Measurement of Lifetime of Cosmic Ray Muons

I. D. Kay
Tim O’Flight

Abstract
Cosmic ray muons were incident on a plastic scintillator and decayed into electrons and neutrinos. Over a run of twelve days, time between incidence and decay was recorded for each of 13,788 events, yielding a distribution of times in the form of a falling exponential. After folding in statistical and systematic uncertainties and an estimated muon-capture correction, the fitted muon lifetime is $2.044 \pm 0.08 \, \mu s$, marginally consistent ($P \approx 5\%$) with previous measurements, whose present world average is $2.197 \, \mu s$ [1]. Interpretation of the result is complicated somewhat by consideration of the effect of surrounding matter on the decay process.

I. INTRODUCTION
In this experiment, we recorded the distribution of decay times of cosmic ray muons, and from the distribution measured the lifetime of the muon. In vacuum, the only decay process for muons is

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

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

Since the observed decays in this experiment took place in the presence of matter, however, the $\mu^-$ particle should have a shorter lifetime because the possibility of being captured by a nucleus gives it the extra decay channel

$\mu^- + p^+ \rightarrow n + \nu_\mu$

Submitted April 1, 2011.
A 12-inch diameter, 6-inch high, cylindrical plastic scintillator and two 1-inch thick, 12-inch wide square scintillators above and below, all optically coupled to photomultiplier tubes (PMTs), were used with logic and delay circuitry to detect the muons and their decay products.\(^1\)

Particle incidence was distinguished from other events by coincidence between the top and central detectors. Particles stopped were distinguished from those passing through by anticoincidence with the lower detector. An event in the central detector following the stoppage of an incident particle by 0.5 to 8.5 $\mu$s was considered a decay event.

II. Theory

In the upper atmosphere, pions are produced when atmospheric protons are hit by protons incident from outer space [2]. Charged pions then decay (with mean lifetime $\tau_{\pi^\pm} = 2.60 \times 10^{-8} \text{s}$) into muons by the processes $\pi^- \rightarrow \mu^- + \bar{\nu}_\mu$ and $\pi^+ \rightarrow \mu^+ + \nu_\mu$, according to the Feynman diagrams below.

If the pions travel\(^2\) at 0.998$c$, a fraction $1/e$ of them will decay in $\gamma\beta c \tau \approx 1950 \text{ m}$, where $\gamma\beta = p/mc$ is the pion’s Lorentz boost factor and $\tau$ is the pion lifetime. So if they are produced at a typical altitude of 8 km, the fraction decaying into muons by sea level will be approximately

---

\(^1\)Reviewer points out that nowhere in this report do we discuss the expected muon incidence and stopping rates and compare them with the observed rates. The $1/\text{cm}^2/\text{minute}$ rule of thumb yields a flux on the order of 12 Hz for the central module’s 730 cm\(^2\) circular cross-section. 15 cm of plastic will stop muons of kinetic energy up to (very roughly) 60 MeV. The energy spectrum is roughly flat below 1 GeV, generally falls with energy, and has a mean around 4 GeV. Let’s crudely assume a uniform distribution from 0 to 8 GeV. Then we stop $60/8000 \approx 0.8\%$ of 12 Hz, or 0.1 Hz. In practice, the experiment stopped 14000 muons in 12 days, or 0.014 Hz. The observed rate is about a factor of 7 lower than the crude calculation.

