

# Analysis of Pion Production Near Threshold for Carbon and Helium

Michael Barnett

*Subatomic Physics Internal Report SPIR-127*

*Dept. of Physics and Engineering Physics*

*University of Saskatchewan*

*August 30, 2004*

---

## Abstract

*This report details the methods used to extract the  $p_3$  reduced amplitude and its first order, energy dependent correction factor  $\alpha$  for pion photoproduction from nuclei, specifically helium and carbon. This was done by examining the pion production cross section at energies within about 25 MeV of threshold, using data taken at the Saskatchewan Accelerator Laboratory (SAL). At these energies,  $p_3$  should exhibit an energy dependence which is approximately linear, of the form  $p_3 \rightarrow p_3(1 + \alpha(E - E_{thr}))$ , where  $E_{thr}$  is the pion production threshold energy. As well as determining  $p_3$  and  $\alpha$ , it was also necessary to determine the shifts in the energy spectrum of each run of data used, corresponding to calibrations of the experimental setup. Using a multi-parameter optimization program, the values for  $p_3$  and  $\alpha$  for Helium were determined to be  $10.82 \pm 0.10$  and  $(3.44 \pm 0.45) \times 10^{-3}/\text{MeV}$  respectively, with  $p_3$  measured in units of  $10^{-3}/m_\pi$ . The results for carbon were  $p_3 = 10.27 \pm 0.09$  and  $\alpha = (2.80 \pm 0.44) \times 10^{-3}$ , which are significantly different from previous results of  $p_3 = 11.24 \pm 0.15$ ,  $\alpha = (1.20 \pm 0.36) \times 10^{-3}$ . This difference was shown to be caused by an improvement in the methods used to filter out background data.*

---

# 1 Introduction and Background

The object of this research is to study the cross section for pion production near threshold for different nuclei to observe the effects, if any, of nuclear size on the  $p_3$  wave amplitude, a parameter in the cross section model. This will provide a test of Quantum Chromodynamics at nuclear energy scales, which are far more accessible than the quark scales it is typically associated with.

The experimental data was collected at the Saskatchewan Accelerator Laboratory at the University of Saskatchewan, using the Igloo detector to measure total pion production cross sections. The analysis of this data to extract the  $p_3$  and  $\alpha$  parameters is an extension of that used by Ru Igarashi and Jack Bergstrom of the University of Saskatchewan.

## 1.1 Pion Production Process

Pions were produced in this experiment by bombarding target nuclei with gamma photons. The produced pions quickly decayed into two photons as shown in Fig. 1, which could be detected by lead glass photon detectors.

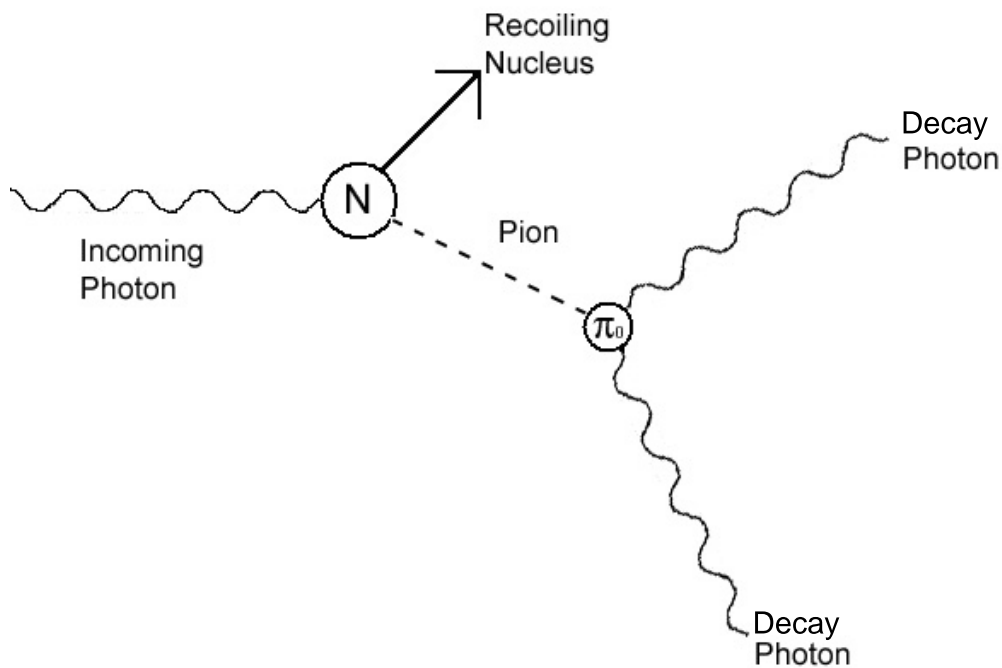


Figure 1: Pion Production and Decay Process

## 1.2 Original SAL Experiment

The Igloo detector consisted of lead glass detectors arranged around a target. In its closed mode, the detectors cover 83% of the  $4\pi$  solid angle around the target (Fig. 2). This arrangement was useful for measuring total cross sections because of its large acceptance angle.

