

2016年度 卒業論文

μ 粒子の寿命及び崩壊電子の
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

市川真有 永松えりな 皆吉遥

2017年3月3日

目次

1	はじめに	4
1.1	μ 粒子とは	4
1.2	研究の目的	5
1.3	本論文の構成	5
2	測定原理とセットアップ	6
2.1	測定で用いるセットアップ	6
2.1.1	通過 μ 粒子と静止後崩壊する μ 粒子の信号の見分け方	7
2.2	シンチレーションカウンター	8
2.2.1	シンチレーションカウンターとは	8
2.2.2	シンチレーションカウンターの配置	9
2.2.3	トリガーカウンター	11
2.2.4	ベトーカーカウンター	11
2.3	NIM モジュール	12
2.4	CAMAC	14
2.4.1	ADC(Analog to Digital Converter)	14
2.4.2	TDC(Time to Digital Converter)	15
2.4.3	CC(Crate Controller)	15
2.4.4	データ収集プログラム	16
3	μ 粒子の寿命測定	19
3.1	原理	19
3.2	CAMAC モジュールのセットアップ	22
3.3	解析	23
3.3.1	イベントの選定	23
3.3.2	TDC 分布に関数を F i t する	23
3.3.3	TDC 時間較正	24
3.4	結果	26
3.5	考察	26
4	μ 粒子から崩壊した電子のエネルギースペクトラムの測定	28
4.1	原理	28
4.1.1	物質中の荷電粒子(電子)のふるまい	30
4.2	CAMAC モジュールのセットアップ	32
4.3	解析	32
4.3.1	イベントの選定	32
4.3.2	ADC エネルギー較正	36
4.4	結果	38
4.5	シミュレーションによる検討	38
4.5.1	Geant4	38
4.5.2	シミュレーション結果	41
4.6	考察	42

5	まとめ	43
5.1	実験のまとめ	43
5.2	謝辞	43
5.3	参考文献	43
A	ele.c	
	CAMAC からのデータ収集	44
B	rootfileOutput.cc	
	CAMAC からのデータを root ファイルへ出力	48
C	histogramOutput.cc	
	ROOT によるヒストグラム作成	50
D	hist_lifetime.cc	
	ROOT による μ 粒子寿命解析	50
E	hist_test.cc	
	ROOT による電子のエネルギースペクトラムの作成	52
F	ExN03PrimaryGeneratorAction.cc	53
G	ExN03EventAction.cc	55
H	ExN03SteppingAction.cc	57

1 はじめに

1.1 μ 粒子とは

μ 粒子とは、地球の大気に入射する宇宙線の一部である。宇宙線とは、宇宙空間から飛来する高エネルギーの放射線のことであり、このうち、地球大気に入射する宇宙線を一次宇宙線と呼ぶ。一次宇宙線は主に高いエネルギーを持った陽子と α 線から成り、大気中の窒素原子核や酸素原子核などと衝突して高エネルギーの中間子などの二次粒子を生成する。生成された二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上に生成される現象を、空気シャワー現象と呼び、生成された新たな粒子を二次宇宙線と呼ぶ。 μ 粒子はこの二次宇宙線の一つである。

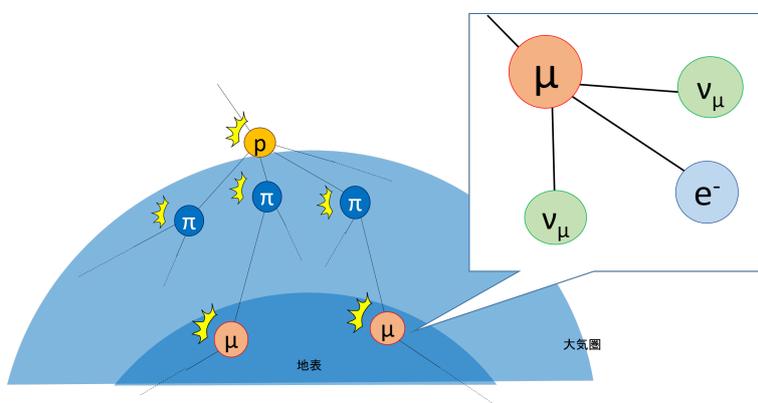


図 1: 宇宙線の模式図

μ 粒子は π 中間子が崩壊して生成される粒子である。 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 $139.57(\text{MeV}/c^2)$ 、 π^\pm の質量は $134.97(\text{MeV}/c^2)$ である。荷電 π 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。

$$\mu^- \rightarrow e^- + \bar{\nu}_e + \nu_\mu \tag{1}$$

ここで、 $\bar{\nu}_e$ 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。

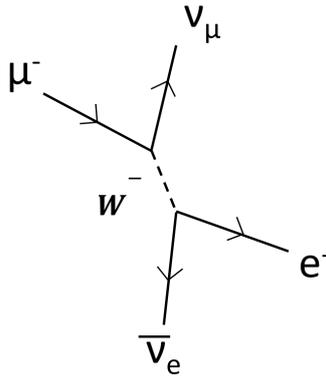


図 2: μ^- 崩壊のファインマン図

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 研究の目的

エネルギー的に不安定な粒子は、より安定な粒子へと崩壊する。宇宙線として降り注ぐ μ 粒子が、シンチレーションカウンター内で静止し崩壊すると電子、電子ニュートリノ、反電子ニュートリノへ崩壊する。この崩壊にかかる時間の平均値は、 μ 粒子の平均寿命と呼ばれる。本実験では μ 粒子の寿命を測定することで、その性質と仕組み、測定器の取り扱いなどについて学ぶ。また崩壊の際に得られる電子のエネルギースペクトルを測定し、その性質についても学ぶ。

1.3 本論文の構成

本論文の以下の構成は次のようになっている。

第 2 章では、測定原理とセットアップについて述べる。

第 3 章では、 μ 粒子の寿命測定について、

第 4 章では、 μ 粒子から崩壊した電子のエネルギースペクトラムの測定について述べる。

最後に、第 5 章で本論文のまとめを行う。

なお、付録 A,B は CAMAC からのデータ収集及び root ファイルへ出力するためのソースコード、付録 C~E は ROOT でグラフを作成するためのソースコード、付録 F~H は Geant4 で用いたソースコードである。

2 測定原理とセットアップ

2.1 測定で用いるセットアップ

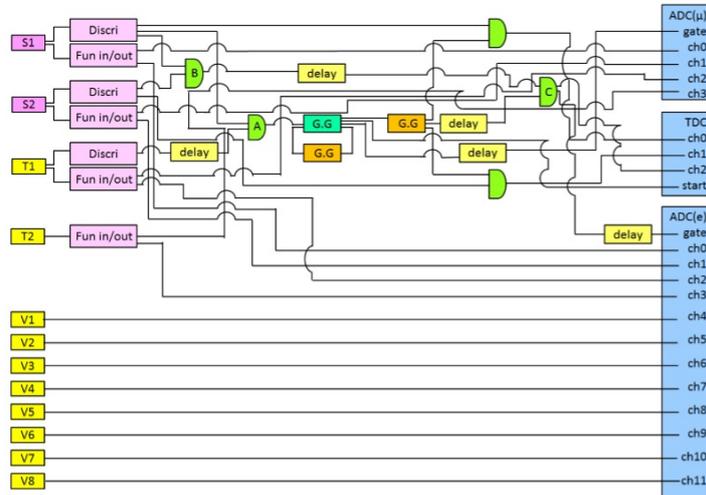


図 3: セットアップの全体図

図 3 は測定で用いるセットアップの全体図である。シンチレーションカウンター S1,S2,T1,T2,V1 V8 から出た信号を ADC のゲートやチャンネルにつないでいる。図 4 は簡略化したセットアップのイメージ図である。シンチレーションカウンターで荷電粒子の信号を読み取り NIM モジュールや CAMAC モジュールを通すことでそれぞれの信号について調整を行い、PC へと送っている。各装置の説明は後ほど行う。

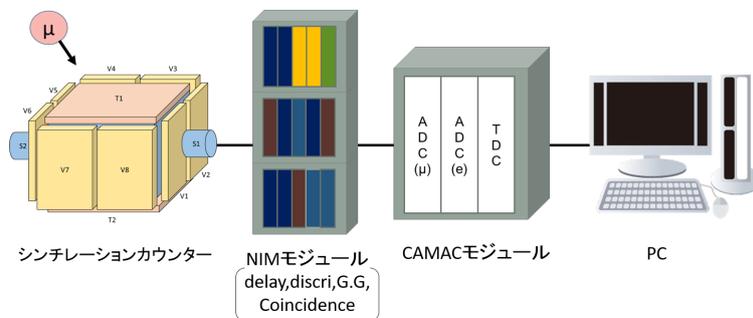


図 4: セットアップの全体の概要図

2.1.1 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時と、 μ 粒子がシンチレーター内で静止した後崩壊した時とは、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 6 に示すように 1 つの山の信号が見られる。

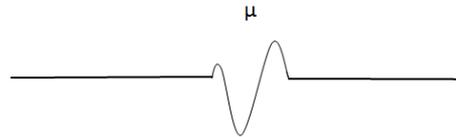
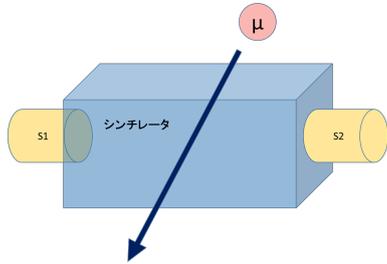


図 6: μ 粒子の波形

図 5: μ 粒子がメインシンチレーターを通過するとき

・ 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 8 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。前者を μ 粒子の波形、後者を電子の波形と呼ぶ。

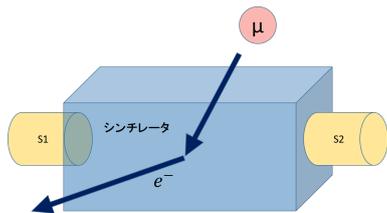


図 7: μ 粒子がメインシンチレーター内で電子に崩壊するとき

図 8: μ 粒子と電子の波形

2.2 シンチレーションカウンター

2.2.1 シンチレーションカウンターとは

シンチレーションカウンターとは、シンチレーターという蛍光物質と光電子増倍管を組み合わせた放射線検出器のことである下の図9は参考として、本実験で用いたシンチレーションカウンターと同じ型のシンチレーションカウンターである。大きさの比較として、約15cmのボールペンを置いている。



図 9: シンチレーションカウンター

実際に本測定に使用した機器は図10のとおり、シンチレーターに余計な光が入らないように、黒いビニールテープで覆ってある。



図 10: 実際に使用したシンチレーションカウンター

以下でシンチレーターと光電子増倍管について述べる。

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。シンチレーターで発生した光子を電気信号に変換するのに光電子増倍管を用いる。シンチレーターには有機シンチレーターと無機シンチレーターがあり、本実験で用いたプラスチックシンチレーターのような有機シンチレーターは、発光時間が ns オーダーで非常に短いため高い時間分解能が得られる。しかしシンチレーション光が弱いためエネルギー分解能は低めである。

光電子増倍管

シンチレーターによって放出された光子を、最終的に読み取り可能な電気信号として外部に読み出すための装置。図 11 は光電子増倍管の仕組みの概略図である。シンチレーターから出た光子を電子に変えるだけでは微弱な信号しか得られないので、電子を増幅する必要がある。まず、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す。飛び出した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出される。