\(^2\)Reviewer correctly points out that the number 0.998 has been pulled out of thin air — and needs justification.
$1 - e^{-4}$, or about 98%. These muons subsequently decay by the processes $\mu^- \rightarrow e^- + \nu_\mu + \bar{\nu}_e$ and $\mu^+ \rightarrow e^+ + \nu_\mu + \nu_e$, corresponding to the Feynman diagrams below.

\begin{center}
\includegraphics[width=0.5\textwidth]{feynman_diagram1.jpg}
\end{center}

In the presence of matter, the situation is further complicated by the fact that negative muons may be captured by nuclei, thus allowing them to decay by another process [3]: $\mu^- + p \rightarrow \nu_\mu + n$, according to the Feynman diagram below.

\begin{center}
\includegraphics[width=0.5\textwidth]{feynman_diagram2.jpg}
\end{center}

Experimental values of the rate of this capture process in $^{12}C$ are on the order of magnitude of $7 \times 10^3 \text{ s}^{-1}$ [4]. This capture rate, combined with the established experimental lifetime of $2.197 \times 10^{-6}$ s for free muons [1], yields a lifetime of about $2.16 \times 10^{-6}$ s for negative muons in carbon. We infer that the muon lifetime measured by stopping a mixture of positive and negative muons in a carbon-dominated detector will be systematically biased $-0.02 \pm 0.02 \times 10^{-6}$ s with respect to the lifetime of muons decaying in vacuum.
III. Method

A. Equipment

Fig. 1 shows a schematic diagram of the experimental setup. Three scintillation modules were arranged vertically and attached to PMTs. Muons or their decay electrons would cause scintillations, which in turn would excite the PMTs, producing a small current, which was sent into discriminators after being terminated into 50 $\Omega$ with a LEMO “tee” connector at the discriminator input. The discriminators had adjustable thresholds and widths, so that any signal from the PMT with a voltage above the set threshold would cause the discriminator to output a pulse of the set width. We set the thresholds by studying the rates at which the discriminators were triggered both singly and in coincidence, and we set the widths according to the requirements of the logic, as described below. The output from the discriminators was then taken through logic circuitry to a time-to-amplitude converter (TAC). The TAC accepted two inputs, a “start” signal and a “stop” signal. If it received a “stop” signal within a specified time (8.5 $\mu$s in our case) after a “start” signal, then it would output a pulse whose height was proportional to the time between the two input signals. To avoid problems of simultaneity, the TAC would not accept any “stop” signal that occurred within 12 ns of a “start.” The “start” and “stop” signals were determined by the discriminators and logic circuitry as described below. The output from the TAC was sent to a multi-channel analyzer (MCA), which counted the pulses in 512 channels, according to their pulse height. Thus, there was a one-to-one correspondence between channel number in the MCA and the time between the “start” and “stop” signals in the TAC.

In order to calibrate the MCA and test the response of the equipment, we fed the output from a pulse generator into a discriminator, and then into the TAC and MCA. We set the pulse generator to give double pulses, so that the first pulse would act as the “start” signal, and the second as the “stop” signal. We measured the time between the pulses by examining the output of the pulse generator directly on an oscilloscope. The results of the calibration are shown in Fig. 2 as a graph of channel number versus time. The correspondence between channel number and time is linear in the measured range of 0.5 $\mu$s to 8 $\mu$s, and is 60.99 $\pm$ 0.55 channels per microsecond, as detailed in the figure caption. Fig. 3 shows the results from our response test. We measured the number of events recorded in 20 seconds at various pulse delay settings. As the figure shows, the response is flat in the measured range. The figure shows no points of less
than 0.5 $\mu$s because at about that point, there seemed to be a drastic falloff in the response, and there were no counts at all. It seems likely that this falloff was due to the discriminator, which perhaps could not be triggered more than once in that span of time: When we directly watched the output from the discriminator on an oscilloscope, the second pulse disappeared at the same point at which the MCA stopped counting events.\textsuperscript{3} It is also possible that for weaker signals from the pulse generator, or different thresholds or widths from the discriminators, there may have been some time-delay dependence to the response. A more sophisticated analysis would have repeated this test for various signal levels using our final logic setup, in order to be sure that there was no bias in our results. Such an analysis, however, would have taken much of our limited time – time that we felt was better spent optimizing the logic.

