

宇宙線観察から学ぶ粒子の崩壊とスピン回転

2022 年 9 月 27 日

1 理論

1.1 ミュオンの生成

ミュオン μ^- は第 2 世代の荷電レプトンであり、地表に到達する 2 次宇宙線の大部分を占める。電荷は ± 1 , スピン $1/2$, 質量 $(105.658\,371\,5 \pm 0.000\,003\,5) \text{ MeV}$ で電子の約 200 倍あり, 寿命は $(2.196\,981\,1 \pm 0.000\,002\,2) \mu\text{s}$ である。2 次宇宙線のミュオンは主に荷電パイオンの崩壊によって生じる。1 次宇宙線の陽子が大気上層で

$$\begin{aligned}p + p &\rightarrow p + n + \pi^+ \\p + n &\rightarrow p + p + \pi^- \\p + n &\rightarrow p + n + \pi^+ + \pi^-\end{aligned}$$

の反応を起こしてパイオンが生じ、

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

と崩壊する。

ニュートリノのヘリシティは負、反ニュートリノは正であるから、角運動量保存よりパイオン静止系にて μ^+ のスピンは運動量と逆、 μ^- は同じ向きとなる。パイオンの崩壊は主に電磁シャワー内部で生じることを考慮すると、実験室系にて地表に降り注ぐ向きの運動量が大い μ^+ はスピンが下向き、 μ^- はスピンが上向きになる割合が多くなると考えられる。

1.2 ミュオンの崩壊

宇宙線ミュオンは地表に降り注ぐ間は固有時間がほとんど進まないが、資料にトラップされると平均寿命 $(2.196\,981\,1 \pm 0.000\,002\,2) \mu\text{s}$ で崩壊し電子または陽電子を放出する。

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

Fermi の黄金則を使えばエネルギーとスピンに依存する (陽) 電子の放出方向の分布が計算でき、以下の式で表される。

$$\frac{dN(v, |b|, t)}{dt} = N_0 \frac{\lambda}{2\pi} y^2 (3 - 2y) \left(1 + \frac{2y - 1}{3 - 2y} P_z \cos \theta \right) e^{-\gamma t} dy d\Omega$$

表 1: 各元素における μ^- の寿命変化

元素	文献値 [μs]
Cu	0.16
Al	0.86
Ca	0.33
C	2.04
O	0.81

積分して、

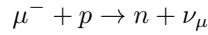
$$N = \iint N_0 \frac{\lambda}{4\pi} A_0 (1 + aP \cos \theta) e^{-\gamma t} dy d\Omega$$

を得る。すなわち、 μ^+ はスピンと同じ向きに陽電子を、 μ^- はスピンと逆向きに電子を出しやすい。

1.1 を合わせて踏まえると、地表に降り注ぐミュオンのうち運動量の大きいものは上向きに、運動量の小さいものは下向きに (陽) 電子を多く放出する。

1.3 負ミュオンの原子核捕獲

一部の μ^- は物質中に侵入すると原子核の Coulomb ポテンシャルにとらわれミュオン原子を形成する。この μ^- は特性 X 線やオージェ電子を放出して次第に低準位へ遷移し、やがて原子核へ落ち込む。原子核にて陽子と弱い相互作用をすることで中性子を放出する。



(陽) 電子だけでなく中性子も検出する装置を使うと、 μ^- の減衰比には通常の崩壊に加えてこの原子核捕獲の確率も加わるため、見かけの寿命は μ^+ に比べて短くなる。寿命変化は原子核に依存し、今回使用した試料に含まれる元素では表 1 のようになる [1]。

ミュオンが試料に停止してから崩壊して (陽) 電子を出すまでの時間 t を計測し、 $(t, t + \Delta t)$ のカウント $N(t)$ をヒストグラムで集計すれば、 μ^- の寿命変化及び計測のバックグラウンド BG を反映して以下の式が成立すると考えられる。

$$N(t) = N_0^+ e^{-t/\tau_+} + N_0^- e^{-t/\tau_-} + BG \quad (1)$$

1.4 ミュオンのスピン偏極検出

1.2 に基づけば、磁場をかけない状態で運動量の比較的大きいミュオンを多くトラップする試料を上下 2 つの検出器で挟むと上側で (陽) 電子をより多く検出する。しかし上下の検出数の差がスピン偏極によるものなのか検出器の特性によるものなのかを判断することは難しい。そこで偏極の検出では一般にミュオンスピン回転 (μSR) と呼ばれる Larmor 歳差運動を用いた方法を採用。

