

2019 年度 卒業論文

μ^- 粒子の寿命測定

奈良女子大学 理学部
数物科学科物理学コース 高エネルギー物理学研究室

下総七海

2020 年 2 月 21 日

目次

第 1 章	宇宙線・ μ 粒子	1
1.1	宇宙線とは	1
1.2	μ 粒子とは	2
1.3	真空中の μ 粒子の崩壊	3
1.4	μ 粒子の崩壊と寿命	4
第 2 章	測定原理とセットアップ	9
2.1	全体の流れ	9
2.2	シンチレーションカウンター	10
2.3	NIM	15
2.4	CAMAC	17
2.5	エレクトロニクスのセットアップ	23
第 3 章	μ 粒子の寿命測定	27
3.1	原理	27
3.2	イベントの選定	30
3.3	TDC の時間較正	36
3.4	結果	39
3.5	考察	43
第 4 章	まとめ	47
参考文献		51

目次

1.1	宇宙線の模式図	1
1.2	μ^+ 粒子の崩壊のファインマン図	4
1.3	μ^- 粒子の崩壊のファインマン図	5
1.4	ミューオン原子の模式図	5
1.5	μ 粒子の原子核捕縛	6
1.6	Compendium of total muon capture results for light nuclei	8
2.1	セットアップの概略図	9
2.2	左図:シンチレーションカウンター 右図:シンチレーションカウンター 模式図	10
2.3	実際のシンチレーターの配置	11
2.4	シンチレーションカウンターの模式図	12
2.5	シンチレーター内の様子と入射 μ 粒子の信号	14
2.6	シンチレーター内の様子と入射 μ 粒子と e^- の信号	14
2.7	左図:本実験で使用している NIM モジュール 右図:主な NIM モジュール	15
2.8	LAM をチェックする行程	18
2.9	LAM をチェックするフローチャート	18
2.10	本実験で使用している CAMAC モジュール	19
2.11	Gate 信号と入力パルス	20
2.12	TDC の仕組み	20
2.13	データ収集プログラムのフローチャート	22
2.14	エレクトロニクスの全体図	23
2.15	ADC(μ) の回路図	24
2.16	ADC(μ) のロジック図	24
2.17	ADC(e) の回路図	25
2.18	ADC(e) のロジック図	25
2.19	TDC の回路図	26
2.20	TDC のロジック図	26
3.1	放射性崩壊の図	28

3.2	TDC 分布の概形図	29
3.3	装置の簡略図	30
3.4	es1 の ADC 分布 (左図: 取得した全イベント、右図: 「磁石を通過」の 選定後)	31
3.5	μ^- 粒子 (電荷の区別あり) の寿命測定の模式図	32
3.6	ADC(μ) に入力した Gate 信号のロジック図	34
3.7	mut3,mut4 の ADC 分布	34
3.8	TDC 時間較正に用いるパルス	36
3.9	TDC 時間較正に用いるロジック	36
3.10	TDC 時間較正	37
3.11	右図: μ 粒子の TDC 分布 (2019 年度)、左図: Pull	39
3.12	右図: μ^- 粒子の TDC 分布 (2019 年度)、左図: Pull	41
3.13	μ 粒子の TDC 分布 (2019 年度)	44
3.14	μ 粒子の TDC 分布 (2018 年度)	44
3.15	μ 粒子の TDC 分布 (2018~2019)	44
3.16	μ^- 粒子の TDC 分布 (2019 年度)	46
3.17	μ^- 粒子の TDC 分布 (2018 年度)	46
3.18	μ^- 粒子の TDC 分布 (2018~2019)	46

表目次

2.1	シンチレーターのサイズ	12
2.2	各シンチレーターの仕様	13
3.1	Delay time と TDC カウント数	37
3.2	μ 粒子の平均寿命	43
3.3	μ^- 粒子の平均寿命	45
4.1	μ 粒子 (電荷の区別あり、なし) の平均寿命	47

第1章

宇宙線・ μ 粒子

1.1 宇宙線とは

宇宙線とは、宇宙空間から絶えず飛来する高エネルギーの放射線である。宇宙では高エネルギーの粒子が飛び交っていて、地球に入射すると大気中の窒素原子や酸素原子の原子核と衝突して様々な粒子を大量に作り出す。このとき、宇宙から大気圏へ入射する前の宇宙線を「一次宇宙線」と呼び、主に陽子と α 粒子から成る。大気圏に入射後、大気中の原子核と相互作用して生成された粒子を「二次宇宙線」といい、生成された中間子は地表に向かって飛んでいる途中で光子、電子、陽電子、 μ 粒子、ニュートリノなどの様々な素粒子に変化しながら雪崩のように数を増やしていく。この現象を「空気シャワー現象」と呼ぶ。このうち地表に到達するのは μ 粒子とニュートリノがほとんどである。

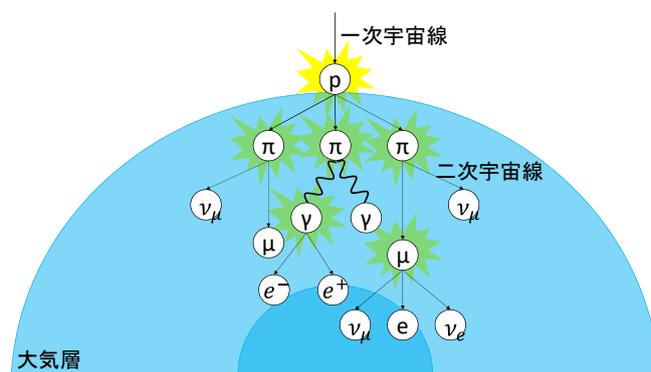


図 1.1 宇宙線の模式図

1.2 μ 粒子とは

1.2.1 π 中間子

π 中間子とは、二次宇宙線の一種であり、高速の陽子が大気中の原子核と衝突することで生成される。電荷をもたない π^0 ($139.57 \text{ MeV}/c^2$)、荷電粒子である π^- 、 π^+ ($134.98 \text{ MeV}/c^2$) が存在し、以下のように崩壊する。

$$\begin{aligned}\pi^0 &\rightarrow \gamma\gamma \\ \pi^- &\rightarrow \mu^- + \bar{\nu}_\mu \\ \pi^+ &\rightarrow \mu^+ + \nu_\mu\end{aligned}$$

1.2.2 μ 粒子

μ 粒子は、 π 中間子の崩壊により生成される。質量は $m_\mu = 105.658369 \pm 0.000009 \text{ MeV}$ 、荷電粒子であり電荷 ± 1 に応じて μ^- と μ^+ の 2 種類がある。 μ 粒子のスピンは $1/2$ で μ^- と μ^+ は互いに粒子と反粒子の関係であるので質量は等しい。 μ 粒子はほぼ 100% の確率で以下のように崩壊する。

$$\begin{aligned}\mu^- &\rightarrow e^- + \bar{\nu}_e + \nu_\mu \\ \mu^+ &\rightarrow e^+ + \nu_e + \bar{\nu}_\mu\end{aligned}$$

本実験ではこの μ 粒子の寿命を測定していく。

1.3 真空中の μ 粒子の崩壊

崩壊幅（最低次）

$$\Gamma = \frac{1}{\tau_{\mu^-}} = \frac{G_F^2 m^5}{192\pi^3} F\left(\frac{m_e^2}{m_\mu^2}\right) \quad (1.1)$$

放射補正

式 (1.1) の最低次の崩壊幅 Γ には、一般的な崩壊の途中で光子の放出や、W ボソンによる影響を補正を加える必要がある。補正した崩壊幅 Γ は以下のものである。

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

ここで、

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

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

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

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

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

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

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、

$$\tau_\mu = 2.197 \times 10^{-6} (sec) \quad (1.3)$$

である。

1.4 μ 粒子の崩壊と寿命

1.4.1 μ^+ 粒子の崩壊

μ^+ は以下のように崩壊する。

$$\mu^+ \rightarrow e^+ + \nu_e + \bar{\nu}_\mu$$

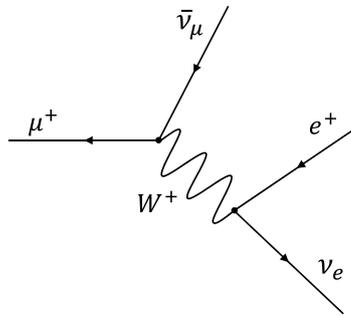


図 1.2 μ^+ 粒子の崩壊のファインマン図

μ^+ は物質中で相互作用を起こさないので、真空中と同様に静止、崩壊する。

1.4.2 μ^- 粒子の崩壊

μ^- は以下のように崩壊する。

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

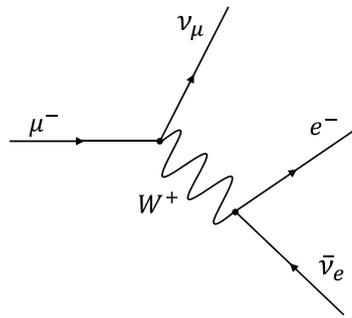


図 1.3 μ^- 粒子の崩壊のファインマン図

真空中の μ^- の振る舞いや平均寿命は μ^+ と等しいが、物質中の μ^- の振る舞いや平均寿命は μ^+ と異なる。それは μ^- が物質中の原子核と相互作用を起こすからである。入射した μ^- 粒子は物質中でミューオン原子を形成する。ミューオン原子は異種原子の一種で、原子中の e^- が μ^- に置き換わったものである。 μ^- (105.66MeV) は電子 (0.51MeV) よりも重いので電子よりも内側の独自の軌道を周回する。

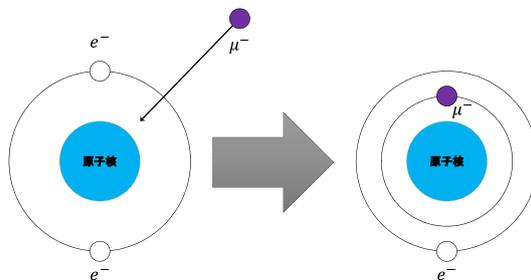


図 1.4 ミューオン原子の模式図

原子核に束縛された μ^- は、以下のどちらかの過程をたどる。

1. 束縛軌道上での崩壊 (Decay In Orbit(DIO))

真空中と同様の過程で崩壊する。平均寿命も真空中と同様である。

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

2. 原子核捕縛

原子核との間の弱い相互作用により、原子核内の陽子が μ^- を吸収し中性子に変じ、ニュートリノを放出する。この結果、 μ^- の寿命が見かけ上縮まったことになる。この様子は以下の式で表される。

$$\mu^- + (A, Z) \rightarrow \nu_\mu + (A, Z - 1)$$

原子核レベルでは、

$$\mu^- + \text{“}p\text{“} \rightarrow \text{“}n\text{“} + \nu_\mu$$

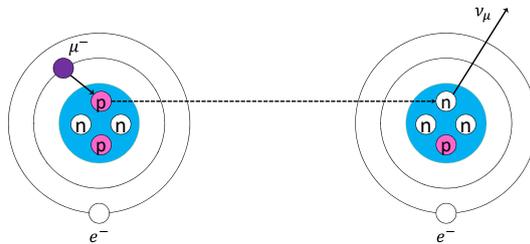


図 1.5 μ 粒子の原子核捕縛

この原子核捕縛が起こる割合 (Capture rate) は原子番号 Z に比例して大きくなるため、金属などの物質中に入射した μ^- は原子核捕縛を起こしやすい。物質中の μ^- は、この 2 通りの崩壊過程のどちらかをたどる。したがって、物質中の μ^- の寿命はこれらの過程の結果として決定される。