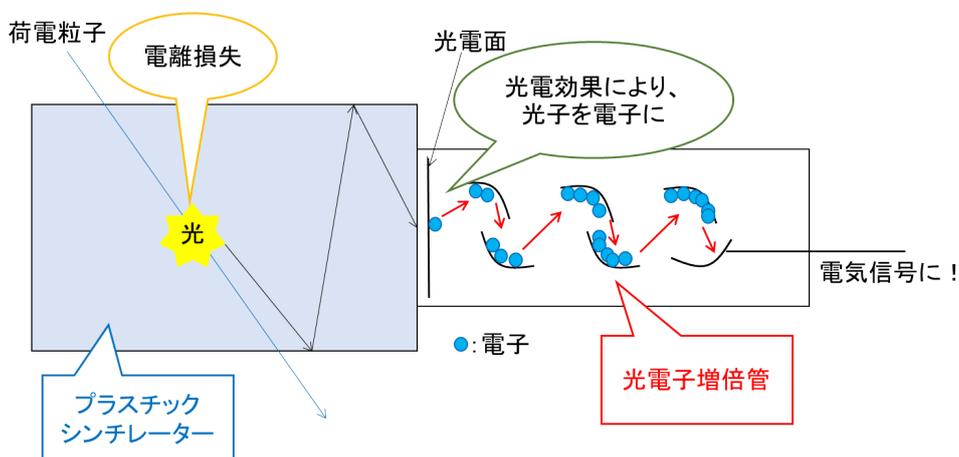


図 11: シンチレーションカウンターによる測定の原理の概要

2.2.2 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 12 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 1 に示す。

表 1: 各シンチレーターの仕様

	名称	横幅 (mm)	縦幅 (mm)	奥行 (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

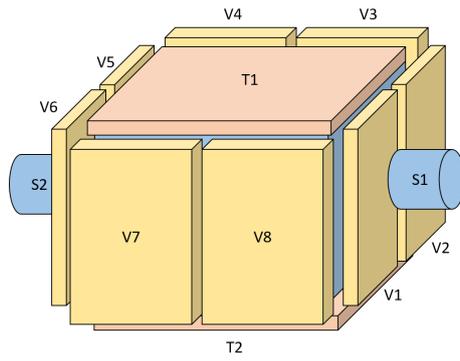


図 12: 全体図

図 12 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 13 と図 14 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ー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。

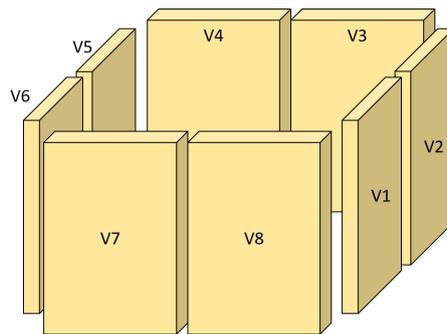


図 13: 分解図 1

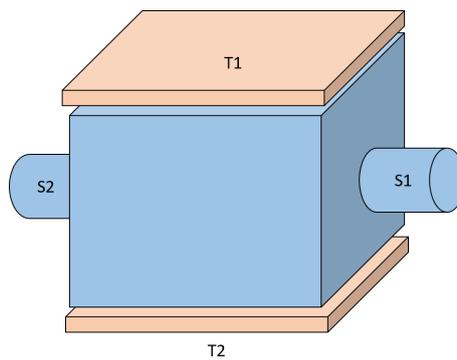


図 14: 分解図 2

2.2.3 トリガーカウンター

μ 粒子が入射し、メインシンチレーター内で崩壊したことを観測するために設置するシンチレーションカウンター。 μ 粒子の寿命を測るためには、メインシンチレーターに入射した μ 粒子が、メインシンチレーター内で静止した後、崩壊していなければならない。そこで、2つのシンチレーター（本実験における T1 と S1）の反応を見る。1つ目のシンチレーター（S1）が反応したとき μ 粒子が入射しているとわかり、その直後に2つ目のシンチレーターが反応すれば、崩壊して出てきた電子に反応しているとわかる。 μ 粒子が崩壊するまでの時間は非常に短いため、この2つの反応は同時とみなすことができる。この2つのシンチレーターをトリガーカウンターとして取り扱う。

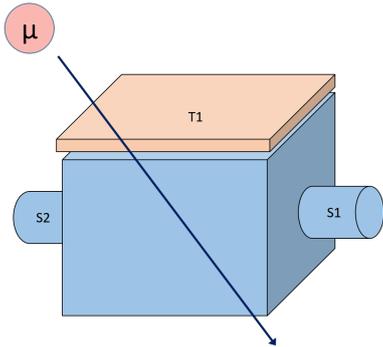


図 15: μ 粒子がシンチレータを貫通する時

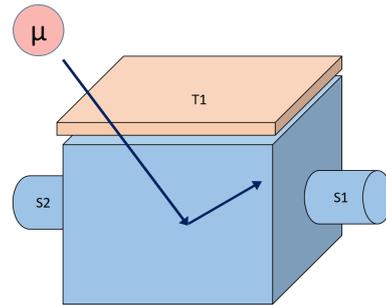


図 16: μ 粒子がシンチレータ内で電子に崩壊する時

2.2.4 ベトーカーカウンター

電子のエネルギースペクトラムを測る際のイベント選定に使用するシンチレーションカウンター。崩壊電子のエネルギースペクトラムを測るためには、崩壊電子がメインシンチレーター内で生成されてから完全にエネルギーを失う様子を測定することが必要である。なので、メインシンチレーターを取り囲むシンチレーションカウンター（本実験における T1, T2, V1 V8）に反応がある、つまり生成されて飛び出した電子が、そのままメインシンチレーター外に出てしまっているイベントを取り除いて測定する。これらのシンチレーションカウンターをベトーカーカウンターとして取り扱う。

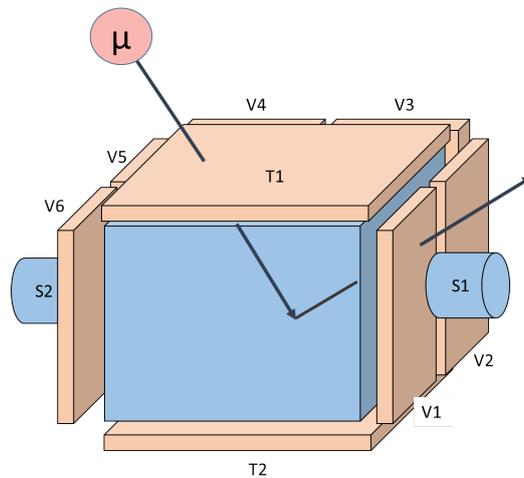


図 17: ベトーカーカウンターの概念図

2.3 NIM モジュール

本実験では NIM 規格に沿って作られた NIM モジュールを使用する。NIM モジュールはある条件の信号が入力されたときにロジック信号を出すようなモジュールのことで、NIM 信号のパルスの大きさは 0.7V で一定である。各装置の説明は以下のとおりである。

- Gate and delay Generator(G.G)

信号が入力されたとき、ロジック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 を任意に調整できる。

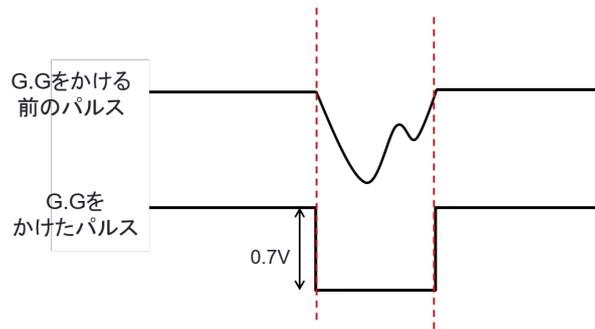


図 18: ゲートジェネレーターによる波形の様子

- Coincidence(C)

複数の信号が同時に入力されているときのみロジック信号を出力する。出力するパルスの時間幅を任意に調整できる。

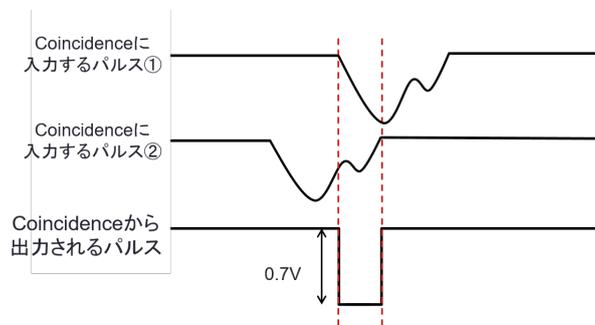


図 19: コインシデンスによる波形の様子

- Discriminator

あるしきい値を超える大きさの信号が入力されたとき、ロジック信号を出力する。

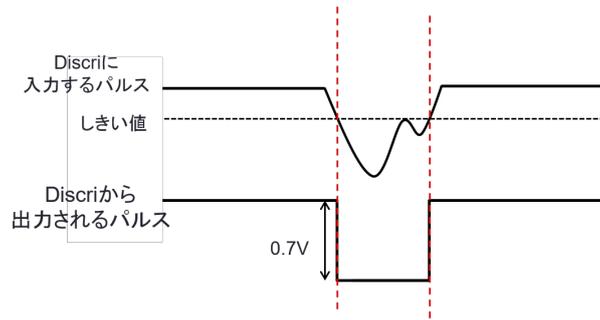


図 20: ディスクリによる波形の様子

- Fun-in Fun-out

入力信号がひとつでも入力されたときロジック信号を出力する。

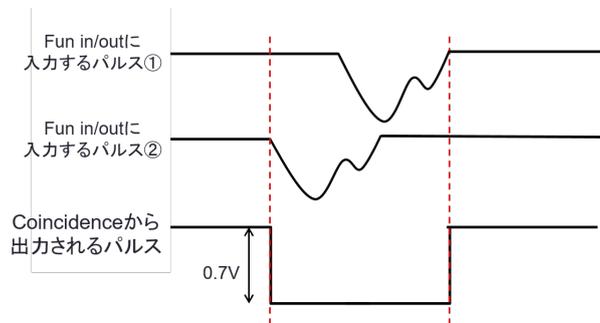


図 21: Fun in/out による波形の様子

- NIM モジュール以外のモジュール (Delay)

入ってきた信号のタイミングを遅らせる。入力信号の形や大きさは変わらない。

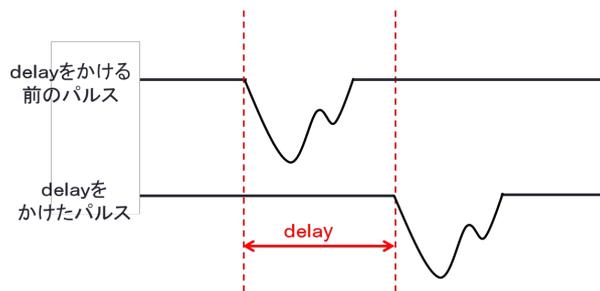


図 22: Delay による波形の様子

2.4 CAMAC

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システム規格のことである。本実験では、ADC(Analog to Digital Converter) と、TDC(Time to Digital Converter)、CC(Crate Controller) を用いる。PC などの計算機から 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。

- アドレス次の C,N,A,F にそれぞれ対応する数字を入れて指定する。
 - C クレート番号。
 - N スロット番号。クレートの何番目にモジュールがあるか。
 - A モジュール内のアドレス (本実験においては ADC のアドレス)。
 - F ファンクション。上記のアドレスについてどのような操作をするか。
- ファンクションモジュールによって使用可能なファンクションは異なるが、多くの場合番号が決まっています、以下その一部よく使われるものである。LAM(Look At Me) 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。
 - F(0),F(2) ReadData
 - F(8) TestLAM
 - F(9) ClearLAM
 - F(24) DisableLAM
 - F(26) EnableLAM

2.4.1 ADC(Analog to Digital Converter)

アナログ電気信号をデジタル電気信号に変えるモジュール。ADC には V モード(ピークホール型 ADC) と Q モード(荷電積分型 ADC) がある。V モードは Gate パルスが開いている間に入ってきた信号のピーク電圧を数値化するもので、Q モードは Gate パルスが開いている間に入ってきた信号を時間積分して数値化するものである。本実験で使用したのは Q モードである。Q モードで測定される電荷の量は

$$I = \int_{dt} i dt \quad (2)$$

となる。i は信号の電流であり、電荷は電流の時間積分なので、図 23 の青色部分の面積に相当する。

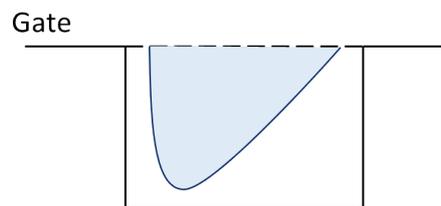


図 23: ADC の仕組み

尚、本実験では μ 粒子と電子についてそれぞれ測定するための ADC を設置しており、それぞれ ADC(μ)、ADC(e) と呼ぶ事にする。

2.4.2 TDC(Time to Digital Converter)

二つのパルスの時間差を測定するモジュール。図 24 は TDC の仕組みの概要図である。start パルスが入力されてから stop パルスが入力されるまで、一定の時間間隔でカウントを刻む。つまり TDC カウント数が多いほど、start パルスから stop パルスまでの時間差が大きいという事になる。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。