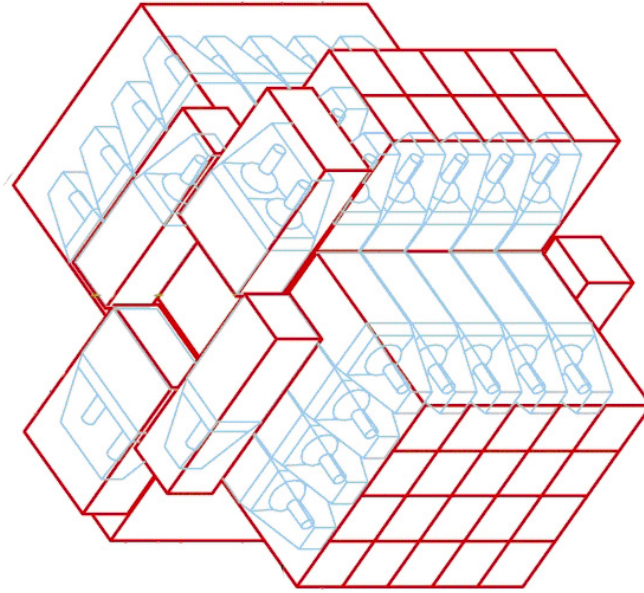


Figure 2: Igloo Detector Schematic

The gamma rays that struck the target were generated by a bremsstrahlung process. Electrons were accelerated and struck a thin aluminum foil target. Some of the electrons shed energy through bremsstrahlung emission, and the relativistic nature of the electrons kept the photon beam well collimated. The scattered electrons then passed into a magnetic field region, and their energy was measured by observing the degree of deflection each electron experienced (Fig. 3). A pion event was said to have occurred when an electron was detected at the same time as the two photons produced by a decaying pion. By matching the electron to the produced photons, the energy of the photon that struck the target could be determined for that event.

The energy shifts that were applied to the data are due to this geometry. The magnetic field strength and the electron beam energy were not the same for all runs, and so two electrons of the same energy could strike different detectors during different experimental runs. The energy shift calibration is a way to account for this effect.