1.4.3 物質中の μ^- 粒子の寿命

原子に束縛された軌道上での μ^- の崩壊の確率 P_{decay} と、 μ^- の原子核捕縛の確率 $P_{capture}$ は以下のように表される。

$$P_{\mu^-total} = P_{capture} + P_{decay} \quad (1.4)$$

Capture rate を Λ_{cap} とすると、上の式はこのように書き換えられる。

$$\begin{aligned} \exp\left(-\frac{t}{\tau_{\mu^-}}\right) &= \Lambda_{cap} t \cdot \exp\left(-\frac{t}{\tau_{\mu^+}}\right) \\ &= \exp\left(-\left(\Lambda_{cap} + \frac{1}{\tau_{\mu^+}}\right)t\right) \end{aligned} \quad (1.5)$$

すなわち、

$$\frac{1}{\tau_{\mu^-}} = \Lambda_{cap} + \frac{1}{\tau_{\mu^+}} \quad (1.6)$$

ここで、 τ_{μ^-} は全ての μ^- の平均寿命を表しており、 τ_{μ^+} は真空中の μ 粒子の平均寿命と同値である。式 (1.6) から分かるように、Capture rate が大きいほど μ^- の寿命は短くなる。

TABLE III. Compendium of total muon capture results for light nuclei. (Z_{eff} is taken from Ref. 77. When underlined it is an estimate.)

Z (Z_{eff})	Element	Mean life (ns)	Total capture rate (s^{-1})	Huff factor	Refs.
Positive muon		2197.03 \pm 0.04			4,5
		2194.903 \pm 0.066			34
1 (1.0)	$^1\text{H}^b$		420 \pm 20	1.00	35
	$^1\text{H}^b$		420 \pm 60	1.00	36
2 (1.98)	^3He		470 \pm 29		37
			2170 + 170 (−430)	1.00	38
	^4He		2140 \pm 200		39
			336 \pm 75		37
		375 + 30 (−300)		60	
		364 \pm 46		1	
3 (2.94)	^6Li	2173 \pm 5	6100 \pm 1400		6
		2175.3 \pm 0.4	4680 \pm 120	1.00	a
3	^7Li	2177.0 \pm 2.0	4180 \pm 450		1
		2194 \pm 4	1800 \pm 1100		6
		2186.8 \pm 0.4	2260 \pm 120		a
4 (3.89)	Be	2188.3 \pm 2.0	1810 \pm 440		40
		2140 \pm 20	18 \pm 10×10^3	1.00	1
		2156 \pm 10	10 \pm 2×10^3		7
		2169.0 \pm 1.0	5.9 \pm 0.2×10^3		a
5 (4.81)	^{10}B	2162.1 \pm 2.0	7.4 \pm 0.5×10^3		1
		2082 \pm 6	26.5 \pm 1.5×10^3	1.00	a
		2070.7 \pm 3.0	27.8 \pm 0.7×10^3		1
		2102 \pm 6	21.8 \pm 1.6×10^3		a
6 (5.72)	C	2096.1 \pm 3.0	21.9 \pm 0.7×10^3		40
		2020 \pm 20	44 \pm 10×10^3	1.00	53
			36 \pm 4×10^3		41
		2043 \pm 3	37.3 \pm 1.1×10^3		42
		2041 \pm 5	36.1 \pm 1.0×10^3		43
		2040 \pm 30	37 \pm 7×10^3		1
		2025 \pm 4	39.7 \pm 1.3×10^3		27
		2035 \pm 8	36.5 \pm 2.0×10^3		44
		2060 \pm 30	30.3 \pm 7×10^3		7
		2030.0 \pm 1.6	37.6 \pm 0.4×10^3		73
		2040 \pm 10	35.2 \pm 2.0×10^3		29
		2029 \pm 3	37.7 \pm 0.7×10^3		a
		2026.3 \pm 1.5	38.8 \pm 0.5×10^3		29
		2045 \pm 2	33.8 \pm 0.4×10^3		a
2029.1 \pm 3.0	37.6 \pm 0.7×10^3		40		
7 (6.61)	N	1860 \pm 20	86 \pm 11×10^3	1.00	1
		1927 \pm 13	65 \pm 4×10^3		7
		1940.5 \pm 2.8	60.2 \pm 0.8×10^3		29
		1910 \pm 3	68.4 \pm 0.8×10^3		a
		1906.8 \pm 3.0	69.3 \pm 0.8×10^3		40
		1640 \pm 30	159 \pm 14×10^3	0.998	1
		1812 \pm 12	98 \pm 3×10^3		44
8 (7.49)	O	1810 \pm 20	98 \pm 5×10^3		73
		1832 \pm 29	95 \pm 8×10^3		a
		1795.4 \pm 2.0	102.6 \pm 0.6×10^3		a
		1844.0 \pm 4.5	88.0 \pm 1.5×10^3		a
		1420 \pm 40	254 \pm 22×10^3	0.998	40
		1450 \pm 20	235 \pm 10×10^3		45
		1458 \pm 13	231 \pm 6×10^3		27
1462.7 \pm 5.0	229 \pm 1×10^3		a		
		(these F data show the lifetime for the lower hf state; see Sec. IV D).			

^aDenotes the results of this experiment.

^bFor hydrogen the capture rate depends on mesomolecular effects (Ref. 25). Thus we have given the most recent articles only.

図 1.6 Compendium of total muon capture results for light nuclei

本実験で用いるプラスチックシンチレーターはポリスチレン ($[\text{C}_6\text{H}_5\text{CHCH}_2]_n$) 製で、H よりも C の電荷が 6 倍大きく、 μ^- に関係していると考えられることから、シンチレーターで観測できる μ^- の寿命は 2020 ± 20 ns(測定値) に近い値を取ると考えられる。これより、電荷を区別しない μ^- 粒子と、電荷を区別した μ^- 粒子の寿命を別々に測定することを試みる。

第 2 章

測定原理とセットアップ

2.1 全体の流れ

本実験で用いる装置の概略図を以下に示す。

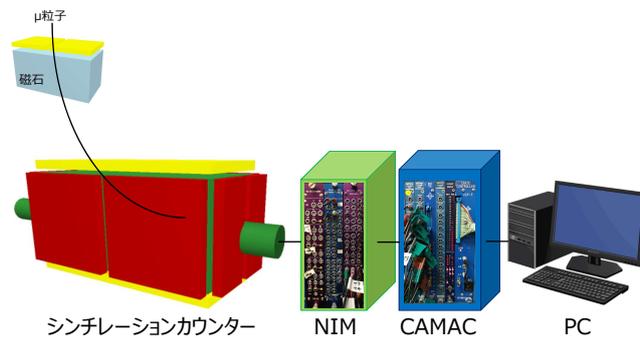


図 2.1 セットアップの概略図

まず上部に設置された磁石にて μ 粒子の電荷の区別を行う。荷電粒子（今回は μ 粒子）がシンチレーションカウンターに入射すると、シンチレーターがシンチレーション光を発生し、光電子増倍管で電子を発生・増幅させ、アナログ信号として出力する。出力された波形は NIM モジュールを経て波形形成され、CAMAC モジュール内でデジタル信号化される。それを PC を用いて解析する。

2.2 シンチレーションカウンター

シンチレーションカウンターとは、シンチレーターとライトガイド、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。図 2.2.3 は、本実験で用いたものと同じ型の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。

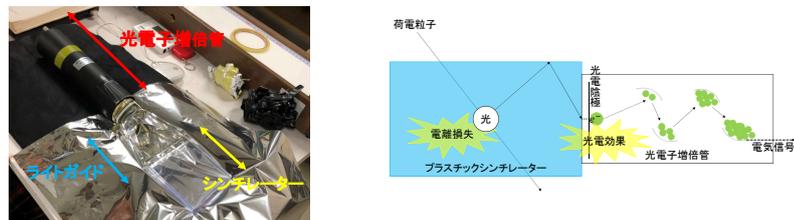


図 2.2 左図：シンチレーションカウンター 右図：シンチレーションカウンター模式図

2.2.1 シンチレーター

シンチレーター (Scintillator) とは、荷電粒子がシンチレーターを通過する際に、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る。本実験では有機シンチレーターの一種であるプラスチックシンチレーターを使用する。シンチレーターの光の強度が弱いため、検出には光信号を増倍する必要がある。そのために、次に説明する光電子増倍管が広く用いられており、直接取り付けられるか、ライトガイドを通して取り付けられる。ライトガイドとは、シンチレーターで発生した光を効率よく光電子増倍管に送るための、着色のない優れた分光透過性を持ったガラス光ファイバーを多数束ねたものである。

- 有機シンチレーター

蛍光の減衰時間（発光時間）が n 秒程度と短く速い反応を示すので、計数率の高い測定や早い信号を要する測定に適している。

- 無機シンチレーター

検出体には NaI や Cs など I を用いる。発光時間が μ 秒程度と長く反応は遅いが、発光量は有機シンチレーターよりも大きいため低エネルギーでの測定に適している。

2.2.2 光電子増倍管

光電子増倍管 (PMT : PhotoMultiplier Tube) とは、ライトガイドを通して送られてきた微弱な光を増幅させ検出可能な電気信号に変える装置である。PMT は光電陰極と複数のダイノード、陽極 (アノード) から成る。シンチレーターで発生した光子は、光電陰極に入射すると光電効果を起こし、電子を発生させる。生じた電子は強い磁場によって加速され、ダイノードに入射するとダイノード内の電子を励起し、二次電子を発生させる。発生した二次電子は順次ダイノードに入射し、電子の数を増幅させていく。増幅された電子は陽極に集まり、電気信号として計測される。

2.2.3 シンチレーションカウンターの配置

本実験では、中央に位置するメインシンチレーターと、それを囲むように配置した 10 個のシンチレーションカウンター (上部 : T1、下部 : T2、側部 : V1~V8)、磁石の上部横並びに T3 と T4 のシンチレーションカウンター、合計 13 個を用いた。メインシンチレーターには直接 2 つの光電子増倍管 (S1, S2) を取り付けている。以下に各シンチレーションカウンターの配置と仕様、サイズを示す。

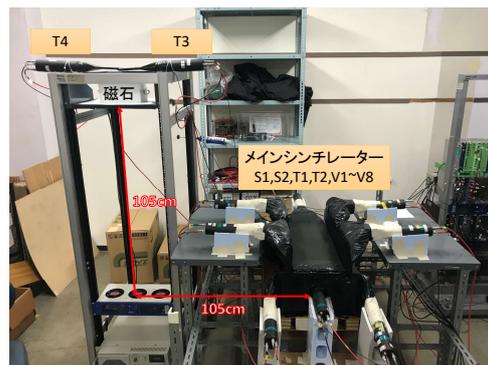


図 2.3 実際のシンチレーターの配置

上の表 2.2 は、本実験で用いた各シンチレーションカウンターの光電子増倍管 (PTM) の型番、それぞれにかけた電圧 (HV:High Voltage) とそのチャンネル、Discriminator でカットした閾値 (Threshold)、各シンチレーターのペDESTAL を表にしたものである。