\textsuperscript{3}Reviewer points out that it would be nice to check this conclusively.
Fig. 2. *Time calibration.* The figure shows channel number vs. time (µs). We assign to each point an uncertainty $1/\sqrt{12}$ on channel number and a fractional uncertainty of 2% on time delay measured by eye from the traces on an analog oscilloscope. To facilitate fitting, these are combined into a vertical error bar $\frac{1}{\sqrt{12}} \oplus 0.02 \cdot t \cdot \frac{dc}{dt}$. The time-to-channel relationship is linear in the measured range of 0.5 µs to 8 µs with a fitted slope of 60.99 ± 0.55 channels/µs, which implies a 0.90% (statistical) fractional uncertainty on the time calibration. In addition, the Tektronix 2215 analog oscilloscope claims ±3% time base accuracy [5], which we add in quadrature with 0.9% to yield a 3.1% time calibration uncertainty.

**B. Logic**

The TAC required both a “start” and a “stop” signal, as explained above. It was our goal in setting up the logic to maximize the chance of detecting any muon stopped in the second scintillation module and its decay, while excluding as many false events as possible. Formally, one would seek to maximize the significance of the muon-decay signal, which should scale as $S / \sqrt{S + B}$, where $S$ is signal counts and $B$ is background counts; in practice, we followed a more heuristic procedure to minimize background while preserving signal efficiency.

Ideally, every “start” signal would indicate a muon that passed through the first scintillation module and was stopped in the second. Thus, our logic only included “start” signals in which the first and the second PMTs were triggered, but the third one was not. By excluding events that triggered the third PMT, we hoped to filter out those events in which a muon traveled completely through the apparatus. The “stop” signal was furnished solely by the second PMT, but the TAC only accepted those “stop” signals that were within 8.5 µs of a valid “start” signal.

In order to ensure that the logic worked correctly, we needed to set the widths of the
discriminators properly. Our logic was of the form \((A \text{ AND } B \text{ AND } \neg C)\), where \(A\), \(B\), and \(C\) represent the first, second, and third discriminators, respectively. Since \((\neg C)\) always registers true except when \(C\) is triggered, it was important that the pulse given by \(C\) completely encompassed those given by \(A\) and \(B\), in order for \(C\) to exclude events in which all three were triggered simultaneously. Thus, we needed to set the width of \(C\) greater than those of \(A\) and \(B\). We set the width of \(A\) at 0.1 \(\mu s\), \(B\) at 0.05 \(\mu s\), and \(C\) at 0.25 \(\mu s\). Furthermore, since the logic elements took a finite time to make the transition from false to true, and since a muon traveling through the apparatus would trigger the first two PMTs before the third, we put the output from \(B\) through a 30 ns delay, in order to put it safely within the pulse from both \(A\) and \(C\). We used the same delay for the “stop” signal for the sake of consistency, though any difference between the time for the “start” and “stop” signals to reach the MCA would only move all the points to the left or the right, and would not affect our analysis in any way.

Before applying this logic to the experiment, we tested it with known signals from a pulse generator to ensure that it was functioning as we expected.
C. Preliminary tests and optimization

All preliminary tests were conducted by performing timed cosmic-ray runs, and counting selected signals. One important quantity we measured this way was the rate at which the second discriminator fired, \( R_B \). If \( N_S \) is the number of “start” signals counted in a given time, and \( T_0 \) is the period of time in which a “stop” signal must occur in order to be counted as an event, then the background count due to random coincidences can be estimated as \( N_S R_B T_0 \). This, of course, assumes that background events are uncorrelated with one another. \( (T_0 \) was 8.5 \( \mu s \) throughout the course of the experiment, while \( R_B \) depended on the setting of the discriminator’s threshold, and \( N_S \) varied from run to run depending on a number of factors.) It was our goal in the course of optimization to maximize the ratio of the event count (signal plus background, \( S + B \)) to this estimated background count \( (B) \), while also maximizing the event count itself — in other words to reduce the background as much as possible while preserving high signal efficiency. This was done by adjusting the thresholds of the discriminators. Although this heuristic process to maximize \( \frac{S+B}{B} \) while preserving large \( S \) succeeded, in retrospect we would have directly optimized the muon signal’s estimated statistical significance, \( \frac{S}{\sqrt{S+B}} \).