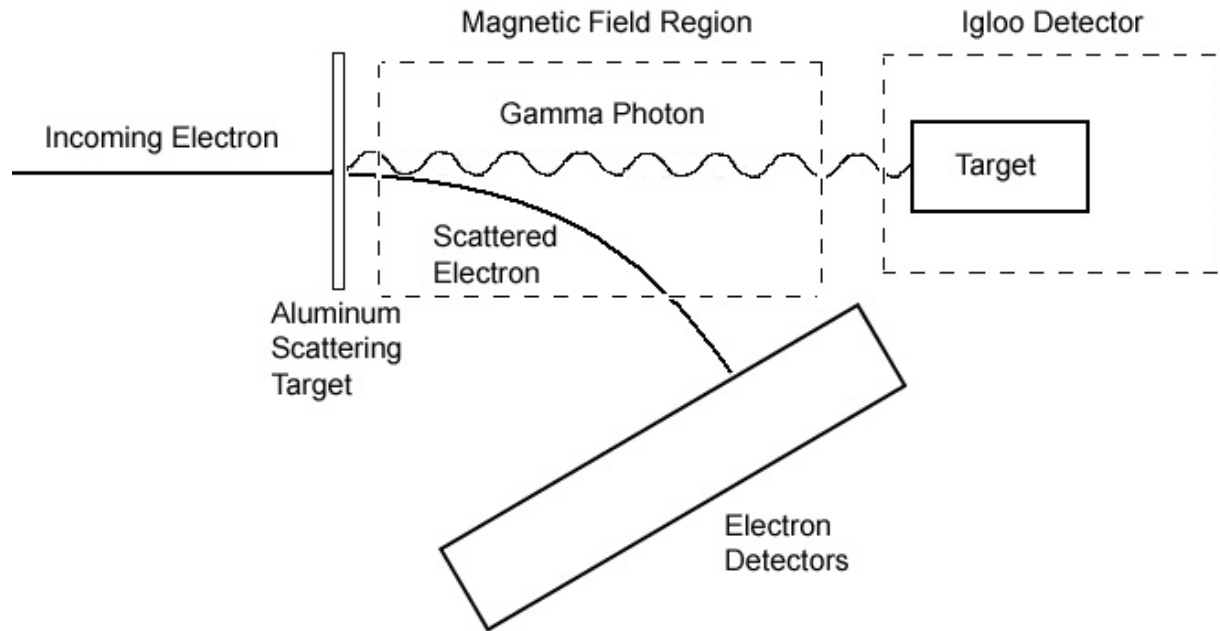


Figure 3: Schematic of Beamline at SAL

## 2 Method of Analysis

The purpose of this analysis was to determine the set of parameters that best fit the experimentally measured cross sections to the theoretical model. This was done in several stages, using a series of coordinated software programs. The first step was to create the theoretical cross sections using *dwa\_mike*, and to develop the experimental cross sections from the raw yield data using *yield\_to\_xs*. Next, two passes were made through the data sets to match up the two sets of cross sections: one pass included only those points within about 10 MeV of threshold, and a second included all data points (up to about 25 MeV above threshold). For the first pass,  $\alpha$  was assumed to be negligible; i.e.  $p_3$  was assumed constant. With this in mind, a fit was applied to the data in order to find the shift in the energy spectrum of each run's data needed to minimize  $\chi^2_\nu$ . Using these shift values for the energy spectra, the second pass was made to find the  $p_3$  and  $\alpha$  values that would minimize  $\chi^2_\nu$  for the complete data set. The  $\chi^2$ -space about the optimal parameter choices was then probed to determine the error in each parameter's value.

The following describes only the high level structure of the algorithms used in the analysis; for a better look into the algorithms used, refer to the software documentation in the Appendices, or the significant internal documentation within the programs' source files.

### 2.1 Generation of Experimental and Theoretical Cross Sections

In order to compare the experimental and theoretical cross sections - and thus find the parameters to create a best fit - the two sets of cross sections first needed to be generated. Experimental cross sections were generated from yield files using the program *yield\_to\_xs*, while theoretical cross

sections were generated in part using code ported into C from the original Fortran source, written by Jack Bergstrom.

The yield data were stored as effective event count rates at each energy probed by the experiment. Two types of yield files existed for use: files of the form “yield.<run>”, which used an energy-dependent physical background subtraction in their creation, and files of the form “yield.flat.<run>”, which used an energy-independent flat background subtraction. The physical background subtraction approach was more physically meaningful, and was used exclusively in the actual analysis. The flat background data was useful for comparison to previous data when testing for program correctness.

To convert yields into cross sections, the raw counts had to be converted into events per square area, using the conversion equation:

$$\sigma = Y \frac{m_{mol}}{\rho_t N_A} \frac{1}{\epsilon(E)} \quad (1)$$

where  $\sigma$  is the cross section,  $Y$  is the yield or event count,  $\rho_t$  is the thickness of the target multiplied by its density,  $N_A$  is Avagadro’s number,  $m_{mol}$  is the molar mass of the target material and  $\epsilon(E)$  is the efficiency of the detection equipment, which is energy dependent. The form of  $\epsilon(E)$  was chosen to be a fourth order polynomial in energy, with the coefficients found by applying a least squares fit to the data in the file “eff.<target>.phys.dat” using the *polyfit* program, where <target> in this case was “he” for helium. These coefficients were stored in the file “efficiency\_file”, and are listed below for the case of a helium target.

$$\begin{aligned} a_0 &= 3.4221 \times 10^2 \\ a_1 &= -8.9116 \times 10^2 \\ a_2 &= 8.7275 \times 10^2 \\ a_3 &= -3.8005 \times 10^2 \\ a_4 &= 6.2074 \times 10^2 \end{aligned}$$

where

$$\epsilon(E) = a_0 + a_1 E + a_2 E^2 + a_3 E^3 + a_4 E^4 \quad (2)$$

Once the file “efficiency\_file” was generated and the appropriate parameters were passed into the *yield\_to\_xs* program, the requested yield file was converted into a cross section file of the form “xs.<run>” or “xs.flat.<run>” depending on the type of yield file used. External documentation for the *yield\_to\_xs* program is found in Appendix C.