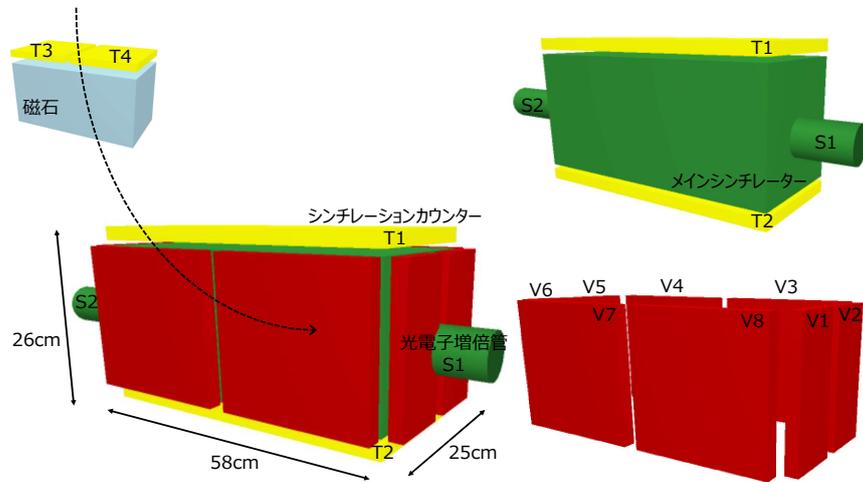


図 2.4 シンチレーションカウンターの模式図

表 2.1 シンチレーターのサイズ

	名称	横幅 (mm)	縦幅 (mm)	奥行き (mm)	個数 (個)
	(S1,S2)	580	260	250	2
プラスチック	T1,T2	582	10	250	2
シンチレーター	T3,T4	100	10	100	2
	V1,V2,V5,V6	131	250	10	4
	V3,V4,V7,V8	280	261	10	4
ライトガイド					12
光電子増倍管					14

2.2.4 トリガーカウンター

トリガーとは、幾つかのカウンターを用いて信号が来たと判断することであり、トリガーに用いるシンチレーションカウンターのことをトリガーカウンターと呼ぶ。宇宙線はあらゆる方向から、絶え間なく降り注いでいる。本実験で用いる μ 粒子も同様に、あらゆる方向からシンチレーションカウンターに絶え間なく入射している。一方向から入射した粒子のみを選定するために、複数のシンチレーションカウンターが同時に反応しているイベントを採用する。本実験ではメインシンチレーター上部に設置しているシン

表 2.2 各シンチレーターの仕様

カウンター	PMT 型番	HV(V)	HV(ch)	閾値 (mV)	ペDESTAL (ADC(μ))	ペDESTAL (ADC(e))
S1	H161	2050	1	160	109	58
S2	H161	2050	2	300	98	63
T1	H161	2250	3	100	89	56
T2	H195	1770	4	100	85	60
T3	H7195	1800	13	200	20	
T4	H3983	1800	14	100	97	
V1	H161	1970	5			67
V2	H7195UV	1660	6			63
V3	H161	1870	7			57
V4	H161	2440	8			63
V5	H161	2400	9			65
V6	H161	2030	10			64
V7	H7195	2210	11			66
V8	H161	2025	12			80

チレーションカウンター T1 とメインシンチレーターに直接取り付けられた光電子増倍管 S1 をトリガーカウンターとしている。この T1 と S1 が同時に信号を発しているとき (T1 と S1 のコインシデンスが ON のとき) に μ 粒子が上部からメインシンチレーター内に入射したとしている。すなわち、T1 と S1 のコインシデンスしたものをトリガー信号としている。

2.2.5 識別

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場合と、 μ 粒子がシンチレーター内で静止した後に崩壊した場合は、観測される信号に違いが生じる。

- μ 粒子がシンチレーターを通過する場合
観測される信号には、以下のように 1 つのパルスがみられる。
- μ 粒子がシンチレーター内で静止した後、崩壊する場合
観測される信号には、以下のように 2 つのパルスがみられる。1 つ目は入射した μ

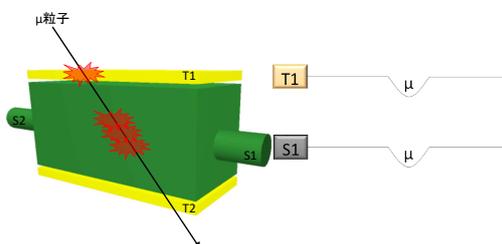


図 2.5 シンチレーター内の様子と入射 μ 粒子の信号

粒子によるもの、2つ目は μ 粒子の崩壊によって放出された電子のによるものである。前者を μ 粒子の波形、後者を電子の波形と呼ぶ。

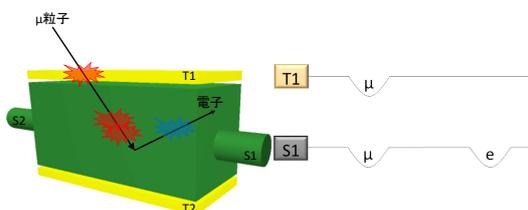


図 2.6 シンチレーター内の様子と入射 μ 粒子と e^- の信号

2.3 NIM

2.3.1 NIM 規格

NIM(Nuclear Instrument Modules)とは、米国原子力委員会(AEC)(米国エネルギー省の前身)において1966年に制定された「放射線測定モジュール標準規格 TID-20893」に準拠した標準規格である。この規格を適用したモジュールであれば、国内外、またはメーカーを問わず物理的及び電氣的に完全な互換性を有する。

この規格に準拠の回路は、NIM モジュールと呼ばれ、原子力関連、高エネルギー加速器用、宇宙線研究用、精密計測システムの構築など広く研究用途に活用されています。弊社 NIM モジュールもこの規格に準拠して製作しております。この規格は、信号レベル・信号線・コネクタ形状・電源およびケースサイズなどについての規定です。NIM モジュールは、NIM ビンと呼ばれる電源クレートに装着して使用されます。電源は、通常 $\pm 24V$ 、 $\pm 12V$ 、 $\pm 6V$ を供給します。

2.3.2 NIM モジュール

NIM モジュールとは、NIM 規格に準拠した回路である。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。

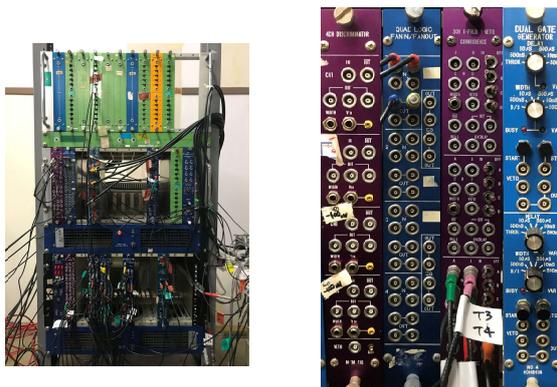


図 2.7 左図：本実験で使用している NIM モジュール 右図：主な NIM モジュール

- Discriminator

ある一定の電圧 (threshold:しきい値) を超えるアナログ信号が入力されたとき、

その電圧を超えた時刻を起点としてデジタルのロジック信号を出力する装置。

- Delay
入力された信号を任意の時間遅らせて出力する装置。
- Coincidence (=AND 回路)
複数の信号が重なったとき、重なった時刻を起点として信号を出力する。出力するパルスの時間幅を任意に調節できる。
- G.G(Gate and delay Generator)
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 Time を任意に調節できる。設定できる。
- Fan-in Fan-out (=OR 回路)
入力信号が一つでも入力されたときロジック信号を出力する。
- Scaler
入力端子に入力されるロジック信号を数える装置。

2.4 CAMAC

2.4.1 CAMAC 規格

CAMAC (Computer Aided Measurement And Control) とは、ハーヴェル原子力研究所を中心にして使用実績のあった規格等が拡張発展され、ヨーロッパ統一規格として誕生したエレクトロニクス規格である。計算機器周辺でのデジタル化された情報の処理を各機能毎にモジュール化して行うことができる。実験装置等からの情報はモジュールの中で処理されると、クレートと呼ばれるプラグインを収納する箱の裏側の配線と結線される。これらの作業はすべてクレートの右端に挿入される CC (Crete Controller) が制御する。この CC は PC からの指示により動作する。

PC から CC への命令には以下のアドレス (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 : クレート番号 (本実験で用いるクレートは 1 台なので C=1)
- N : ステーション番号 (クレートの左端から順に数える)
- A : モジュール内のサブアドレス
- F : 動作 (Function)

- ファンクション

今回主に使用したファンクションを記載する。

- F(0) : Read Data
- F(2) : Read and Clear Data
- F(8) : Test LAM
- F(9) : Clear Data
- F(10) : Clear LAM
- F(24) : Disable LAM
- F(26) : Enable LAM
- F(27) : Test status

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になったことを PC 側に知らせる信号のことである。図 2.8 は LAM チェックをする行程を示している。

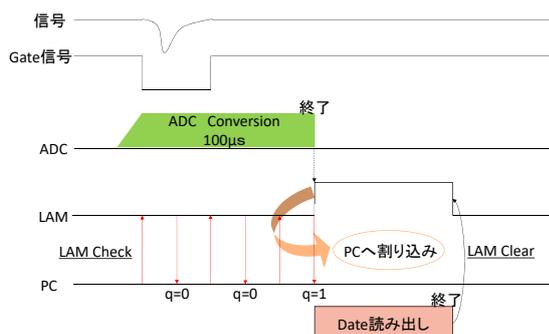


図 2.8 LAM をチェックする行程

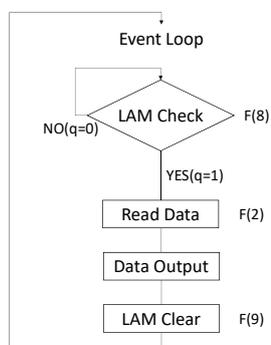


図 2.9 LAM をチェックするフローチャート

2.4.2 CAMAC モジュール



図 2.10 本実験で使用している CAMAC モジュール

- CC(Crate Controller)
 - ADC や TC など制御して、データ収集を PC などで行うためのモジュール。
- 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。
 - V モード (ピークホール型 ADC)
 - 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。
 - Q モード (荷電積分型 ADC)
 - 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。

今回実験で扱うプラスチックシンチレータは時間応答が速く、ピークの電圧値を取ることが難しいので Q モードを使用している。無機シンチレータは遅いので V モードでもピークの高さをとらえることができる。Q モードで測定される電荷量 Q は、信号の電流を i とすると

