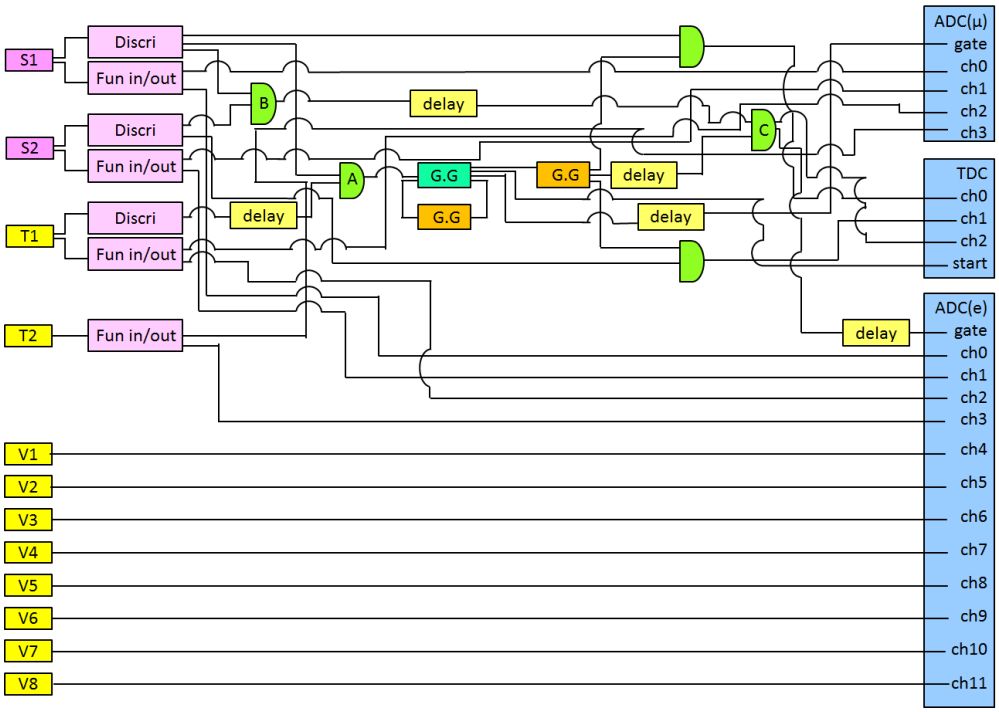


図 3.4.1　エレクトロニクスの全体図

以下、初めに、使用するエレクトロニクスの専門用語について解説する。その後、本実験で用いた個々のセットアップについて説明する。

3.4.2 NIM

NIM(Nuclear Instrument Module) とは、『放射線測定モジュール標準規格 TID-20893』に準拠したエレクトロニクスの規格のことである。同規格では、信号レベル・コネクタ形状・電源・サイズなどについて規格されている。この企画に準拠した装置は NIM モジュールと呼ばれている。本実験では、以下の NIM モジュールを使用した。以下、簡単に解説する。

ディスクリミネーター (Discriminator)
: あらかじめ設定した threshold（しきい値）よりも大きな信号が入力されたとき、その threshold を超えた時刻を起点として矩形波（NIM 信号）を出力するモジュール。NIM 規格で出力するパルスの大きさは-0.7V と決められている。
本実験では threshold 以下のノイズを除去するために用いた。

コインシデンス (Coincidence)
: 最大 4 つまでの複数の信号が同時に入力され重なったとき、重なった時刻を起点としてパルス（NIM 信号）を出力する（複数の信号の AND を取る）モジュール。出力するパルスの時間幅を任意に調整することができる。

ゲート・ジェネレーター (Gate and delay Generator)
: 信号が入力されたとき、矩形波（NIM 信号）を出力するモジュール。出力するパルスの時間幅と Delay を任意に調整することができる。

ファンイン ファンアウト (Fan-in Fan-out)
: 最大 4 つまでの複数の入力信号についてのアナログ和をとり、その結果のアナログ信号を最大 4 つまで複製して出力するモジュール。
本実験では T1、T2、S1、S2 それぞれからの信号を複製するために使用した。

3.5　CAMAC

3.5.1　CAMAC の概要

CAMAC(Computer Aided Measurements And Control) とは、アナログ入力信号を値に数量化するエレクトロニクスのシステム規格のことである。アメリカの NIM とヨーロッパの ESONE 委員会のジョイントで提供された。この規格は世界中のほとんどの素粒子・原子核研究所や、多くの工場現場で使用されている。CAMAC は計算機周辺のデジタル化された情報の処理を機能ごとにモジュール化して行えるようにできている。実験装置などの外部からの情報は、プラグイン・ユニットまたはモジュールのパネルからコネクターを通じて取り込まれる。この情報はプラグイン・ユニットの中で処理されると、プラグイン・ユニットを収容するクレートと呼ばれる箱のバックプレーン及びデータウェイに流れる。これらはクレートコントローラーが制御する。また、クレートコントローラー自身は計算機の指示に従って制御されている。プラグイン・ユニットは内部の回路基板のエッジがコネクターとなって飛び出していて、クレートに差し込むと自動的にクレート裏側のコネクターを通じてデータウェイと接続する。そこから電源やデータ・制御信号の受け渡しが行われる。計算機から CAMAC への命令は、アドレス（住所）とファンクション（動作）を指定する。

アドレス

C,N,A,F の 4 つの数字で指定する。

C　クレートコントローラーの番号。本実験で使用しているクレートコントローラーは 1 つだけなので、C=1 である。

N　スロット番号

A　サブアドレス。モジュール内の信号チャンネル。

F　動作（Function）

ファンクション (Function)

モジュールによって使用可能な Function は異なるが、多くの場合、番号は決まったものがある。

F(0),F(2)　Read Data

F(8)　Test LAM

F(9)　Clear LAM

F(24)　Disable LAM

F(26)　Enable LAM

ここで、LAM(Look At Me) とは、モジュールからデータが読み出し可能となったことをコンピューターに知らせる信号のことである。

3.5.2 ADC

ADC(Analog to Digital Converter) とは、アナログ電気信号をデジタル電気信号に変換する電子回路のことである。プラスチックシンチレーターシンチレーター内で崩壊した μ 粒子と電子からの信号を ADC で数値化し、PC で解析を行った。ADC はアナログ信号を数値化する方法の違いによって大きく、Q モードと V モードに分類される。本実験では Q モードを使用した。Q モードとは、荷電積分型 ADC で Gate パルスが持続している間に入力された信号の積分値を測定するモードである。測定される電荷の量は

$$Q = \int_{dt} idt$$

となる。i は信号の電流であり、電荷は電流の時間積分である。

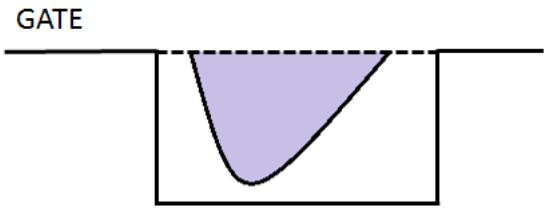


図 3.5.1 ADC の仕組み

3.5.3 TDC

TDC(Time to Digital Converter) とは、スタート信号の入力で内部のクロックをスタートさせ、ストップ信号の入力時までのクロックの出力数をカウントする電子回路のことである。図 3.5.2 のようにして、スタート信号からストップ信号までの時間を計測する。本実験で使用した TDC は REPIC 社 Rpc-060 型 (25psec/count select) で、最大 118ns まで測定することができる。

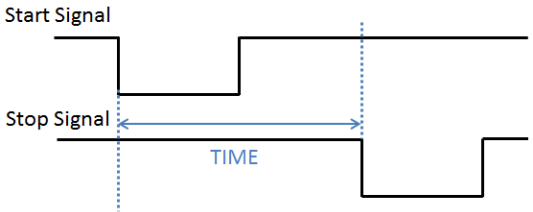


図 3.5.2 TDC の仕組み

3.6　各 CAMAC のモジュールについてのセットアップ

本実験では以下に示す 3 つの測定が必要である。

(1) μ 粒子のエネルギー測定

(2) μ 粒子が入射してから電子が放出されるまでの時間の測定 (μ 粒子の寿命測定)

(3) μ 粒子の崩壊で放出される電子のエネルギースペクトラムの測定

図 3.4.1 に示した全セットアップの中で、(1)〜(3) の各測定におけるセットアップについて詳細を示す。

3.6.1　入射した μ 粒子のエネルギー損失の測定 (ADC(μ) のセットアップ)

入射した μ 粒子のエネルギー損失を測定する ADC を ADC(μ) とし、セットアップ図を図 3.6.1 に示す。

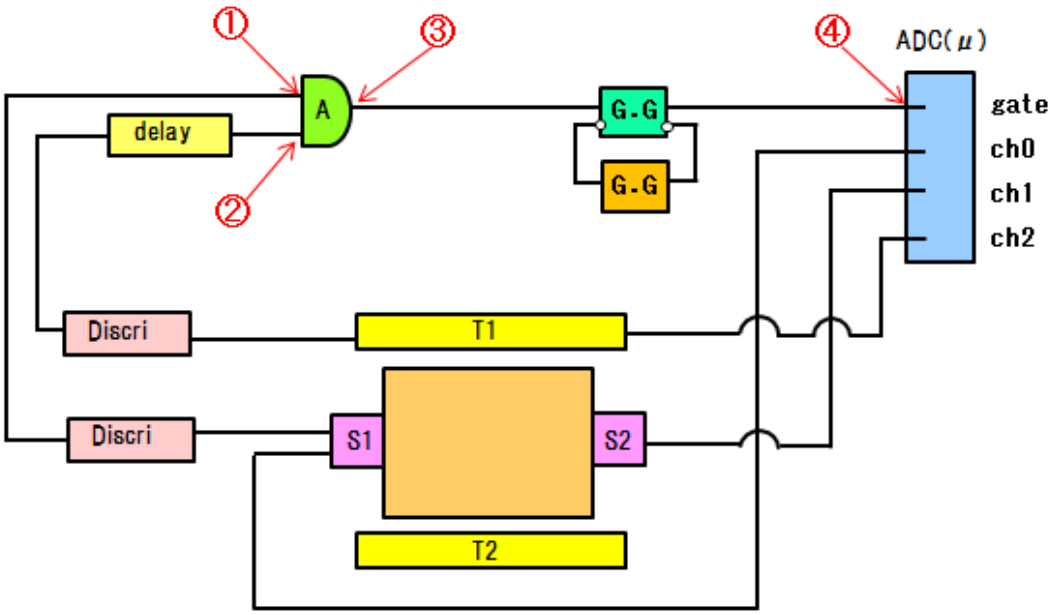


図 3.6.1　ADC(μ) のセットアップ

μ 粒子が入射したとみなす条件は、T1 と S1 のシンチレーターが ON のときとする。T1 と S1 の Coincidence で gate をつくり、μ 粒子のエネルギー損失を測定した。

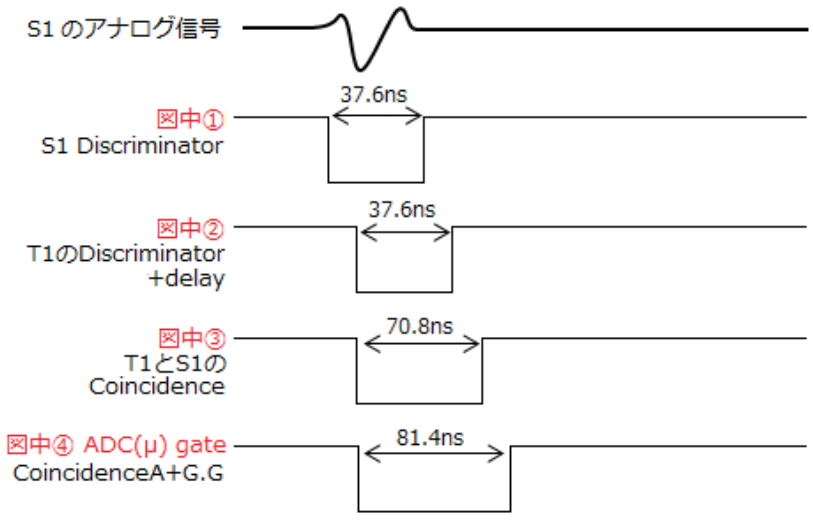


図 3.6.2　ADC(μ) のセットアップにおける各モジュールの入力または出力信号

3.6.2　μ 粒子の寿命測定 (TDC のセットアップ)

TDC のセットアップ図を図 3.6.3 に示す。

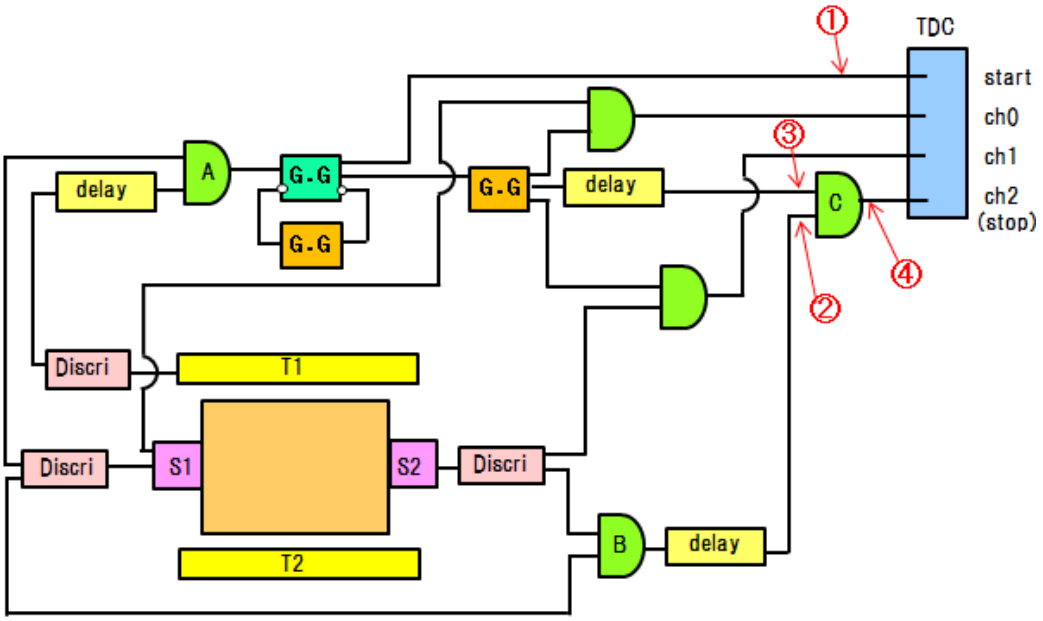


図 3.6.3　TDC のセットアップ

A の Coincidence(μ 粒子がシンチレーターに入射) の信号を TDC の start 信号とし、C の Coincidence(μ 粒子が崩壊した後電子を放出) の信号を TDC の stop 信号とした。これにより、start 信号から stop 信号までの時間を μ 粒子の寿命として測定した。