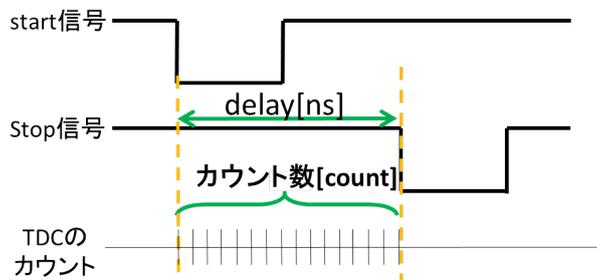


図 24: TDC の仕組み

2.4.3 CC(Crate Controller)

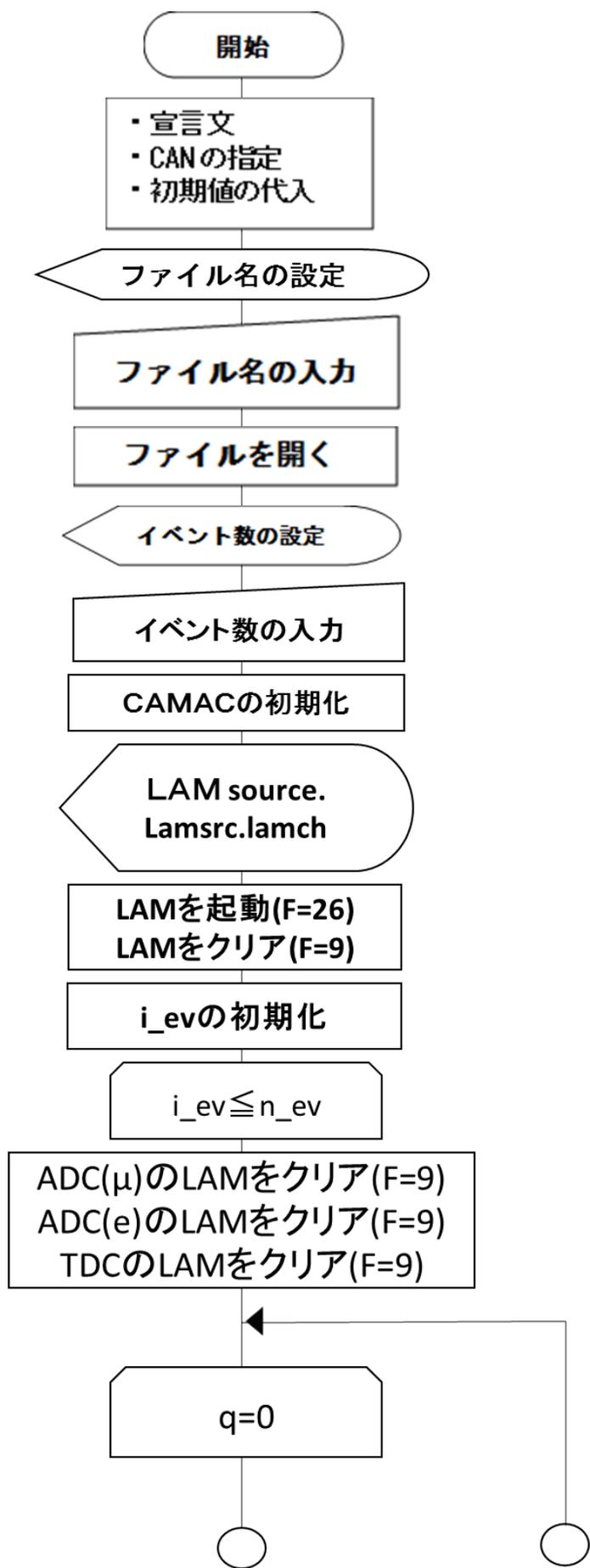
ADC や TDC 等を制御して、データ収集を PC 等で行う為のモジュール。本実験で用いたシンチレーションカウンターの光電子増倍管の型番、光電子増倍管にかけた電圧 (HV)、Discriminator の Threshold(しきい値)、各カウンターの ADC(μ)・ADC(e) の値を表にする。

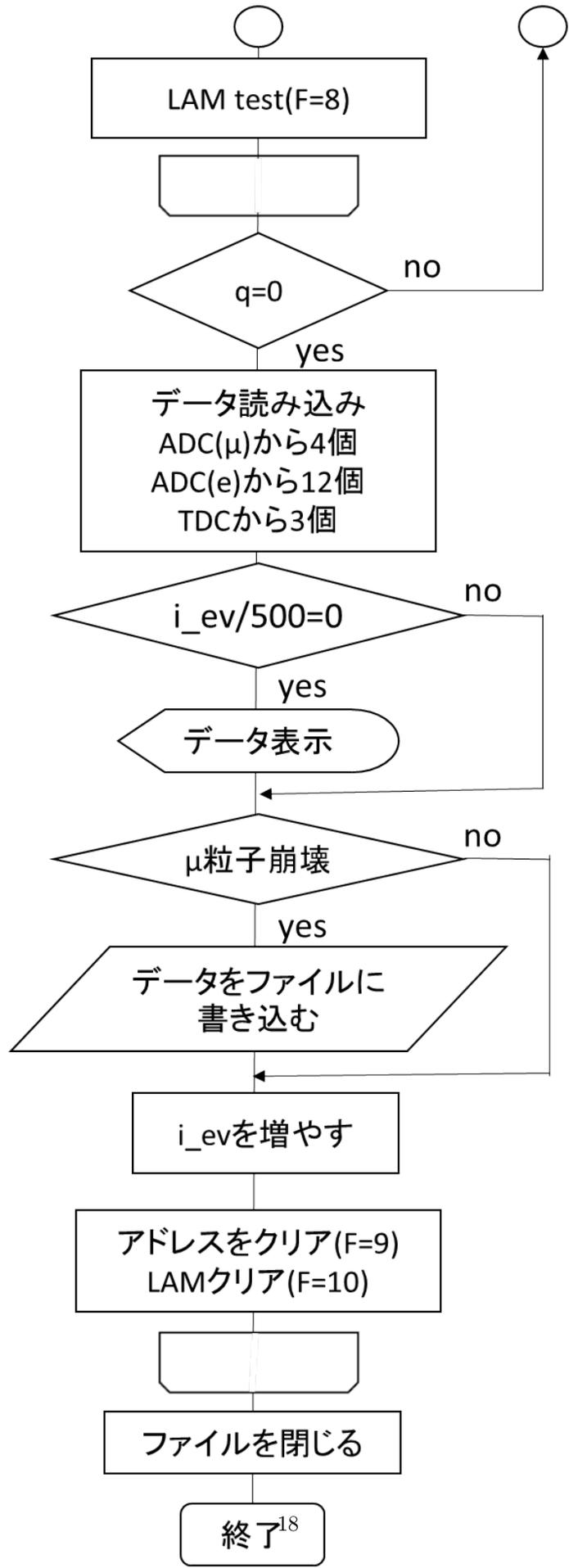
表 2: 各シンチレーターの仕様

カウンター	PMT 型番	HV(V)	Threshold(mV)	ペDESTアル ADC(μ)	ペDESTアル ADC(e)
S1	H1161	2050	286	109	58
S2	H1161	2050	98.9	98	63
T1	H1161	2250	158.7	89	56
T2	H7195	1770	197.6	85	60
V1	H1161	1970			55
V2	H7195UV	1660			63
V3	H1161	1870			57
V4	H1161	2440			63
V5	H1161	2400			65
V6	H1161	2030			64
V7	H7195	2210			66
V8	H1161	2025			80

2.4.4 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。はじめに、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。





3 μ 粒子の寿命測定

3.1 原理

μ 粒子の寿命の逆数 τ_μ^{-1} は

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

と表せる。ここで、

$$x = \frac{m_e^2}{m_\mu^2} \quad (4)$$

$$F(x) = 1 - 8x + 8x^3 - x^4 - 12x^2 \ln(x) = 0.999813 \quad (5)$$

$$H_1(x) = \frac{25}{8} - \frac{\pi^2}{2} - (9 + 4\pi + 12 \ln(x))x + 16\pi^2 x^{\frac{3}{2}} + O(x^2) = -1.8079 \quad (6)$$

$$H_2(x) = \frac{156815}{5184} - \frac{518}{81}\pi^2 - \frac{895}{36}\zeta(3) + \frac{67}{720}\pi^4 + \frac{53}{6}\pi^2 \ln 2 - \frac{5}{4}\pi^2 \sqrt{x} + O(x) = 6.7 \quad (7)$$

$$\hat{\alpha}(m_\mu)^{-1} = \alpha^{-1} - \frac{2}{3\pi} \ln\left(\frac{m_\mu}{m_e}\right) + \frac{1}{6\pi} = 135.9 \quad (8)$$

$$G_F = 1.16637 \times 10^{-5} (\text{GeV}^{-2}) \quad (9)$$

$$(10)$$

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、

$$(\tau_\mu)_{\text{理論値}} = 2.197 \times 10^{-6} (\text{sec}) \quad (11)$$

である。

本実験では、式 (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。

μ 粒子崩壊の時間的振る舞いは、放射性崩壊の指数関数的法則に従う。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。

$$dN = -\lambda N(t) dt \quad (12)$$

式 (12) を積分すると、

$$\begin{aligned} \int_{N_0}^N \frac{dN}{N} &= -\lambda \int_{t_0}^t dt \\ [\ln N]_{N_0}^N &= -\lambda [t]_{t_0}^t \\ \ln \frac{N}{N_0} &= -\lambda (t - t_0) \end{aligned} \quad (13)$$

$t=0$ の時、式 (13) は、

$$N(t) = N_0 e^{-\lambda t} \quad (14)$$

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 $t \rightarrow t + dt$ の間の微小時間 dt で崩壊する粒子の個数は式 (8) で表されるので、 N_0 個すべての粒子の生存時間を足し合わせたものは次式となる。

$$L = \int_0^{\infty} tN(t)\lambda dt \quad (15)$$

$$= \int_0^{\infty} tN_0e^{-\lambda t}\lambda dt \quad (16)$$

$$= \left[-\frac{1}{\lambda}N_0e^{-\lambda t} \right]_0^{\infty} \quad (17)$$

$$= \frac{N_0}{\lambda} \quad (18)$$

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

$$\tau = \frac{L}{N_0} = \frac{1}{\lambda}$$

$$\lambda = \frac{1}{\tau} \quad (19)$$

である。これを式 (14) に代入すると、

$$N(t) = N_0e^{-\frac{t}{\tau}} \quad (20)$$

が得られる。式 (20) を図 25 に示す。

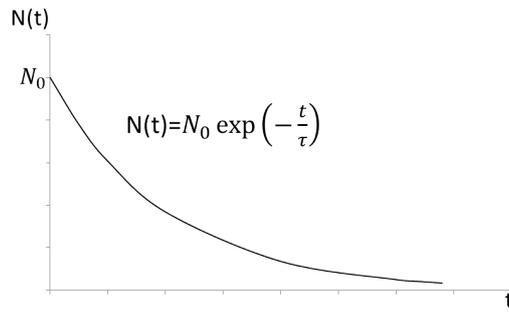


図 25: 放射性崩壊の図

また、式 (20) を t で微分すると、単位時間当たりの崩壊数を表す式が得られる。

$$\frac{dN}{dt} = -\frac{N_0}{\tau} \exp\left(-\frac{t}{\tau}\right) \quad (21)$$

dN/dt は $t \rightarrow t+dt$ 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。 $t \rightarrow t+dt$ 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 (21) は

$$\frac{dN_{\text{崩壊}}}{dt} = \frac{N_0}{\tau} \exp\left(-\frac{t}{\tau}\right) \quad (22)$$