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

と表せ、以下の図の色部分の面積に相当する。一つのパルスに一つの値（これを

チャンネルと呼ぶ) を対応させる。Gate 内に波形が入るようにするために信号側に Delay を入れる。

本実験では ADC を 2 台使用する。以降 μ 粒子のエネルギー測定に用いる

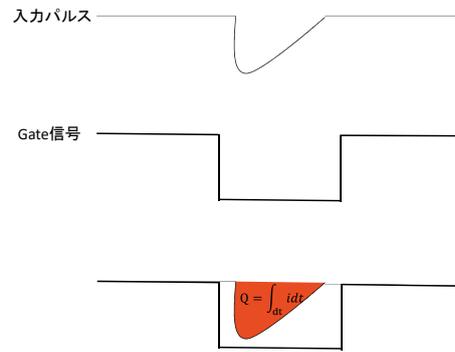


図 2.11 Gate 信号と入力パルス

ADC(μ)、 e^- のエネルギー測定に用いる ADC(e) と呼称する。

- TDC(Time to Digital Converter)

時間というアナログ量をデジタル信号に変換する装置のこと。Start 信号が入力されてから、Stop 信号が入力されるまで、一定の時間間隔でカウントを刻み、そのカウント数で時間間隔を表す。

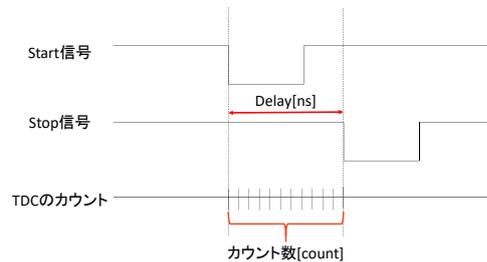


図 2.12 TDC の仕組み

2.4.3 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。まず PC から ADC にテスト LAM の信号を送り、LAM OK の Q 信号 ($q \neq 0$) が返ってきたら ADC・TDC のデータを読み込む。データを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 の信号を送り、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を参照。

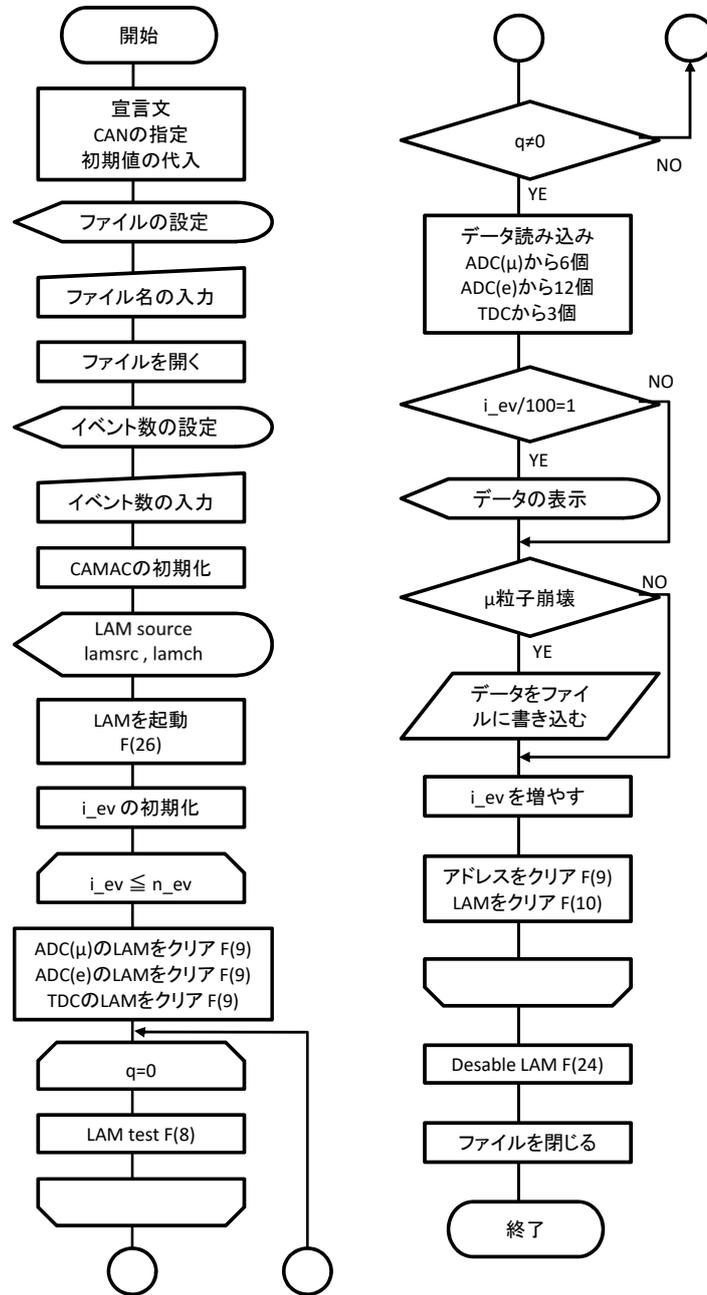


図 2.13 データ収集プログラムのフローチャート

2.5 エレクトロニクスのセットアップ

本実験では ADC 2 台、TDC 1 台を使用した。ADC(μ) は μ 粒子のエネルギー測定、ADC(e) は μ 粒子が静止崩壊後に放出する電子のエネルギー測定、TDC は μ 粒子の寿命を測定する目的でセットした。図 2.14 は測定に用いるセットアップ全体のエレクトロニクスである。

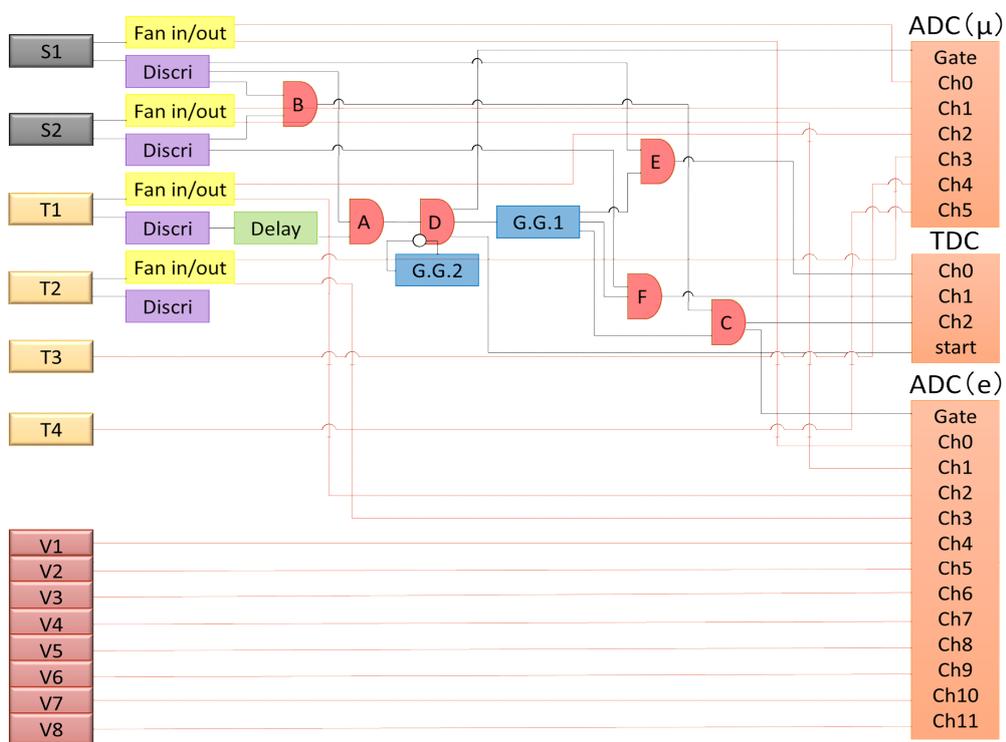


図 2.14 エレクトロニクスの全体図

2.5.1 入射した μ 粒子のエネルギー測定 ($ADC(\mu)$ のセットアップ)

μ 粒子がメインシンチレーター内に入射した時の信号を ADC で測定するために、 $ADC(\mu)$ の Gate に入力するアナログ信号を作成する。以下に回路図 (図 2.15) とパルスのロジック図 (図 2.16) を示す。

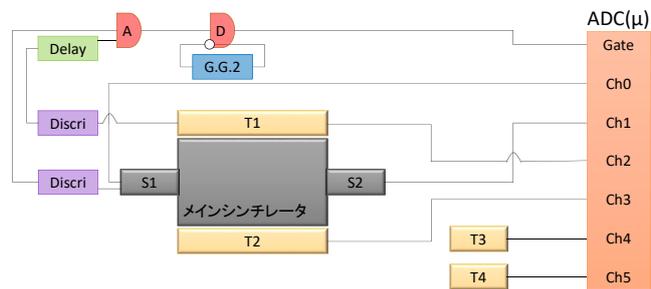


図 2.15 $ADC(\mu)$ の回路図

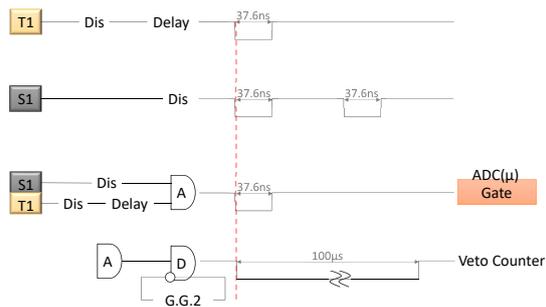


図 2.16 $ADC(\mu)$ のロジック図

Coincidence D は Veto カウンターの役割を果たしており、この Veto カウンターが開いている約 10ms の間は次の μ 粒子が入射した信号を受け付けない。これは、ADC のデータ取集中に次の信号が発生しないための対策である。

2.5.2 入射した μ 粒子が静止後、崩壊して生成された電子のエネルギー測定 (ADC(e) のセットアップ)

μ 粒子がメインシンチレーター内に入射し静止後、崩壊して放出した電子の信号を ADC で測定するために、ADC(e) の Gate に入力するアナログ信号を作成する。以下に回路図 (図 2.17) とパルスのロジック図 (図 2.18) を示す。

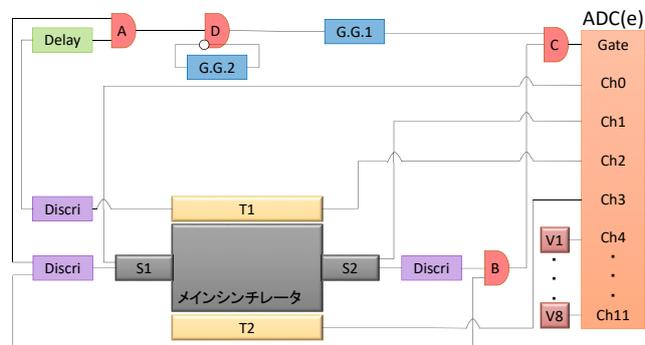


図 2.17 ADC(e) の回路図

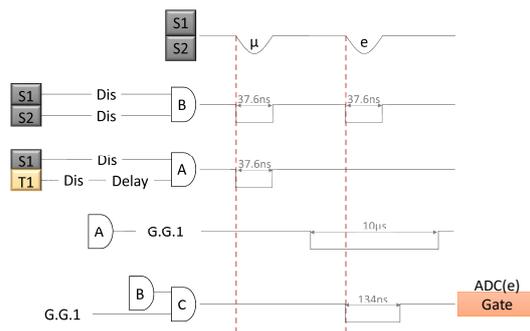


図 2.18 ADC(e) のロジック図

2.5.3 μ 粒子の寿命測定 (TDC のセットアップ)