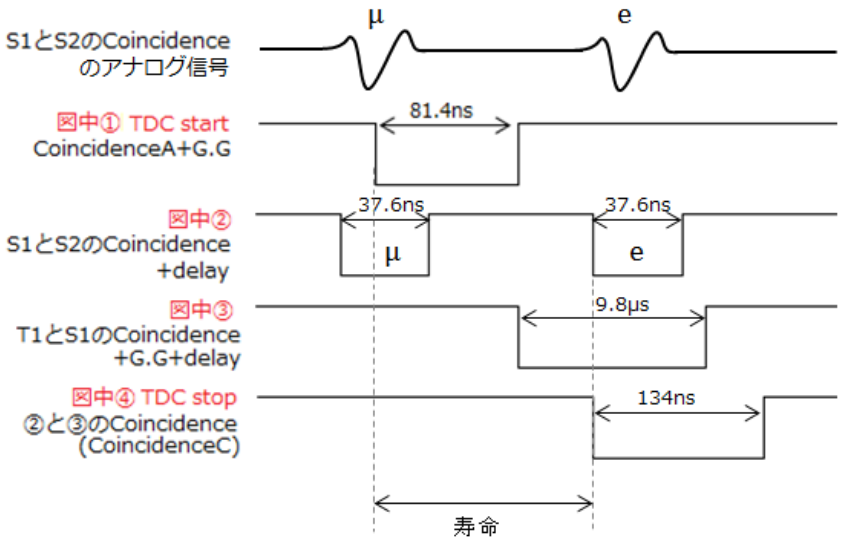


図 3.6.4　TDC の start 信号と stop 信号

3.6.3　電子のエネルギースペクトラムの測定 (ADC(e) のセットアップ)

電子のエネルギーを測定する ADC を ADC(e) とし、セットアップ図を図 3.6.5 に示す。

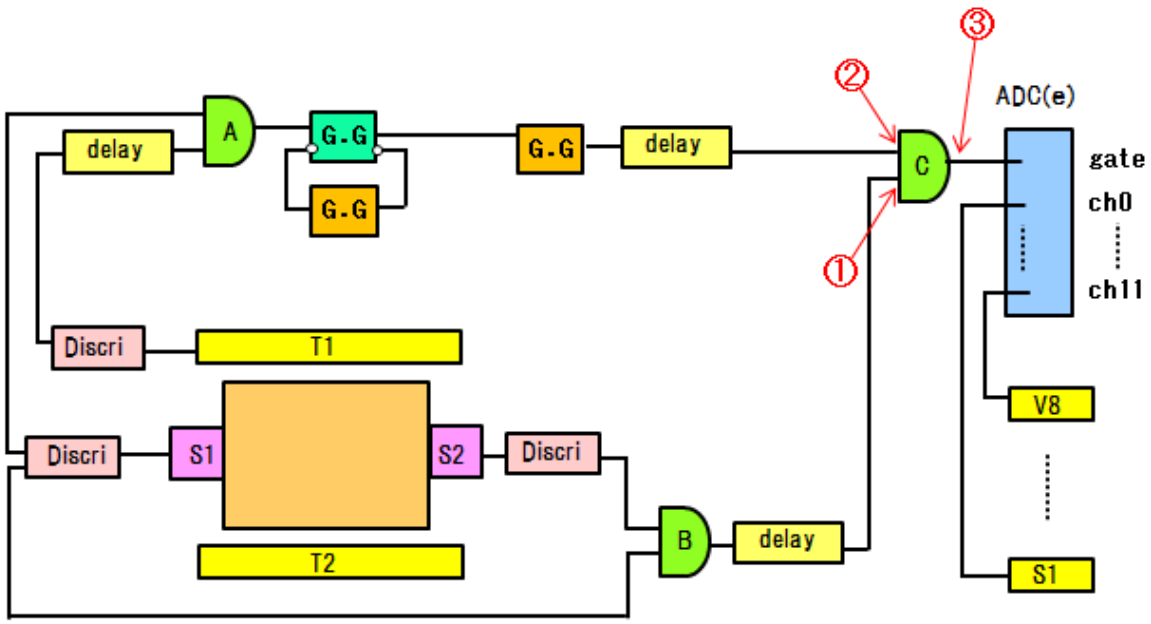


図 3.6.5　ADC(e) のセットアップ

S1 と S2 の Coincidence を gate とするが、計測する μ 粒子以外に入ってきた μ 粒子を取り除くために C の Coincidence で gate generator の信号と AND をとって電子の信号のみを取り出して gate とした。これにより、μ 粒子が崩壊して放出した電子のエネルギー損失を測定した。

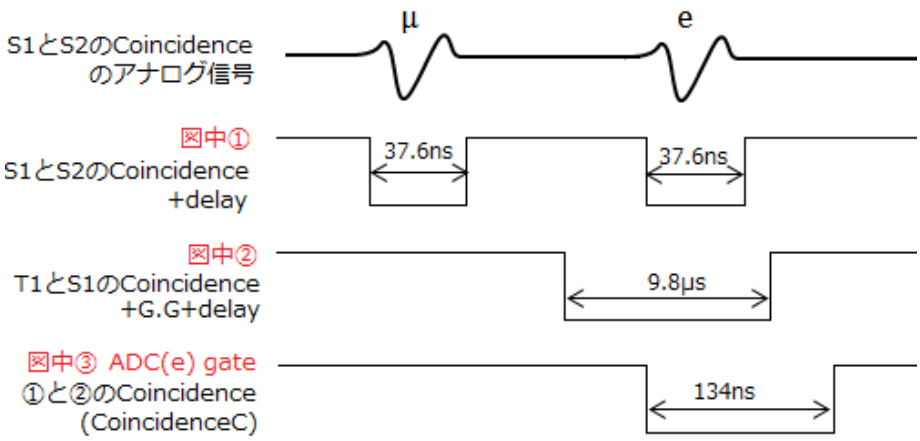


図 3.6.6　ADC(e) のセットアップにおける各モジュールの入力または出力信号

3.7　エレクトロニクスの設定

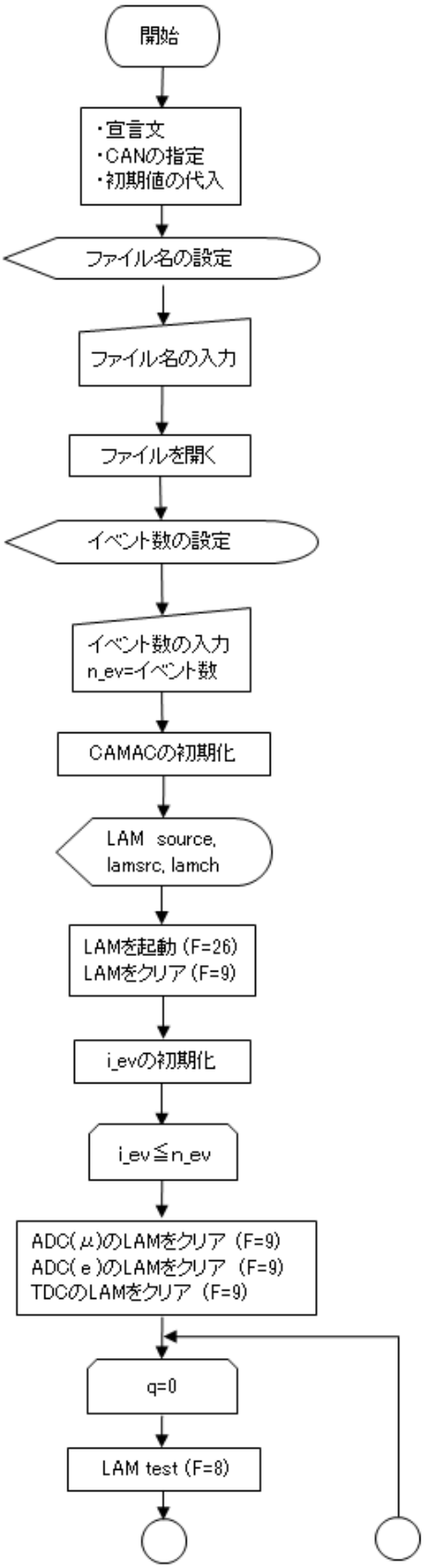
本実験で用いたカウンターの光電子増倍管の型番、光電子増倍管にかけた電圧 (ＨＶ)、Discriminator の Threshold、各カウンターの ADC(μ)・ADC(e) のペデスタルの値を表にまとめる。

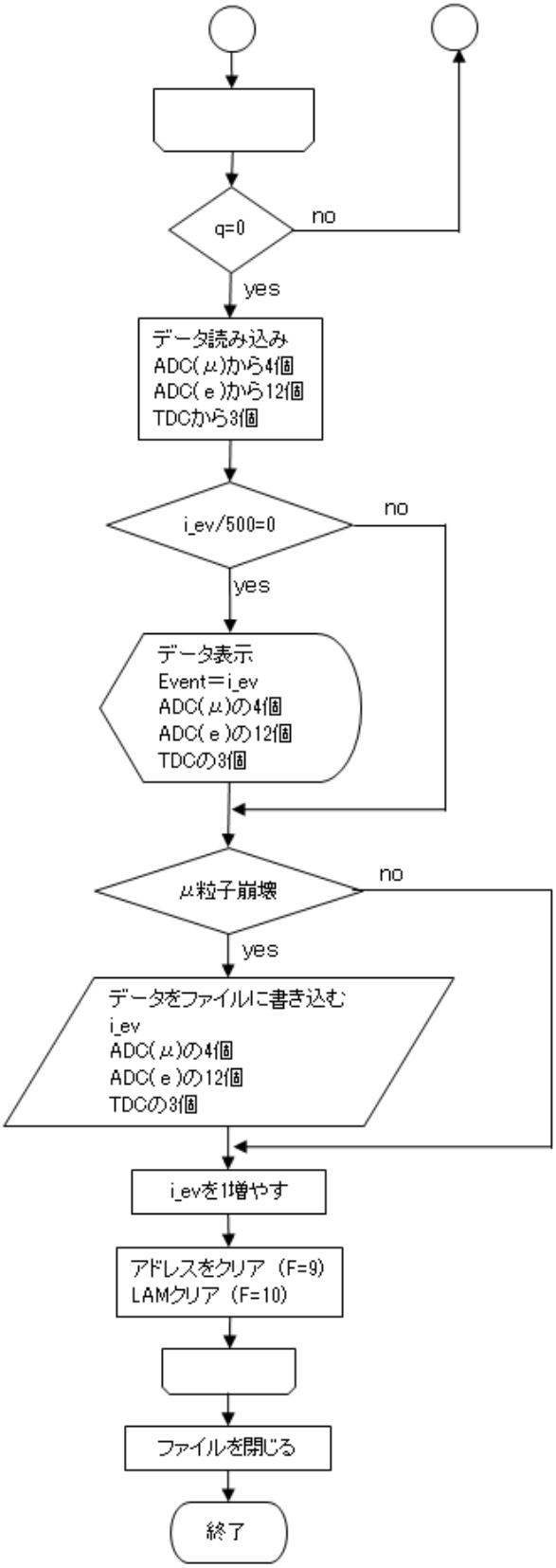
カウンター	PMT 型番	HV(V)	Threshold(mV)	ペデスタル ADC(μ)	ペデスタル ADC(e)
S1	H1161	2050	160	106	57
S2	H1161	2050	300	96	63
T1	H1161	2250	100	88	57
T2	H7195	1770	100	85	61
V1	H1161	1970			68
V2	H7195UV	1660			64
V3	H1161	1870			64
V4	H1161	2440			63
V5	H1161	2400			65
V6	H1161	2030			65
V7	H7195	2210			66
V8	H1161	2025			74

3.8 データ収集プログラム

本実験では、CAMAC のデータ収集プログラムに C 言語を用いた。初めに、コンピューターから ADC にテスト LAM の信号を送る。LAM OK の Q 信号が返ってきたら ADC・TDC のデータを読み込む。ファイルに書き込んだ後、データと LAM のクリアを行い、再び ADC にテスト LAM の信号を送る。これをイベントの数だけ繰り返し行う。

次のページに、用いたデータ収集プログラムのフローチャートを示す。プログラムについては巻末の付録 A を参照。


開始
・宣言文
・CANの指定
・初期値の代入
ファイル名の設定
ファイル名の入力
ファイルを開く
イベント数の設定
イベント数の入力
n_ev=イベント数
CAMACの初期化
LAM source,
lamsrc, lamch
LAMを起動 (F=26)
LAMをクリア (F=9)
i_evの初期化
i_ev≦n_ev
ADC(μ)のLAMをクリア (F=9)
ADC(e)のLAMをクリア (F=9)
TDCのLAMをクリア (F=9)
q=0
LAM test (F=8)


q=0
no
yes
データ読み込み
ADC(μ)から4個
ADC(e)から12個
TDCから3個
i_ev/500=0
no
yes
データ表示
Event＝i_ev
ADC(μ)の4個
ADC(e)の12個
TDCの3個
μ粒子崩壊
no
yes
データをファイルに書き込む
i_ev
ADC(μ)の4個
ADC(e)の12個
TDCの3個
i_evを1増やす
アドレスをクリア（F=9）
LAMクリア（F=10）
ファイルを閉じる
終了

第 4 章

μ粒子の寿命測定

本章では、μ粒子の寿命測定の結果について報告する。本実験では約 20 日間かけてμ粒子がメインシンチレーター内に入射したイベントのデータを採集した。集めたデータは全部で 19098430 イベントとなった。これらのデータを解析することでμ粒子の寿命を測定する。

4.1 原理

まず、解析の原理を説明する。放射性崩壊についての原理はセクション**??**を参照。

TDC で測定した、μ粒子が入射したときの信号と、μ粒子崩壊後シンチレーター内で放出された電子の信号の時間差の分布は、以下の放射性崩壊の指数関数分布に従う。

$$\frac{dN_{\text{崩壊}}}{dt} = \frac{N_0}{\tau} \exp\left(-\frac{t}{\tau}\right) \tag{2.8}$$

ここで、実験で得た TDC 分布に以下の実験式を Fit する。

$$y(t) = p_0 \exp\left(-\frac{t}{p_1}\right) \tag{4.1}$$

式 2.8 と式 4.1 を比較すると、式 4.1 の p_1 と式 2.8 の τ が比例関係にあることがわかる。よって、Fit により p_1 を求めることで、平均寿命 τ を算出できる。

4.2　解析

4.2.1　イベントの選定

μ粒子の寿命測定においてデータを使用するイベントは「μ粒子がメインシンチレーター内で崩壊したイベント」である。約 20 日間で取得した 19098430 イベントのうち、次の条件に合うイベントを選定する。

選定条件

1. μ粒子がメインシンチレーター内に入射
   T1,S1 コインシデンスが ON になっている。すなわちデータが取得できている。
2. μ粒子がメインシンチレーターで崩壊
   メインシンチレーター内で電子が放出されている。つまり ADC(e) で取得した S1 の値（以下 es1 と呼ぶ）が、カウンターが鳴っているとみなす値を上回っているイベントである。es1 のヒストグラムを図 4.2.1 に示す。ただし、縦軸は対数表示となっている。