磁気回転比 γ のミュオンに磁場 B がかったとき、

$$\omega = \gamma B \quad (2)$$

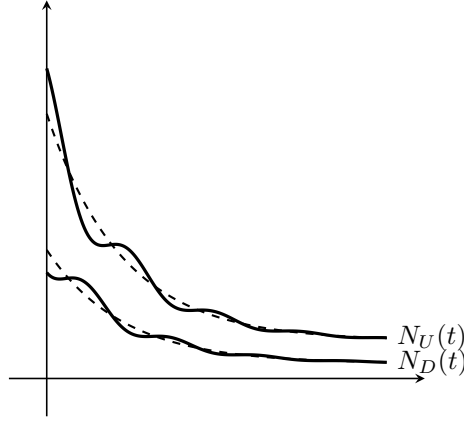


図 1: Larmor 歳差運動下で予想される粒子検出数。破線は磁場をかけない場合。

でスピンは歳差運動する。したがって 1.2 によれば、放出 (陽) 電子の角度分布も磁場を軸に回転し、粒子の検出数が崩壊時間によって変化する (図 1)。

上下の検出器でのカウントをそれぞれ N_U, N_D とし、(2) の ω の大きさを ω とすれば、1.3 で述べた μ^- の原子核捕獲も考慮して、以下の式が成り立つと考えられる。

$$\begin{aligned} N_U(t) &= N_{U0}^+(1 + P^+ \cos \omega t)e^{-t/\tau_+} + N_{U0}^-(1 + P^- \cos \omega t)e^{-t/\tau_-} + BG_D \\ N_D(t) &= N_{D0}^+(1 - P^+ \cos \omega t)e^{-t/\tau_+} + N_{D0}^-(1 - P^- \cos \omega t)e^{-t/\tau_-} + BG_D \end{aligned} \quad (3)$$

ここで τ_- は原子核捕獲のため一般に小さく、 $t \gg \tau_-$ にて (3) 各式の第 2 項は無視できる。加えて μ^- は試料に侵入すると周囲の電子との間で相互作用してスピン偏極が破れるので N_{U0}^-, N_{D0}^- は小さい。(3) 各式の第 2 項を無視すれば、 $N'_U = N_U - BG_U, N'_D = N_D - BG_D$ 及び実数 α を使って、

$$\mathcal{A} = \frac{\alpha N'_D - N'_U}{\alpha N'_D + N'_U} = \frac{(\alpha N_{D0}^+ - N_{U0}^+) + (\alpha N_{D0}^+ + N_{U0}^+)P^+ \cos \omega t}{(\alpha N_{D0}^+ + N_{U0}^+) + (\alpha N_{D0}^+ - N_{U0}^+)P^+ \cos \omega t} \quad (4)$$

と表される非対称度 \mathcal{A} は、 $\alpha N_{D0}^+ - N_{U0}^+ = 0$ なる α にて角振動数 ω の三角関数となる。

2 実験・解析方法

本演習では、

- ミュオンの寿命
- 負ミュオンの寿命変化
- スピン偏極

の測定を目的とする。

2.1 寿命測定

図 2 に示すように試料を上下からプラスチックシンチレータ (PSc) で挟み光電子増倍管 (PMT) に接続する。試料には Cu, Al の板を重ね合わせたものと、大理石の板とを使った。PSc+PMT は上層から順に T,U,D とし、各層でシンチレーション光を別々に検出できるようにする。