μ 粒子の寿命を測定するために、TDC の Start には μ 粒子がメインシンチレーター内に入射した時の信号を、TDC の Stop には μ 粒子がメインシンチレーター内に入射し静止後、崩壊して電子を放出した時の信号を入力する。これは Start 信号には ADC(μ) の Gate 信号と同様の信号を、Stop 信号には ADC(e) の Gate 信号と同様の信号を入力している。以下に回路図 (図 2.19) とパルスのロジック図 (図 2.20) を示す。

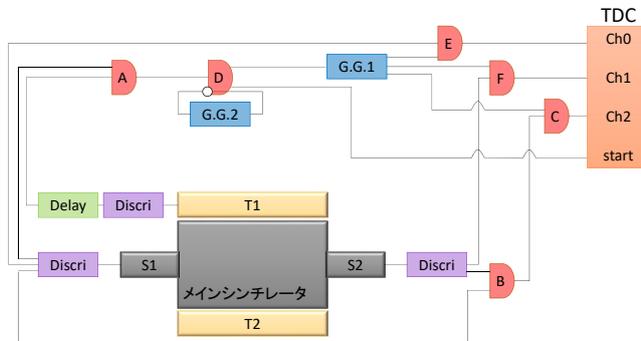


図 2.19 TDC の回路図

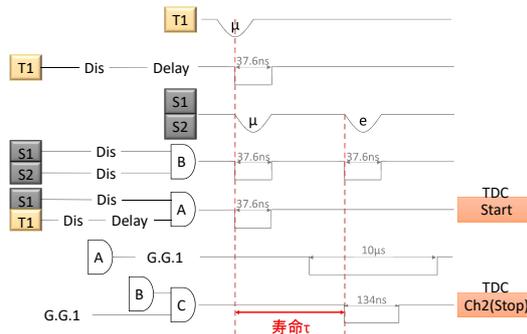


図 2.20 TDC のロジック図

第3章

μ 粒子の寿命測定

3.1 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 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 とし（これを崩壊定数という）、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。

$$dN = -\lambda N(t)dt \quad (3.1)$$

式 (3.1) を積分すると、

$$\begin{aligned} \int_{N_0}^N \frac{dN}{N} &= -\lambda \int_{t_0}^t dt \\ [\ln N]_{N_0}^N &= -\lambda [t]_{t_0}^t \\ \ln \frac{N}{N_0} &= -\lambda(t - t_0) \end{aligned} \quad (3.2)$$

$t_0=0$ の時、式 (3.2) は、

$$N(t) = N_0 e^{-\lambda t} \quad (3.3)$$

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 $t \rightarrow t + dt$ の間の微小時間 dt で崩壊する粒子の個数は式 (3.1) で表

されるので、 N_0 個すべての粒子の生存時間を足し合わせたものは次式となる。

$$\begin{aligned} L &= \int_0^{\infty} tN(t)\lambda dt \\ &= \int_0^{\infty} tN_0e^{-\lambda t}\lambda dt \\ &= \left[-\frac{1}{\lambda}N_0e^{-\lambda t} \right]_0^{\infty} \\ &= \frac{N_0}{\lambda} \end{aligned}$$

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、

$$\begin{aligned} \tau &= \frac{L}{N_0} = \frac{1}{\lambda} \\ \lambda &= \frac{1}{\tau} \end{aligned} \tag{3.4}$$

である。

これを式 (3.3) に代入すると、

$$N(t) = N_0e^{-\frac{t}{\tau}} \tag{3.5}$$

が得られる。

式 (3.5) を図 3.1 に示す。

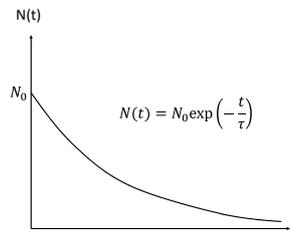


図 3.1 放射性崩壊の図

また、式 (3.5) 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。

$$\frac{dN}{dt} = -\frac{N_0}{\tau} \exp\left(-\frac{t}{\tau}\right) \quad (3.6)$$

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_{decay} とすると、 $dN_{decay} = -dN$ なので、式 (3.6) は

$$\frac{dN_{decay}}{dt} = \frac{N_0}{\tau} \exp\left(-\frac{t}{\tau}\right) \quad (3.7)$$

となる。

μ 粒子がシンチレーターに入射、内部で静止後、崩壊して電子を放出するまでの時間を TDC で測定することで得られる TDC 分布が式 (3.7) に従っている。実験で得た TDC 分布を実験式

$$y(t) = p_0 \exp\left(-\frac{t}{p_1}\right) \quad (3.8)$$

で Fit することにより平均寿命 τ が算出できる。

図 3.2 は式 (3.7) 及び TDC 分布の概形である。

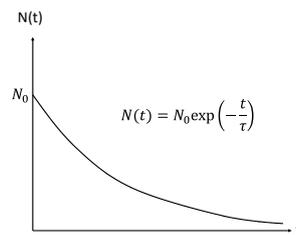


図 3.2 TDC 分布の概形図

3.1.1 ローレンツ力

磁石を置いたことにより、メインシンチレーター内に μ^+ 、 μ^- 粒子のどちらが入射されているのかを考える。

ローレンツ力は以下の式である。

$$\mathbf{F} = q(\mathbf{v} \times \mathbf{B}) \quad (3.9)$$

ここで、 \mathbf{F} は粒子にかかる力、 q は粒子の電荷、 \mathbf{v} は入射粒子の速度、 \mathbf{B} は磁石が作る磁束密度である。本実験の簡略図 (3.3) は以下の通りである。

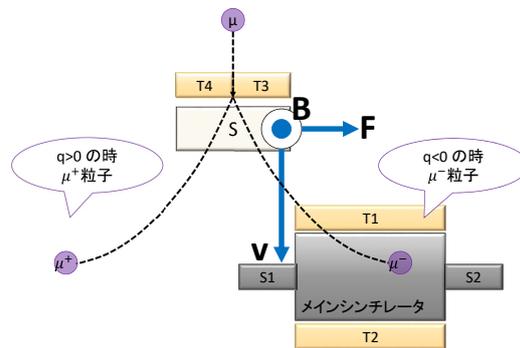


図 3.3 装置の簡略図

$q > 0$ のとき、力 F は上図において左向きにかかり、 $q < 0$ のとき、力 F は右向きにかかる。つまり、本実験においてメインシンチレーター内に入射しているのは μ^- 粒子である、と分かる。

3.2 イベントの選定

3.2.1 μ 粒子の寿命測定 (電荷の区別なし)

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。実験で取得した全 507,761 イベントのうち、次の条件に合うイベントを選定する。

0. 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

トリガーカウンターである T1 と S1 の Coincidence 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内で崩壊

メインシンチレーター内で電子が放出される。つまり ADC(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 のヒストグラムを以下 (図 3.4) に示す。

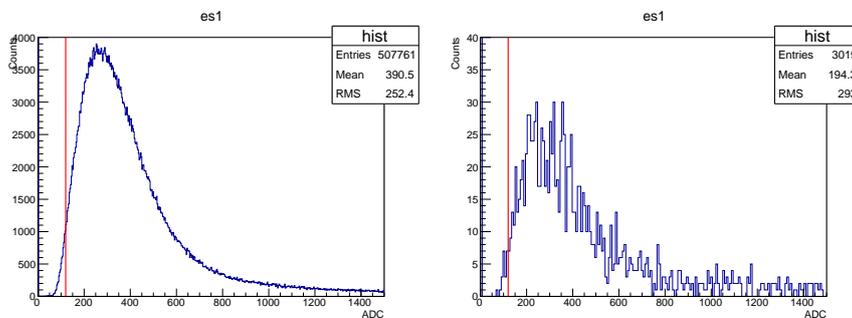


図 3.4 es1 の ADC 分布 (左図: 取得した全イベント、右図: 「磁石を通過」の選定後)

上の S1 の ADC 分布 (図 3.4) から、 $120 < es1$ の時に μ 粒子がメインシンチレーター内で崩壊したとみなす。

TDC がタイムアウトしていない。TDC がタイムアウトした場合、戻り値は 4000 以上の値を示す。したがって具体的な選定条件は TDC カウント数 < 4000 である。

以上 2 つの選定条件に合致したものは全部で 458,336 イベントである。

3.2.2 μ^- 粒子の寿命測定 (電荷の区別あり)

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「磁石を通過し、区別した μ^- 粒子のうち、メインシンチレーター内で崩壊したイベント」である。実験で取得した全 507,761 イベントのうち、次の条件に合うイベントを選定する。

電荷の区別ありの μ^- 粒子の測定に用いた装置の模式図は以下のようなものである。

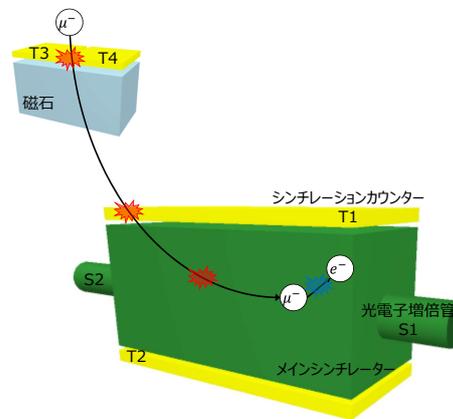


図 3.5 μ^- 粒子 (電荷の区別あり) の寿命測定の様式図

0. 選定条件

条件の選出方法

1. μ 粒子が磁石を通過

磁石を通過し、 μ 粒子の電荷の区別がされている。つまり磁石の上部に設置されたプラスチックシンチレーター T3,T4 において、 $ADC(\mu)$ で取得した T3 か T4 どちらかの値 ($mut3, mut4$) 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。

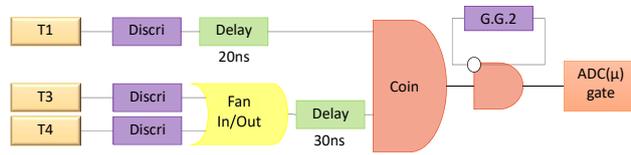


図 3.6 ADC(μ) に入力した Gate 信号のロジック図

磁石を通過し、その後メインシンチレーターに入射した μ 粒子の ADC 分布を選出に用いた。このとき、T3 または T4 と、T1 をトリガーカウンターとし、T3 と T4 の Fan in/out と、T1 をコインシデンスしたものを ADC(μ) の Gate 信号 (上図 3.6) に入力した。より多くのデータを集めるために、 μ 粒子のメインシンチレーター内での崩壊に関わらずデータを収集するように、データ収集プログラムを変更した。T3 の値 (mut3) と T4 の値 (mut4) それぞれのヒストグラムを以下 (図 3.7) に示す。