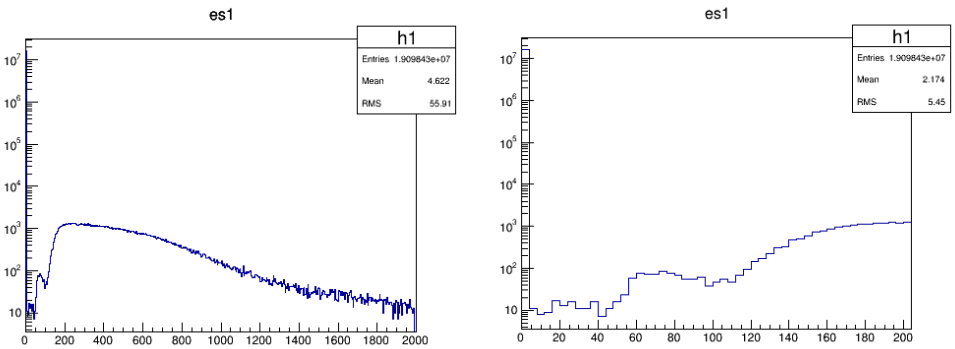


図 4.2.1　左：es1 の分布　右：es1 の分布の上端を拡大したもの

   ここから、es1 $>$ 120 のとき、メインシンチレーター内でμ粒子が崩壊したとみなす。
3. TDC がタイムアウトしていない
   TDC がタイムアウトしたとき、戻り値は 4000 以上の値を示す。よって、具体的な選定条件は TDC カウント数 $<$ 4000 である。

以上の選定条件に合致したものは、全部で 167574 イベントとなった。

4.2.2　TDC 分布に関数を Fit

TDC 分布に Fit した関数は、先述の通り以下の式である。

$$y(t) = p_0 \exp\left(-\frac{t}{p_1}\right) \tag{4.1}$$

Fit 範囲について、立ち上がりのふらつきやオーバーフローしたデータを省くため、TDC カウント数 60〜2000 の間で Fit を行った。Fit したときの TDC 分布を図 4.2.2 に示す。この Fit には ROOT を用いた。Fit のソースコードについては付録 D を参照。

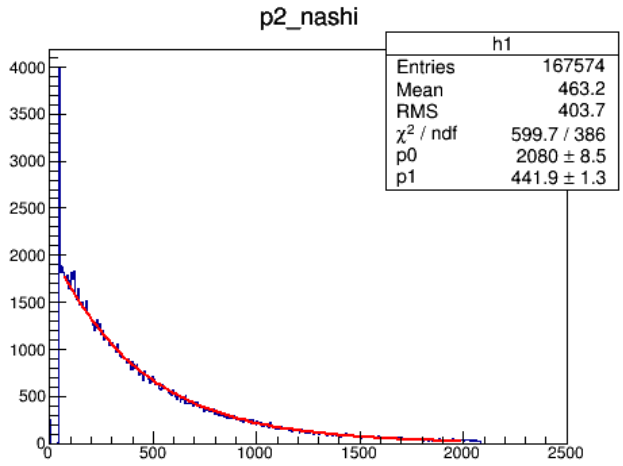


図 4.2.2　式 4.1 を Fit したときの TDC 分布

ここから、

$$p_1 = 441.9 \pm 1.3[\text{count}] \tag{4.2}$$

となった。ただし、得られた p_1 はあくまで TDC のカウント数であって、秒ではない。従って、TDC の時間較正を行い、TDC カウント数を秒に換算する必要がある。

4.2.3 TDC 時間較正

前章で得られた TDC カウント数 p_1 を秒に換算するために、TDC カウント数と実際の時間の対応を調べる。

それぞれ Clock Generator で生成した、1 つ目のパルスを start 信号、ある時間だけ delay させた 2 つ目のパルスを stop 信号として、その間の時間を TDC で測定する（図 4.2.3）。

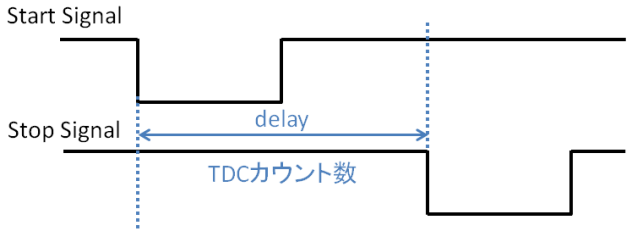


図 4.2.3 TDC 時間較正に用いるパルス

パルスを delay させる時間を変化させ、それに応じた TDC のカウント数の変化を見ることで、TDC1 カウントあたりの時間を算出することができる。TDC 時間較正に用いるロジックを図 4.2.4 に示す。

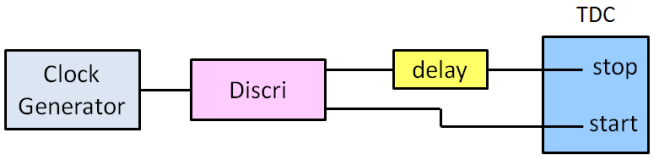


図 4.2.4 TDC 時間較正に用いるロジック

実際に測定した delay 時間と TDC カウント数の関係を表 4.2.1 に示す。このカウント数は 1000 イベントのデータの平均値である。表 4.2.1 を、縦軸を delay、横軸を TDC カウント数としてグラフにしたものは図 4.2.5 である。

図 4.2.5 から、delay 時間と TDC カウント数は比例関係にあることがわかる。線形近似 $y = ax + b$ （ y, x は変数、 a, b は定数）のパラメータ a, b は、それぞれ図 4.2.5 の p_0, p_1 に対応する。よって、 a, b は

$$a = 5.003 \pm 0.004$$
$$b = 0.5 \pm 1.3$$

表 4.2.1　delay-TDC カウント数

delay[ns]	10	50	108	145	218	245	332	436
TDC[counts]	1.59	9.689	21.35	29.044	43.69	49.02	65.67	87.82
delay[ns]	475	500	550	600	700	800	900	1000
TDC[counts]	95.88	100.1	109.7	119.1	139.7	159.6	180	199.6
delay[ns]	1500	2000	2500	3000	3500	4000		
TDC[counts]	299.6	399.5	499.5	599.7	699	799.8		

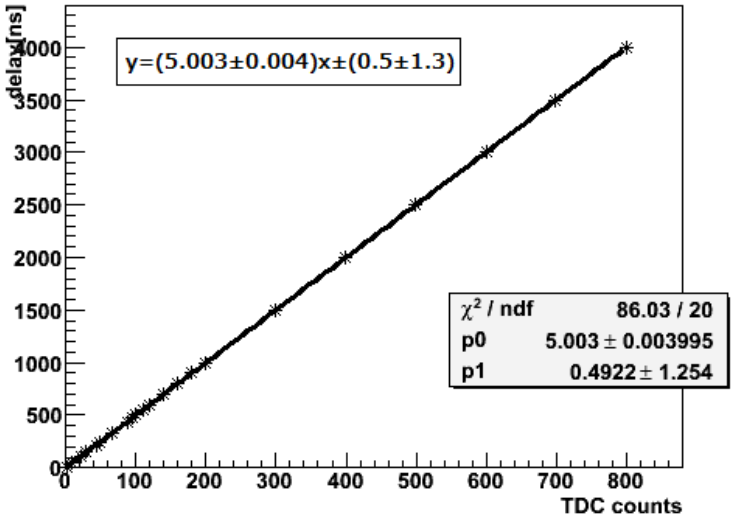


図 4.2.5　TDC 時間較正

となった。この傾き a は TDC1 カウントあたりの時間 α にあたる。よって、

$$\alpha = 5.003 \pm 0.004[\text{ns/count}] \tag{4.3}$$

と求められる。線形近似及び誤差の算出には最小二乗法を用いた。また、 χ^2/ndf は以下となった。

$$\chi^2/\text{ndf} = 86.03/20$$

4.3 結果

μ粒子の寿命を表すTDCカウント数 p_1 と、TDC1カウントあたりの時間αを用いて、μ粒子の平均寿命τは次のように求められる。

$$\tau[\mathrm{ns}] = p_1[\mathrm{ns}] \times \alpha[\mathrm{ns/count}] \tag{4.4}$$

また、平均寿命τの誤差$\Delta\,\tau$は間接誤差の誤差伝播から以下の式で求められる。

$$\frac{\Delta\tau}{\tau} = \sqrt{\left(\frac{\Delta\alpha}{\alpha}\right)^2 + \left(\frac{\Delta p_1}{p_1}\right)^2} \tag{4.5}$$

ここで、$\Delta\ p_1$ は p_1 の誤差、$\Delta\,\alpha$はαの誤差である。

さて、測定から得られた $p_1 = 441.9 \pm 1.3[\mathrm{ns}]$, $\alpha = 5.003 \pm 0.004[\mathrm{ns/count}]$ より、本実験で求めるμ粒子の平均寿命τは以下となった。

$$\tau = 2.211 \pm 0.007[\mu\mathrm{s}] \tag{4.6}$$

τの測定誤差はスロープαの統計誤差と、キャリブレーション p_1 の係数の系統誤差を含んでいる。ここで、 p_1 とαの相対誤差はそれぞれ

$$\frac{\Delta p_1}{|p_1|} = 0.0029$$
$$\frac{\Delta\alpha}{|\alpha|} = 0.0008$$

となる。よって、平均寿命τへの誤差の寄与は p_1 の方が大きい。

また、Particle Data Book(Particle Data Group,2010,Review of Particle Physics,p27)に掲載されているμ粒子の平均寿命は $\tau_{\mathrm{PDG}} = 2.197034 \pm 0.000021[\mu\mathrm{s}]$ である。実験結果と比較すると以下のようになった。

$$\frac{\tau_{\text{観測}}}{\tau_{\mathrm{PDG}}} = \frac{2.211}{2.197} = 1.006 \tag{4.7}$$

よって、文献値とのずれは0.6 %である。

4.4　考察

TDC 分布はただ 1 つの式 2.8 に従うため、実験式 4.1 の Fit により得られる p_1 は本来 Fit 範囲に依らない。そこで、TDC カウント数を 100〜1900 の範囲で 200 ずつ、9 つのブロックに分けてそれぞれ Fit を行った。これにより求められる p_1 及び平均寿命 τ の変化は図 4.4.1 の通りである。

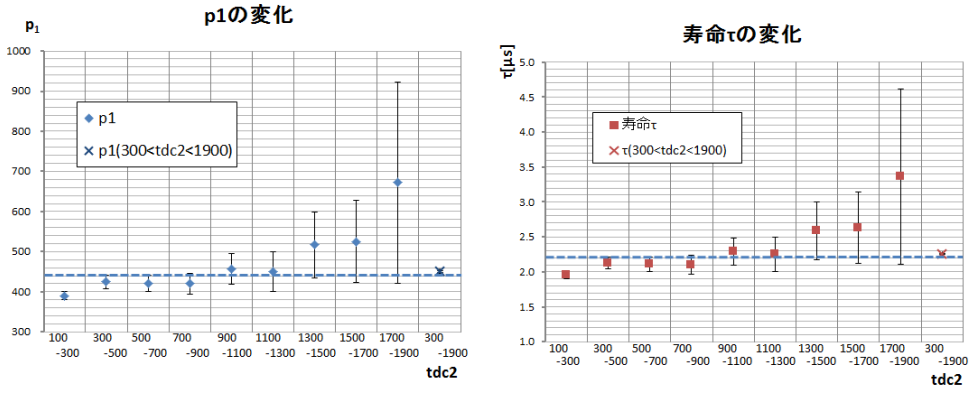


図 4.4.1　TDC 分布への Fit 範囲と p_1 及び平均寿命 τ の変化（点線はセクション 4.3 で示した実験値）

図 4.4.1 から、Fit 範囲と平均寿命 τ に相関関係があるように見られる。ただし、TDC カウント数 300〜1900 範囲は誤差が大きく、エラーの範囲に実験値が含まれるため断定はできない。また、TDC カウント数 100〜300 の範囲のみが実験値から有為にずれている。そこで、100〜300 の範囲を除いた 300〜1900 の範囲で Fit 行い、求めた p_1 と $\tau_{300-1900}$ を以下に示す。また、図 4.4.1 にはそれぞれ右端に示した。

$$\begin{aligned} p_1 &= 450.7 \pm 1.9 \\ \tau_{300-1900} &= 2.255 \pm 0.010 \end{aligned} \tag{4.8}$$

$\tau_{300-1900}$ と、Particle Data Book より $\tau_{\mathrm{PDG}} = 2.197034 \pm 0.000021[\mu\mathrm{s}]$ を比較すると以下のようになった。

$$\frac{\tau_{300-1900}}{\tau_{\mathrm{PDG}}} = \frac{2.255}{2.197} = 1.026$$

よって、文献値とのずれは 2.6 %である。

Fit 範囲と平均寿命 τ の相関関係の原因究明は今後の課題である。

第 5 章

μ粒子から崩壊した電子スペクトラムの測定

5.1 解析

5.1.1 データの解析方法

μ粒子の崩壊により生成される電子のエネルギー分布を測定するために行うことは、主に以下に示す 2 つである。

1. イベントの選別
   測定により得られたデータのうち、今回使用するデータは「μ粒子が崩壊し、かつ放出された電子がメインシンチレーター内にとどまったイベント」のデータである。この条件を満たすイベントを選出しそのデータを用いて解析を行う。
2. ADC のエネルギー較正
   電子のエネルギ-分布を測定する際、S1 と S2 の値の和の ADC 分布を用いる。ADC(e) の channel 数とエネルギー (MeV) の対応を調べ、結果の解析を行う。

以下で、これらについて説明する。

5.1.2　イベントの選別

測定により得られたデータのうち使用するデータを選出する。選出するイベントの条件と、その条件を満たすイベントの選出方法を以下で述べる。

条件

1. μ粒子がメインシンチレーター内に入射する。
2. μ粒子がメインシンチレーター内で崩壊し、電子が放出される。
3. 放出された電子がメインシンチレーター内にとどまっている。

このイベントが選出されるようにヒストグラムを作成するプログラム内で指定した。

条件の選出方法

1. μ粒子がメインシンチレーター内に入射する。
   4.2.1 で示した条件を用いた。
2. μ粒子がメインシンチレーター内で崩壊し、電子が放出される。
   4.2.1 で示した条件を用いた。
3. 放出された電子がメインシンチレーター内にとどまっている。
   ベトーカウンターがなっていないイベントを選出した。つまり、T1,T2,V1～V8 の ADC 分布からカウンターが反応しているとみなす値を決定し、その値を下回っているイベントを選出した。以下の図 5.1.1～図 5.1.10 にそれぞれのカウンターの ADC 分布を示す。横軸が channel 数、縦軸が log スケールでエントリー数を示している。