Our process of optimization was rather straightforward. We would adjust the discriminator thresholds, measuring those thresholds by the rate at which the discriminator triggered. Then we would conduct a test run, in which we would count both the “start” signals and the events. From these counts, as well as the time elapsed, we could calculate the ratio of events to estimated background, as well as the event rate.\(^4\)

There were, of course, some problems with our optimization process. First of all, we had limited laboratory time in which to conduct test runs. Since the event rate ranged only from one to three per minute, depending on the threshold settings, brief runs would not give enough counts to make very definite conclusions. On the other hand, if we took longer runs, then we could test fewer settings. In the end, we took most of our test runs over time spans of five to ten minutes, except for one longer one which we were able to leave running for two days. Given more time to conduct this experiment, it would have been useful to make more, longer test runs, and especially more overnight ones.

\(^4\)Reviewer points out that including some numbers from the optimization process would be good. If there is time, I may be able to dig them out of our 20-year-old logbook!
A useful check on our methods was to test the effect of removing the output of the third discriminator from the logic setup (i.e. using \( A \) AND \( B \) for the “start” signals). Since we set the rate of the third discriminator \( R_C \) to about 15 counts per second, and since its width was 0.25 \( \mu s \), the logic from C should only screen out about one out of every 270,000 “start” signals at random. However, by directly counting \((A \) AND \( B \) AND (NOT \( C \))) and \((A \) AND \( B \)) at the same time, we found that C actually screened out about two-thirds of the “start” signals. (For example, in one ten-minute run, the addition of C reduced the number of “start” signals from 1500 to 440.) Therefore, we concluded that the signals that were being screened did indeed represent some particle that traveled through the apparatus, triggering all three PMTs. Since such an event could not give us a measurable decay, any detected “decays” that would have been screened out by C must be part of the background. Thus, by measuring the effect that removing C had on the observed event rate, we could obtain another rough estimate of the background level, independently of our measurements of \( R_B \). In practice, our test runs were too short to obtain as precise an estimate of the background as we could get by measuring \( R_B \), since the expected effect of removing C was negligible. However, this was still a useful check, since it did not depend on any assumptions about whether the output from the second discriminator was uncorrelated. In fact, this method helped us to detect a “double pulsing” phenomenon in our circuit (caused by a small reflection), which was giving us much higher background levels than we expected. The double pulsing was fixed by adjusting the threshold on the second discriminator.

**D. Gathering data**

We conducted one run over a period of twelve days. During this time, the discriminators were set so that \( R_A \), \( R_B \), and \( R_C \) were 50 Hz, 127 Hz, and 20.3 Hz, respectively. This value of \( R_B \) corresponded to a random-coincidence background rate of 1.08 per 1000 “start” signals. This estimate of the background, however, is subject to error, in that it assumes that \( R_B \) was constant over the twelve-day period, when in fact we made no effort to exclude possible time dependence of \( R_B \).

Using a signal generator and an oscilloscope, we measured the voltage thresholds of the discriminators at 0.50 volts, 0.058 volts, and 0.92 volts. On reflection, the thresholds of the first two discriminators were probably set lower than necessary, resulting in a higher background count than would be optimally possible. Given more time to conduct optimization runs, we may
have been able to achieve better results.

At the conclusion of the run, we downloaded the data from the MCA to an IBM computer for statistical analysis.

E. Why muons?

How can we be confident that the particles whose lifetime we are measuring are muons? First, other experiments have shown that about 75% of cosmic ray particles at sea level are muons [1].

Second, there is no other known particle with decay time in the range 0.5–8.5 µs. The pion, which we might expect to have a small contribution to the particle flux, has a decay rate 100 times as fast, which our equipment would not have been able to detect.