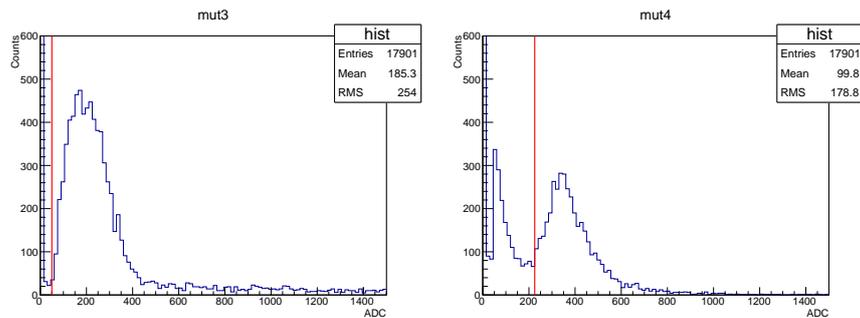


図 3.7 mut3,mut4 の ADC 分布

上の T3,T4 の ADC 分布 (図 3.7) から、 $50 < \text{mut3}$, $225 < \text{mut4}$ の時に μ 粒子が T3,T4 を通過したとみなす。

2. μ^- 粒子がメインシンチレーターに入射
電荷の区別がないときと同じ条件の選出方法である。
3. μ^- 粒子がメインシンチレーター内で崩壊

電荷の区別がないときと同じ条件の選出方法である。

以上 3 つの選定条件に合致したものは全部で 1,253 イベントである。

3.3 TDC の時間較正

前節で得られた 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との対応を調べる。

Clock Generator で作成した 1 つ目のパルスをも start 信号、任意の時間だけ Delay させた 2 つ目のパルスをも stop 信号として、その間の時間を TDC で測定する。

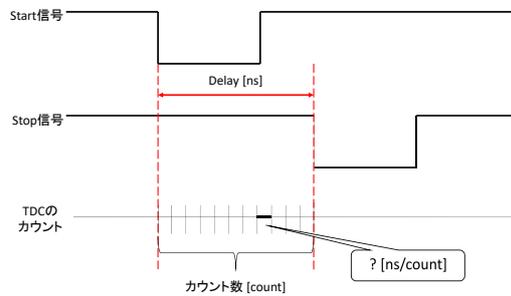


図 3.8 TDC 時間較正に用いるパルス

パルスを Delay させる時間を変化させ、それに応じた TDC のカウントの変化を見ることで、TDC の 1 カウント当たりの時間を算出することができる。TDC の時間較正に用いるロジックを以下の図に示す。

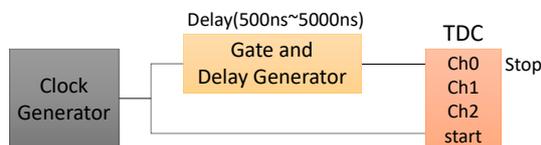


図 3.9 TDC 時間較正に用いるロジック

Gate and Delay Generator (G.G.) で Stop に入力する 100Hz のパルスの Delay time を 500ns~5000ns まで 500ns ずつ変化させていく。Delay time の読み取り誤差は 10ns

としている。

実際に測定した Delay time と TDC カウント数の関係を以下に示す。このカウント数は 1000 イベントのデータの平均値である。この結果を横軸が Delay time [ns]、縦軸が TDC カウント数 [count] としてグラフにしたものが図 (3.10) である。右図は、「Pull = TDC カウント数 - Fit 関数の値」を算出しプロットしたものである。

表 3.1 Delay time と TDC カウント数

Delay time[ns]	500 ± 10	1000 ± 10	1500 ± 10	2000 ± 10	2500 ± 10
TDC[count]	99.590 ± 0.013	199.514 ± 0.018	299.653 ± 0.012	399.543 ± 0.016	499.703 ± 0.013
	3000 ± 10	3500 ± 10	4000 ± 10	4500 ± 10	5000 ± 10
	599.457 ± 0.017	699.551 ± 0.014	799.500 ± 0.016	899.798 ± 0.014	999.486 ± 0.015

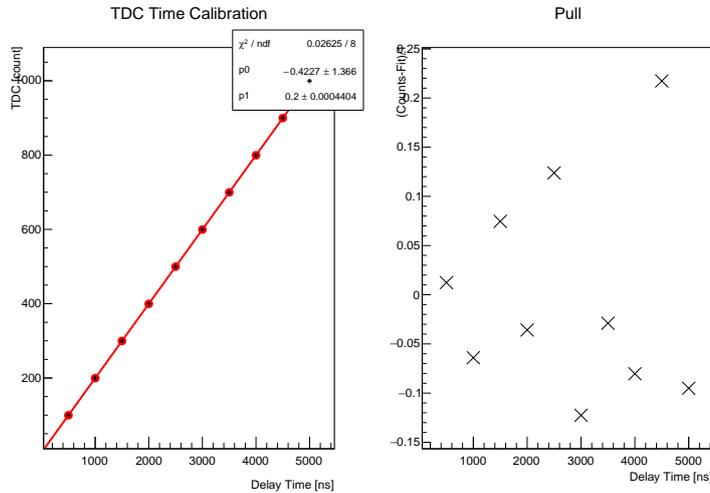


図 3.10 TDC 時間較正

図 (3.10) から Delay time と TDC カウント数は比例関係にあることが分かる。線形近似 $y=ax+b$ (x,y は変数、 a,b は定数) のパラメータ a,b は、それぞれ図 (3.10) の統計ボックス内に表示された p_1, p_0 に対応する。よって a,b は

$$a = 0.2 \pm 0.0004404$$

$$b = -0.4227 \pm 1.366$$

となった。 $\frac{1}{\text{傾き } a}$ は TDC 1 カウント当たりの時間 $\alpha[\text{ns}]$ に相当する。これにより、

$$\alpha = \frac{1}{0.2} = 5.000 \pm 0.011[\text{ns/count}] \tag{3.10}$$

と求められる。

3.4 結果

3.4.1 μ 粒子の平均寿命 (電荷の区別なし)

イベントの選定を行った μ 粒子の TDC 分布を第 3.1 節で述べたように式 (3.8) で Fit する。

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

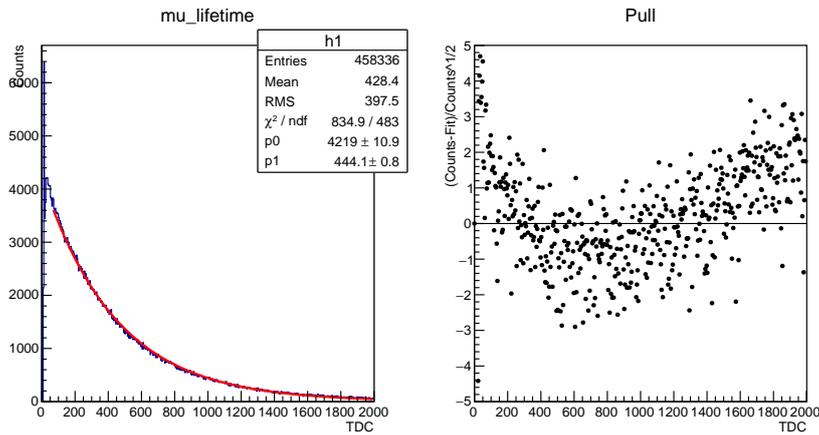


図 3.11 右図： μ 粒子の TDC 分布 (2019 年度)、左図：Pull

ここで左図 Pull とは、

$$Pull = \frac{\text{ヒストグラムの値 } y[\text{counts}] - \text{Fit 関数の値 } y_{Fit}}{\sqrt{\text{ヒストグラムの値 } y[\text{counts}]}}$$

を算出しプロットしたものである。

μ 粒子の平均寿命を表す TDC カウント数 $p_1[\text{count}]$ 、TDC 1 カウント当たりの時間 $\alpha[\text{ns}]$ を用いて μ 粒子の寿命 $\tau_\mu[\text{ns}]$ は次のように求められる。

$$\begin{aligned}\tau_\mu[\text{ns}] &= p_1[\text{count}] \times \alpha[\text{ns}/\text{count}] \\ &= 444.1 \times 5.000 \\ &= 2220.5 \text{ ns} \\ &= 2.2205 \text{ } \mu\text{s}\end{aligned}\tag{3.11}$$

また、平均寿命 τ_μ の誤差 $\delta\tau_\mu$ は間接誤差の誤差伝播を考慮して以下の式を用いる。

$$\begin{aligned}\frac{\delta\tau_\mu}{\tau_\mu} &= \sqrt{\left(\frac{\delta\alpha}{\alpha}\right)^2 + \left(\frac{\delta p_1}{p_1}\right)^2} \\ &= \sqrt{\left(\frac{0.011}{5.000}\right)^2 + \left(\frac{0.8}{444.1}\right)^2} \\ &= 0.028 \\ \therefore \delta\tau_\mu &= 2.2205 \times 0.028 = 0.0063 \text{ } \mu\text{s}\end{aligned}\tag{3.12}$$

ここで、 δp_1 は p_1 の誤差、 $\delta\alpha$ は α の誤差である。

測定で求められた p_1, α より、本実験で得られた結果から導き出される μ 粒子の寿命 τ_μ は、

$$\tau_\mu = 2.221 \pm 0.006 \text{ } \mu\text{s}$$

である。

3.4.2 μ^- 粒子の平均寿命 (電荷の区別あり)

前小節と同様に TDC 分布を Fit する。

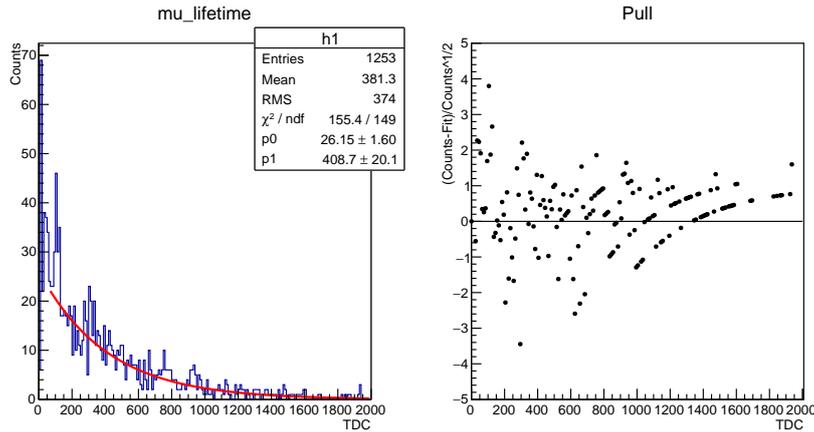


図 3.12 右図： μ^- 粒子の TDC 分布 (2019 年度)、左図：Pull

前小節と同様に

$$\begin{aligned}
 \tau_{\mu^-} [ns] &= p_1 [count] \times \alpha [ns/count] \\
 &= 408.7 \times 5.000 \\
 &= 20435 \text{ ns} \\
 &= 2.0435 \mu s
 \end{aligned}$$