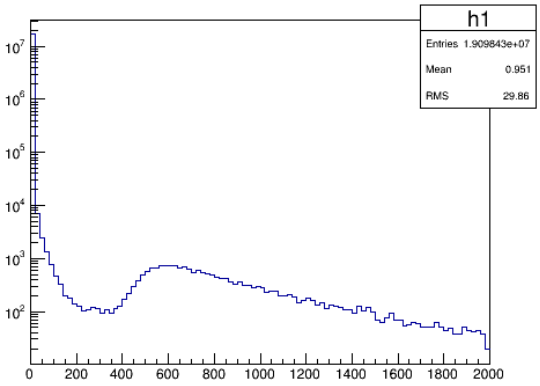


図 5.1.1 T1 の ADC 分布

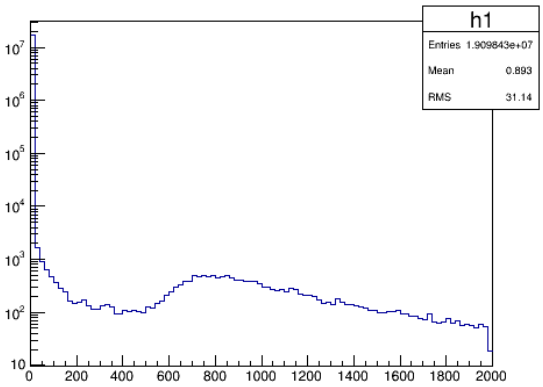


図 5.1.2 T2 の ADC 分布

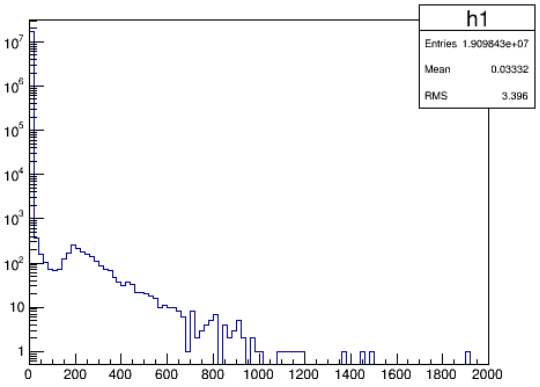


図 5.1.3 V1 の ADC 分布

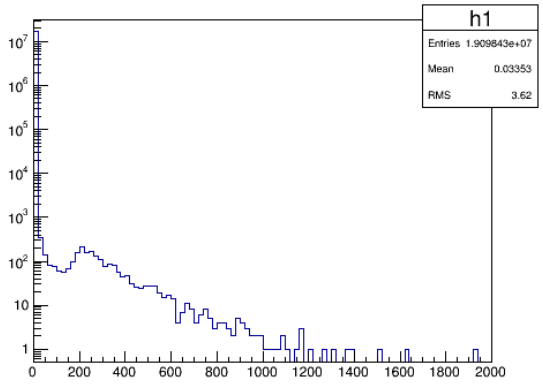


図 5.1.4 V2 の ADC 分布

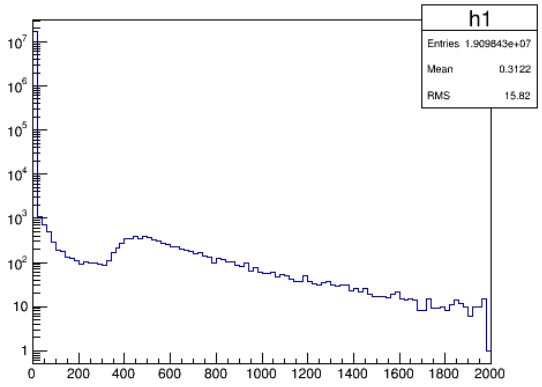


図 5.1.5 V3 の ADC 分布

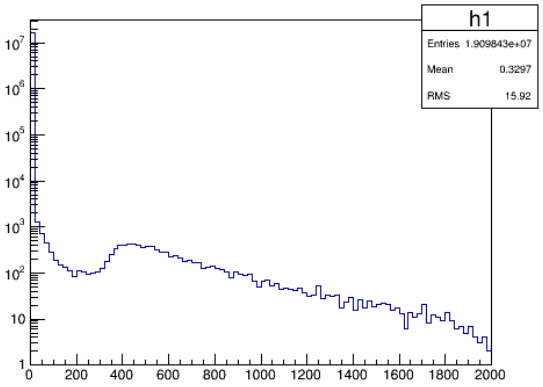


図 5.1.6 V4 の ADC 分布

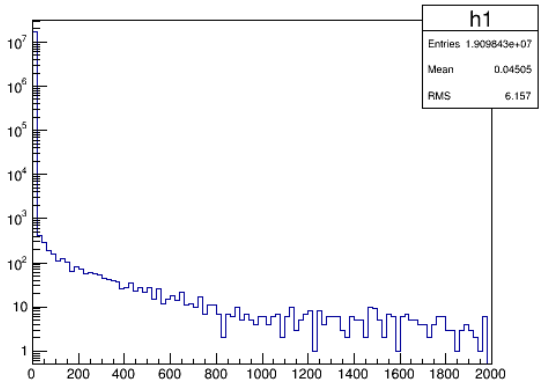


図 5.1.7　V5 の ADC 分布

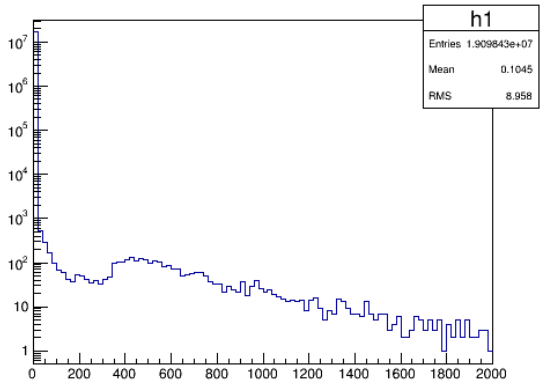


図 5.1.8　V6 の ADC 分布

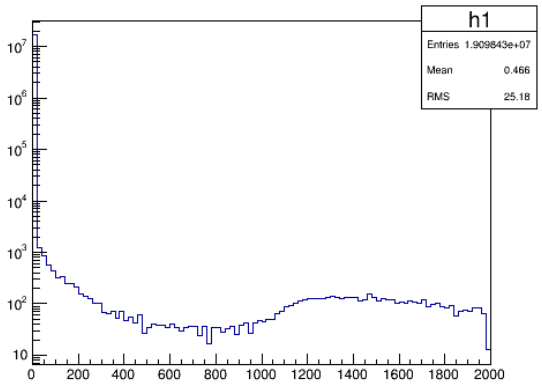


図 5.1.9　V7 の ADC 分布

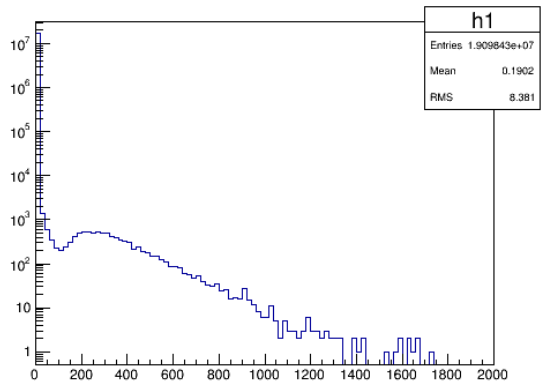


図 5.1.10　V8 の ADC 分布

よって、ベトーカウンターが反応したとみなす ADC 値は、

T1	T2	V1	V2	V3	V4	V5	V6	V7	V8
110	170	80	70	100	90	85	130	210	90

である。全てのベトーカウンター T1,T2,V1～V8 の ADC 値がそれぞれの値を下回ったとき、電子がメインシンチレーター内でとどまっているとみなした。

各条件でのイベントの数とμ粒子がメインシンチレーター内に入射したときを全体としたときの割合を以下の表に示す。

	イベント数	割合 (%)
μ粒子がメインシンチレーター内に入射する	19098430	
μ粒子がメインシンチレーター内で崩壊する	174237	0.91
電子がメインシンチレーター内にどまっている	100717	0.53

5.1.3 通過した μ 粒子を用いた ADC のエネルギー較正

ADC(e) についてエネルギー較正を行うために、μ 粒子がメインシンチレーターを貫通した際の μ 粒子のエネルギー損失を測定した。使用するデータは「μ 粒子が崩壊せずにメインシンチレーターを通過したイベント」のデータである。このイベントの条件と、条件を満たすイベントの選出方法を以下で示す。ここで、宇宙線の角度分布と運動量分布は無視し、ADC による個体差をなくすために ADC(e) で μ 粒子のエネルギーを測定した。(ADC(μ) では電子のエネルギーを測定した。)

条件

1. μ 粒子がメインシンチレーター内に入射する。
2. μ 粒子がメインシンチレーター内で崩壊しない。(電子が放出されない。)
3. μ 粒子が T2 のカウンターを通過する。

このイベントが選出されるようにヒストグラムを作成するプログラム内で指定した。

条件の選出方法

1. μ 粒子がメインシンチレーター内に入射する。
   4.2.1 で示した条件を用いた。
2. μ 粒子がメインシンチレーター内で崩壊しない。(電子が放出されない。)
   ADC(μ) の S1 の ADC 分布からカウンターが反応しているとみなす値を決定し、その値を下回っているイベントを選出した。図 5.1.11 に S1 の ADC 分布を示す。
3. μ 粒子が T2 のカウンターを通過する。
   ADC(e) の T2 の ADC 分布からカウンターが反応しているとみなす値を決定し、その値を上回っているイベントを選出した。図 5.1.12 に T2 の ADC 分布を示す。

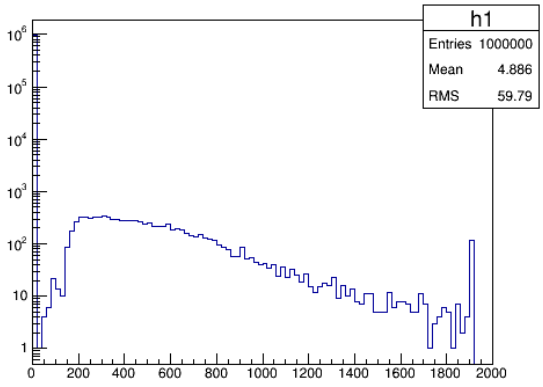


図 5.1.11 ADC(μ) の S1 の ADC 分布

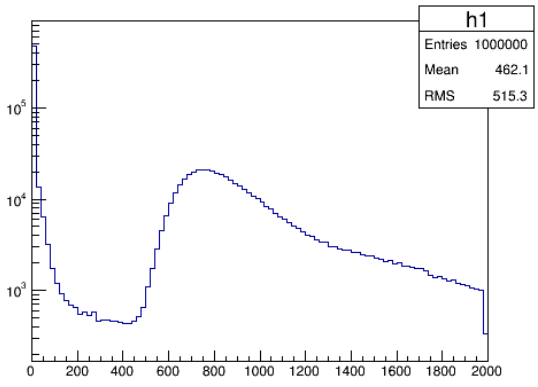


図 5.1.12 ADC(μ) の T2 の ADC 分布

これより、S1 は channel 数 140 以上のとき反応しているとみなし、T2 は channel 数 500 以上のとき反応しているとみなした。

よって、上記の条件 1〜3 を満たすイベント数は 10 万イベント中で 483293 イベントであった。μ 粒子のエネルギー分布、つまり ADC(e) の S1 と S2 の値の和の ADC 分布を図 5.1.13 に示す。

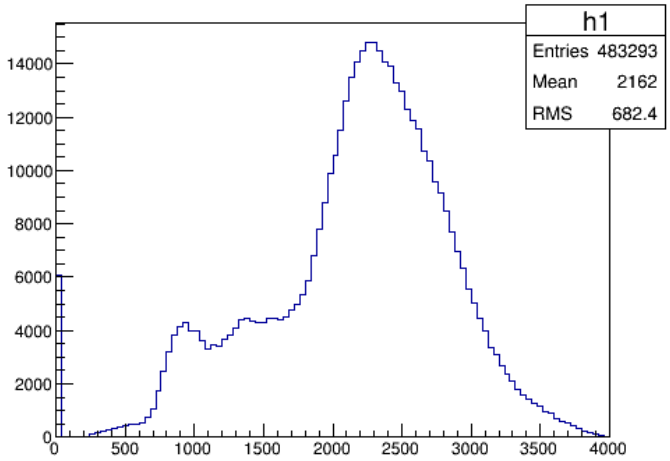


図 5.1.13　メインシンチレーターを通過した μ 粒子のエネルギー損失の分布

図 5.1.13 より、channel 数が 2300 あたりに分布のピークがあることがわかった。原理で示したように、μ 粒子がメインシンチレーターを垂直に通過した際のエネルギー損失は約 53.3MeV であった。よって図 5.1.13 で示した μ 粒子のエネルギー分布のピーク位置である chanel 数 2300 あたりのエネルギーが約 50MeV であることがわかった。

5.2　結果

μ 粒子の崩壊により生成された電子のエネルギー分布の測定結果を図 5.2.1 に示す。一方、図 5.2.2 は式 2.19 で与えられる静止した μ 粒子から放出される電子のエネルギースペクトラムの理論値である。

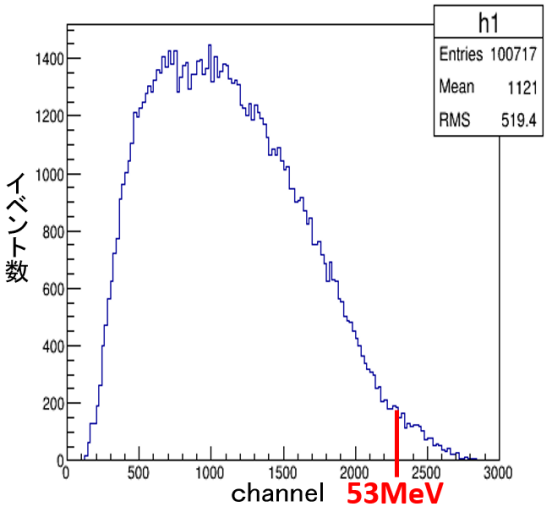


図 5.2.1　電子のエネルギースペクトラムの測定結果

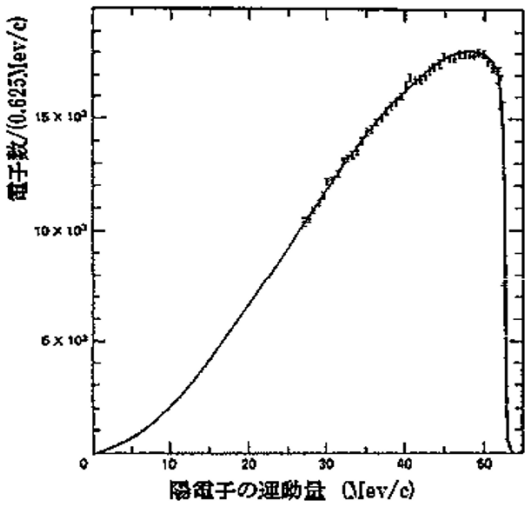


図 5.2.2　電子のエネルギースペクトラムの理論値

図 5.2.2 と図 5.2.1 を比較すると、理論値と測定結果ではピーク位置も分布の概形も異なっていることがわかった。ピーク位置については、理論値の分布が約 50MeV であるのに対し、測定結果の分布では 50MeV よりも低いエネルギーの位置にある。また分布の概形については、図 5.2.1 の約 50MeV のイベント数がピーク位置に比べてかなり低く、測定結果の分布は理論値と比べて高エネルギー部の割合が少ない結果となった。