The theoretical cross sections were calculated in two stages to greatly shorten the running time of the program. A lookup table was first generated of the theoretical cross sections at every useable energy, with a factor of  $(p_3)^2$  removed. In this way, many choices for the value of  $p_3$  and  $\alpha$  could be tried without the need to repeat the intensive calculations needed to compute the rest of the cross section. The useable energies were from 120 MeV to 170 MeV, in 0.01 MeV increments. The increment size was as chosen since the energies reported in the yield files (and thus in the cross section files) were only listed to 0.01 MeV accuracy. In this way, lookup tables of cross sections at 5001 energy values were created and stored in the file “xs.theor.<target>”, where <target>

was “he” for helium in this case. These lookup tables take a relatively long time to generate, but this only needs to be done once for any given material (for example, the carbon and oxygen tables each took about 2 hours to produce and the helium table took about 10 seconds). The tables are produced by the program *dwba\_mike* if the appropriate option is selected during the interactive portion of that program. This option only needs to be used once, and exists to make subsequent trials with the same material much faster.

## 2.2 Optimization of Parameter Subsets

Once the experimental cross sections and the theoretical lookup tables are prepared, the actual parameter optimization was handled by *dwba\_mike*, whose input parameters were managed by *mikefit.sh*. The first pass through the data optimized the energy spectrum shifts for each run and a temporary value for  $p_3$  used to bind the data sets together. The second pass optimized only  $p_3$  and  $\alpha$ , with the shifts left fixed to the values determined in the first pass through the data.

Each parameter to be optimized in a given pass was defined by a search region and resolution. The search range was specified by a central value and the width of the search region; for example a central value of 10 and a search range of 5.0 for  $p_3$  would have searched all  $p_3$  values from 7.5 to 12.5. The resolution was specified by the precision value, which corresponded to the number of decimal places desired in the parameters’ final values. For example, a precision of 2 created a search increment of 0.01. The search regions were probed exhaustively at the specified region; all possible combinations of all parameters were attempted. This slowed the program’s execution time, but greatly reduced the chance for error in finding the actual optimal values for the parameters. Note that the energy shifts were never probed beyond an accuracy of 0.01, since this was the limit of the energy resolution in the original data sets. Also note that the search region for alpha is never specified directly; instead the parameter *pct* was used, corresponding to  $\alpha \times 2500$ , which was a carryover from earlier code written by Jack Bergstrom and Ru Igarashi. The program still reports final results in terms of  $\alpha$ , but this input quirk is important to note.

To optimize the shifts, three nested loops were employed. The outer loop iterated through all possible values for  $p_3$  within the specified range. The second loop iterated through each of the data sets, one per experimental run (four sets were used for helium, for example). The innermost loop iterated through all possible values of the energy spectrum shift for the current file. Once a single file had attempted all possible shift values, the value which yielded the lowest  $\chi^2$  value for that file (at the current choice of  $p_3$ ) was saved. Once each file had found its ideal shift value (again for the current value of  $p_3$ ), the lowest total  $\chi^2$  for all files at the current  $p_3$  value was calculated. This was repeated for all possible values of  $p_3$  until the minimum value for  $\chi^2$  was found. The shift values for this case were saved to a file to be reused in the second pass through the data, while the optimal  $p_3$  value was discarded.

In order to apply energy shifts, the energies listed in the experimental cross section data sets were all shifted by the specified constant amount. Thus a positive shift corresponds to shifting the plot of cross section vs. energy to the right, while a negative shift moves the cross sections to the left. Once the spectrum was shifted, a mirror spectrum of theoretical values was created by looking up each new energy value in the lookup tables and noting the corresponding cross section value. The theoretical cross section values were then multiplied by the factor  $p_3^2[1 + \alpha(E - E_{thr})]^2$ , noting that  $\alpha$  was set to zero for the first pass through the data. The two cross sections, experimental and theoretical, were then compared point by point to determine the  $\chi^2$  value for their agreement.

The second pass through the data optimized only  $p_3$  and  $\alpha$ . The optimal shift values were loaded from the file that they were saved to at the end of the first pass, and were not modified during the second pass. The same procedure was again used to find the ideal combination of  $p_3$  and  $\alpha$  using two nested loops.

The significant addition to the procedure for the second pass was the determination of the errors associated with  $p_3$ ,  $\alpha$  and the shift values. This was done by probing values of the parameters around their optimal values and examining the effect this had on  $\chi^2$ . A  $1 - \sigma$  error bound is associated with a  $\chi^2$  value one greater than the minimal value, so the parameters were adjusted until this criterion was met. The values of the parameters satisfying  $\chi^2 = \chi_{\min}^2 + 1.0$  ideally form an  $n$ -dimensional ellipsoid where  $n$  is the number of free parameters. Because of this, finding the extremal points of the ellipsoid takes a large amount of computation ( the process has a timing analysis of  $\mathcal{O}(2^n n!)$ ). To keep this problem manageable, the shift parameters were assumed to be totally independent of each other, so that the ellipsoid was never more than 3-dimensional:  $p_3$ ,  $\alpha$  and one of the shift parameters.