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 (22) に従う。実験で得た TDC 分布を以下の実験式でフィットする。

$$y(t) = p_0 \exp\left(-\frac{t}{p_1}\right) \quad (23)$$

式 (22) と式 (24) を比較すると、式 (24) の p_1 と式 (22) の τ が比例関係にあることが分かる。よって、 p_1 を求めることにより、平均寿命 τ を算出することができる。図 26 は、式 (22) 及び TDC 分布の概形である。

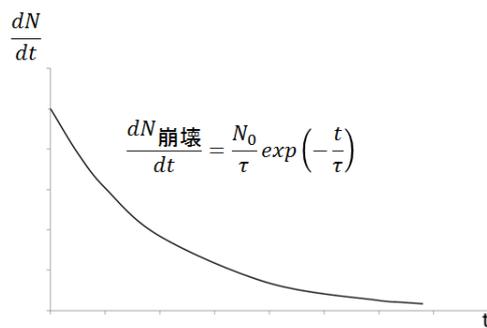


図 26: TDC 分布の概形図

3.2 CAMAC モジュールのセットアップ

μ 粒子の寿命測定の際のセットアップの概略図を図 27 に示す。A の Coincidence(μ 粒子がシンチレーター

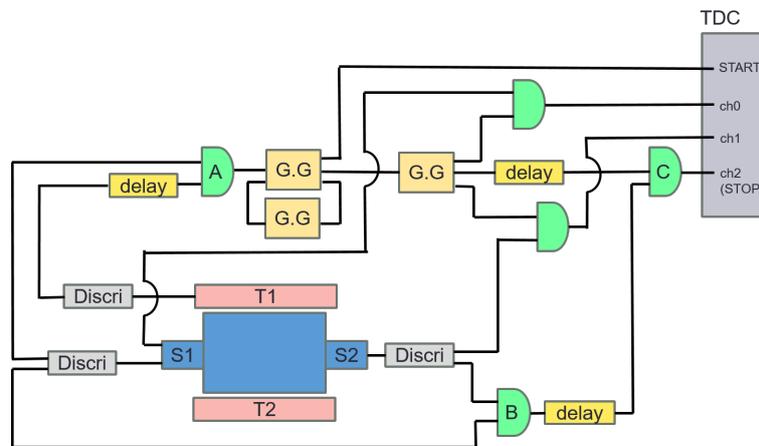


図 27: μ 粒子の寿命測定の際のセットアップ概略図

に入射したことを示すパルス) の信号を TDC の START 信号として入力し、C の Coincidence(μ 粒子が崩壊したあと電子を放出したことを示すパルス) の信号を TDC の STOP 信号として入力した。つまり、TDC の START 信号が鳴ってから STOP 信号が鳴るまでの時間は、 μ 粒子が入射してからシンチレーター内で崩壊するまでの時間となる。この時間を横軸にとり、その時間で崩壊したイベントの数を縦軸に取ることによって、崩壊曲線が得られる。

3.3 解析

3.3.1 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。実験で取得した 8590352 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

条件の選出方法

1. μ 粒子がメインシンチレーターに入射

T 1、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る。つまり ADC(c) 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 28,29 に示す。ただし、縦軸は対数表示となっている。ここから、es1>120 のとき、メインシンチレーター

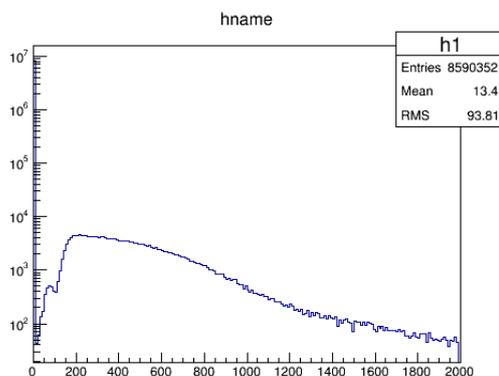


図 28: es1 の分布

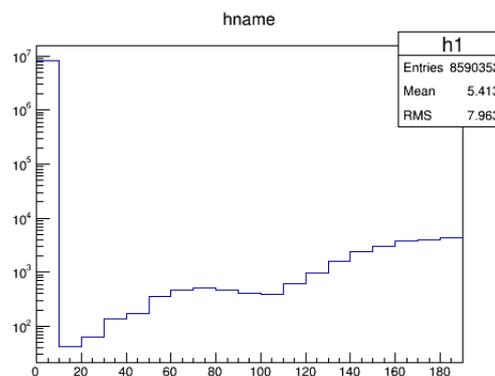


図 29: es1 の分布の上端を拡大したもの

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 223514 イベントとなった。

3.3.2 TDC 分布に関数を Fit する

TDC 分布に Fit した関数は、先述の通り以下の式である。

$$y(t) = p_0 \exp\left(-\frac{t}{p_1}\right) \quad (24)$$

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 60~1900 の間で Fit を行った。Fit したときの TDC 分布を図 30 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。

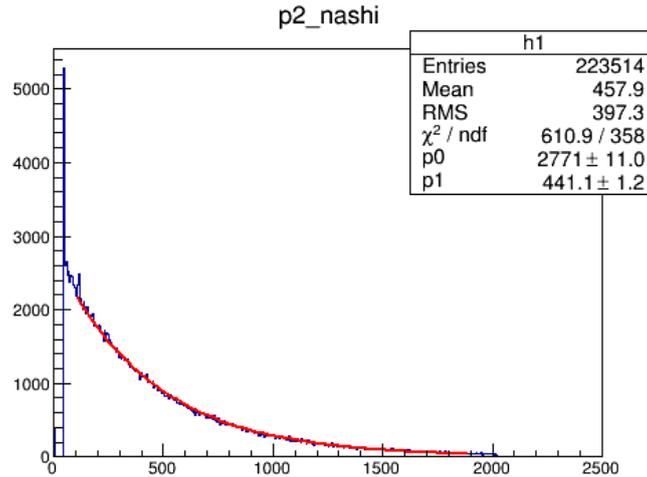


図 30: 実験式を Fit したときの TDC 分布

ここから、

$$p1 = 441.1 \pm 1.2[\text{count}] \quad (25)$$

となった。ただし、得られた p1 はあくまで TDC のカウント数であって、秒ではない。したがって、TDC の時間較正を行い、TDC カウント数を秒に換算する。

3.3.3 TDC 時間較正

前章で得られた TDC カウント数 p1 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 31)。

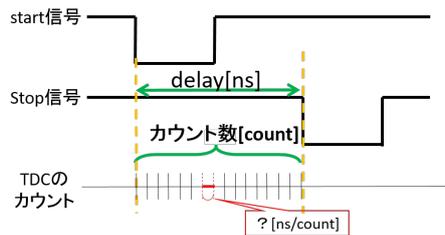


図 31: TDC 時間較正に用いるパルス

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 32 に示す。

実際に測定した delay 時間と TDC カウント数の関係を表 3.3.3 に示す。このカウント数は 1000 イベントのデータ平均値である。表 3.3.3 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 33 である。33 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 33 の p_0 , p_1 に対応する。よって、 a , b は

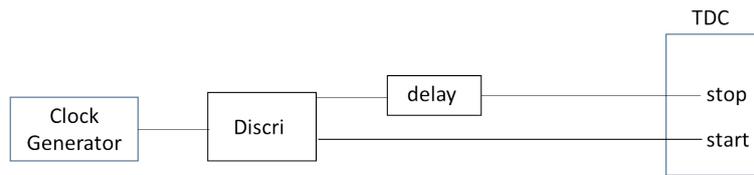


図 32: TDC 時間較正に用いるロジック

$$a = 4.982 \pm 0.01 \quad (26)$$

$$b = 0.565 \pm 0.17 \quad (27)$$

delay[ns]	0	10	20	30	100	200	225	300
TDC[count]	0.196	1.867	3.909	5.975	19.798	40.528	45.097	61.378
delay[ns]	500	560	800	1000	1500	2000	2500	3000
TDC[count]	100.097	111.835	159.985	199.947	300.934	400.617	501.403	600.497
delay[ns]	3500	3600	3900					
TDC[count]	698.905	715.813	788.194					

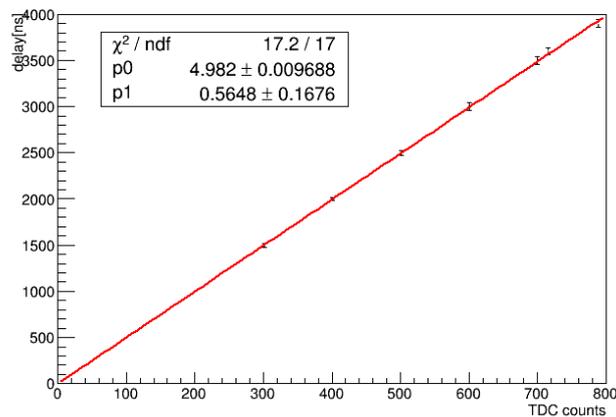


図 33: TDC 時間較正

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、

$$p1 = 5.003 \pm 0.004 [ns/count] \quad (28)$$

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、 $\frac{\chi^2}{ndf}$ は以下となった。

$$\frac{\chi^2}{ndf} = 17.2/17 \quad (29)$$

3.4 結果

μ 粒子の寿命を表す TDC カウント数 p_1 と、TDC1 カウント当たりの時間 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。

$$\tau[ns] = p_1[ns] \times \alpha[ns/count] \quad (30)$$

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。

$$\frac{\Delta\tau}{\tau} = \sqrt{\left(\frac{\Delta\alpha}{\alpha}\right)^2 + \left(\frac{\Delta p_1}{p_1}\right)^2} \quad (31)$$

ここで、 Δp_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 $p_1 = 437.6 \pm 1.4[ns]$ 、 $\alpha = 4.982 \pm 0.01[ns/count]$ 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。

$$\tau = 2.197 \pm 0.007 \quad (32)$$

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 の係数の統計誤差を含んでいる。

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 $\tau_{PDG} = 2.197034 \pm 0.000021[ns]$ である。実験結果と比較すると

$$\frac{\tau_{観測}}{\tau_{PDG}} = \frac{2.19756}{2.197034} = 1.00024 \quad (33)$$

よって文献値とのずれは 0.02 % である。

3.5 考察

TDC 分布は式 (21) に従うため、実験式 (22) 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 Fit 範囲を区切った。Fit 範囲による寿命の変化を以下の図 34 に示す。

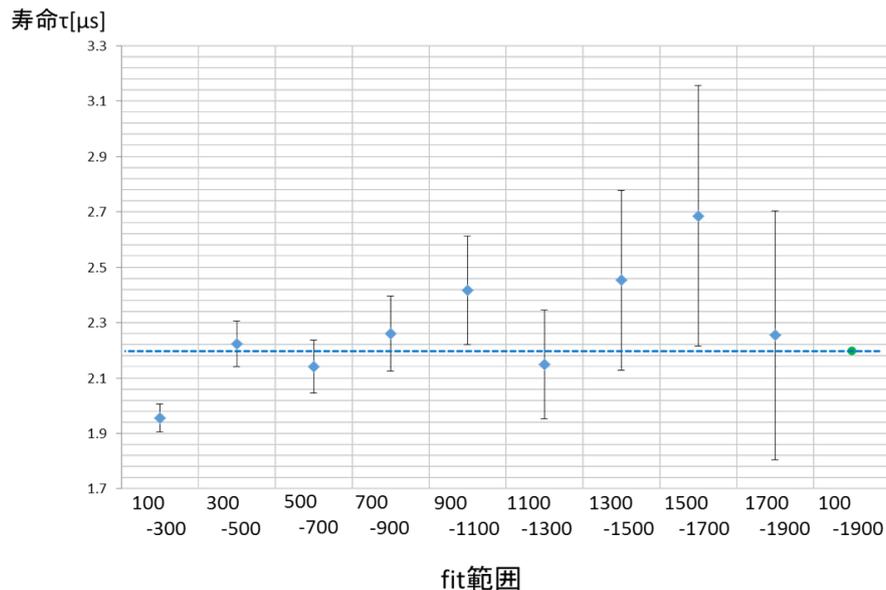


図 34: Fit 範囲と寿命の関係

図 34 の青色の点線は Fit 範囲 100~1900 の時の寿命 (前セクションで示した実験値) の値を表している。図 34 より Fit 範囲と寿命に相関関係は見られないことが分かる。また、Fit 範囲 100~300 の範囲のみ寿命が明らかに短くなっており、エラーの範囲で実験値の寿命とも一致していない。Fit 範囲に相関関係が見られないことと Fit 範囲 100~1900 で寿命が短くなっていることの原因究明は今後の課題である。