5.3　シミュレーションによる検討

実験結果より、電子のエネルギー分布の理論値と測定結果には違いがみられた。そこで Geat4 を用いて本実験を再現し、μ 粒子の崩壊過程をシミュレーションする。シミュレーション結果を用いて、理論値に比べて測定結果のエネルギー分布で高エネルギー部の割合が少ない原因について検討する。以下でシミュレーション方法について説明する。

5.3.1　Geant4

Geant4(http://geant4.web.cern.ch/geant4/) は、粒子と物質との相互作用のシミュレーションを行うプログラムである。特徴としては、幾何学的な記述が優れ、新しい反応過程組み込みが容易で、ソースが公開されている。高エネルギー実験のみでなく、粒子線治療、宇宙ステーション、火星基地などの宇宙計画、宇宙線、原子核、地下実験などへの応用に用いられている。今回行うシミュレーションは、μ 粒子をプラスチックシンチレーターに入射させ、e、ν_μ、$\overline{\nu}_e$ に崩壊する事象とする。Geant4 を用いて本実験のセットアップをシミュレートしたプログラムを付録 F に示す。

5.3.2　シミュレーションの設定

シミュレーションを行うにあたって、入射粒子である μ 粒子のエネルギーや進行方向、検出器の種類を設定する必要がある。以下で今回のシミュレーションの設定条件を示す。

1. 検出器の種類
   実験で使用したメインシンチレーターを再現する。58cm×26cm×25cm のプラスチックシンチレーターを検出器として設定した。図 5.3.1 に Geant4 で設定した検出器の模式図を示す。

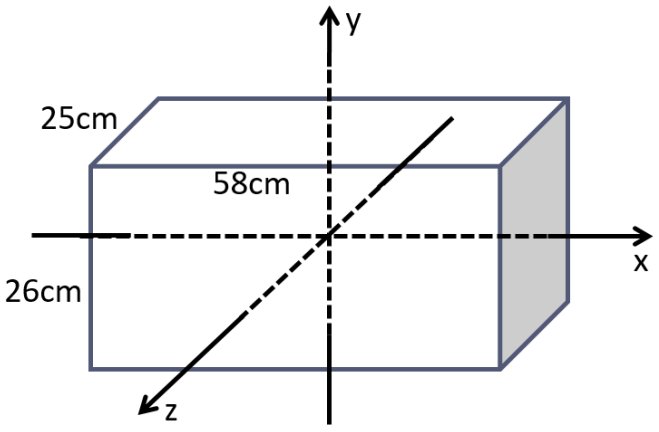


図 5.3.1　Geant4 で定義した検出器の模式図 (プラスチックシンチレーター)

2. 入射粒子の位置

入射位置 (x , y , z) は検出器の真ん中を 0 とした座標で指定する。今回は図 5.3.2 に示すように、検出器の上 3cm の高さでの xz 平面上の範囲で入射位置を指定した。x 座標は-29cm～29cm の範囲での一様乱数で指定し、y 座標は 16cm の位置で固定、 z 座標は-12.5cm～12.5cm の範囲での一様乱数で指定した。

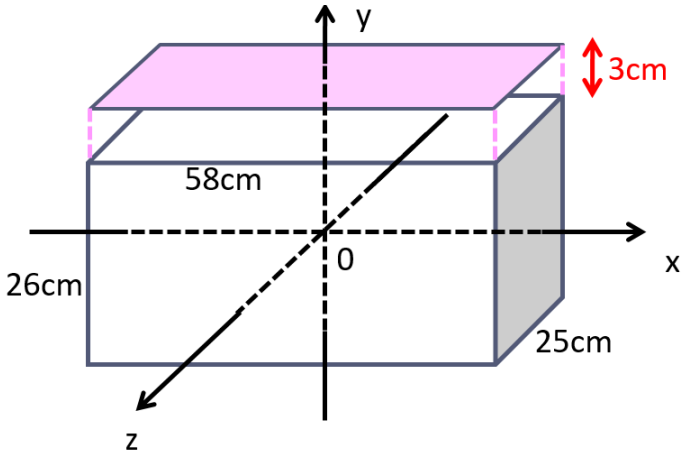


図 5.3.2　入射粒子の位置の範囲

3. 入射粒子の進行方向

入射粒子の進行方向は、入射粒子の運動量方向 (xm,ym,zm) をベクトルとして指定する。xm、ym、zm はそれぞれ x 軸、y 軸、z 軸方向のベクトルの大きさである。今回、 y 軸方向は下向きの大きさ 1 の単位ベクトルで固定した。図 5.3.3 に示すように角度 θ、ϕ を指定し、xm、ym、zm を決定した。

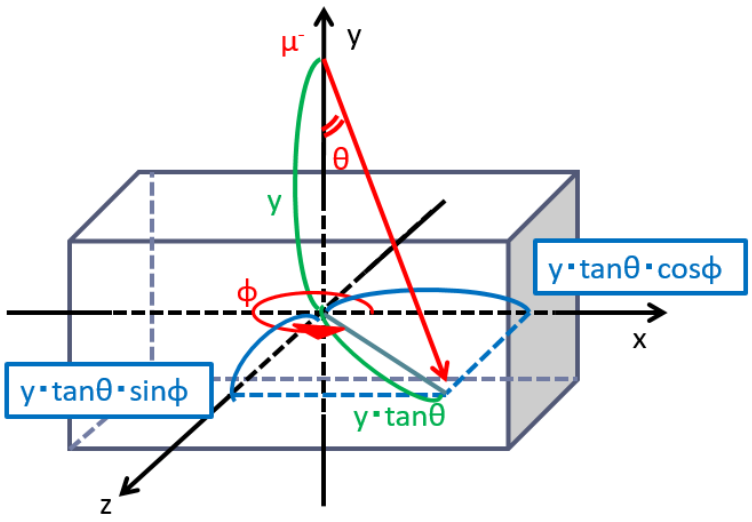


図 5.3.3　入射粒子の運動量方向

図 5.3.3 より、運動量方向の成分 xm、ym、zm は以下の式でそれぞれ求められる。

$$\mathrm{xm} = \mathrm{ym} \times \tan\theta \times \cos\phi$$
$$\mathrm{ym} = \text{-}1$$
$$\mathrm{zm} = \mathrm{ym} \times \tan\theta \times \sin\phi$$

ここで宇宙線の入射角について考える。図 5.3.4 で示したように鉛直上向きを 0° とした角度を天頂角といい、宇宙線の天頂角を θ とする。地表に到達する宇宙線の数は天頂角 θ=0 が最大で、これは大気を通過する距離が短いために崩壊せずに地表まで辿り着くからである。

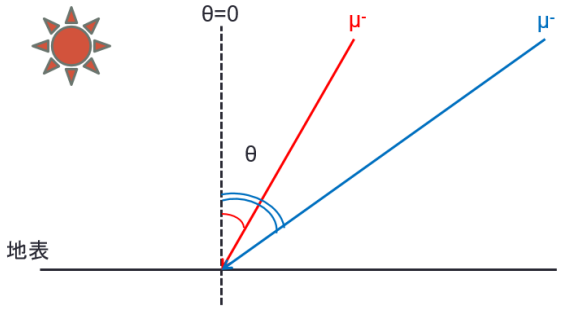


図 5.3.4　宇宙線の地表付近の様子

単位面積・単位時間・単位立体角あたりの粒子の個数を強度 J(θ) とすると、天頂角 θ から飛来する μ 粒子の強度 J(θ) は

$$\mathrm{J}(\theta) = \mathrm{J}(0)\ \cos^2\theta$$

と表される。ここで J(0) は鉛直方向から飛来する μ 粒子の強度であり、

$$\mathrm{J}(0) = 0.83\times\ 10^{-2}(\text{個}/\mathrm{cm}^2\cdot\mathrm{sec}\cdot\mathrm{sr})$$

である。よって、飛来する μ 粒子の個数 J(θ) は $\cos^2\theta$ に比例することがわかる。

これより、天頂角 θ を 0〜$\frac{\pi}{2}$ の範囲で $\cos^2\theta$ に従う分布で乱数を発生させて指定した。角度 ϕ は 0〜2π の範囲で一様乱数を発生させて指定した。

4. 入射粒子のエネルギー

原理で示したように、μ 粒子がメインシンチレーターを垂直に通過した際のエネルギー損失は約 53.3MeV であった。地表に到達する宇宙線の数は天頂角 θ=0 が最大であるので、ほとんどが垂直にメインシンチレーターに入射していると考え、シミュレーションでの入射粒子のエネルギーを 53.3MeV $\fallingdotseq$ 50MeV とした。

5.3.3　シミュレーション結果

5.3.2 で示したように検出器や入射粒子を決定し、約 7 万発の粒子を検出器に入射させ、シミュレーションを行った。結果を以下に示す。図 5.3.5 は μ 粒子が崩壊した直後の電子の運動エネルギー、図 5.3.6 は電子のエネルギー損失の合計である。

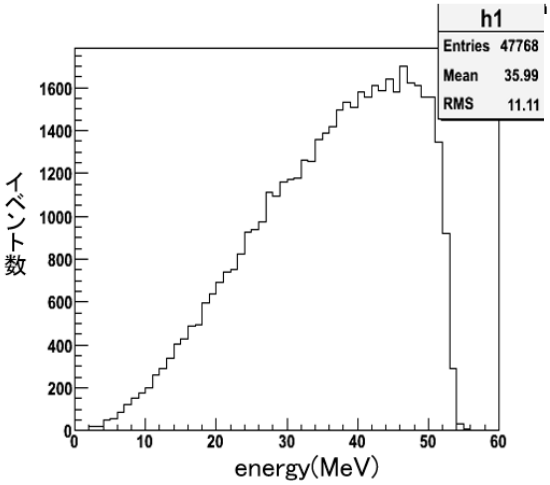


図 5.3.5　μ 粒子が崩壊した直後の電子の運動エネルギー

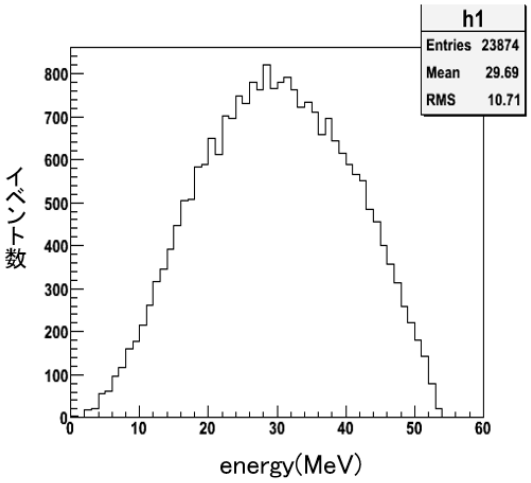


図 5.3.6　電子のエネルギー損失の合計

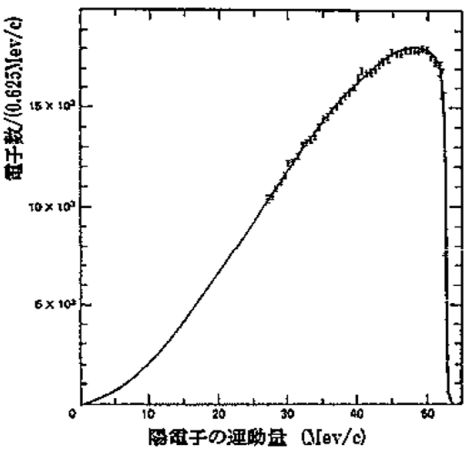


図 5.3.7　電子のエネルギー分布の理論値

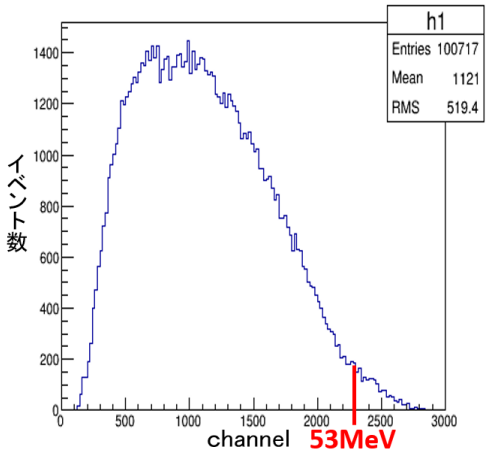


図 5.3.8　電子のエネルギー分布の測定結果

図 5.3.5 は崩壊直後に電子が持っている運動エネルギーで、図 5.3.7 とよく一致している。図 5.3.6 は本実験を再現したもので、図 5.3.5 と比較してエネルギーが下がっている主な理由は次の考察でも述べるように、制動放射の影響である。図 5.3.6 と図 5.3.5 が異なる分布になる原因を探ることで、理論値と測定結果が異なる原因を考察する。

5.4　考察

測定結果は理論値に比べて高エネルギー部が少ない分布になっていた。この原因について、シミュレーション結果を用いて考察する。

1. 制動放射の影響
   50MeV の μ 粒子を検出器に入射させた際の μ 粒子崩壊にともなう各粒子の振る舞いを図 5.4.1 に示す。

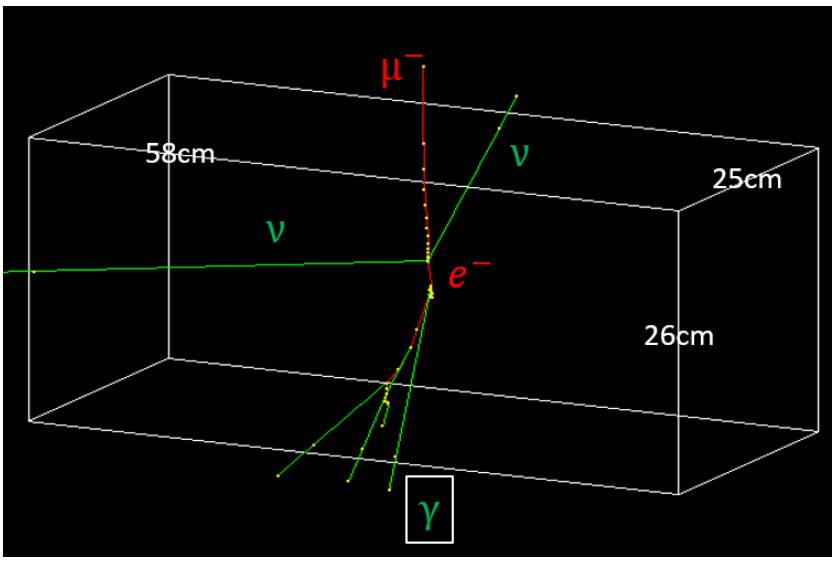


図 5.4.1　μ 粒子を検出器に入射させた際の崩壊の様子

図 5.4.1 より、μ 粒子の崩壊によって放出された電子はメインシンチレーター内を進む中で γ 線を放出していることがわかった。これは制動放射の影響で、電子はメインシンチレーターを進む際に γ 線からエネルギーを奪われていると考えられる。