平均寿命 τ_{μ^-} の誤差 $\delta\tau_{\mu^-}$ は

$$\begin{aligned}
 \frac{\delta\tau_{\mu^-}}{\tau_{\mu^-}} &= \sqrt{\left(\frac{\delta\alpha}{\alpha}\right)^2 + \left(\frac{\delta p_1}{p_1}\right)^2} \\
 &= \sqrt{\left(\frac{0.011}{5.000}\right)^2 + \left(\frac{20.1}{408.7}\right)^2} \\
 &= 0.049 \\
 \therefore \delta\tau_{\mu^-} &= 2.0435 \times 0.049 = 0.1006 \mu s
 \end{aligned}$$

本実験で得られた結果から導き出される μ^- 粒子の寿命 τ_{μ^-} は、

$$\tau_{\mu^-} = 2.04 \pm 0.10 \mu s$$

である。

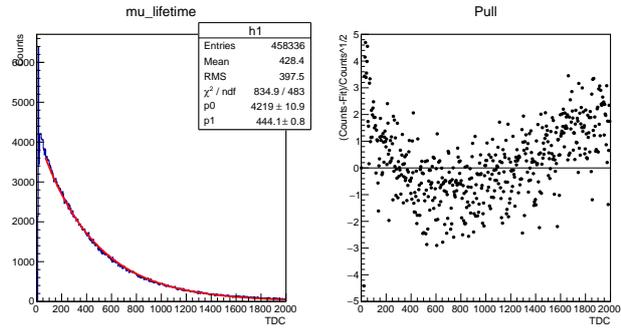
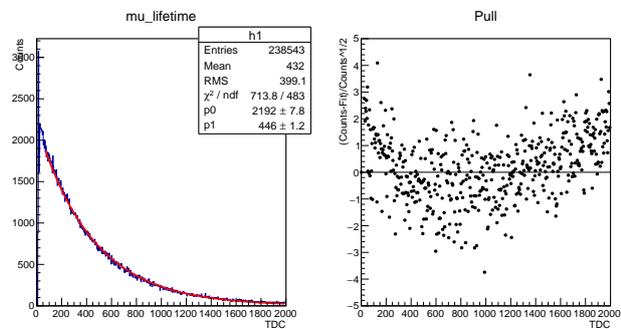
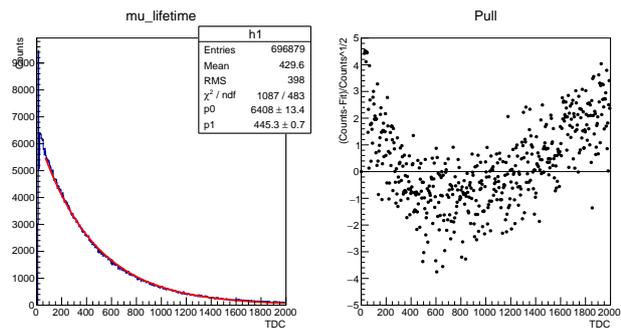
3.5 考察

3.5.1 μ 粒子の寿命解析

本実験で得られた μ 粒子の TDC 分布と平均寿命と昨年の先行実験によるものとの比較を行った。本実験では先行実験と同じセットアップ、データ収集方法で行った。本実験と先行実験とはイベントの選定条件が異なるが、先行実験で得られたデータを本実験での選定条件で選定し平均寿命を求めた。先行実験のみ (2018 年度) のデータによる μ 粒子の寿命と先行実験と本実験 (2018 年度 +2019 年度) のデータによるものを以下にまとめ、比較する。

表 3.2 μ 粒子の平均寿命

	測定時間 [h]	エントリー数	$\tau_{\mu} [\mu s]$
2019 年度	1388	458336	2.221 ± 0.006
2018 年度	336	238543	2.230 ± 0.008
2018 年度 +2019 年度	1674	696879	2.227 ± 0.006
文献値			2.19703 ± 0.00002

図 3.13 μ 粒子の TDC 分布 (2019 年度)図 3.14 μ 粒子の TDC 分布 (2018 年度)図 3.15 μ 粒子の TDC 分布 (2018~2019)

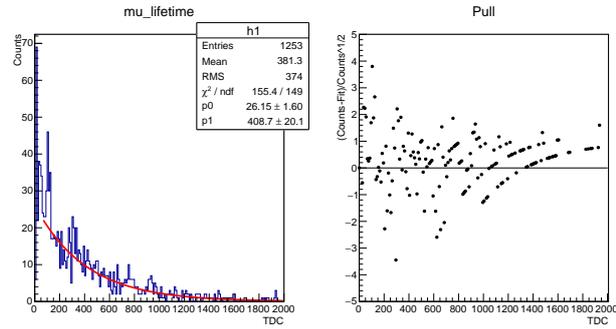
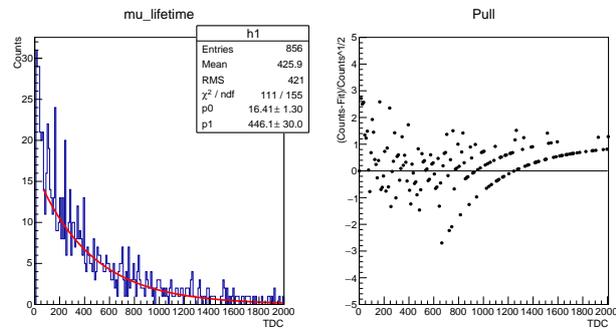
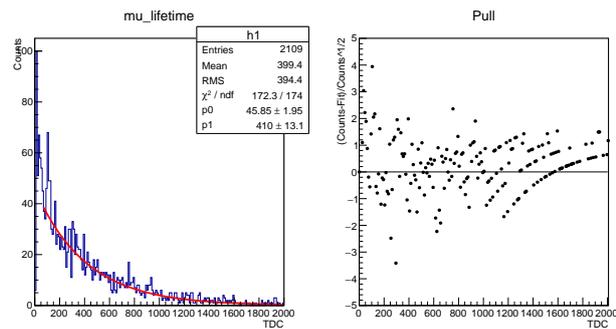
上図より TDC 分布とそれに Fit させた関数、Pull を比較する。Pull は TDC 分布に対する Fit がうまく行われているかを確認するために用いた。Pull が 0 であるとき、Fit が一致している。どのデータでも Pull が 1 に同じような関係がみられた。電荷の区別をしていないため、寿命の異なる μ 粒子と μ^- 粒子が混在していることを考慮した Fit 関数を用いることで Fit の精度を上げることができると思う。

3.5.2 μ^- 粒子の寿命解析

前小節と同様に、本実験で得られた μ^- 粒子の TDC 分布と平均寿命と昨年 of 先行実験によるものとの比較を行った。

表 3.3 μ^- 粒子の平均寿命

	測定時間 [h]	エントリー数	τ_{μ^-} [μs]
2019 年度	1388	1253	2.04 ± 0.10
2018 年度	336	856	2.23 ± 0.15
2018 年度 + 2019 年度	2109	1674	2.05 ± 0.07
文献値			2.02 ± 0.02

図 3.16 μ^- 粒子の TDC 分布 (2019 年度)図 3.17 μ^- 粒子の TDC 分布 (2018 年度)図 3.18 μ^- 粒子の TDC 分布 (2018~2019)

上図より TDC 分布とそれに Fit させた関数、Pull を比較する。Pull の TDC カウント数 600~2000 に見られる曲線は、ヒストグラムの値が 0,1,2,... と少なく、Fit 関数の形の影響をそのまま受けていると考える。Pull の関係を見るためにはより多くのデータを集めることが必要であると考えられる。

第4章

まとめ

以下の表に各イベント数と平均寿命をまとめる。

表 4.1 μ 粒子 (電荷の区別あり、なし) の平均寿命

	測定時間 [h]	取得した全イベント数	$\tau_{\mu}[\mu s]$ (イベント数)	$\tau_{\mu^-}[\mu s]$ (イベント数)
2019 年度	1,388	507,761	2.221 ± 0.006 (458,336)	2.04 ± 0.10 (1,253)
2018 年度	336	386,521	2.230 ± 0.008 (238,543)	2.23 ± 0.15 (856)
2018+2019	2109	894,282	2.227 ± 0.006 (696,879)	2.05 ± 0.07 (1,674)
文献値			2.19703 ± 0.000021	2.02 ± 0.02

μ^- 粒子の平均寿命 τ_{μ^-} は $2.04 \pm 0.10[\mu s]$ となった。この結果は文献値 2.02 ± 0.02 と誤差の範囲で一致した。

今後の課題として考えられることを以下に示す。

今回測定した、メインシンチレータに入射した μ^- 粒子は 1 時間に約 1 イベントと、とても少なかった。また本実験と先行実験とでイベント数の比較をすると、測定時間（データ収集プログラムを回し、データを取得していた時間）に対して取得したイベントが少なかった。データ取得の効率を高めるために、セットアップの検討が求められる。また、磁石の上部に設置されているシンチレーションカウンター T4 の鳴っているとみなされるイベントが少なく、これについても検討が必要であると考えられる。

謝辞

卒業研究を進めるにあたり、ご指導やご協力いただいた皆様に深く感謝いたします。お忙しい中熱心に、丁寧にご指導いただいた林井先生をはじめ、的確なアドバイスやご指導、進捗状況を気にかけてくださりました、宮林先生、下村先生、蜂谷先生、先輩方、大変お世話になりました。卒業研究を進めることができたのは皆様のおかげです。本当にありがとうございました。心より感謝と敬意を。

参考文献

- [1] T.Suzuki and D. F. Measday, “Total nuclear capture rates for negative muons “
Physical Review C, 35, 2212, June 1987.
- [2] 市川真有 永松えりな 皆吉遥 「 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 卒業論文 2016.
- [3] 石黒稚可子 藪内晶友美 「 μ 粒子の寿命及び放出電子のエネルギースペクトラム測定」 奈良女子大学理学部数物科学科物理学コース 高エネルギー物理学研究室 卒業論文 2017.
- [4] 奥田真利子 「負電荷ミュー粒子と正電荷ミュー粒子の寿命測定」 奈良女子大学理学部数物科学科物理学コース 高エネルギー物理学研究室 卒業論文 2018.