4 μ 粒子から崩壊した電子のエネルギースペクトラムの測定

4.1 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の原理について述べる。 μ 粒子崩壊のファインマン図と崩壊式を以下に示す。

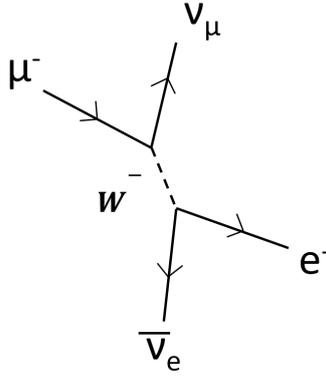


図 35: μ^- 崩壊のファインマン図

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

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 36 のようになる。図 36

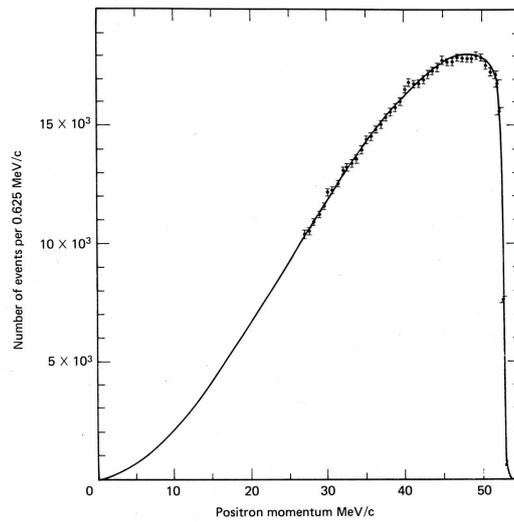


図 36: 電子のエネルギースペクトラムの理論値

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分かる。また、ここでは電子の質量を無視している。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。

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 (34) となる。

$$M = \frac{g_\omega^2}{8(M_\omega c)^2} [u(3)\gamma^\mu(1-\gamma^5)u(1)][u(4)\gamma_\mu(1-\gamma^5)u(2)] \quad (34)$$

ここで、 g_ω 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、

$$\langle [M^2] \rangle = 2\left(\frac{g_\omega}{M_\omega c}\right)^4 (P_1 \cdot P_2)(P_3 \cdot P_4) \quad (35)$$

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、 $P_1 = (m_\mu c, \vec{0})$ となり、 P_1 と P_2 の内積 $P_1 \cdot P_2$ は

$$P_1 \cdot P_2 = m_\mu E_2 \quad (36)$$

となる。また、エネルギー運動量保存則より

$$(P_3 + P_4)^2 = M_\mu^2 c^2 - 2P_1 \cdot P_2 \quad (37)$$

よって、

$$P_3 \cdot P_4 = \frac{(m_\mu^2 - m_e^2)^2 c^2}{2} - m_\mu E_2 \quad (38)$$

となる。式 (35) に式 (36)、(38) を代入すると、振幅 M は

$$M = \left(\frac{g_\omega}{M_\omega c}\right)^2 m_\mu^2 E_2 (m_\mu c^2 - 2E_2) \quad (39)$$

となる。式 (39) より、崩壊幅は次のように与えられる。

$$d\Gamma = \frac{\langle [M^2] \rangle c}{(4\pi)^4 h m_\mu} dE_2 \frac{d^3 \vec{p}_4}{E_4^2} \quad (40)$$

また、

$$d^3 \vec{p}_4 = 4\pi \left(\frac{E_4}{c}\right)^2 \frac{dE_4}{c} \quad (41)$$

なので、式 (40) に式 (41) を代入すると、

$$\frac{d\Gamma}{dE} = \left(\frac{g_\omega}{M_\omega c}\right)^4 \frac{m_\mu^2 E^2}{2h(4\pi)^3} \left(1 - \frac{4E}{3m_\mu c^2}\right) \quad (42)$$

を得られる。式 (42) 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 (42) で期待される分布より様々な理由で変形を受ける。その一つが高次の輻射補正で、もう一つが制動放射である。

4.1.1 物質中の荷電粒子 (電子) のふるまい

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、電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 $(\frac{Z}{m_e})^2$ 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。単位長さ当たり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。

$$\frac{dE}{dx} = \frac{NEZ^2\gamma_e^2}{137} \left(4 \ln \frac{183}{z^{\frac{1}{3}}} + \frac{2}{9} \right) \quad (43)$$

ここで、 N は物質の単位体積当たりの原子数 ($N = \frac{N_0\rho}{A}$)、 E は入射電子のエネルギー、 γ_e は電子の古典半径 ($\gamma_e = \frac{e_0^2}{m_e c^2} = 2.81 \times 10^{-13}(\text{cm})$)、 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 は物質の密度を示す。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(43)より

$$-\frac{dE}{E} = \frac{dx}{x_0} \quad (44)$$

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するとき使用する。

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、物質中の電子と衝突し、電離や励起を繰り返しながらエネルギーの一部を失う。これを電離損失という。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。

$$\frac{dE}{dx} = 4\pi N_a \gamma_e^2 m_e c^2 \rho \frac{Z}{A} \frac{z^2}{\beta^2} \left[\ln \left(\frac{2m_e \gamma^2 v^2 W_{max}}{I^2} \right) - 2\beta^2 - \delta - 2\frac{C}{Z} \right] (\text{MeV} \cdot \text{cm}^2/\text{g}) \quad (45)$$

ここで、

γ_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のエネルギー

ρ :物質の密度 δ :密度補正
 C :殻補正

である。

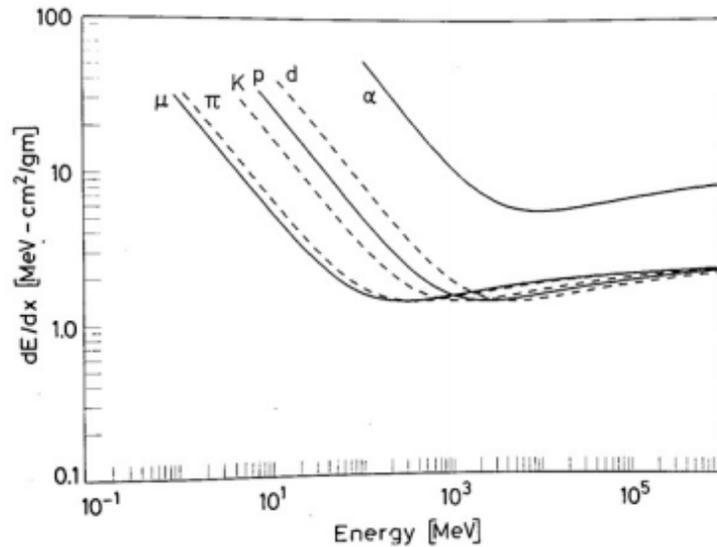


図 37: 電離損失

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

$$\frac{dE}{dx} \propto 1/\beta^2 \quad (46)$$

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。この領域で電離及び最小値のことを Minimum Ionization と呼ぶ。(宇宙で起きている。) 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、 \log の中の項が効くので、電離損失は $\log \gamma$ でエネルギーが増加すると上昇する。

$$\frac{dE}{dx} \propto \log\left[\frac{1}{1-\beta^2}\right] \quad (47)$$

$$\gamma = \frac{1}{1-\beta^2} = \frac{E}{m} \quad (48)$$

荷電粒子がシンチレーター内に入射したとき、シンチレーター内の分子を電離及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。

4.2 CAMAC モジュールのセットアップ

崩壊電子のエネルギースペクトラムを図る際のセットアップの概略図を図 38 に示す。S1 と S2 の Coin-

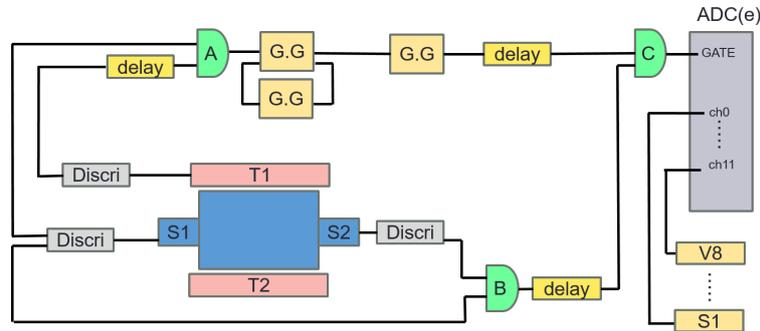


図 38: 崩壊電子のエネルギースペクトラム測定の際のセットアップ概略図

cidence(メインシンチレーター内を荷電粒子が通過した反応) を gate として ADC(e) に入力する。この時、メインシンチレーターは崩壊していない μ 粒子にも反応してしまうので、この反応を取り除きたい。そこで、T1 と S1 の Coincidence(メインシンチレーター内で μ 粒子が崩壊していることを表す信号) と、先ほどの S1 と S2 の Coincidence でさらに Coincidence をとる。すると μ 粒子がメインシンチレーター内で崩壊した反応についてのみの gate が作れる。

4.3 解析

4.3.1 イベントの選定

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に入射する。
2. μ 粒子がメインシンチレーターに入射し、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留まっている。

この 1~3 の条件を満たす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に入射する。
3.3.1 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に入射し、電子が放出される。
3.3.1 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留まっている。
ベーターカウンターが鳴っていないイベントを選定した。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、その値を下回るイベントを選出した。以下の図 (39)~図 (48) 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。