ここで、γ 線にエネルギーを奪われる前の電子のエネルギー分布と γ 線にエネルギーを奪われた後の電子のエネルギー分布、つまり図 5.3.5 と図 5.3.6 を比較する。

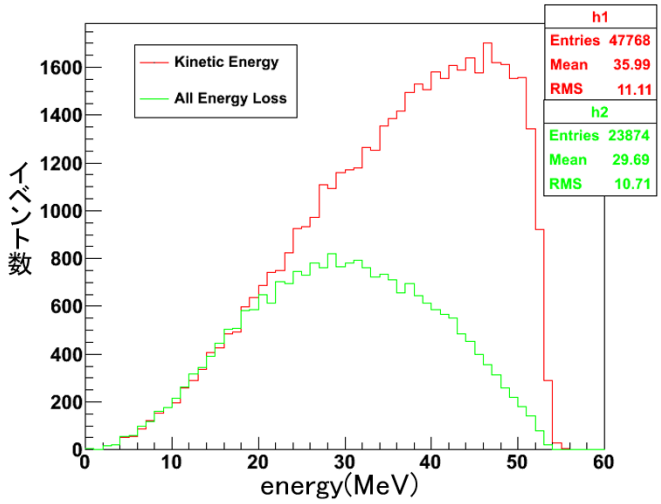


図 5.4.2 崩壊直後の電子のエネルギーの分布と電子のエネルギー損失の合計の分布

図 5.4.2 は、赤が図 5.3.5、緑が図 5.3.6 を示している。γ 線にエネルギーを奪われる前後では明らかにエネルギーの変化があり、メインシンチレーターを進む中で電子が γ 線にエネルギーを奪われていることがわかる。理論値に比べて測定結果の高エネルギー部のイベントが少なくなっている原因の 1 つであると考えられる。

2. 検出器外に飛び出た電子

図 5.3.6 および図 5.3.8 の測定では、放出された電子がメインシンチレーター内にとどまっているイベントのデータのみを扱っている。そこで、電子がメインシンチレーター内にとどまっているイベントにおける崩壊直後の電子の運動エネルギーについて考える。

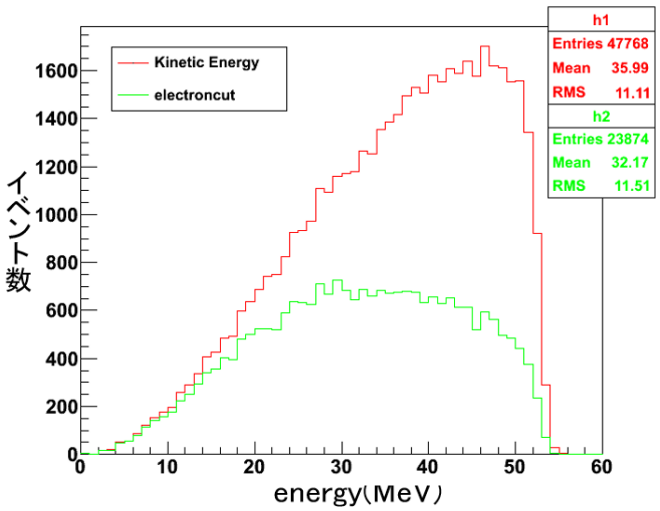


図 5.4.3　μ 粒子崩壊直後の電子のエネルギー分布

図 5.4.3 は、赤が全イベントにおける崩壊直後の電子の運動エネルギーの分布、緑が電子がメインシンチレーターにとどまっているイベントにおける崩壊直後の電子の運動エネルギーの分布を示している。これより、崩壊直後に高い運動エネルギーを持っていたものがメインシンチレーターから飛び出しやすいことがわかる。実験ではそのイベントをカットしたので、これは理論値に比べて測定結果の高エネルギー部のイベントが少なくなっている原因の 1 つであると考えられる。

以上より、測定結果の電子のエネルギー分布が理論値に比べて高エネルギー部が少ない分布になっている原因として、制動放射とメインシンチレーターから飛び出た電子のイベントのカットの影響が考えられた。

しかしながら、Geant4 のシミュレーションと実験値を比較すると、シミュレーションによる本実験の再現性は良くない。原因としては、入射粒子のエネルギーを全て 50MeV と指定した点が考えられる。実際にメインシンチレーターに入射する宇宙線はエネルギーにばらつきがあるので、その点を考慮する必要がある。その他の源因も考え、シミュレーションの再現性を高めることによって測定結果と理論値が異なる原因を究明することが今後の課題である。

第 6 章

まとめ

6.1　実験のまとめ

1. μ 粒子の寿命測定
   μ 粒子の寿命 τ は $2.211 \pm 0.007(\mu\mathrm{s})$ となった。
   また、文献値 $\tau_{\mathrm{PDG}} = 2.197034 \pm 0.000021[\mu\mathrm{s}]$ と比較すると、0.6 %のずれが生じた。Fit 範囲の決定は今後の課題である。

2. 電子のエネルギースペクトラムの測定
   理論値と実験結果では分布の概形もピーク位置も異なった。理論値に比べて実験結果の高エネルギー部が少ない分布になる原因は、制動放射の影響と放出された電子がメインシンチレーターを飛び出た電子のイベントをカットした影響が考えられた。他にも要因があると思われ、シミュレーションの再現性を高めることで理論値と実験結果が異なる要因を究明することが今後の課題である。

6.2　謝辞

私たちの卒業研究のため、お忙しい中熱心にご指導してくださった林井先生をはじめ、研究するにあたって相談にのっていただいた宮林先生、様々な視点からアドバイスを下さった下村先生、そしていつも気にかけてくださった諸先輩方に感謝しております。この 1 年で学んだことを大学院でも生かし、さらに精進していきたいと思います。本当にありがとうございました。

6.3　参考文献

[1]「2014 年度卒業論文 μ 粒子の寿命と電子エネルギースペクトラムの測定」 奈良女子大学 北内久美 太地彩 武田明莉

[2]「2013 年度卒業論文 μ 粒子の寿命と電子エネルギースペクトラムの測定」 奈良女子大学 新井智穂 長谷川香織

[3]「2012 年度卒業論文 μ 粒子の寿命と電子エネルギースペクトラルの測定」 奈良女子大学 小川加奈 田中恵梨香

[4]「Geant4 の使い方入門」 奈良女子大学 修了 岩下友子

[5]「2001 年度卒業論文 Geant4 による素粒子実験シミュレーション」 信州大学 入江隆平

[6]「Geant4 Tutorial of install and set」 金賀史彦

[7]「宇宙線を目で見よう スパークチェンバーの制作」 高エネルギー加速器研究機構サマーチャレンジ演習課題 2

[8]「Technique for Nuclear and Particle Physics Experiments」 William R.Leo

[9]「理科年表 平成 24 年・第 85 冊」p.489 国立天文台

[10]「Review of Particle Physics」p.27,127 Particle Data Group

付録 A

ele.c
CAMAC からのデータ収集

```
/***** ele.c ***** created 2012/Feb./1th**********************
 * Original version was written by S.Ono 2002/Jan./27th
 * This is simplified version having only CAMAC control/IO part.
 * LAM clear was moved to outside of "if(q!=0)". 2003/Jul./9th
 * Update for stopped muon spectrum reading. 2013/11/27
 * Delete not need sentence. 2015/11/6
 ************************************************************/
/**** original title comment *****
  *******************************/
#include <fcntl.h>
#include <sys/time.h>
#include <unistd.h>
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <sys/errno.h>
#include "camlib.h"
#include <time.h>
#include <netinet/in.h>

FILE *fp; /* The file to save the taken data.*/

int main(){
  int i_ev, n_ev;
  int q,x,data21,data22,data23,data24,data31,data32,data33; //,data34; //data34:TDC 時間
較正
  int data1,data2,data3,data4,data5,data6,data7,data8,data9,data10,data11,data12;
  int cadcqn1,cadcqn2,ctdcqn;
  int ch0,ch1,ch2,ch3,ch4,ch5,ch6,ch7,ch8,ch9,ch10,ch11;
  int lamsrc;
  int lamch;

  char fname[36];
```

```
int qq;
int head;

 cadcqn1=1; /* muon ADC-Qmode module number */
 cadcqn2=2; /* electron  ADC-Qmode module number */
 ctdcqn=3; /* TDC  module number */
 ch0=0;
 ch1=1;
 ch2=2;
 ch3=3;
 ch4=4;
 ch5=5;
 ch6=6;
 ch7=7;
 ch8=8;
 ch9=9;
 ch10=10;
 ch11=11;

 /*=====
  * Ask the file name to save the taken data.
  * Also open the data file.
  *======*/
    printf("File name to save data?\n");
      scanf("%s",fname);
   fp=fopen(fname, "w");

 /*=====
  * How many events do you take?
  *=====*/
         printf("Number of events?\n");
        scanf("%d",&n_ev);
        fprintf(fp,"%d\n",n_ev);

  /*=====
   * Open CCP interface device file.
   * If it fails, exit.
   *======*/
 if(COPEN()){
    printf("ccp open error\n");
    exit(-1);
    }

  /*=====
   * Initialize CAMAC.
   *======*/
 CSETCR(0);
 CGENZ();
 CGENC();
 CREMI();

 lamsrc=cadcqn1;
 //lamsrc=cadcqn2;
 lamch=ch0;
```

```
 printf("LAM source : %d,%d\n" ,lamsrc,lamch);

/*=====
 * Again send enable command to prepare the first event.
 *======*/
CAMAC(NAF(cadcqn1,ch0,26),&data1,&q,&x); /* F=26 is enable. */
CAMAC(NAF(cadcqn2,ch0,26),&data1,&q,&x); /* F=26 is enable. */
CAMAC(NAF(ctdcqn,ch0,26),&data1,&q,&x); /* F=26 is enable. */

/*=====
 * send message to the user.
 *======*/
printf("CAMAC initilize done.\n");
printf("Number of event = %d\n",n_ev);

/*====
 * Event loop.
 *=====*/
i_ev = 1;
while( i_ev <= n_ev )
{
 CAMAC(NAF(cadcqn1,ch1,9),&data1,&q,&x); /* F=9 LAM and module clear. */
 CAMAC(NAF(cadcqn2,ch0,9),&data1,&q,&x); /* F=9 LAM and module clear. */
 CAMAC(NAF(ctdcqn,ch0,9),&data1,&q,&x); /* F=9 LAM and module clear. */
}

   /*---------
   * Test LAM.
   *---------*/
  do {
  CAMAC(NAF(lamsrc,lamch,8),&data1,&q,&x);  /* F=8 is test LAM.*/
  } while ( q==0);

  /*------
   * If no event comes yet, q is set to be 0,
   * otherwise, the digitized event is there!
   *-----*/

    /*-----
     * Read the digitized data from the register.
     *----*/
    CAMAC(NAF(cadcqn1,ch0,0),&data21,&qq,&x);
    CAMAC(NAF(cadcqn1,ch1,0),&data22,&qq,&x);
    CAMAC(NAF(cadcqn1,ch2,0),&data23,&qq,&x);
    CAMAC(NAF(cadcqn1,ch3,0),&data24,&qq,&x);
    CAMAC(NAF(cadcqn2,ch0,0),&data1,&qq,&x);
    CAMAC(NAF(cadcqn2,ch1,0),&data2,&qq,&x);
    CAMAC(NAF(cadcqn2,ch2,0),&data3,&qq,&x);
    CAMAC(NAF(cadcqn2,ch3,0),&data4,&qq,&x);
    CAMAC(NAF(cadcqn1,ch4,0),&data5,&qq,&x); //ch4~ch11 が cadcqn1 のとき ADC(μで V1～
V8 を読む
    CAMAC(NAF(cadcqn1,ch5,0),&data6,&qq,&x);
    CAMAC(NAF(cadcqn1,ch6,0),&data7,&qq,&x);
```

```
CAMAC(NAF(cadcqn1,ch7,0),&data8,&qq,&x);
CAMAC(NAF(cadcqn1,ch8,0),&data9,&qq,&x);
CAMAC(NAF(cadcqn1,ch9,0),&data10,&qq,&x);
CAMAC(NAF(cadcqn1,ch10,0),&data11,&qq,&x);
CAMAC(NAF(cadcqn1,ch11,0),&data12,&qq,&x);
CAMAC(NAF(ctdcqn,ch0,2),&data31,&qq,&x);
CAMAC(NAF(ctdcqn,ch1,2),&data32,&qq,&x);
CAMAC(NAF(ctdcqn,ch2,2),&data33,&qq,&x);
//CAMAC(NAF(ctdcqn,ch3,2),&data34,&qq,&x); //TDC 時間較正

/*-----
 * Update the event counter(i_ev), and send message for
 * every 50 events or there are data for big electron signal.
 *-----*/
if( i_ev%100 == 1 )
 {
  printf("Event= %d\n",i_ev);
  printf("data21= %d\n",data21);
  printf("data22= %d\n",data22);
  printf("data23= %d\n",data23);
  printf("data24= %d\n",data24);
  printf("data1= %d\n",data1);
  printf("data2= %d\n",data2);
  printf("data3= %d\n",data3);
  printf("data4= %d\n",data4);
  printf("data5= %d\n",data5);
  printf("data6= %d\n",data6);
  printf("data7= %d\n",data7);
  printf("data8= %d\n",data8);
  printf("data9= %d\n",data9);
  printf("data10= %d\n",data10);
  printf("data11= %d\n",data11);
  printf("data12= %d\n",data12);
  printf("data31= %d\n",data31);
  printf("data32= %d\n",data32);
  printf("data33= %d\n",data33);
  //printf("data34= %d\n",data34); //TDC 時間較正
  }

  head=-1;

/*-----
 * The read data is written into the file.
 *-----*/
  // if( data1>10 ) /* Cut of event of not decay to electron */
  {
fprintf(fp," %d",head);
fprintf(fp," %d",i_ev);
fprintf(fp," %d",data21);
fprintf(fp," %d",data22);
fprintf(fp," %d",data23);
fprintf(fp," %d",data24);
fprintf(fp," %d",data1);
fprintf(fp," %d",data2);
```

```
      fprintf(fp," %d",data3);
      fprintf(fp," %d",data4);
      fprintf(fp," %d",data5);
      fprintf(fp," %d",data6);
      fprintf(fp," %d",data7);
      fprintf(fp," %d",data8);
      fprintf(fp," %d",data9);
      fprintf(fp," %d",data10);
      fprintf(fp," %d",data11);
      fprintf(fp," %d",data12);
      fprintf(fp," %d",data31);
      fprintf(fp," %d",data32);
      fprintf(fp," %d\n",data33);
      //fprintf(fp," %d\n",data34); //TDC 時間較正
        }

    /*-----
     * Clear LAM to wait for the next event.
     *   fuc=9 ; LAM clear for usual module
     *   FADC
     *    fuc=9 ; address clear
     *       =10 ; LAM clear
     *-----*/

    i_ev++;

   }/*   end of event loop  */

/*=======
 Instructions for termination.
*======*/
  CAMAC(NAF(lamsrc,lamch,24),&data1,&q,&x); /* F=24 is desable lam.*/
  CAMAC(NAF(cadcqn1,ch1,24),&data1,&q,&x); /* F=24 is desable lam.*/
  CAMAC(NAF(cadcqn2,ch1,24),&data1,&q,&x); /* F=24 is desable lam.*/
  CAMAC(NAF(ctdcqn,ch1,24),&data1,&q,&x); /* F=24 is desable lam.*/

  CCLOSE();                          /* CAMAC close. */
  fclose(fp);                        /* Close data file.*/

  return 0;
```