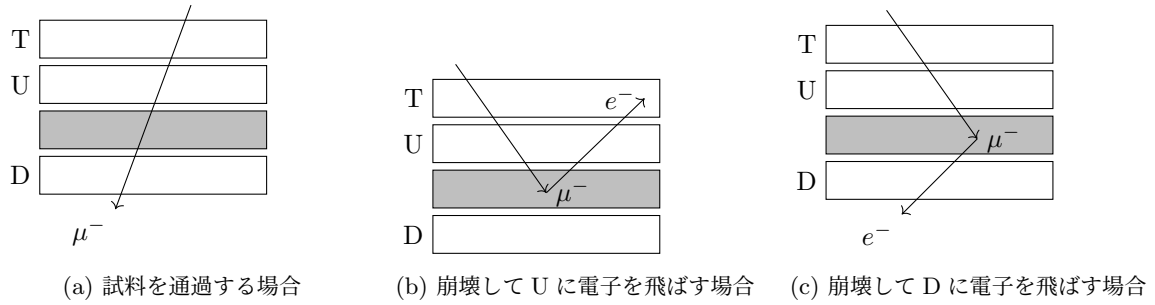


図 2: PSc+PMT の配置と粒子の飛程。着色部が試料を表す。

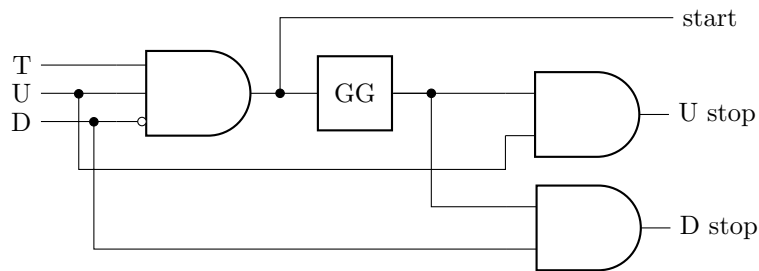


図 3: 回路の概要。GG は gate generator を表し、入力から所定の時間 (本演習では 20 μ s) 出力を続ける。

宇宙線及び崩壊粒子^{*1}の飛程を考慮すると、以下の事象が考えられる。

- (a). 粒子が試料を通過
- (b). 崩壊時に粒子を上方へ放出
- (c). 崩壊時に粒子を下方へ放出

事象 (a) では TUD 全ての PSc+PMT が反応するが、事象 (b), (c) ではまず TU が反応し D が反応しない。その後時間を置いて粒子が崩壊すると、事象 (b) では U が、事象 (c) では D が反応する。

各事象を弁別するためおおよそ図 3 に示す論理回路を組む。^{*2}左端の and ゲートでミュオンが試料に停止したか検出し、停止すれば gate generator を作動、時間測定を開始する。この gate generator が作動している間に U または D で信号を検出すれば、崩壊によって粒子が放出されたものとして同図右側の and gate で測定を止める。

Cu, Al 及び及び大理石それぞれの試料で計測は 18 h, 51 h にわたって行った。

U, D 両側で測定したカウント N_U, N_D を合計して、時間を横軸としてヒストグラムにする。 $N_U + N_D$ の分布は (1) に従うはずであるから、この式によってフィッティングし寿命 τ_+, τ_- を求める。

^{*1} μ^- の原子核捕獲で生じる中性子も速度が \sim となって PSc の検出下限 \sim を上回るの、これには中性子も含む。

^{*2} 実際に構成した論理回路は付録 A を参照。2 種類の試料を同時に測定するため、入出力が 2 系統あるのに加えて制御系統が複雑になっている。

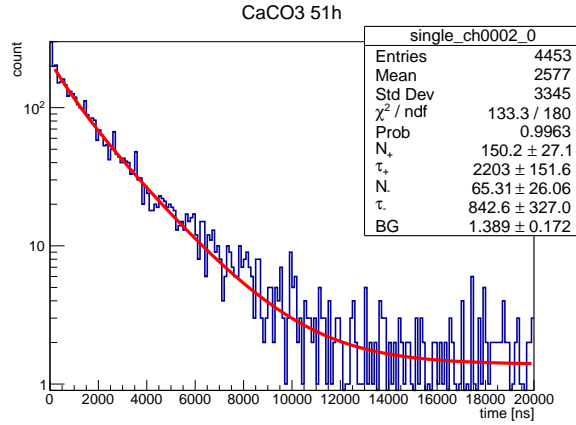


図 4: 大理石

2.2 スピン偏極測定

2.1 での Cu, Al の実験と同様にセットアップした上で、ソレノイドコイルによって水平に磁場を印加、崩壊時間を測定する。

解析にあたっては、まず上下別々にヒストグラムを描いて

$$\begin{aligned} N_U(t) &= N_U^0 e^{-t/\tau_U} + BG_U \\ N_D(t) &= N_D^0 e^{-t/\tau_D} + BG_D \end{aligned} \quad (5)$$