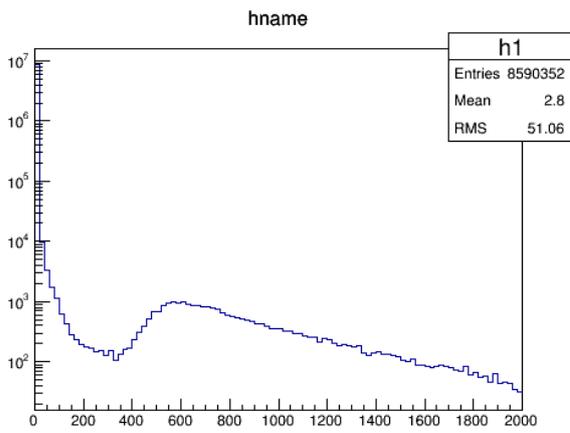


図 39: T1 の ADC 分布

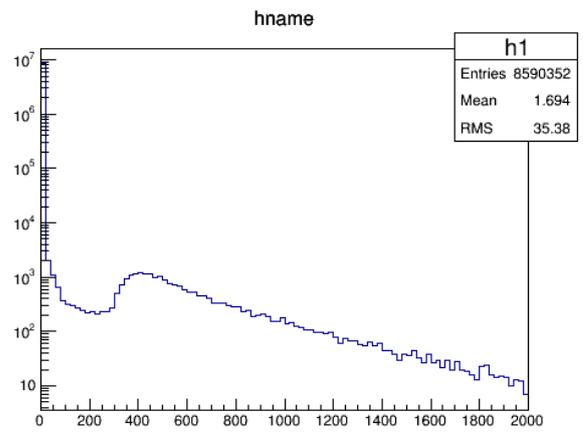


図 40: T2 の ADC 分布

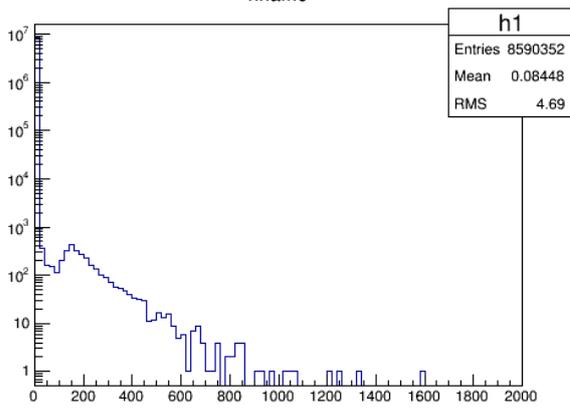


図 41: V1 の ADC 分布

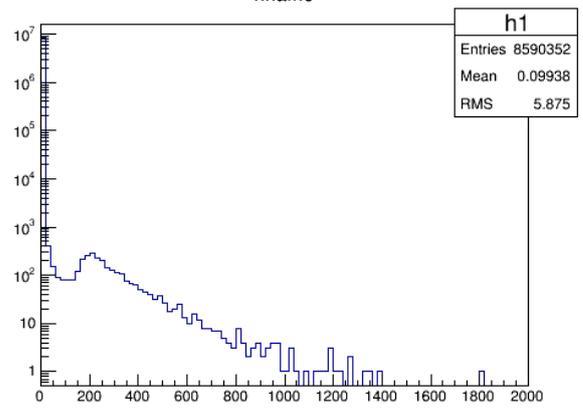


図 42: V2 の ADC 分布

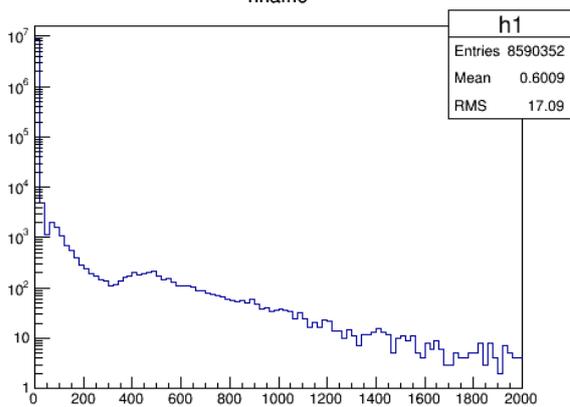


図 43: V3 の ADC 分布

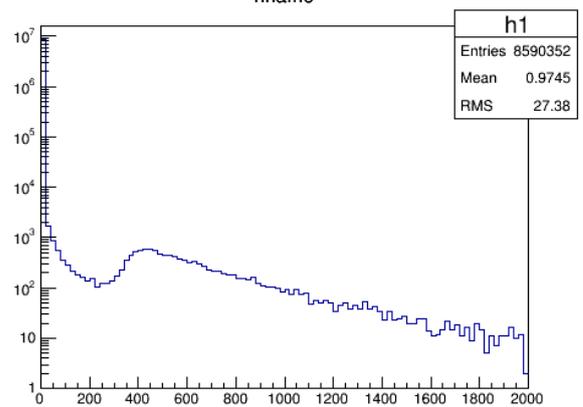


図 44: V4 の ADC 分布

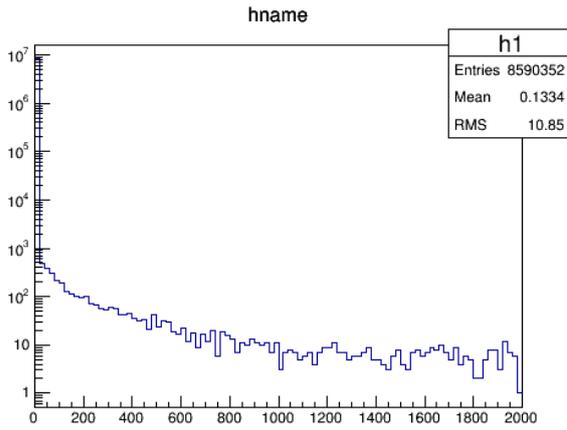


図 45: V5 の ADC 分布

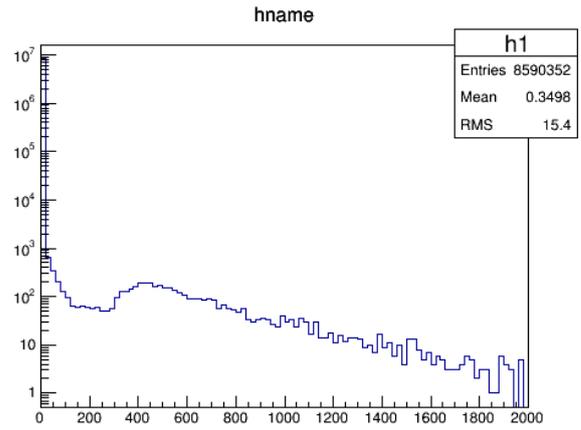


図 46: V6 の ADC 分布

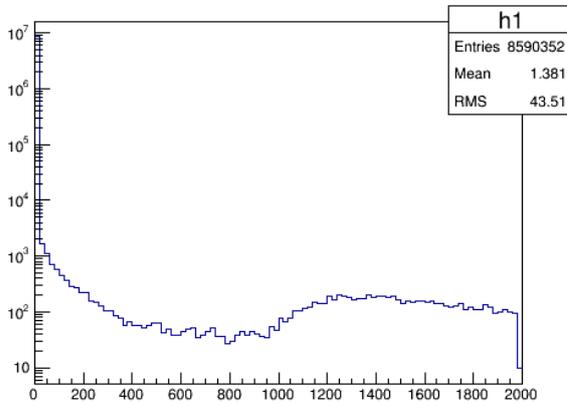


図 47: V7 の ADC 分布

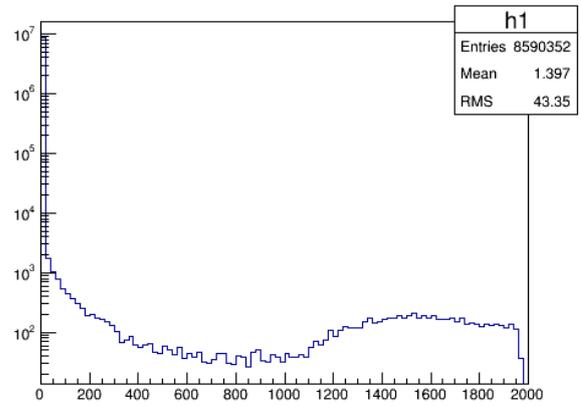


図 48: V8 の ADC 分布

よって、ベトーカーが反応したとみなす値は、

T1	T2	V1	V2	V3	V4	V5	V6	V7	V8
120	30	30	30	50	50	50	130	50	50

である。すべてのカウンターが上記の値を下回った時、電子がメインシンチレーター内で留まっているとみなした。

4.3.2 ADC エネルギー較正

ADC(e) を用いて電子のエネルギースペクトラムを測定するために、ADC(e) のエネルギー較正を行う。プラスチックシンチレーターを通過した際の μ 粒子のエネルギー損失は、

$$\Delta E(\text{MeV}) = \frac{dE}{dx}(\text{MeV} \cdot \text{cm}^2/\text{g}) \times \rho(\text{g}/\text{cm}^3) \times t(\text{cm}) \quad (49)$$

で求められる。ここで、 $\frac{dE}{dx}|_{min}$ は、minimum ionization の粒子に対する厚さ $1(\text{g}/\text{cm}^3)$ 辺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 $\frac{dE}{dx}|_{min} = 1.936(\text{MeV} \cdot \text{cm}^2/\text{g})$ 、 $\rho = 1.06(\text{g}/\text{cm}^3)$ とする。これより、本実験で使用した厚さ 26cm を通過するプラスチックシンチレーターのエネルギー損失は

$$\Delta E = 1.936 \times 1.06 \times 26 = 53.3(\text{MeV}) \quad (50)$$

となる。この値を用いて、ADC (e) のエネルギー較正を行う。

まず、ADC(e) を用いて μ 粒子のエネルギースペクトラムを測定する。その時に用いるイベントの選出条件は以下の通りである。

条件

1. μ 粒子がメインシンチレーターに入射する。
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 1~3 の条件を満たす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に入射する。
3.3.1 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 (49) 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 (50) に T2 の ADC 分布を示す。

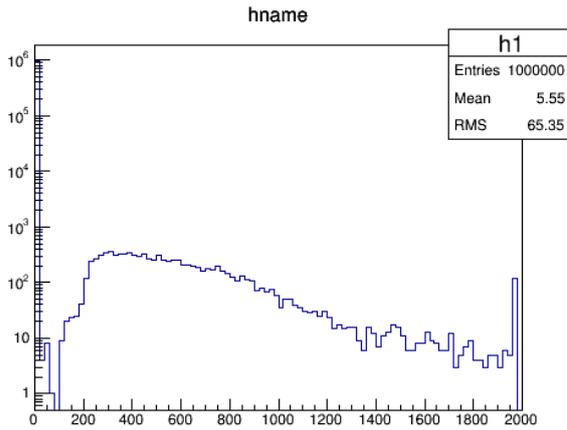


図 49: ADC(μ) の S1 の ADC 分布

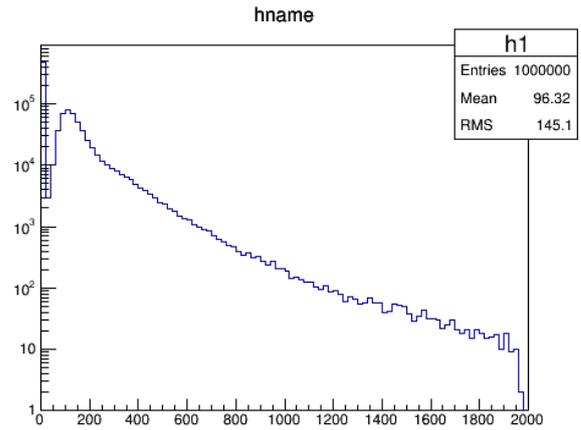


図 50: ADC(e) の T2 の ADC 分布

図 (49), 図 (50) より、S1 は channel 数 150 以下、T2 は channel 数 120 以上の値を用いた。

上記の条件 1~3 を満たすイベント数は 100 万イベント中、314060 イベントであった。ADC(e) を用いて測定した μ 粒子のエネルギー分布を図 (51) に示す。この図は ADC(e) の S1 と S2 の値の和の ADC 分布である。

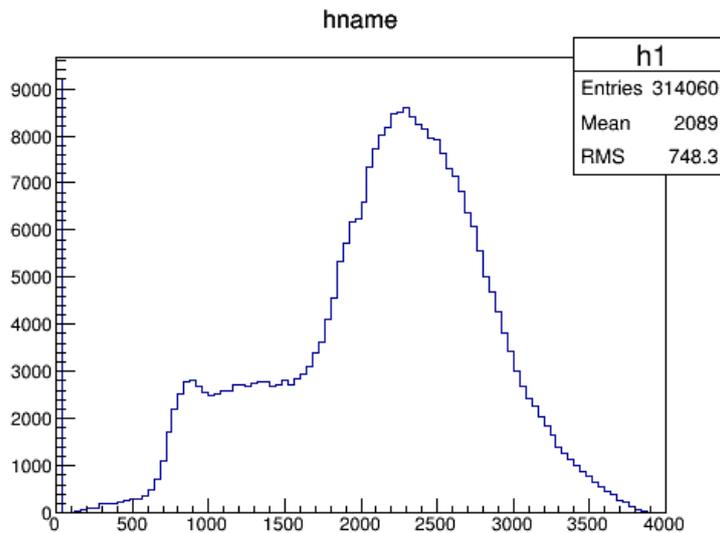


図 51: メインシンチレータを通過した μ 粒子のエネルギーの分布

図 (51) より、channel 数 2300 辺りに分布のピークがあることが分かった。式 (50) で示したように、 μ 粒子がメインシンチレータを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 (51) の 2300ch 辺りのエネルギーが約 50MeV であることが分かった。

4.4 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 (52) に示す。図 (53) は式 (42) で得られる静止した 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。