### 2.3 Complete Algorithm

The program is initiated by executing the script *mike\_fullfit.sh* with the appropriate arguments: the first is the target type to be analyzed (e.g. “he” for helium), the second is the type of background subtraction used to generate the yield files (“flat” or “physical”, with “physical” being better and “flat” supported only to compare results with older, less correct results) and the energy cutoff to be used in the first pass through the data to determine the shifts, in MeV.

*mike\_fullfit.sh* calls the script *mikefit.sh* twice. The first call is with the specified energy cutoff, and is used to determine the best choices for the energy shifts. The second pass loads these shifts from a file and uses the full data sets in the fit to determine  $p_3$  and alpha.

*mikefit.sh* used the two C programs written for this software package; *yield\_to\_xs*, which converted the experimental yields into cross sections and *dwa\_mike*, which actually performed the variable optimization necessary to fit the data to the theoretical model.

### 3 Analysis of Helium and Carbon Data

The program was verified through analysis of carbon data, which had been previously analyzed and published in [1]. The program was then passed new data, the experimental data for a helium target. The primary parameters that had to be adjusted by the user of the software were the cutoff energy for the first pass through the data set, the search regions for probing the values of  $p_3$ ,  $\alpha$  and the energy shifts, and whether to use a physical or flat background subtraction. Some of the approaches to fitting the data which were not successful are also discussed below.

#### 3.1 Choosing the Cutoff Energy for the First Pass

To properly determine the values for the energy shifts during the first pass through the data, the cutoff energy had to be chosen carefully. If the cutoff was too low, too few data points would remain and the data would become relatively noisy. If too many data points were used, the assumption that  $\alpha$  was approximately zero would no longer be valid. Ultimately, the condition to determine the optimal cutoff energy was decided to be the cutoff which yielded the lowest reduced  $\chi^2$  fit in the second pass through the data. This should correspond to the best possible choice for the shift parameters. Cutoff energies between 5 and 11 MeV above threshold were attempted in 1 MeV increments.

#### 3.2 Flat or Physical Backgrounds

In the raw yield files, data from the SAL experiments had already been filtered to remove extraneous background events through a series of cuts and background subtractions. The cuts removed unphysical pairs of detected photons by noting that the relative angle between them had to be less than 180 degrees to conserve momentum.

The background subtraction was originally done by averaging all of the count rates below threshold (which should be zero by definition), and subtracting this amount from the data points above threshold. This was known as the flat background subtraction, since it was the same for all energies.

A more physically meaningful background subtraction was slightly stricter, removing additional counts at higher energies. This method was done by Ru Igarashi, and relied on the fact that at higher energies the angle between the produced photons should decrease due to the relativistic speed of the produced pion. The data at low energies was examined to see how many events would meet this narrower angular requirement, and an energy dependent background subtraction function was developed. This was known as the physical background subtraction because of its more realistic physical requirements.

Because more events were removed in this case, the value for the  $p_3$  amplitude was lower for the physical background subtraction than it was for the flat background subtraction. The  $\chi^2$  values for the fits were also about 20% lower for physical background subtractions than for flat background subtractions, which demonstrated their effectiveness in isolating the events which more likely corresponded to pion production events.



### 3.3 Unsuccessful Fitting Attempts

Several methods were attempted to determine the values for  $p_3$ ,  $\alpha$  and the energy shifts, but most were not successful. The most prominent failed attempts are discussed here, along with mention of how they led to more effective methods used later.

The initial plan for fitting the data to the model was to simultaneously fit  $p_3$ ,  $\alpha$  and all of the energy shift parameters, in a single global search. This was very time intensive, but more importantly, it did not produce results which agreed with earlier findings for carbon. It is believed that there were too many degrees of freedom in this fit, and the fit it arrived at was not that physically meaningful. This led to the idea of first finding only the shift parameters by setting  $\alpha$  to zero and then searching over  $p_3$  and  $\alpha$  with the shift parameters fixed to those values. The two step method was able to very closely reproduce the earlier published data for carbon, when all other factors were properly repeated.