でフィッティングする。得られたバックグラウンドを除いて $N'_U = N_U - BG_U$, $N'_D = N_D - BG_D$ とし、 α を 0.01 ずつ変えて非対称度 (4) を計算する。ただし時間原点を揃える必要があるため、回路に組み込んだ遅延を考慮して、

$$\mathcal{A}(\alpha, t_{\text{offset}}) = \frac{\alpha N'_D(t + t_{\text{offset}}) - N'_U(t)}{\alpha N'_D(t + t_{\text{offset}}) + N'_U(t)}, \quad t_{\text{offset}} = -30 \text{ ns} \quad (6)$$

の $A \cos(2\pi f t)$ によるフィッティングが最適なものを探す。フィッティングの χ^2 が最小になる α における f を振動数の測定値とする。

検証のため測定値 f と (2) から磁場の強さ B を逆算して、ソレノイドコイルの設計上期待される値 $B_{\text{design}} = 2 \text{ mT}$ と比較する。

3 結果・考察

3.1 ミュオンの寿命

測定データを図 4 に掲げる。フィッティングした曲線に基づき、それぞれの試料におけるミュオンの寿命は表 2 のとおりになる。

表 2: μ^\pm の寿命測定結果

試料	粒子	寿命 [μs]	文献値 [μs]
Cu, Al	μ^+		2.20
	μ^-		Cu: 0.16, Al: 0.86
大理石	μ^+	2.20	2.20
	μ^-	0.84	Ca: 0.33, C: 2.04, O: 0.81

表 3: (5) によってフィッティングしたときの変数値

N_U^0	$4.63817e + 02 \pm 8.72257e + 00$
τ_U	$2.07050e + 03 \pm 3.18642e + 01$
BG_U	$6.39559e + 00 \pm 2.74083e - 01$
N_D^0	$2.14776e + 02 \pm 6.68531e + 00$
τ_D	$2.00158e + 03 \pm 5.62010e + 01$
BG_D	$8.27831e + 00 \pm 2.91137e - 01$

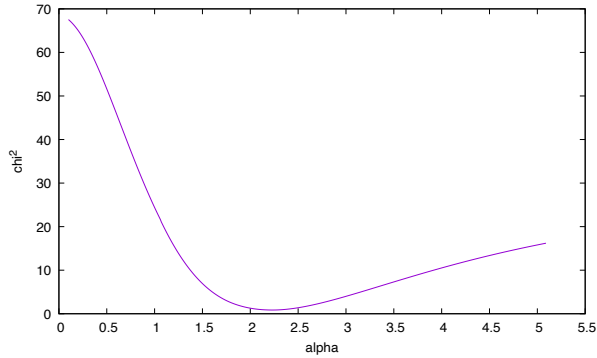


図 5: α を変えて非対称度 (6) を三角関数でフィッティングしたときの χ^2 値

3.2 スピン偏極

測定結果に (5) でフィッティングしたものを図に掲げる。磁場をかけることでカウントは (3) に従うはずであり (5) によるフィッティングは本来妥当でないが、この操作はバックグラウンドを差し引くためのものなので、 BG_U の標準偏差及び χ^2 の上側累積確率などフィッティングの信頼性は特に留意しない。各変数は表 3 のようになる。

次いで α を 0.01 ずつ変えて非対称度 α を計算し、 $A \cos(2\pi ft)$ にフィッティングさせたときの χ^2 値を図 5 に示す。極値は $(\alpha, \chi^2) = (2.23, 0.85)$ であり、上側累積確率は 74.7% である。そこで $\alpha = 2.23$ を採用して (6) を描き \cos でフィッティングすると図 6 の通りになる。

これより振動周期 $T = (4.217 \pm 0.517)\text{s}$ を得て、(2) より磁場を計算すると $B = (1.75 \pm 0.22)\text{mT}$ となる。