IV. RESULTS

Over a period of 12 days, 13,788 decay events were output by the TAC and recorded by the multichannel analyzer (MCA), where they were divided into 512 bins, each of width $\frac{1}{61.0\pm1.9}$ µs. During the same time, 740,306 start signals were sent to the TAC, giving an estimated background level of 1.6 counts per bin, based on the measured random stop signal rate reported in Section III-D. Cutting out the first 13 bins, where the MCA did not respond properly, 5499 bins of data remained for statistical analysis.

Theory predicts a distribution of decay events, as a function of decay time, of the form

$$N(t) = \frac{A}{\tau} e^{-t/\tau} + B$$

where we expect $B$ to be close to 1.6 per channel, as discussed above.

To extract the most probable such function $N(t)$ from the data, we seek to minimize the function

$$\chi^2 = \sum_i \left( \frac{n_i - N(t_i)}{\sigma_i} \right)^2$$

where the $n_i$ are the sampled values of the dependent variable, the $t_i$ are the values of the independent variable for which the $n_i$ are measured, and the $\sigma_i$ are the standard deviations of the parent distributions from which the $n_i$ are sampled. Since the experiment involved counting events, the appropriate parent distribution is the Poisson distribution, giving $\sigma_i = \sqrt{N(t_i)}$ [6].

Reviewer asks why did it not respond properly!
Muon lifetime fit. A weighted least-squares fit (described in text) returns a fitted lifetime $\tau = 2.024 \pm 0.043 \, \mu s$, background level $B = 2.5 \pm 0.4$ events/bin, and $\chi^2 = 463$ for 496 d.o.f. (probability 85%).

To minimize $\chi^2$ we use the Levenberg-Marquardt method of non-linear least squares, as discussed and implemented by Press et al. [7]. Estimated parameters $a_\alpha$ are given with 68% confidence intervals $a_\alpha \pm \sigma_\alpha$, where the $\sigma_\alpha = \left(C_{\alpha\alpha}\right)^{1/2}$ are the square roots of the diagonal elements of the formal covariance matrix $C_{\alpha\beta}$ of the chi-square fit. These confidence intervals are appropriate only in the limit that the Poisson parent distribution can be approximated by a Gaussian.

With 499 data points and three free parameters, the fit converged to values

$$A = 253.8 \pm 3.8$$

$$\tau = 2.024 \pm 0.043 \, \mu s$$

$$B = 2.5 \pm 0.4 \, \text{counts/bin}$$

$$\chi^2 = 463$$

with $\chi^2$ probability value $P(\chi^2 = 463; \nu = 496) = 85\%$ of observing a randomly selected sample with a larger $\chi^2$. Fig. 4 shows the data and fitted curve.
As a check on the least-squares fit procedure, we also perform a binned maximum-likelihood fit. (See Fig. 5.) We aim to maximize the product $L = \prod_i N(t_i) \frac{n_i e^{-N(t_i)}}{n_i!}$ as a function of $A$, $\tau$, and $B$, on which $N(t)$ depends. In practice, we minimize the sum $-\log \delta L = \sum_i (N(t_i) - n_i \log N(t_i))$, where we have dropped the $\log(n_i!)$ term, which is independent of the fit parameters. An advantage of the binned likelihood fit is that it properly handles Poisson statistics, even when the expected number of events per bin is very small. The $\chi^2$ fit, by contrast, uses the Gaussian approximation to the Poisson distribution. A disadvantage to the likelihood fit is that no goodness-of-fit quantity is provided, so one typically calculates $\chi^2$ in any case. In the present analysis, an additional annoyance of using the likelihood fit is that we have not yet implemented the extraction of a fit covariance matrix in our likelihood fit software. So we use the likelihood fit merely as a check on the central values obtained with the $\chi^2$ fit. These central values are $A = 254.3$, $\tau = 2.032 \mu s$, and $B = 2.0$ counts/bin, in good agreement with the $\chi^2$ fit results.