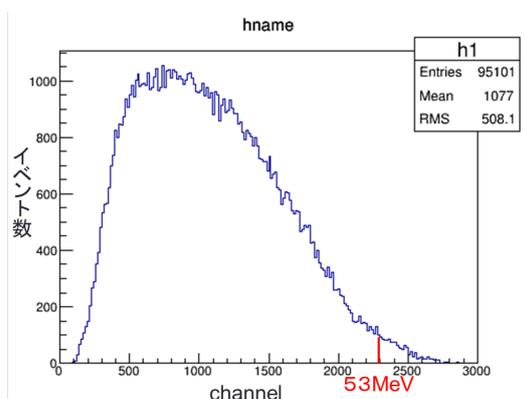


図 52: 電子のエネルギースペクトラムの測定結果

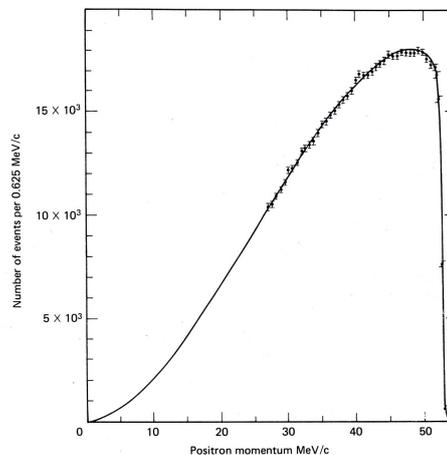


図 53: 電子のエネルギースペクトラムの理論値

4.5 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 Geant4 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。

4.5.1 Geant4

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 、 $\bar{\nu}_e$ に崩壊する事象とする。

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を検出器として設定した。図 54 に Geant4 で設定した検出器の模式図を示す。

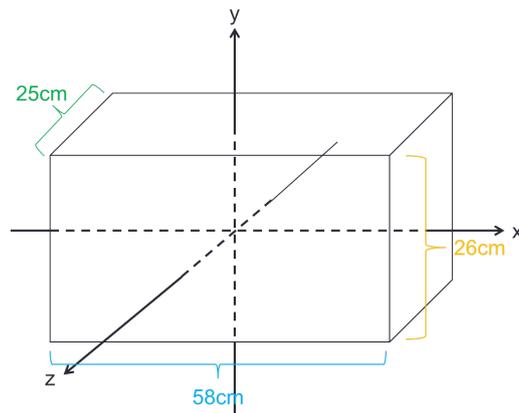


図 54: Geant4 で設定した検出器

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 55 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。

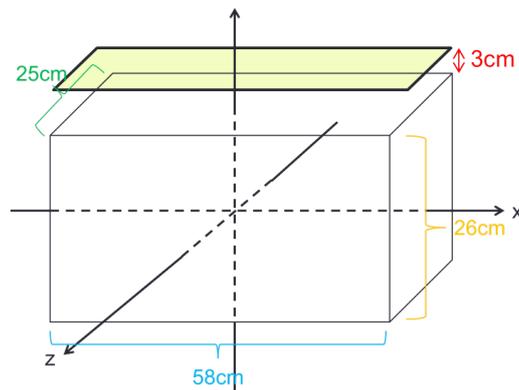


図 55: 入射粒子の位置

3. 入射粒子の進行方向

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 (x_m, y_m, z_m) をベクトルとして指定する。 x_m 、 y_m 、 z_m 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 軸方向は下向きに大きさの 1 の単位ベクトルで固定した。図 56 に示すように角度 θ 、 Φ を指定し、 x_m 、 y_m 、 z_m を決めた。図 56 より、運動量方向の成分 x_m 、 y_m 、 z_m は以下の式でそれぞれ求められる。

$$x_m = y_m \times \tan\theta \times \cos\phi \quad (51)$$

$$y_m = -1 \quad (52)$$

$$z_m = y_m \times \tan\theta \times \sin\phi \quad (53)$$

ここで宇宙線について考える。図 57 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。

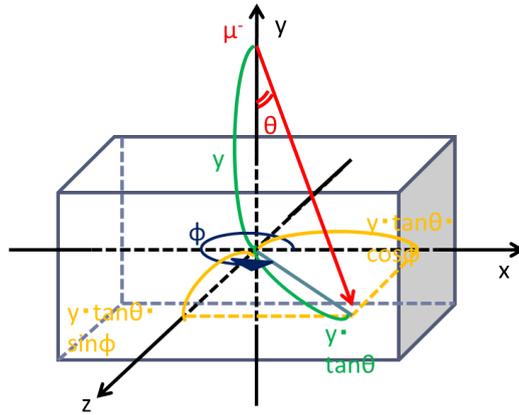


図 56: 入射粒子の運動量方向

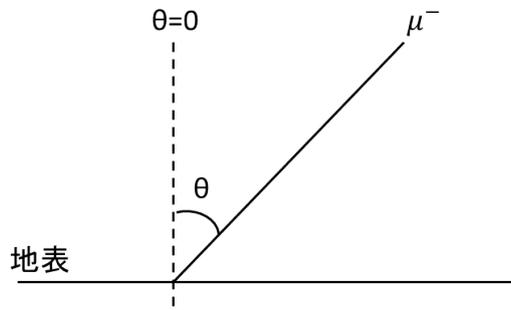


図 57: 宇宙線の地表付近の様子

地表に到達する宇宙線の数は天頂角 $\theta = 0$ 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たどり着くからである。

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 $J(\theta)$ とすると、天頂角 θ から飛来する μ 粒子の強度 $J(\theta)$ は

$$J(\theta) = J(0)\cos^2\theta \quad (54)$$

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、

$$J(0) = 0.83 \times 10^{-2} (\text{個}/\text{cm}^2 \cdot \text{sec} \cdot \text{sr}) \quad (55)$$

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

これより、天頂角 θ を $0 \sim \frac{\pi}{2}$ 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 $0 \sim 2\pi$ 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

入射粒子のエネルギーは、シンチレーター内で静止して、電子を放出する時の μ 粒子のエネルギー分布より決定した。その時の ADC 分布を以下に示す。縦軸がエントリー数、横軸が ADC のチャンネル数である。図 58 から分かるように、エネルギーは 500ch \sim 3000ch に分布していることが分かる。

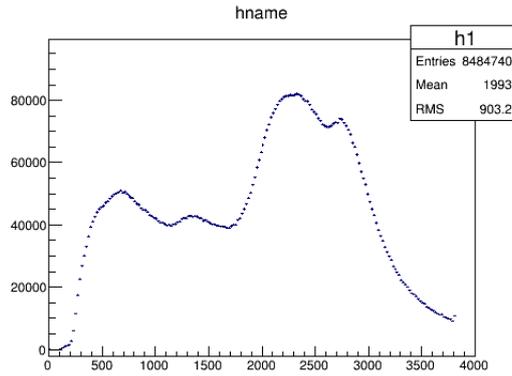


図 58: μ 粒子が電子に崩壊する時の ADC 分布

4.3.2 で求めたように、ADC の 2300ch が約 53.3MeV である。つまり 1ch 当たり 0.02665MeV であり、これより 500ch は約 13MeV、3000ch は約 80MeV と計算できるため、13MeV~80MeV の間で一様乱数を発生させた。

4.5.2 シミュレーション結果

前セクションで示したように、検出器や入射粒子を決定し 10 万発の粒子を入社させた時のシミュレーション結果を以下に示す。図 59 は μ 粒子が検出器内で電子に崩壊した直後の電子の運動エネルギーである。縦軸がエントリー数、横軸がエネルギー (MeV) を表している。図 60 は静止した 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 59 から分かるように、検出器内で崩壊した直後は理論値と

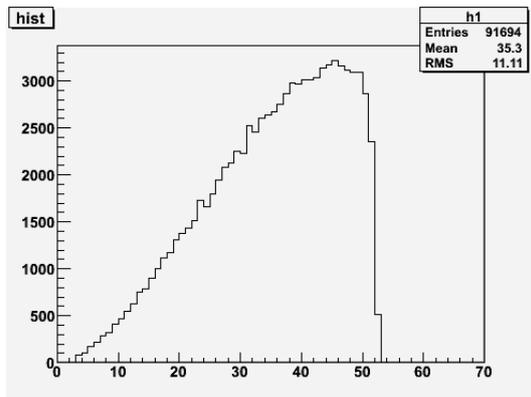


図 59: 電子の運動エネルギーのシミュレーション結果

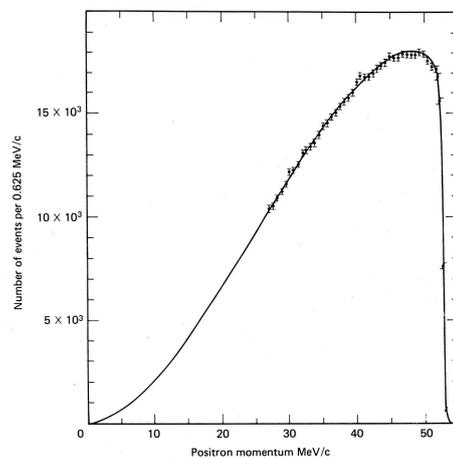


図 60: 電子のエネルギースペクトラムの理論値

概形、ピーク位置とも一致することが分かった。

4.6 考察

本実験での崩壊電子のエネルギースペクトラムの測定結果は高エネルギー部が少なく、ピーク位置は約35MeVであった。理論値と比較すると、ピーク位置、概形共に異なる原因は電子の制動放射により検出器の外に逃げて行った γ 線のエネルギーを捉えきれていないためと考えられる。つまり本実験では、検出器内の電子のエネルギーのみの分布を測定していると考えられる。

5 まとめ

5.1 実験のまとめ

1. μ 粒子の寿命測定

μ 粒子の寿命 $\tau = 2.198 \pm 0.007 [\mu\text{s}]$ となった。これは、文献値 $\tau_{PDG} = 2.197034 \pm 0.000021 [\mu\text{s}]$ と比較すると、0.02 % のズレであり、精度の良い結果が得られた。TDC分布のFit範囲100~300までの間で、寿命が短くなったことの原因究明が今後の課題である。

2. 電子のエネルギースペクトラムの測定崩壊電子のエネルギースペクトラムについては、約20MeVの位置にピークが見られたものの、理論値と実験結果を比較すると、分布の概形が異なりピーク位置に約30MeVの差があった。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であると考えられる。したがって、今後、理論値と違う結果が見られた原因をシミュレーションの再現性を高め、考察する必要があると考えられる。

5.2 謝辞

私たちの卒業研究の為、お忙しい中熱心に指導して下さった林井先生をはじめ、研究するにあたって相談に乗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、プログラムの動かし方、Geant4の動かし方等、手取り足取り教えて下さった先輩方に感謝しております。本当にありがとうございました。

5.3 参考文献

1. 「2015年度卒業論文 μ 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
池田侑加 坂本朋子
2. 「2014年度卒業論文 μ 粒子の寿命と崩壊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
3. 「INTRODUCTION TO ELEMENTARY PARTICLES」 DAVID GRIFFITHS p.308
4. 「素粒子実験(ミューオンの寿命測定実験)」 九州大学 織田勸

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/I/O 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;  
    int data1,data2,data3,data4,data5,data6,data7,data8,data9,data10,data11,data12;  
    int cadcq1,cadcq2,ctdcq;  
    int ch0,ch1,ch2,ch3,ch4,ch5,ch6,ch7,ch8,ch9,ch10,ch11;  
    int lamsrc;  
    int lamch;  
  
    char fname[36];  
  
    int qq;  
    int head;  
  
    cadcq1=1; /* muon ADC-Qmode module number */  
    cadcq2=2; /* electron ADC-Qmode module number */  
    ctdcq=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);

/*=====
 * Enable LAM and Clear it.
 *=====*/
/*CAMAC(NAF(lamsrc,lamch,26),&data1,&q,&x); F=26 is enable. */
/*CAMAC(NAF(lamsrc,lamch, 9),&data1,&q,&x); F=9 is clear. */
/*CAMAC(NAF(cadcqn2,ch1,26),&data1,&q,&x); F=26 is enable. */
/*CAMAC(NAF(ctdcqn,ch1,26),&data1,&q,&x); F=26 is enable. */
/*CAMAC(NAF(cadcqn2,ch1,9),&data1,&q,&x); clear LAM. */
/*CAMAC(NAF(ctdcqn,ch1,9),&data1,&q,&x); clear LAM */