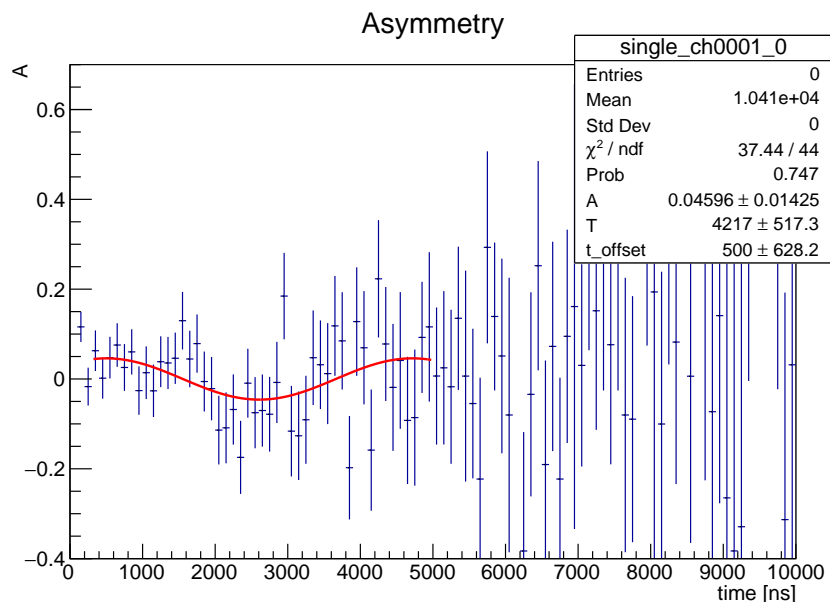


図 6: 非対称度 \mathcal{A} と \cos によるフィッティング

$B_{design} = 2 \text{ mT}$ と比較して、この結果が妥当であることが確かめられる。

付録 A 論理回路

実験で実際に用いた NIM モジュールの回路を図 7 に掲げる。主に Cu, Al からの信号の入力部 (Cu, Al)、大理石からの入力部 (CaCO_3)、制御部 (GATES)、出力部 (OUTPUTS) に大別できる。

入力は TUD 各層それぞれ 3 組の PSc+PMT から引いている。それぞれ T1, T2, T3, U1, ... と番号を振っている。Cu, Al の T に描いたような or 接続を Cu, Al の U 以下全ての入力端子でとっているが、簡単のため Cu, Al の T 以外は省略した。なお、この or 接続以前で delay を置いているのは Cu, Al の T のみである。

制御部 GATES は Cu, Al と大理石の信号が干渉しないようにスイッチの役割を果たす。具体的には、Cu, Al 側の GG が起動している間は大理石の信号が入っても stop と判定しない。逆もまた然りである。