付録 B

rootfileOutput.cc CAMAC からのデータを root ファイルへ出力

```
#include "fstream"
#include "TNtuple.h"
#include "TH1.h"
#include "TH2.h"
#include "TCanvas.h"
#include "TFile.h"
#include <string>
#include "TTree.h"
using namespace std;

void rootfileOutput(){

 string    fname=“2016-01-29-thre-016”; //参照ファイル名
 string   directory1="/home/2015-b4/ug2015/data2015/new/1000000Events/";
 string   directory2="/home/2015-b4/ug2015/data2015/new/1000000Events/rootfile/";
 string   full_name1 = directory1+ fname; //データファイルの絶対パス
 string   full_name2 = directory2+ fname+ ".root"; //root ファイルの絶対パス
 cout <<" f-name"<< full_name1<<endl;

 ifstream data(full_name1.c_str()); //ファイルを開く
 float
one,ev,mus1,mus2,mut1,mut2,es1,es2,et1,et2,ev1,ev2,ev3,ev4,ev5,ev6,ev7,ev8,tdc0,tdc1,tdc2;
 //型宣言
 TTree *mu_t=new TTree("mu_t","title"); //Tree を定義

 mu_t->Branch("one", &one, "one/F"); //Branch(箱) を定義

 mu_t->Branch("ev", &ev, "ev/F");
 mu_t->Branch("mus1", &mus1, "mus1/F");
 mu_t->Branch("mus2", &mus2, "mus2/F");
 mu_t->Branch("mut1", &mut1, "mut1/F");
 mu_t->Branch("mut2", &mut2, "mut2/F");
 mu_t->Branch("es1", &es1, "es1/F");
```

```
mu_t->Branch("es2", &es2, "es2/F");
mu_t->Branch("et1", &et1, "et1/F");
mu_t->Branch("et2", &et2, "et2/F");
mu_t->Branch("ev1", &ev1, "ev1/F");
mu_t->Branch("ev2", &ev2, "ev2/F");
mu_t->Branch("ev3", &ev3, "ev3/F");
mu_t->Branch("ev4", &ev4, "ev4/F");
mu_t->Branch("ev5", &ev5, "ev5/F");
mu_t->Branch("ev6", &ev6, "ev6/F");
mu_t->Branch("ev7", &ev7, "ev7/F");
mu_t->Branch("ev8", &ev8, "ev8/F");
mu_t->Branch("tdc0", &tdc0, "tdc0/F");
mu_t->Branch("tdc1", &tdc1, "tdc1/F");
mu_t->Branch("tdc2", &tdc2, "tdc2/F");

data>>one; //一番上の合計イベント数だけ読む（ループから抜くため）
int i;
while( data >> one >> ev >> mus1 >> mus2 >> mut1 >> mut2 >> es1 >>
es2 >> et1 >> et2 >> ev1 >> ev2 >> ev3 >> ev4 >> ev5 >>
ev6 >> ev7 >> ev8 >> tdc0 >> tdc1 >> tdc2){

  mus1=mus1-106; //ペデスタルを引く
  if(mus1<0){mus1=0;} //ペデスタルを引いて負になる場合 0 にする
  mus2=mus2-96;
  if(mus2<0){mus2=0;}
  mut1=mut1-88;
   if(mut1<0){mut1=0;}
  mut2=mut2-85;
  if(mut2<0){mut2=0;}
  es1=es1-57;
   if(es1<0){es1=0;}
  es2=es2-63;
   if(es2<0){es2=0;}
  et1=et1-57;
   if(et1<0){et1=0;}
  et2=et2-61;
   if(et2<0){et2=0;}
  ev1=ev1-68;
   if(ev1<0){ev1=0;}
  ev2=ev2-64;
   if(ev2<0){ev2=0;}
  ev3=ev3-64;
   if(ev3<0){ev3=0;}
  ev4=ev4-63;
   if(ev4<0){ev4=0;}
  ev5=ev5-65;
   if(ev5<0){ev5=0;}
  ev6=ev6-65;
   if(ev6<0){ev6=0;}
  ev7=ev7-66;
   if(ev7<0){ev7=0;}
  ev8=ev8-74;
   if(ev8<0){ev8=0;}
```

```
   if( i%100000==1 ){
   cout<<"i="<<i<<" one="<<one<<" ev="<<ev<<" mus1="<<mus1<<" mus2="<<mus2<<endl; //画
面へ出力
}
   mu_t->Fill();
   i++;
 };

 data.close(); //ファイル閉じる
 TFile *rtfilemt = new TFile(full_name2.c_str(),“RECREATE”); //root ファイルを作成
 mu_t->Write();
 rtfilemt->Close(); //root ファイルを閉じる

 TFile *f= new TFile(full_name2.c_str(),“”); //root ファイルを開く
 f->ls();

 return ;
}
```

付録 C

histogramOutput.cc
ROOT によるヒストグラム作成

```
#include "fstream"
#include "TNtuple.h"
#include "TH1.h"
#include "TH2.h"
#include "TCanvas.h"
#include "TFile.h"
#include "TTree.h"
#include <string>
using namespace std;

void histogramOutput(){
  gStyle->SetOptFit(1);
  string    fname="2016-01-29-thre-016"; //参照先
  string   directory1="/home/2015-b4/ug2015/data2015/new/1000000Events/";
  string   directory2="/home/2015-b4/ug2015/data2015/new/1000000Events/rootfile/";
  string   full_name1 = directory1+ fname; //データファイルの絶対パス
  string   full_name2 = directory2+ fname+ ".root"; //root ファイルの絶対パス

 TCanvas *c1 = new TCanvas("c1", "ctitle", 0, 0, 500, 400); //新しいキャンバスを作成
 TFile *f=TFile::Open(full_name2.c_str()); //.root を読み込む
 TH1F *h1= new TH1F("h1", "hname",100,0,2000); //新しいヒストグラムを作成

 TTree *t=(TTree*)f->Get("mu_t"); //Tree を読み込む

 TCut positiveCut = "100<es1 && et1<80 && et2<65
 && ev1<45 && ev2<40 && ev3<50 && ev4<50 && ev5<45
 && ev6<20 && ev7<85 && ev8<60";
 TCut tdc = "100<es1 && tdc2<4000";

 c1->SetLogy(0); //y 軸を log スケールに
 t->Draw("tdc2>>h1",tdc,""); //ヒストグラムを出力
 //  TF1*f_gaus=new TF1("f_gaus","gaus");
 //h1->Fit("gaus","","",700,3500); //gaus で Fit
 //h1->Fit("f_gaus","","",500,2000); //exp で Fit
 //  h1->Fit("gaus");
```

```
  return ;
}
```

付録 D

hist_lifetime.cc
ROOT による μ 粒子寿命解析

```
#include "fstream"
#include "TNtuple.h"
#include "TH1.h"
#include "TH2.h"
#include "TCanvas.h"
#include "TFile.h"
#include "TTree.h"
#include "TChain.h"
using namespace std;

void hist_lifetime(){

 TCanvas *c1 = new TCanvas("c1", "ctitle", 0, 0,500, 400); //新しいキャンバスを作成
 //c1->Divide(2,1); //キャンバスを分割

 TChain *mu_ch=new TChain("mu_t","mu_title"); //Chain を作成
 mu_ch->Add("/home/2015-b4/ug2015/data2015/new/1000000Events/rootfile/test2016-01-06/*.root");
 //Chain に入れる

 TCut e_decay="10<es1 && tdc2<4000";
 TCut E_block="750<es1+es2 && es1+es2<1000";

 // TH1F *h1= new TH1F("h1","tdc2",500,0,2500); //新しいヒストグラムを作成

 /*TDC-ADC(e) 相関関係*/
 //TH2F *h2= new TH2F("h2","es1+es2:tdc2",500,0,2500,500,0,2500); //二次ヒストグラムを作成
 //mu_ch->Draw("es1+es2:tdc2>>h2",e_decay,""); //ヒストグラムを出力
/*μ粒子寿命測定/
 //c1->cd(1); //Pad を移動
 TH1F *h1= new TH1F("h1", "p2nashi",500,0,2000); //新しいヒストグラムを作成
 mu_ch->Draw("tdc2>>h1",e_decay && E_block,""); //ヒストグラムを出力
 /*
 c1->cd(2);
 TH1F *h2= new TH1F("h2", "p2ari",500,0,2000); //新しいヒストグラムを作成
 mu_ch->Draw("tdc2>>h2",e_decay && E_block,""); //ヒストグラムを出力
 */
```

```
 // c1->cd(1);
 TF1 *f_exp_y=new TF1("f_exp_y","[0]*exp(-x/[1])");
 //f_exp_y=p0+exp(-x/p1)
 f_exp_y->SetParameter(0,665); //p2 無しパラメータの初期値を設定 i 番目, 初期値)
 f_exp_y->SetParameter(1,435);
 h1->Fit("f_exp_y","","",125,1940); //f_exp_y で Fit
 gStyle->SetOptFit(1); //キャンバスにパラメータを表示

 /*
 c1->cd(2);
 TF1 *f_exp_n=new TF1("f_exp_n","[0]*exp(-x/[1])+[2]");
 //f_exp_n=p0+exp(-x/p1)+p2
 f_exp_n->SetParameter(0,665); //p2 ありパラメータの初期値を設定 i 番目, 初期値)
 f_exp_n->SetParameter(1,435);
 f_exp_n->SetParameter(2,0);
 h2->Fit("f_exp_n","","",125,1940); //f_exp_n で Fit
 gStyle->SetOptFit(1); //キャンバスにパラメータを表示
 */

 Double_t p0_y = f_exp_y->GetParameter(0); //p0 を取得
 Double_t p1_y = f_exp_y->GetParameter(1); //p1 を取得
 /*
 Double_t p0_n = f_exp_n->GetParameter(0); //p0 を取得
 Double_t p1_n = f_exp_n->GetParameter(1); //p1 を取得
 Double_t p2_n = f_exp_n->GetParameter(2); //p2 を取得 */

 cout << " p0_y=" << p0_y << " p1_y=" << p1_y << endl; //パラメータを表示 p2 無し
 //cout << " p0_n=" << p0_n << " p1_n=" << p1_n << " p2_n=" << p2_n << endl; //パラメータを表示 p2 あり

 return ;
}
```

付録 E

hist_test.cc
ROOT による電子エネルギースペクトラム作成

```
#include "fstream"
#include "TNtuple.h"
#include "TH1.h"
#include "TH2.h"
#include "TCanvas.h"
#include "TFile.h"
#include "TTree.h"
#include "TChain.h"
using namespace std;

void hist_test(){

 gStyle->SetOptFit(1); //キャンバスにパラメータを表示

 TCanvas *c1 = new TCanvas("c1", "ctitle", 0, 0, 500, 400); //新しいキャンバスを作成

 TChain *mu_ch=new TChain("mu_t","mu_title"); //Chain を作成
 mu_ch->Add("/home/2015-b4/ug2015/data2015/new/1000000Events/rootfile/*.root"); //Chain
に入れる

 TH1F *h1= new TH1F("h1", "hname",500,0,2500); //新しいヒストグラムを作成

 TCut EnergyCut = "10<es1 && et1<300 && et2<400 && ev1<120 && ev2<150
 && ev3<40 && ev4<220 && ev5<50 && ev6<180 && ev7<360 && ev8<70";

 mu_ch->Draw("es1+es2>>h1",EnergyCut,""); //ヒストグラムを出力

 return ;
}
```

付録 F

ExN03PrimaryGeneratorAction.cc

```
//
// ********************************************************************
// * License and Disclaimer                                           *
// * *
// * The  Geant4 software  is  copyright of the Copyright Holders  of  *
// * the Geant4 Collaboration.  It is provided  under  the terms  and  *
// * conditions of the Geant4 Software License,  included in the file  *
// * LICENSE and available at  http://cern.ch/geant4/license .  These  *
// * include a list of copyright holders.                              *
// *                                                                   *
// * Neither the authors of this software system, nor their employing  *
// * institutes,nor the agencies providing financial support for this  *
// * work  make  any representation or  warranty, express or implied,  *
// * regarding  this  software system or assume any liability for its  *
// * use.  Please see the license in the file  LICENSE  and URL above  *
// * for the full disclaimer and the limitation of liability.          *
// *                                                                   *
// * This  code  implementation is the result of  the  scientific and  *
// * technical work of the GEANT4 collaboration.                       *
// * By using,  copying,  modifying or  distributing the software (or  *
// * any work based  on the software)  you  agree  to acknowledge its  *
// * use  in  resulting  scientific  publications,  and indicate your  *
// * acceptance of all terms of the Geant4 Software license.           *
// ********************************************************************
//
//
// $Id: ExN03PrimaryGeneratorAction.cc,v 1.8 2006/06/29 17:49:07 gunter Exp $
// GEANT4 tag $Name: geant4-09-02 $
//
//

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

#include "ExN03PrimaryGeneratorAction.hh"

#include "ExN03DetectorConstruction.hh"
#include "ExN03PrimaryGeneratorMessenger.hh"
```

```
#include "G4Event.hh"
#include "G4ParticleGun.hh"
#include "G4ParticleTable.hh"
#include "G4ParticleDefinition.hh"
#include "Randomize.hh"
#include "math.h"

#include <iostream>
#include <string>
#include <fstream>
std::ofstream ofp;


//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

ExN03PrimaryGeneratorAction::ExN03PrimaryGeneratorAction(
                                            ExN03DetectorConstruction* ExN03DC)
:ExN03Detector(ExN03DC),rndmFlag("off")
{
  G4int n_particle = 1;
  particleGun  = new G4ParticleGun(n_particle);

  //create a messenger for this class
  gunMessenger = new ExN03PrimaryGeneratorMessenger(this);

  // default particle kinematic

  G4ParticleTable* particleTable = G4ParticleTable::GetParticleTable();
  G4String particleName;
  G4ParticleDefinition* particle
                   = particleTable->FindParticle(particleName="e-");
  particleGun->SetParticleDefinition(particle);
  particleGun->SetParticleEnergy(50.*MeV);
}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

ExN03PrimaryGeneratorAction::~ExN03PrimaryGeneratorAction()
{
  delete particleGun;
  delete gunMessenger;
}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

ExN03PrimaryGeneratorAction::~ExN03PrimaryGeneratorAction()
{
  delete particleGun;
  delete gunMessenger;
}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

void ExN03PrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
```

```
{
  //this function is called at the begining of event
  //

  /*
  if (rndmFlag == "on")
     {
      G4double x0 = (ExN03Detector->GetCalorThickness())*(G4UniformRand()-0.5);
      G4double y0 = (ExN03Detector->GetCalorSizeY())*(G4UniformRand()-0.5);
      G4double z0 = (ExN03Detector->GetCalorSizeZ())*(G4UniformRand()-0.5);
     }
  */

 //particle position
  G4double x0,y0,z0;
  x0 = (G4UniformRand()-0.5)*58;
  y0 = 16;
  z0 = (G4UniformRand()-0.5)*25;
  particleGun->SetParticlePosition(G4ThreeVector(x0*cm,y0*cm,z0*cm));

 //tenchoukaku theta phi
  G4double xt,yt,t,p,xm,ym,zm ;
  yt = G4UniformRand();
  xt = pow(yt,1./3.);
  t  = acos(xt);//theta

ofp.open("test.dat", std::ios::app);
  ofp << t << "    " << xt <<std::endl;
ofp.close();

  p  = G4UniformRand()*2.*M_PI;//phi

  ym = -1;
  xm = ym*tan(t)*cos(p);
  zm = ym*tan(t)*sin(p);
  particleGun->SetParticleMomentumDirection(G4ThreeVector(xm,ym,zm));

  particleGun->GeneratePrimaryVertex(anEvent);
}
//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
```