The fitted muon lifetime is $2.024 \pm 0.043 \mu s$ (statistical). To account for the systematic bias due to muon-capture estimated in Section II, we shift the fitted lifetime $+0.02 \pm 0.02 \mu s$. The time calibration uncertainty derived in Section III-A is $3.1\%$, or $0.063 \mu s$. Combining the uncertainties in quadrature yields $0.043 \oplus 0.02 \oplus 0.063 \mu s = 0.079 \mu s$. Hence, we report a measured muon
lifetime $2.044 \pm 0.08 \, \mu s$, which is $1.9 \sigma$ ($P \approx 5\%$) below the world-average value $2.197 \, \mu s$.

V. Conclusion

The decay time we measure is marginally consistent $P \approx 5\%$ with the accepted value $2.197 \, \mu s$, though it does seem low. One reason for our low measured value may be an underestimate of the effect of $\mu^-$ capture in matter. A potential improvement to this experiment that would directly quantify the muon-capture effect would be to use a magnetic field to distinguish $\mu^+$ from $\mu^-$ in incident cosmic ray muons. In addition, the dominant systematic uncertainty — time calibration — could be reduced to negligible size by repeating the calibration with a modern digital oscilloscope.

References

[1] Particle Properties Data Booklet (Particle Data Group, 2010).

Appendix A

Table I shows the data used to determine the relationship between two-pulse delay and MCA channel number. The third column shows the number of delayed pulses recorded in a 20-second interval, as a check that efficiency is uniform across the range of MCA channels. Table II shows the raw MCA data used in the muon lifetime fit. The first 13 channels were excluded from the fit.

Listing 1 shows the Python source code (using SciPy/PyLab libraries) used to make the data graphs in this report.
<table>
<thead>
<tr>
<th>Delay (µs)</th>
<th>MCA channel</th>
<th># pulses recorded</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>22</td>
<td>201881</td>
</tr>
<tr>
<td>1.0</td>
<td>53</td>
<td>207436</td>
</tr>
<tr>
<td>2.0</td>
<td>114</td>
<td>203132</td>
</tr>
<tr>
<td>3.0</td>
<td>175</td>
<td>203300</td>
</tr>
<tr>
<td>4.0</td>
<td>236</td>
<td>203800</td>
</tr>
<tr>
<td>5.0</td>
<td>296</td>
<td>200600</td>
</tr>
<tr>
<td>6.0</td>
<td>356</td>
<td>201800</td>
</tr>
<tr>
<td>7.0</td>
<td>417</td>
<td>207600</td>
</tr>
<tr>
<td>8.0</td>
<td>480</td>
<td>200000</td>
</tr>
</tbody>
</table>

**TABLE I**
MCA time calibration and efficiency check

**TABLE II**
Raw MCA data used in muon lifetime fit.
from numpy import *
from pylab import *
from scipy import optimize

def fitfunc_fig2(p, x):
    return p[0]+x*p[1]

def errfunc_fig2(p, x, y):
    sigma_y = 1.0/sqrt(12.0)
    sigma_x = 0.02*x
    dydx = p[1]
    fig2_sigmay = hypot(sigma_y, sigma_x*dydx)
    return (fitfunc_fig2(p, x) - y)/fig2_sigmay

def figure2():
    usec = array([0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0])
    chnl = array([ 22, 53, 114, 175, 236, 296, 356, 417, 480])
    p0 = [0.0, 0.0]
    p1, cov, info, msg, status = optimize.leastsq(errfunc_fig2, p0[:], args=(usec, chnl), full_output=True)
    print "fitted parameters = ", p1
    print "covariance matrix = ", cov
    offset = p1[0]
    sigma_offset = sqrt(cov[0,0])
    slope = p1[1]
    sigma_slope = sqrt(cov[1,1])
    xpoints = linspace(0.0, 8.5, num=100)
    ypoints = fitfunc_fig2(p1, xpoints)
    plot(xpoints, ypoints, 'r-', label='linear fit')
    plot(usec, chnl, 'bo', label='measured data')
    errorbar(usec, chnl, yerr=fig2_sigmay, fmt='bo', capsize=5)
    xlabel("time delay \[\mu s\]", fontsize=14)
    ylabel("MCA channel number", fontsize=14)
    legend(loc='upper left')
    axis([0, 8.5, 0, 500])
    grid(True)
    savefig("muon_figure2.pdf")
def fitfunc_fig3(p, x):
    return p[0]*x+p[1]