DAQ-PC では短時間に複数の start 信号を受けたとき最初の信号のみを使用し、それ以降の信号は無視する。

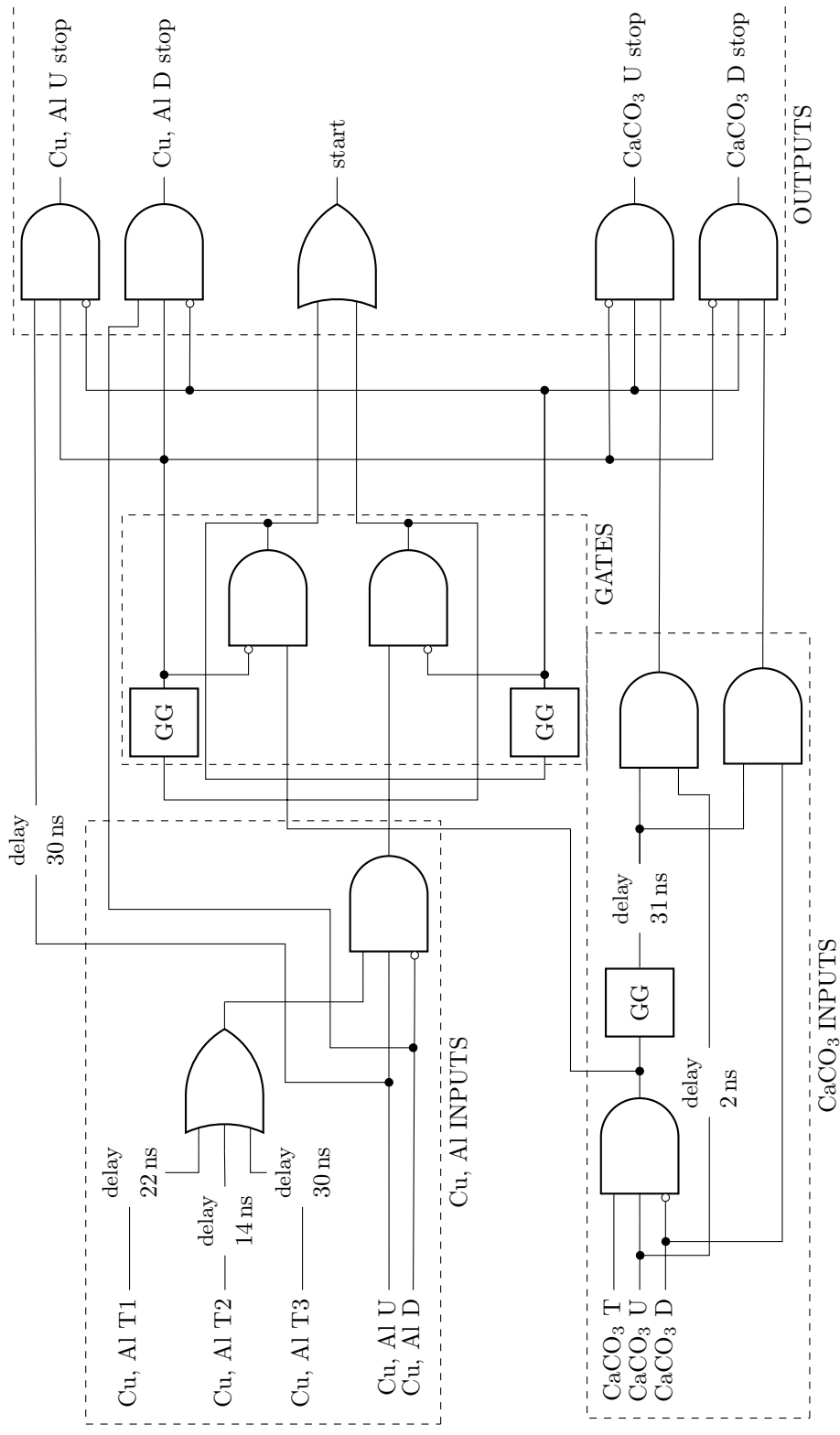


図 7: 実験で用いた論理回路

付録 B 解析コード

ここでは解析に用いたコードの概要を示す。変数の宣言やヒストグラムの取得などは省略しているため、表示の通りに実装しても作動しない。実際に解析で使用したものは <https://github.com/LowToneVoice/ksc16/tree/main/group7/analysis> を参照のこと。また測定データを <https://github.com/LowToneVoice/ksc16/tree/main/group7/data> に置いている。

B.1 寿命測定

Listing 1: life.C

```
1 void life()
2 {
3     // sum of hists
4     hu->Add(hd);
5     // configure bin of hist
6     hu->Rebin(100);
7     // max and min of axis
8     hu->GetXaxis()->SetRangeUser(0, 20000);
9     hu->GetYaxis()->SetRangeUser(0.1, 300);
10    hu->Draw();
11
12    // fitting curve
13    TF1 *f1 = new TF1("f1", "[0]*exp(-(x/[1]))+[2]*exp(-(x/[3]))+[4])");
14    f1->SetParameters(150, 2000.0, 120, 900.0, 2.0);
15    f1->SetParNames("N_{+}", "#tau_{+}", "N_{-}", "#tau_{-}", "BG");
16    hu->Fit("f1", "", "", 100, 20000);
17    hu->SetLineWidth(2.0);
18    // y axis to log scale
19    c1->SetLogy();
20    // label displayed with the graph
21    hu->SetTitle("CaC03_51h;time[ns];count;");
22 }
```

B.2 スピン偏極測定

まず expFit.C で (5) のフィッティングを行いバックグラウンドを求める。(6) の \cos によるフィッティングの χ^2 値を複数の α で計算する。 χ^2 が極小となる α を選び (6) のフィッティングパラメーターを結果とする。