/*=====
 * Again send enable command to prepare the first event.
 *=====*/
    CAMAC(NAF(lamsrc,lamch,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(lamsrc,lamch,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(cadcqn2,ch4,0),&data5,&qq,&x);
CAMAC(NAF(cadcqn2,ch5,0),&data6,&qq,&x);
CAMAC(NAF(cadcqn2,ch6,0),&data7,&qq,&x);
CAMAC(NAF(cadcqn2,ch7,0),&data8,&qq,&x);
CAMAC(NAF(cadcqn2,ch8,0),&data9,&qq,&x);
CAMAC(NAF(cadcqn2,ch9,0),&data10,&qq,&x);
CAMAC(NAF(cadcqn2,ch10,0),&data11,&qq,&x);
CAMAC(NAF(cadcqn2,ch11,0),&data12,&qq,&x);
//2014/11/18:
// Use new adc at slot 5, channel 0, since we can not read the adc
// at channel 11.
//CAMAC(NAF(4,0,2),&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 time calibration*/

/*-----
 * Update the event counter(i_ev), and send message for
 * every 50 events or there are data for big electron signal.
 *-----*/
//if(data12>100)
if( i_ev%100 == 1 )
// if( i_ev%500 == 1 || data1>50 )
//if( data1>10 ) /* Cut of event not decay to electron */
{
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 time calibration*/

}

head=-1;

/*-----
 * The read data is written into the file.

```

```

*-----*/

if( data1>10 ) /* Cut of event 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 time calibration*/
}

/*-----
* 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++;

/*CAMAC(NAF(4,0,9),&data1,&q,&x);
CAMAC(NAF(lamsrc,lamch,9),&data1,&q,&x);
CAMAC(NAF(lamsrc,lamch,10),&data1,&q,&x); */

}/*   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;
}

```

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="2017-01-11-1000000ev.dat"; /*Source File name*/

    string    directory1="/home/2015-b4/ug2016/data2016/";
    string    directory2="/home/2015-b4/ug2016/data2016/root_file/";
    string    full_name1 = directory1+ fname;
    string    full_name2 = directory2+ fname+ ".root";
    cout <<" f-name"<< full_name1<<endl;

    ifstream data(full_name1.c_str()); /*open file*/
    float one, ev, mus1, mus2, mut1, mut2, es1, es2, et1, et2, ev1, ev2, ev3, ev4, ev5, ev6, ev7, ev8, tdc0, tdc1, tdc2;
    /*declare type*/

    TTree *mu_t=new TTree("mu_t","title"); /*definition Tree*/
    mu_t->Branch("one", &one, "one/F"); /*definition Branch*/
    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;
    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=mus1-109; /*subtract pedestal*/
        if(mus1<0){mus1=0;}
        mus2=mus2-98;
        if(mus2<0){mus2=0;}
        mut1=mut1-89;
        if(mut1<0){mut1=0;}
        mut2=mut2-85;
        if(mut2<0){mut2=0;}
        es1=es1-58;
        if(es1<0){es1=0;}
        es2=es2-63;
        if(es2<0){es2=0;}
        et1=et1-56;
```

```

    if(et1<0){et1=0;}
    et2=et2-60;
    if(et2<0){et2=0;}
    ev1=ev1-67;
    if(ev1<0){ev1=0;}
    ev2=ev2-63;
    if(ev2<0){ev2=0;}
    ev3=ev3-57;
    if(ev3<0){ev3=0;}
    ev4=ev4-63;
    if(ev4<0){ev4=0;}
    ev5=ev5-65;
    if(ev5<0){ev5=0;}
    ev6=ev6-64;
    if(ev6<0){ev6=0;}
    ev7=ev7-66;
    if(ev7<0){ev7=0;}
    ev8=ev8-80;
    if(ev8<0){ev8=0;}

    if( i%500000==1 ){
    cout<<"i="<<i<<" one="<<one<<" ev="<<ev<<" mus1="<<mus1<<" mus2="<<mus2<<endl; 出力//
    }
    mu_t->Fill();
    i++;
};

data.close(); /*file close*/

TFile *rtfilemt = new TFile(full_name2.c_str(),"RECREATE"); /*make root file*/
mu_t->Write();
rtfilemt->Close(); /*close root file*/

TFile *f= new TFile(full_name2.c_str(),""); /*open root file*/
f->ls();

return ;
}

```

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 "TChain.h"
using namespace std;

void histogramOutput(){
  gStyle->SetOptFit(1);
  TCanvas *c1 = new TCanvas("c1", "ctitle", 0, 0, 500, 400); /*make new canvas*/
  TChain *t=new TChain("mu_t","mu_title"); /*make Chain*/
  t->Add("/home/2015-b4/ug2016/data2016/root_file/*.root"); /add Chain*/
  TH1F *h1= new TH1F("h1", "hname",100,0,2000); /*make new histogram*/

  //TCut positiveCut = "120<es1 && et1<120 && et2<30 && ev1<30 && ev2<30 && ev3<50 &&
  ev4<50 && ev5<50 && ev6<130 && ev7<50 && ev8<50 && mus1<120 && mut2>60";

  //TCut positiveCut = " mut2>120 && es1<150 ";
  //TCut tdc = "tdc2<4000";

  c1->SetLogy(); /*set y axis log scale*/
  t->Draw("ev1>>h1","",""); /*output histogram*/
  c1->Print( "/home/2015-b4/ug2016/anal2016/counterhistgram/ev1.png" ); /*save graph image*/
}
}
```

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); /*make new canvas*/
  //c1->Divide(2,1); /*divide canvas*/

  TChain *mu_ch=new TChain("mu_t","mu_title"); /*make Chain*/
  mu_ch->Add("/home/2015-b4/ug2016/data2016/root_file/*.root"); /*add Chain*/

  TCut e_decay="120<es1 && tdc2<4000";
  TH1F *h1= new TH1F("h1", "p2_nashi",500,0,2500); /*make new histogram*/
  mu_ch->Draw("tdc2>>h1,e_decay,"); /*output histogram*/

  /* Fit */
  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);
  f_exp_y->SetParameter(1,435);
  h1->Fit("f_exp_y","","",1660,1900); /*fit f_exp_y*/
  gStyle->SetOptFit(1); /*view the parameters on the canvas*/
}
```

```
Double_t p0_y = f_exp_y->GetParameter(0); /*get p0*/
Double_t p1_y = f_exp_y->GetParameter(1); /*get p1*/

cout << " p0_y=" << p0_y << " p1_y=" << p1_y << endl; /*view the parameters*/

c1->Print( "hist_lifetime01-24.png" );/*save graph image*/

return ;
}
```

E hist_test.cc

ROOTによる電子のエネルギースペクトラムの作成

```
#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); /*view the parameters on the canvas*/

    TCanvas *c1 = new TCanvas("c1", "ctitle", 0, 0, 500, 400); /*make new canvas*/

    TChain *mu_ch=new TChain("mu_t","mu_title"); /*make Chain*/
    mu_ch->Add("/home/2015-b4/ug2016/data2016/ADC_energy_calibration/*.root"); /*add Chain*/

    TH1F *h1= new TH1F("h1", "hname",100,0,4000); /*make new histogram*/

    //TCut EnergyCut = "100<es1 && et1<120 && et2<30 && ev1<30 && ev2<30 && ev3<50 &&
ev4<50 && ev5<50 && ev6<130 && ev7<50 && ev8<50";
    mu_ch->Draw("mus1+mus2>>h1",EnergyCut);
    h1->Fit("gaus","","",1600,3500);/*gaus fit between 1600-3500*/

    c1->Print( "/home/2015-b4/ug2016/anal2016/mus1+mus2_fit.png" ); /*save graph image*/

    return ;
}
```

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 $
//
//
//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

#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"

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

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((G4UniformRand()*(80.0-13.0)+13.0)*MeV);
    particleGun->SetParticlePosition(G4ThreeVector(0.*cm,30.*cm,0.*cm));//const

    /* ComentOut->RandomPosition Comand->FixedPosition */
    particleGun->SetParticleMomentumDirection(G4ThreeVector(0.,-1.,0.));//const
```

```

}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

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

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

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);//-29,29
    y0 = 16;
    z0 = ((G4UniformRand()-0.5)*25);//-12.5,12.5

    particleGun->SetParticlePosition(G4ThreeVector(x0*cm,y0*cm,z0*cm));

    //particle momentum direction
    /*
    //ichiyouransuu
    G4double xm = (G4UniformRand()-0.5)*cm;//-0.5,0.5
    G4double ym = (G4UniformRand()*(-0.5)-0.5)*cm;//-1,-0.5
    G4double zm = (G4UniformRand()-0.5)*cm;//-0.5,0.5
    */
    //gauss
    /*
    G4double xm = (G4RandGauss(*CLHEP::HepRandom::getTheEngine()).shoot(0.,0.1));//ave0,0.5
    G4double ym = -1.;
    G4double zm = (G4RandGauss(*CLHEP::HepRandom::getTheEngine()).shoot(0.,0.1));//ave0,0.5
    particleGun->SetParticleMomentumDirection(G4ThreeVector(xm,ym,zm));
    */

    /* ComentOut->FixedPosition Comand->RandomPosition */
    //tenchoukaku theta phi

    G4double rt=0.,ct=0.,t,p,xm,ym,zm ; //kakudo random
    //theta
    while ( ct <= rt )
    {
        rt = G4UniformRand();
        t = (G4UniformRand()-0.5)*M_PI;
        ct = cos(t)*cos(t);
    }

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

    ym = -1;
    xm = ym*tan(t)*cos(p);
    zm = ym*tan(t)*sin(p);

    particleGun->SetParticleMomentumDirection(G4ThreeVector(xm,ym,zm));
}

```

```

    particleGun->GeneratePrimaryVertex(anEvent);
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

```

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 $
//
//
//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

#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;

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

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("2017-02-28-100ev.dat", std::ios::out);
    if(! ofs.good())
    {
        G4cout<<" EventAction; error opening 2017-02-28-100ev.dat. ofs.good="<<ofs.good()<<G4endl;
    }
}

```

```

}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

ExN03EventAction::~ExN03EventAction()
{
    ofs.close();
    G4cout<<" EventAction: 2017-02-28-100ev.dat file is closed"<<G4endl;
    delete eventMessenger;
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

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.;
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

void ExN03EventAction::EndOfEventAction(const G4Event* evt)
{
    G4double energy = ExN03SteppingAction::Instance()->GetEnergy();
    G4double fEnergySum += energy;
    G4double fEnergy2Sum +=energy*energy;

    //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
            << "    Gap: total energy: " << std::setw(7)
                << G4BestUnit(EnergyGap,"Energy")
            << "    total track length: " << std::setw(7)
                << G4BestUnit(TrackLGap,"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;
    }
}

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

```

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 $
//
//
//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....
//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

#include "ExN03SteppingAction.hh"
#include "ExN03DetectorConstruction.hh"
#include "ExN03EventAction.hh"
#include "G4Step.hh"
#include <fstream>
std::ofstream ofss;
#include "ExN03SteppingVerbose.hh"
#include "G4SteppingVerbose.hh"

//...ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

ExN03SteppingAction::ExN03SteppingAction(ExN03DetectorConstruction* det,
                                           ExN03EventAction* evt)
:detector(det), eventaction(evt)
{
    /* edep -> energy of just after decay of muon */
    /* ofss.open("edep.dat", std::ios::out);
    if(! ofss.good()){

```

```

    G4cout<<" SteppingAction; error opening edep.dat. ofs.good="<<ofs.good()<<G4endl;
  }
*/
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

ExN03SteppingAction::~ExN03SteppingAction()
{ /*
  ofs.close();
  G4cout<<" SteppingAction: edep.dat file is created"<<G4endl;
  */
}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

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()->GetLogicalVolume();
  // 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()
    == "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,step1);
if (volume == detector->GetGap())      eventaction->AddGap(edep,step1);

/* Kill out of detector particle */
/*
if (aTrack->GetDefinition()->GetParticleName()
    == "e-"){
    if( 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)
        { aTrack->SetTrackStatus(fSuspend);
          aTrack->SetTrackStatus(fKillTrackAndSecondaries);};
    }
*/
//example of saving random number seed of this event, under condition
//if (condition) G4RunManager::GetRunManager()->rndmSaveThisEvent();

}

//....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....ooo00000ooo.....

```