In applying the two step fit, two possibilities were considered for how to determine the shift parameters for the different data sets for a given target type. The first was to do a set of individual fits, by finding the shift value and a (temporary)  $p_3$  value for each data set. The alternative was to work with all of the data sets for a target type at the same time, and find a common temporary  $p_3$  value for them. The results using a group fit produced  $\chi^2$  values about three times lower than the single fits, so the single fit method was discarded from further use. It is likely that by having a common value for  $p_3$ , the separate data sets were brought into closer line with each other, reducing noise in the composite data set once the shifts were applied.

Before the lookup table for the partial theoretical cross sections described earlier was used, these cross sections were created and stored every time new shifts were applied to the energy spectra. This was the simplest way to construct the program, but it made the program to time intensive to probe the shift parameters down to the 0.01 MeV accuracy that it now does. It was also not possible to accurately determine the errors in the shift parameters, since the cross sections had to be recalculated every time new shift values were probed. Introducing the lookup table had an initial time penalty that was more than made up for after three or four executions of the software.

The calculation of the errors associated with each parameter was one of the more complicated tasks in developing this software. The original algorithm was to simply probe each variable individually, keeping all others constant. This underestimated the errors greatly, as there was some correlation between most of the parameters involved. The next attempt was to probe for several points on the ellipsoid forming the  $\chi^2 + 1$  boundary, and to then develop the ellipsoid's equation from this. This was difficult mathematically, and appeared to also underestimate the error ranges, likely due to the assumption that the boundary was perfectly ellipsoidal. The most effective method found so far, the one used in the final analysis of this report, was to probe every combination of variables, incrementing first one and then another variable to find the maximal deviation allowed from the optimal parameter choices.

## 4 Results

The results of the analysis of carbon and helium are best divided into two groups: results used to show agreement with earlier published values, and results using the new refined techniques. It is also interesting to look at just how effective the choice of fit parameters was by looking at the quality of fits produced with one parameter removed.

### 4.1 Effects of $\alpha$ and Energy Shifts

Although the  $p_3$  amplitude is the major feature being determined in these fits,  $\alpha$  and the energy shifts are vital to bring the experimental results into agreement with the model.

The effect of the energy shifts can be clearly seen by looking at its effect on a single data set. The data from the c1 experimental run required an energy shift of  $-1.46 \pm 0.04$  MeV, and the change in the cross section as a function of energy can be clearly seen in Fig. 4.

The parameter  $\alpha$  acted as a nonlinear correction to the  $p_3$  amplitude, and was able to greatly improve the quality of the fits to the experimental data. The fit below shows the data from the c1 experimental run only for clarity, although all of the carbon data were used in the fit. The same energy shift is used as in the final results for consistency. Using  $\alpha$  in the fit produced a  $p_3$  value of 10.27 with a reduced  $\chi^2$  of 1.18 for 310 data points or 303 degrees of freedom. Holding  $\alpha$  to zero gave a  $p_3$  amplitude of 10.79 (significantly higher than when  $\alpha$  was used) with a reduced  $\chi^2$  of 3.23 for 310 data points or 304 degrees of freedom. Clearly, the presence of  $\alpha$  greatly improves the quality of the fit, which can be seen visually in Fig. 5.

### 4.2 Connection to Earlier Results

Although the new results presented below appear quite different from the earlier results published in [1], these discrepancies can be totally accounted for. The most drastic change is the use of a physical background subtraction, which depressed the  $p_3$  amplitudes by about 10%. The other change was the refinement in finding the ideal energy shifts, which slightly altered the results as well. All connections to earlier results were demonstrated using the carbon target, since no prior results exist for the helium targets.

The earlier results were taken from the trial c032, corresponding to experiment 032 with a carbon target. The new data used trials c1 to c5, a different set of experimental runs.

The old published results of the c032 data used a flat background subtraction. The energy shift for the published data set, which we will call cfinal, was 0 MeV. The published values for  $p_3$  and  $\alpha$  were:

$$\begin{aligned} p_3 &= (11.24 \pm 0.15) \times 10^{-3}/m_\pi \\ \alpha &= (1.20 \pm 0.36) \times 10^{-3}/\text{MeV} \end{aligned}$$

To see if these results could be repeated with the developed software, the cfinal data was analyzed several times. The first pass forced the shifts to be zero, which should produce extremely similar values for  $p_3$  and  $\alpha$ . The results do not have error bars associated with them; this was due to the limits of the program, which does not currently handle error analysis for fixed shifts. The reduced  $\chi^2$  for this fit was 0.695 for 51 data points (49 degrees of freedom).