Listing 2: expFit.C

```

1 void expFit()
2 {
3     gStyle->SetOptStat(0000);
4     gStyle->SetOptFit(0000);
5     gStyle->SetFuncWidth(2);
6
7     const char *fp = "../data/4013.root";
8     TFile *file = new TFile(fp);
9     TH1D *hu;
10    TH1D *hd;
11    TCanvas *c1 = new TCanvas("c1", "c1", 600, 600);
12    file->GetObject("single_ch0000_0", hu);
13    file->GetObject("single_ch0001_0", hd);
14    hu->Rebin(100);
15    hd->Rebin(100);
16
17    // fitting curve
18    TF1 *fu = new TF1("fu", "[0]*exp(-x/[1])+[2]");
19    TF1 *fd = new TF1("fd", "[0]*exp(-x/[1])+[2]");
20    fu->SetParameters(200.0, 2000.0, 0.0);
21    fu->SetParNames("N_U", "#tau_U", "BG_U");
22    fd->SetParameters(100.0, 2000.0, 0.0);
23    fd->SetParNames("N_D", "#tau_D", "BG_D");
24    hu->Fit("fu", "", "", 300, 20000);
25    hu->SetLineWidth(2.0);
26    hd->Fit("fd", "", "", 300, 20000);
27    hd->SetLineWidth(2.0);
28
29    // graph and axis label
30    hu->SetTitle("Cu, Al with B33h; time [ns]; count;");
31    hd->SetTitle("Cu, Al with B33h; time [ns]; count;");
32    hu->Draw();
33    hu->SetLineColor(kRed);
34    hu->GetXaxis()->SetRangeUser(0, 20000);
35    hu->GetYaxis()->SetRangeUser(0.9, 500);
36    hu->Draw("same");
37    hu->GetXaxis()->SetRangeUser(0, 20000);
38    hu->GetYaxis()->SetRangeUser(0.9, 500);

```

```
39      // y axis to log scale
40      c1->SetLogy();
41  }
```

Listing 3: findPar.C

```
1  void findPar()
2  {
3      TH1D *h1;
4      TH1D *h2;
5      TH1D *h3;
6      TH1D *h4;
7      TH1D *h1_init;
8      TH1D *h2_init;
9      const char *fp = "../.../data/4013.root";
10     TFile *file = new TFile(fp);
11     // parameters
12     int i, j, k;
13     int offset = -30;
14     double alpha;
15     // results
16     Double_t erfun;
17     Double_t ndf;
18     Double_t prob;
19     // BGs
20     const double bg_u = 6.39559e+00;
21     const double bg_d = 8.27831e+00;
22     // output files
23     std::ofstream outputfile_chi2("param_chi2.dat");
24     std::ofstream outputfile_prob("param_prob.dat");
25     // hists
26     file->GetObject("single_ch0000_0", h1);
27     file->GetObject("single_ch0001_0", h2);
28     h1_init = (TH1D *)h1->Clone();
29     h2_init = (TH1D *)h2->Clone();
30
31     for (i = 0; i < 500; i++)
32     {
33         alpha = 0.1 + i * 0.01;
34     }
```

```

35     h1 = (TH1D *)h1_init->Clone();
36     h2 = (TH1D *)h2_init->Clone();
37
38     // move h2 left for offset
39     for (k = 0; k < h2->GetXaxis()->GetNbins(); k++)
40     {
41         if (offset + k < 0)
42         {
43             h2->SetBinContent(k, 0);
44         }
45         else
46         {
47             h2->SetBinContent(k, h2_init->GetBinContent(k +
48                 offset));
49         }
50     }
51
52     // rebin
53     h1->Rebin(100);
54     h2->Rebin(100);
55
56     // for error evaluation
57     h1->Sumw2();
58     h2->Sumw2();
59
60     // delete BGs
61     for (k = 0; k < h1->GetXaxis()->GetNbins(); k++)
62     {
63         h1->SetBinContent(k, h1->GetBinContent(k) - bg_u);
64     }
65     for (k = 0; k < h2->GetXaxis()->GetNbins(); k++)
66     {
67         h2->SetBinContent(k, h2->GetBinContent(k) - bg_d);
68     }
69     h3 = (TH1D *)h1->Clone();
70     h4 = (TH1D *)h2->Clone();
71
72     // h2 times alpha
73     for (k = 0; k < h2->GetXaxis()->GetNbins(); k++)
74     {