# kludge: store error bars in a global variable so that they can be computed
# in function called by fitter and then re-used in graphing the data below
fig3_sigmay = []

def errfunc_fig3(p, x, y):
    # use quadrature sum of sqrt(n) for counting statistics
    # and n*0.25/20 for 0.25 second accuracy in measuring 20-second
    # interval using stopwatch
    global fig3_sigmay
    fig3_sigmay = hypot(sqrt(y),y*0.25/20)
    return (fitfunc_fig3(p, x) - y)/fig3_sigmay

def figure3():
    # data
    data = array([(0.5, 201881),
                  (1.0, 207436),
                  (2.0, 203132),
                  (3.0, 203300),
                  (4.0, 203800),
                  (5.0, 200600),
                  (6.0, 201800),
                  (7.0, 207600),
                  (8.0, 200000)])
    usec = data[:,0]
    resp = data[:,1]
    # clear out any previous graph, to start new plot
    clf()
    # initial guess for fit parameters (offset, slope)
    p0 = [0.0, 0.0]
    # invoke the fitter and print out the results
    p1, cov, inf, msg, status = optimize.leastsq(errfunc_fig3, p0[:], args=(usec, resp), full_output=True)
    print "fitted parameters = ", p1
    print "status code = ", status
    assert(status in [1,2,3,4])
    print "covariance matrix = 
    print cov
    offset = p1[0]
    sigma_offset = sqrt(cov[0,0])
    print "constant term = %.2f +/- %.2f counts"%(offset, sigma_offset)
    slope = p1[1]
    sigma_slope = sqrt(cov[1,1])
    print "linear term = %.2f +/- %.2f counts/microsecond"%(slope, sigma_slope)
    # evaluate the fit function at 100 evenly spaced points
    xp0ints = linspace(0.0, 8.5, num=100)
    ypoints = fitfunc_fig3(p1, xp0ints)
    plot(xpoints, ypoints, 'r-', label='linear fit')
    yavg = mean(resp)
    plot([0.0, 8.5], [yavg, yavg], 'g-', label='constant response')
    # plot the data points with error bars
    errorbar(usec, resp, yerr=fig3_sigmay,
             fmt='bo', capsize=5, label='measured data')
    xlabel("time delay [\mu s]", fontsize=14)
    ylabel("response [counts]", fontsize=14)
    legend(loc='upper left')
    # set the axis limits to values that frame the data nicely
    axis((0, 8.5, 190000, 220000))
    # for this plot, graph-paper-like grid lines may be helpful
    grid(True)
    # save to pdf file
    savefig("muon_figure3.pdf")
def load_muon_data():
    fnam = '/data2/ashmansk/berkeley/class/p191/lab2/fit_orig.dat'
    counts = []
    for line in open(fnam).readlines():
        words = line.strip().split(',')
        nums = [int(x) for x in words]
        assert(nums[0]==len(counts))
        counts += nums[1:]
        assert(len(counts)==512 or len(counts)==515)
    counts = counts[:512]
    return counts

def fitfunc_fig4(p, t):
    ampl = p[0]
    tau = p[1]
    bkg = p[2]
    tmax = 2.5
    norm = tau*(1-exp(-tmax/tau))
    f = (ampl/norm)*exp(-t/tau)+bkg
    return f

def errfunc_fig4(p, t, counts):
    sigma_counts = sqrt(fitfunc_fig4(p, t))
    return (fitfunc_fig4(p, t) - counts)/sigma_counts