付録 G

ExN03EventAction.cc

```
//
// ********************************************************************
// * License and Disclaimer    *
// *    *
// * The  Geant4 software  is  copyright of the Copyright Holders  of  *
// * the Geant4 Collaboration.  It is provided  under  the terms  and  *
// * conditions of the Geant4 Software License,  included in the file  *
// * LICENSE and available at  http://cern.ch/geant4/license .  These  *
// * include a list of copyright holders.                              *
// *                                                                   *
// * Neither the authors of this software system, nor their employing  *
// * institutes,nor the agencies providing financial support for this  *
// * work  make  any representation or  warranty, express or implied,  *
// * regarding  this  software system or assume any liability for its  *
// * use.  Please see the license in the file  LICENSE  and URL above  *
// * for the full disclaimer and the limitation of liability.          *
// *                                                                   *
// * This  code  implementation is the result of  the  scientific and  *
// * technical work of the GEANT4 collaboration.                       *
// * By using,  copying,  modifying or  distributing the software (or  *
// * any work based  on the software)  you  agree  to acknowledge its  *
// * use  in  resulting  scientific  publications,  and indicate your  *
// * acceptance of all terms of the Geant4 Software license.           *
// ********************************************************************
//
//
// $Id: ExN03EventAction.cc,v 1.29 2008/01/17 17:31:32 maire Exp $
// GEANT4 tag $Name: geant4-09-02 $
//
//

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

#include "ExN03EventAction.hh"

#include "ExN03RunAction.hh"
#include "ExN03EventActionMessenger.hh"
```

```
#include "G4Event.hh"
#include "G4TrajectoryContainer.hh"
#include "G4VTrajectory.hh"
#include "G4VVisManager.hh"
#include "G4UnitsTable.hh"

#include "Randomize.hh"
#include <iomanip>

#include <fstream>
std::ofstream ofs;

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

ExN03EventAction::ExN03EventAction(ExN03RunAction* run)
  :runAct(run),printModulo(1),eventMessenger(0)
{
  eventMessenger = new ExN03EventActionMessenger(this);
  /* edep2 -> sum of lost energy of electron and muon , and Kinetic Energy */
   ofs.open("edep2.dat", std::ios::out);
   if(! ofs.good()){
      G4cout<<" EventAction; error opening edep2.dat. ofs.good="<<ofs.good()
<<G4endl;
  }

}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo.....


ExN03EventAction::ExN03EventAction(ExN03RunAction* run)
  :runAct(run),printModulo(1),eventMessenger(0)
{
  eventMessenger = new ExN03EventActionMessenger(this);
  /* edep2 -> sum of lost energy of electron and muon , and Kinetic Energy */
   ofs.open("edep2.dat", std::ios::out);
   if(! ofs.good()){
      G4cout<<" EventAction; error opening edep2.dat. ofs.good="<<ofs.good()
<<G4en\
dl;
  }

}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

ExN03EventAction::~ExN03EventAction()
{
   ofs.close();
   G4cout<<" EventAction: edep2.dat file is closed"<<G4endl;
  delete eventMessenger;
}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
```

```
void ExN03EventAction::BeginOfEventAction(const G4Event* evt)
{
  G4int evtNb = evt->GetEventID();
  if (evtNb%printModulo == 0) {
    G4cout << "\n---> Begin of event: " << evtNb << G4endl;
    CLHEP::HepRandom::showEngineStatus();
//--
// initialize energy sum
//--
    EdepEl=0.;
    EdepMu=0.;
    KinetE=0.;
    // EdepAll=0.; //2016-02-16
    OutOfMu=0.; //2016-02-23
    OutOfEl=0.; //2016-02-18
    PosY=0.; //2016-02-19
  }

 // initialisation per event
 EnergyAbs = EnergyGap = 0.;
 TrackLAbs = TrackLGap = 0.;
}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
void ExN03EventAction::EndOfEventAction(const G4Event* evt)
{
  //accumulates statistic
  //
  runAct->fillPerEvent(EnergyAbs, EnergyGap, TrackLAbs, TrackLGap);

  //print per event (modulo n)
  //
  G4int evtNb = evt->GetEventID();
  if (evtNb%printModulo == 0) {
    G4cout << "---> End of event: " << evtNb << G4endl;

    G4cout
        << "   Absorber: total energy: " << std::setw(7)
                                         << G4BestUnit(EnergyAbs,"Energy")
       << "       total track length: " << std::setw(7)
                                         << G4BestUnit(TrackLAbs,"Length")
       << G4endl;

//--
//  write energy sum to file.
//--
    G4cout <<" Eventaction write EdelEl and EdepMu"<<G4endl;
     ofs <<EdepEl<<"    "
         <<EdepMu<<"    "
         <<KinetE<<"    "
       //<<EdepAll<<"    " //2016-02-16
         <<OutOfMu<<"    "//2016-02-23
```

```
          <<OutOfEl<<"    "//2016-02-18
          <<PosY<<"    "<<G4endl; //2016-02-19

      G4cout <<" EdepEl, EdepMu, KineE="<< EdepEl<<", "<<EdepMu<<", "
             <<KinetE<<G4endl;
  }
}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
```

付録 H

ExN03SteppingAction.cc

```
//
// ********************************************************************
// * License and Disclaimer                                           *
// *                                                                  *
// * The  Geant4 software  is  copyright of the Copyright Holders  of *
// * the Geant4 Collaboration.  It is provided  under  the terms  and *
// * conditions of the Geant4 Software License,  included in the file *
// * LICENSE and available at  http://cern.ch/geant4/license .  These *
// * include a list of copyright holders.                             *
// *                                                                  *
// * Neither the authors of this software system, nor their employing *
// * institutes,nor the agencies providing financial support for this *
// * work  make  any representation or  warranty, express or implied, *
// * regarding  this  software system or assume any liability for its *
// * use.  Please see the license in the file  LICENSE  and URL above *
// * for the full disclaimer and the limitation of liability.         *
// *                                                                  *
// * This  code  implementation is the result of  the  scientific and *
// * technical work of the GEANT4 collaboration.                      *
// * By using,  copying,  modifying or  distributing the software (or *
// * any work based  on the software)  you  agree  to acknowledge its *
// * use  in  resulting  scientific  publications,  and indicate your *
// * acceptance of all terms of the Geant4 Software license.          *
// ********************************************************************
//
//
// $Id: ExN03SteppingAction.cc,v 1.15 2006/06/29 17:49:13 gunter Exp $
// GEANT4 tag $Name: geant4-09-02 $
//
//
//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

#include "ExN03SteppingAction.hh"

#include "ExN03DetectorConstruction.hh"
#include "ExN03EventAction.hh"

#include "G4Step.hh"
```

```
#include <fstream>

#include "ExN03SteppingVerbose.hh"
#include "G4SteppingVerbose.hh"


//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo.....
ExN03SteppingAction::ExN03SteppingAction(ExN03DetectorConstruction* det,
                                         ExN03EventAction* evt)
:detector(det), eventaction(evt)
{
}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

ExN03SteppingAction::~ExN03SteppingAction()
{
}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

void ExN03SteppingAction::UserSteppingAction(const G4Step* aStep)
{
  // get volume of the current step

  // G4cout<<" we are in UserSteppingAction"<<G4endl;

  // const G4StepPoint* preStepPoint = aStep->GetPreStepPoint();

  G4Track * aTrack=aStep->GetTrack();
  G4VPhysicalVolume* volume
  = aStep->GetPreStepPoint()->GetTouchableHandle()->GetVolume();

  // collect energy and track length step by step
  G4double edep = aStep->GetTotalEnergyDeposit();
  G4double Kine = aTrack->GetKineticEnergy()+ edep;
  G4double stepl = 0.;

  G4ThreeVector pos = aTrack->GetPosition();
  G4double ParentID = aTrack->GetParentID();

  if (aStep->GetTrack()->GetDefinition()->GetPDGCharge() != 0.)
    stepl = aStep->GetStepLength();

  /* Out put to display */
 G4cout<<"  SteppingAction TrkID = "<<std::setw(6)<< aTrack->GetTrackID()
       <<"  Parent ID ="<<std::setw(6)<< ParentID
      <<"   name       ="<<std::setw(6)<< aTrack->GetDefinition()->GetParticleName();
  G4cout<<"     edep   ="<<std::setw(6)<< edep
       <<" stepl="<<std::setw(6)<< stepl
       <<"  kine ="<<std::setw(6)<<Kine
       << G4endl;
```

```
 /* Muon is out or inside of detector */
 if (aTrack->GetDefinition()->GetParticleName()
     =="mu-" && ( 29*cm<pos.x()     || pos.x()<-29*cm
                                    || pos.y()<-13*cm
               || 12.5*cm<pos.z()   || pos.z()<-12.5*cm) )
   {
     eventaction->OutMuadd(1.);
     // aTrack->SetTrackStatus(fSuspend);
     // aTrack->SetTrackStatus(fKillTrackAndSecondaries);
   }

  /* Electron is out or inside of detector */
 if (aTrack->GetDefinition()->GetParticleName()
     =="e-" && (29*cm<pos.x()   || pos.x()<-29*cm
     || 13*cm<pos.y()   || pos.y()<-13*cm
     || 12.5*cm<pos.z() || pos.z()<-12.5*cm) )
      {
       eventaction->OutEladd(1.);
       // aTrack->SetTrackStatus(fSuspend);
       // aTrack->SetTrackStatus(fKillTrackAndSecondaries);
       }


 /* Add lost energy of electron*/
if (aTrack->GetDefinition()->GetParticleName()
             =="e-")
    eventaction->Eladd(edep);

 /* Add lost energy of muon*/
if (aTrack->GetDefinition()->GetParticleName()=="mu-")
     eventaction->Muadd(edep);

/* Kinetic energy of just after decay of muon */
if (aTrack->GetDefinition()->GetParticleName()
if (aTrack->GetDefinition()->GetParticleName()
    =="e-" && aTrack->GetParentID()==1
    &&  aTrack->GetCurrentStepNumber()==1
    //     &&  (4.33*cm<pos.y() && pos.y()<13.0*cm)
    ){
     eventaction->Kineadd(Kine);
     G4cout<<"Kineadd is called"<<G4endl;

     if(4.33*cm<pos.y() && pos.y()<13.0*cm)
       { eventaction->positionY(1.);};
     if(-4.33*cm<pos.y() && pos.y()<4.33*cm)
       { eventaction->positionY(2.);};
     if(-13.0*cm<pos.y() && pos.y()<-4.33*cm)
       { eventaction->positionY(3.);};


     }

 if (volume == detector->GetAbsorber()) eventaction->AddAbs(edep,stepl);
```

```
  if (volume == detector->GetGap())      eventaction->AddGap(edep,stepl);
}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
```

付録 I

ExN03EventAction.hh

```
//
// ********************************************************************
// * License and Disclaimer                                           *
// *                                                                  *
// * The  Geant4 software  is  copyright of the Copyright Holders  of *
// * the Geant4 Collaboration.  It is provided  under  the terms  and *
// * conditions of the Geant4 Software License,  included in the file *
// * LICENSE and available at  http://cern.ch/geant4/license .  These *
// * include a list of copyright holders.                             *
// *                                                                  *
// * Neither the authors of this software system, nor their employing *
// * institutes,nor the agencies providing financial support for this *
// * work  make  any representation or  warranty, express or implied, *
// * regarding  this  software system or assume any liability for its *
// * use.  Please see the license in the file  LICENSE  and URL above *
// * for the full disclaimer and the limitation of liability.         *
// *                                                                  *
// * This  code  implementation is the result of  the  scientific and *
// * technical work of the GEANT4 collaboration.                      *
// * By using,  copying,  modifying or  distributing the software (or *
// * any work based  on the software)  you  agree  to acknowledge its *
// * use  in  resulting  scientific  publications,  and indicate your *
// * acceptance of all terms of the Geant4 Software license.          *
// ********************************************************************
//
//
// $Id: ExN03EventAction.hh,v 1.12 2007/07/02 13:22:08 vnivanch Exp $
// GEANT4 tag $Name: geant4-09-02 $
//
//

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

#ifndef ExN03EventAction_h
#define ExN03EventAction_h 1

#include "G4UserEventAction.hh"
#include "globals.hh"
```

```
class ExN03RunAction;
class ExN03EventActionMessenger;

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

class ExN03EventAction : public G4UserEventAction
{
public:
  ExN03EventAction(ExN03RunAction*);
  virtual ~ExN03EventAction();

  void  BeginOfEventAction(const G4Event*);
  void    EndOfEventAction(const G4Event*);

  void AddAbs(G4double de, G4double dl) {EnergyAbs += de; TrackLAbs += dl;};
  void AddGap(G4double de, G4double dl) {EnergyGap += de; TrackLGap += dl;};

  void SetPrintModulo(G4int    val)  {printModulo = val;};
  void Eladd(G4double  de){ EdepEl += de;};
  void Muadd(G4double  de){ EdepMu += de;};
  void Kineadd(G4double Kine1){KinetE += Kine1;};
  void OutMuadd(G4double  de){ OutOfMu  += de;}; //2016-02-23
  void OutEladd(G4double  de){ OutOfEl  += de;}; //2016-02-18

  void positionY(G4double  de){ PosY = de;}; //2016-02-19

private:
//energy deposit of electron and muon.
   G4double  EdepEl;
   G4double  EdepMu;
   G4double  KinetE;

  G4double  OutOfMu; //2016-02-23
  G4double  OutOfEl; //2016-02-18

  G4double  PosY; //2016-02-19

   ExN03RunAction*  runAct;

   G4double  EnergyAbs, EnergyGap;
   G4double  TrackLAbs, TrackLGap;

   G4int     printModulo;

   ExN03EventActionMessenger*  eventMessenger;


};

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
#endif
```