```

```

74         h2->SetBinContent(k, (h4->GetBinContent(k) * alpha));
75         h2->SetBinError(k, (h4->GetBinError(k) * alpha));
76     }
77
78     // D-U/D+U
79     h3 = (TH1D *)h1->Clone();
80     h4 = (TH1D *)h2->Clone();
81     h4->Add(h3, -1);
82     h2->Add(h1);
83     h4->Divide(h2);
84
85     // fitting by cos
86     TF1 *f = new TF1("f", "[0]*cos(2*[1]*TMath::Pi()*x-[2])
        [1])");
87     f->SetParNames("A", "T", "t_offset");
88     f->SetParameters(.1, 3500, .0);
89     f->SetParLimits(0, .01, .5);
90     f->SetParLimits(1, 2500, 4500);
91     f->SetParLimits(2, -500., 500.);
92     h4->Fit("f", "", "", 300, 5000);
93
94     // output chi^2/ndf and prob
95     erfun = h4->GetFunction("f")->GetChisquare();
96     ndf = h4->GetFunction("f")->GetNDF();
97     prob = h4->GetFunction("f")->GetProb();
98     outputfile_chi2 << alpha << " " << erfun / ndf << " ";
99     outputfile_prob << alpha << " " << prob << " ";
100
101     // change the lines of
102     outputfile_chi2 << endl;
103     outputfile_prob << endl;
104 }
105 outputfile_chi2.close();
106 outputfile_prob.close();
107 }

```

Listing 4: fitAsym.C

```

1 void fitAsym()
2 {

```

```

3      gStyle->SetOptStat(1111);
4      gStyle->SetOptFit(1111);
5      gStyle->SetFuncWidth(2);
6
7      TH1D *h1;
8      TH1D *h2;
9      TH1D *h3;
10     TH1D *h4;
11     TH1D *h1_init;
12     TH1D *h2_init;
13     const char *fp = "../.../data/4013.root";
14     TFile *file = new TFile(fp);
15     // parameters
16     int i;
17     int offset = -30;
18     double alpha = 2.23;
19     // BGs
20     const double bg_u = 6.39559e+00;
21     const double bg_d = 8.27831e+00;
22     // hists
23     file->GetObject("single_ch0000_0", h1);
24     file->GetObject("single_ch0001_0", h2);
25     h1_init = (TH1D *)h1->Clone();
26     h2_init = (TH1D *)h2->Clone();
27
28     // move h2 left for offset
29     for (i = 0; i < h2->GetXaxis()->GetNbins(); i++)
30     {
31         if (offset + i < 0)
32         {
33             h2->SetBinContent(i, 0);
34         }
35         else
36         {
37             h2->SetBinContent(i, h2_init->GetBinContent(i + offset))
38             ;
39         }
40     }
41     // rebin

```

```

42     h1->Rebin(100);
43     h2->Rebin(100);
44
45     // for error evaluation
46     h1->Sumw2();
47     h2->Sumw2();
48
49     // delete BGs
50     for (i = 0; i < h1->GetXaxis()->GetNbins(); i++)
51     {
52         h1->SetBinContent(i, h1->GetBinContent(i) - bg_u);
53     }
54     for (i = 0; i < h2->GetXaxis()->GetNbins(); i++)
55     {
56         h2->SetBinContent(i, h2->GetBinContent(i) - bg_d);
57     }
58     h3 = (TH1D *)h1->Clone();
59     h4 = (TH1D *)h2->Clone();
60
61     // h2 times alpha
62     for (i = 0; i < h2->GetXaxis()->GetNbins(); i++)
63     {
64         h2->SetBinContent(i, h4->GetBinContent(i) * alpha);
65         h2->SetBinError(i, h4->GetBinError(i) * alpha);
66     }
67
68     // D-U/D+U
69     h3 = (TH1D *)h1->Clone();
70     h4 = (TH1D *)h2->Clone();
71     h4->Add(h3, -1);
72     h2->Add(h1);
73     h4->Divide(h2);
74
75     // fitting by cos
76     TF1 *f = new TF1("f", "[0]*cos(2*TMath::Pi()*x[2])/[1]");
77     f->SetParNames("A", "T", "t_offset");
78     f->SetParameters(.1, 3500, .0);
79     f->SetParLimits(0, .01, .5);
80     f->SetParLimits(1, 2500, 4500);

```

```
81      f->SetParLimits(2, -500., 500.);
82      h4->Fit("f", "", "", 300, 5000);
83
84      h4->SetTitle("Asymmetry;time[ns];A");
85      h4->Draw();
86      h4->GetXaxis()->SetRangeUser(0, 10000);
87      h4->GetYaxis()->SetRangeUser(-.4, .7);
88  }
```

参考文献

- [1] 伊藤泰男, 鍛冶東海, 田畑米穂, 吉原賢二, 「素粒子の化学」学会出版センター (1985)