def figure4():
    counts = load_muon_data()
    dt_binwidth = 1.0/60.99
    t = dt_binwidth*arange(512)
    fitpar, cov, info, msg, status = optimize.leastsq(func=errfunc_fig4, x0=(200.0,2.0,1.0),
                                                   args=(t[13:],counts[13:]), full_output=True)
    set_printoptions(precision=3, suppress=True)
    print "fitted parameters = ", fitpar
    print "status code = ", status
    assert(status in [1,2,3,4])
    print "covariance matrix = 
    print cov
    print "correlation matrix ="
    cor = array(cov)
    for i in range(len(cor)):
        for j in range(len(cor)):
            cor[i,j] /= sqrt(cov[i][i]*cov[j][j])
    print cor
    print "parameter uncertainties (1 sigma) ="
    fitsigma = sqrt(diag(cov))
    print fitsigma
    chiq = sum(errfunc_fig4(fitpar, t[13:], counts[13:])**2)
    ndof = len(t[13:])-len(fitpar)
    print "chi^2 = %.1f / %d d.o.f."%(chisq, ndof)
    clf()
    plot(t, counts, 'b.', label='MCA counts per channel')
    plot(t[:13], counts[:13], 'ro', label='rejected points')
    plot(t, fitfunc_fig4(fitpar, t, 'g-', label='fitted curve', linewidth=2)
    axis([0,9, -1, 140])
    xlabel("time [\mu s]", fontsize=14)
    ylabel(r"counts / %.3f \mu s"%(dt_binwidth), fontsize=14)
    text(4.7, 100.0, r'\textbf{f(t) = (A/\tau) e^{t/\tau} + B}', fontsize=16)
    label = r'$A = %.1f \pm %.1f$'%(fitpar[0],fitsigma[0])
    text(5.0, 90.0, label, fontsize=16)
    label = r'$\tau = %.3f \pm %.3f \mu s$'%(fitpar[1],fitsigma[1])
    text(5.1, 79.0, label, fontsize=16)
    label = r'$B = %.1f \pm %.1f\text{ counts/bin}$'%(fitpar[2],fitsigma[2])
    text(5.0, 70.0, label, fontsize=16)
    savefig("muon_figure4.pdf")
    print fitpar
def fitfunc_fig5(p, t):
    ampl = p[0]
    tau = p[1]
    bkg = p[2]
    f = (ampl/tau)*exp(-t/tau)+bkg
    return f

def objfunc_fig5(p, *args):
    counts = args[0]
    t = args[1]
    assert(len(args)==2)
    assert(len(t)==len(counts))
    n = counts
    mu = fitfunc_fig5(p, t)
    minus_logL = sum(mu - n*log(mu))
    return minus_logL

def figure5():
    counts = load_muon_data()
    dt_binwidth = 1.0/60.99
    t = dt_binwidth*arange(512)
    fitpar = optimize.fmin(
        func=objfunc_fig5, x0=(200.0,2.0,1.0), args=(counts[13:],t[13:]))
    clf()
    plot(t, counts, 'b.', label='MCA counts per channel')
    plot(t[:13], counts[:13], 'ro', label='rejected points')
    plot(t, fitfunc_fig5(fitpar, t), 'g-', label='fitted curve', linewidth=2)
    axis([0,9, -1, 140])
    xlabel("time \[\mu s\]", fontsize=14)
    ylabel(r"counts / %.3f $\mu$s"%(dt_binwidth), fontsize=14)
    text(4.5, 100.0, r'$f(t) = (A/\tau) e^{-t/\tau} + B$', fontsize=16)
    label = r'$A = %.1f \pm x$'%(fitpar[0])
    text(5.0, 90.0, label, fontsize=16)
    label = r'$\tau = %.3f \pm x \mu s$'%(fitpar[1])
    text(5.1, 79.0, label, fontsize=16)
    label = r'$B = %.1f \pm x$ counts/bin'%(fitpar[2])
    text(5.0, 70.0, label, fontsize=16)
    legend()
    savefig("muon_figure5.pdf")
    print fitpar

if __name__=="__main__":
    main()