付録

ele-mag.c

CAMAC からのデータ収集

```

/***** ele_mag.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;
    int data1,data2,data3,data4,data5,data6,data7,data8,data9,data10,
    data11,data12;
    int data25,data26;
    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;
//lamsrc=ctdcqn; /* time calibration */
lamch=ch0;

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

/*=====
 * Send enable command to prepare the first event.
 *=====*/
//CAMAC(NAF(cadcqn1 ,ch0 ,26),&data1,&q,&x); /* F=26 is enable. */
CAMAC(NAF(lamsrc ,lamch ,26),&data1,&q,&x); /* F=26 is enable. */
// CAMAC(NAF(cadcqn2 ,ch0 ,26),&data1,&q,&x); /* F=26 is enable.
*/
// CAMAC(NAF(ctdcqn ,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);
time_t current = time(NULL);
struct tm *timer = localtime(&current);
printf("start time %d/%d/%d %d:%d:%d\n" ,
timer->tm_year+1900,
timer->tm_mon+1,
timer->tm_mday,
timer->tm_hour,
timer->tm_min,
timer->tm_sec);

/*=====
 * Event loop.
 *=====*/
i_ev = 1;
while( i_ev <= n_ev )

```

```

{
    CAMAC(NAF(lamsrc , lamch , 9) ,&data1 ,&q ,&x) ; /* F=9 LAM and ADC(mu)
clear. */
    CAMAC(NAF(cadcqn2 , ch0 , 9) ,&data1 ,&q ,&x) ; /* F=9 LAM and ADC(e) clear
. */
    CAMAC(NAF(ctdcqn , ch0 , 9) ,&data1 ,&q ,&x) ; /* F=9 LAM and TDC clear. */
//CAMAC(NAF(cadcqn1 , ch0 , 9) ,&data1 ,&q ,&x) ; /* time calibration */

    CAMAC(NAF(lamsrc , lamch , 9) ,&data1 ,&q ,&x) ;
    CAMAC(NAF(cadcqn2 , ch0 , 9) ,&data1 ,&q ,&x) ;
    CAMAC(NAF(ctdcqn , ch0 , 9) ,&data1 ,&q ,&x) ;

    /*-----
    * 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) ;
// - 20181220 add -
    CAMAC(NAF(cadcqn1 , ch4 , 0) ,&data25 ,&qq ,&x) ;
    CAMAC(NAF(cadcqn1 , ch5 , 0) ,&data26 ,&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);

/*-----
 * 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 */
{ time_t current = time(NULL);
  struct tm *timer = localtime(&current);
  printf("%d/%d/%d %d:%d:%d\n",
         timer->tm_year+1900,
         timer->tm_mon+1,
         timer->tm_mday,
         timer->tm_hour,
         timer->tm_min,
         timer->tm_sec);
  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);
  //-20181220 add ---
  printf(" data25= %d\n", data25);
  printf(" data26= %d\n", data26);
  //-
  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);

}

head=-1;

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

//---
    if( data1>10 ) /* write events if muon decays 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);
// -20181220 add ---
fprintf(fp, " %d", data25);
fprintf(fp, " %d", data26);
//---
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);

    }
/*-----
 * 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.*/

    printf(">>>>>>FINISH!<<<<<<\n\n");
    return 0;
}

```

rootfileOutput-mag

CAMAC からデータを 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_mag() {

    string    fname="2020_02_10_2400000"; //参照元ファイル名

    string    directory1="/home/2015-b4/ug2019/data2019/";
    string    directory2="/home/2015-b4/ug2019/data2019/";
    string    full_name1 = directory1+ fname+ ".dat"; //データファイルの絶対パス
    string    full_name2 = directory2+ fname+ ".root"; //ファイルの絶対パス root
    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, mut3, mut4, 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"); //定義
    mu_t->Branch("one", &one, "one/F"); //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("mut3", &mut3, "mut3/F"); //20181221-----
    mu_t->Branch("mut4", &mut4, "mut4/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 >> mut3 >>
mut4 >> es1 >> es2 >> et1 >> et2 >> ev1 >> ev2 >> ev3 >> ev4 >> ev5 >>
ev6 >> ev7 >> ev8 >> tdc0 >> tdc1 >> tdc2){

    mus1=mus1-106; //ペDESTALを引く
    if (mus1<0){mus1=0;} //ペDESTALを引いて負になる場合はに揃える0
    mus2=mus2-96;
    if (mus2<0){mus2=0;}
    mut1=mut1-88;
    if (mut1<0){mut1=0;}
    mut2=mut2-85;
    if (mut2<0){mut2=0;}
    mut3=mut3-20; //20181221---
    if (mut3<0){mut3=0;}
    mut4=mut4-97;
    if (mut4<0){mut4=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;
    }
    mu_t->Fill ();
    i++;
};

data.close (); //ファイル閉じる

TFile *rtfilemt = new TFile(full_name2.c_str(),"RECREATE"); //ファイルを作
成
mu_t->Write ();
rtfilemt->Close (); //ファイルを閉じる root

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

return ;
}

```

tdc-calibration2019

ROOT による TDC 時間較正

```

#include "fstream"
#include "TH1.h"
#include "TCanvas.h"
#include "TFile.h"
#include "TTree.h"
#include "TChain.h"
#include "TGraph.h"
#include "TMath.h"

void tdc_calibration2019() {
    c1 = new TCanvas("c1", "A Simple Graph with error bars", 200, 10, 700, 500)
    ;
    c1->Divide(2, 1);
    //c1->SetFillColor(42);
    //c1->SetGrid();
    //c1->GetFrame()->SetFillColor(21);
    //c1->GetFrame()->SetBorderSize(12);
    const Int_t n=100;
    Double_t x[n]
    ]={99.590,199.514,299.653,399.543,499.703,599.457,699.551,799.500,899.798,999.486};

    Double_t y[n]={500,1000,1500,2000,2500,3000,3500,4000,4500,5000};
    Double_t ex[n]
    ]={0.013,0.018,0.012,0.016,0.013,0.017,0.014,0.016,0.014,0.015};
    Double_t ey[n]={10,10,10,10,10,10,10,10,10,10};

    c1->cd(1);
    gr = new TGraphErrors(n, x, y, ex, ey);

    func = new TF1("func", "pol1");
    func->SetParameter(0, 2);
    func->SetParameter(1, 5);
    gr->Fit("func");

    gr->SetMarkerStyle(8);
    gr->SetMarkerColor(2);
    //gr->SetMarkerSize(1.5);
    gr->GetXaxis()->SetTitle("TDC [count]");
    gr->GetYaxis()->SetTitle("Delay Time [ns]");
    gr->GetYaxis()->SetTitleOffset(1.3);
    gr->GetXaxis()->SetTitleOffset(1.3);
    gr->SetTitle("TDC Time Calibration");
    gr->Draw("AP");

```

```
gStyle->SetOptFit(1);

Double_t yy[10];
Double_t p0 = func->GetParameter(0);
Double_t p1 = func->GetParameter(1);
//cout << "p0=" << p0 << "p1=" << p1 << endl;

for (Int_t i = 0; i<n; i++){
    yy[i] =p0 + p1*x[i] - y[i];
};

c1->cd(2);
TGraph* Gr = new TGraph(n,x,yy);
Gr->Draw("AP");
Gr->SetMarkerStyle(5);
Gr->SetMarkerSize(1.5);
Gr->SetTitle("Fit Difference");
//Gr->SetMarkerColor(2);
Gr->GetXaxis()->SetTitle("TDC [count]");
Gr->GetYaxis()->SetTitle("Delay Time [ns]");

return;
}
```

histgramOutput2019

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 histgramOutput2019() {
    gStyle->SetOptFit(1);

    TCanvas *c1=new TCanvas("c1"," ctitle",0,0,500,400);
    c1->Divide(4,3);
    TChain *mu_t=new TChain("mu_t"," mu_title");
    mu_t->Add("/home/2015-b4/ug2019/data2019/*.root");

    /* cut*/
    TCut e_decay="120<es1 && tdc2<4000";
    TCut mag ="mut3<50 || mut4<225";

    c1->cd(1);
    TH1F *h1=new TH1F("h1"," mut1",500,0,1500);
    //c1->SetLogy();
    mu_t->Draw("mut1>>h1",e_decay,"");

    c1->cd(2);
    TH1F *h2=new TH1F("h2"," mut2",100,0,1500);
    mu_t->Draw("mut2>>h2",e_decay,"");

    c1->cd(3);
    TH1F *h3=new TH1F("h3"," mut3",100,0,1500);
    mu_t->Draw("mut3>>h3",e_decay,"");

    c1->cd(4);
    TH1F *h4=new TH1F("h4"," mut4",100,0,1500);
    mu_t->Draw("mut4>>h4",e_decay,"");

    c1->cd(5);
    TH1F *h5=new TH1F("h5"," es1",100,0,1500);
    mu_t->Draw("es1>>h5",e_decay,"");

```

```
c1->cd(6);
TH1F *h6=new TH1F("h6", "es2", 100, 0, 1500);
mu_t->Draw("es2>>h6", e_decay, "");

c1->cd(7);
TH1F *h7=new TH1F("h7", "mus1", 100, 0, 1500);
mu_t->Draw("mus1>>h7", e_decay, "");

c1->cd(8);
TH1F *h8=new TH1F("h8", "mus2", 100, 0, 1500);
mu_t->Draw("mus2>>h8", e_decay, "");

c1->cd(9);
TH1F *h9=new TH1F("h9", "et1", 100, 0, 1500);
mu_t->Draw("et1>>h9", e_decay, "");

c1->cd(10);
TH1F *h10=new TH1F("h10", "et2", 100, 0, 1500);
mu_t->Draw("et2>>h10", e_decay, "");

c1->cd(11);
TH1F *h11=new TH1F("h11", "ev1", 100, 0, 1500);
mu_t->Draw("ev1>>h11", e_decay, "");

c1->cd(12);
TH1F *h12=new TH1F("h12", "tdc2", 100, 0, 1500);
mu_t->Draw("tdc2>>h12", e_decay, "");

return;
}
```

histgramOutput2019

ROOT による μ 粒子の寿命解析

```

#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", "c1title", 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"); //作成
    mu_ch->Add("/home/2015-b4/ug2019/data2019/*.root"); //入れる

    TCut e_decay="120<es1 && tdc2<4000";
    TCut mag="120<es1 && tdc2<4000 && (50<mut3 || 225<mut4)";
    TCut nonmag="120<es1 && tdc2<4000 && (mut3<50 && mut4<225)";

    /*  $\mu$  粒子寿命測定 */
    c1->cd(1); //移動
    TH1F *h1= new TH1F("h1", "mu_lifetime", 200, 0, 2000); //新しいヒストグラムを作成
    //c1->SetLogy();
        mu_ch->Draw("tdc2>>h1", mag, ""); //ヒストグラムを出力

    /* Fit */
        TF1 *f_exp_y=new TF1("f_exp_y", "[0]*exp(-x/[1])", 0, 2000); //f-exp-y=
        p0+exp(-x/p1)
    f_exp_y->SetParameter(0, 665); //パラメータの初期値を設定 番目初期値(,)
    f_exp_y->SetParameter(1, 435);

    h1->Fit("f_exp_y", "", "", 60, 2000);

        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
    Double_t p1_y = f_exp_y->GetParameter(1); //p1

    cout << " p0-y=" << p0_y << " p1-y=" << p1_y << endl;

```

```
const Int_t n=500;
Double_t x[n],y[n],Y[n],y-Y[n],X[n],pull[n];
for (Int_t i=1 ; i<n ; i++){
    x[i]= h1->GetBinCenter(i);
    y[i]= h1->GetBinContent(i);
    Y[i]= p0-y*exp(-x[i]/p1-y);
    pull[i]=(y[i]-Y[i])/(TMath::Power(y[i],0.5));
};

c1->cd(2);
TGraph* g=new TGraph(n,x,pull);
g->GetYaxis()->SetRangeUser(-5,5);
g->GetXaxis()->SetRangeUser(0,2000);

g->Draw("AP");

g->SetMarkerStyle(7);
g->SetTitle("Pull");

return ;
}
```