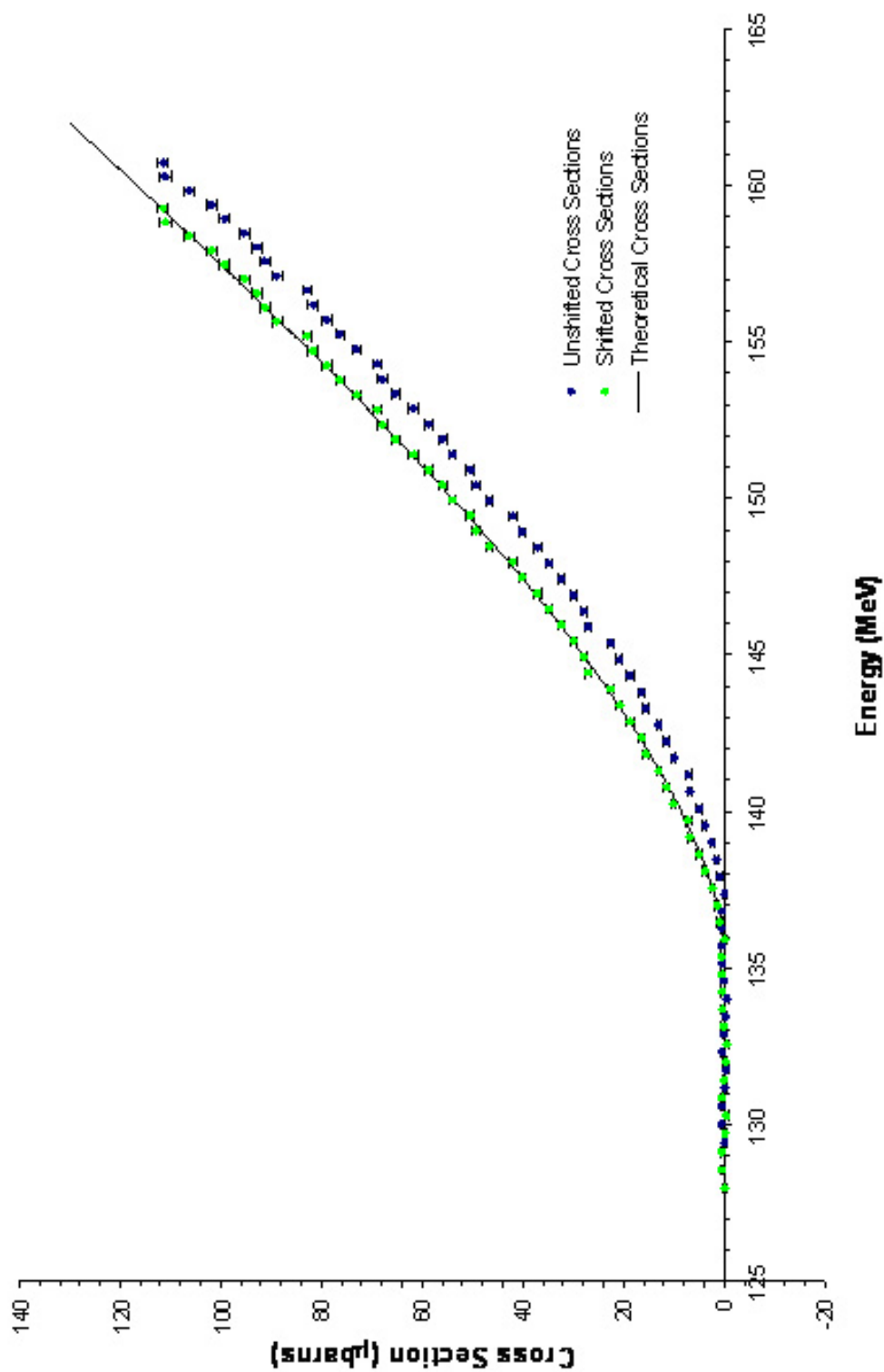


Figure 4: Cross Section vs. Energy Showing Effect of Energy Shift

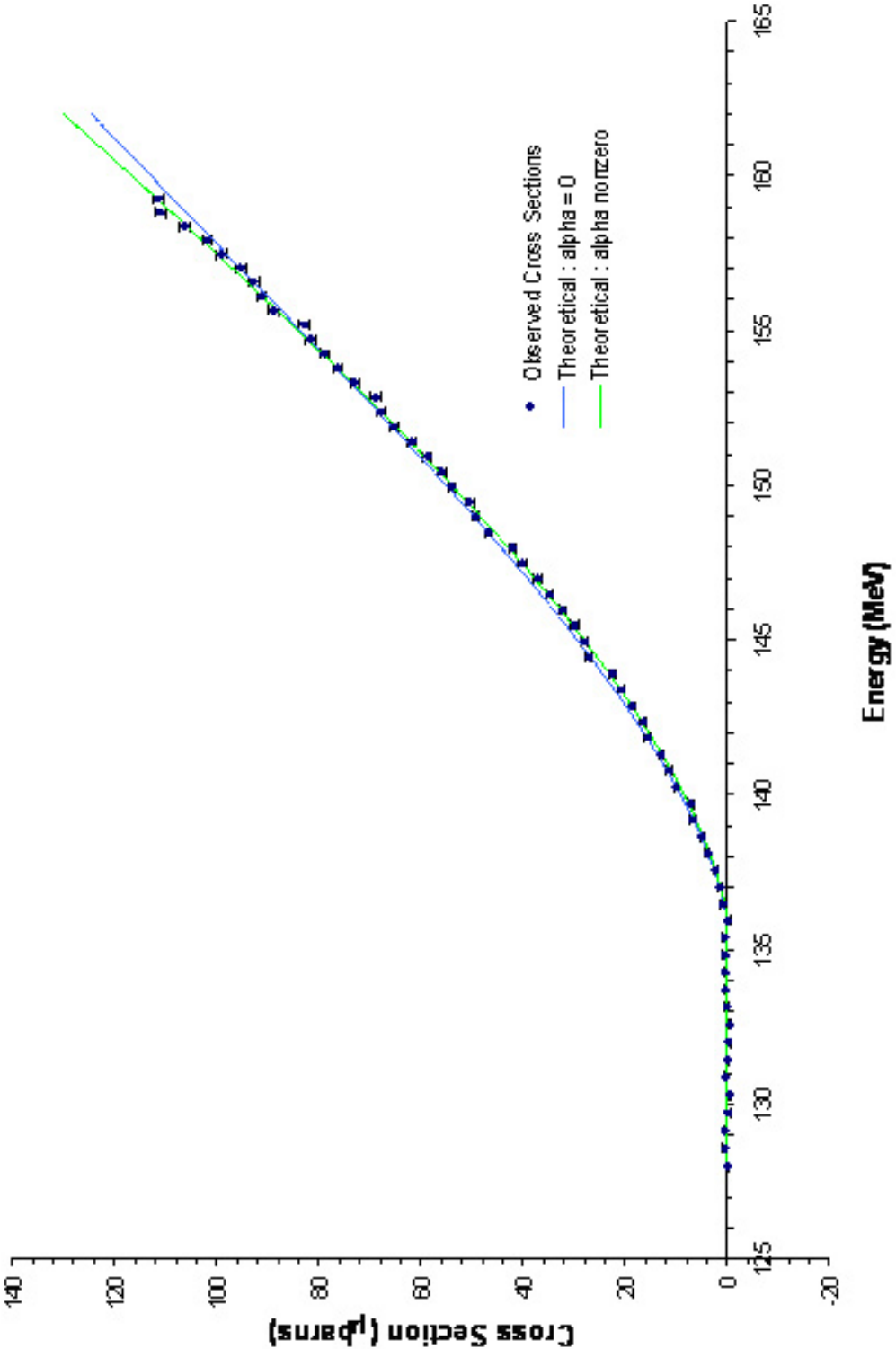


Figure 5: Cross section vs. Energy,  $\alpha = 0$

$$p_3 = 11.22 \times 10^{-3}/m_\pi$$

$$\alpha = 1.36 \times 10^{-3}/\text{MeV}$$

These values agree within error to the published results, which confirms that the program can at least handle  $p_3$  and  $\alpha$  optimization.

Permitting energy shifts on the published data produced interesting results. If the cfinal results were in fact optimal, then the shift should have naturally been zero. This was not the case, implying that the new algorithm is more sensitive to locating optimal values of parameters. The reduced  $\chi^2$  for this fit was 0.548 for 51 data points or 48 degrees of freedom.

$$p_3 = (10.91 \pm 0.26) \times 10^{-3}/m_\pi$$

$$\alpha = (2.40 \pm 1.00) \times 10^{-3}/\text{MeV}$$

$$\text{Shift} = +0.22 \pm 0.17\text{MeV}$$

These results show a slightly lower value for  $p_3$  than was earlier reported, as well as a larger correction term  $\alpha$ . Both values still agree with the published results within error, however.

By using the c032 data set with a flat background subtraction, which is a slightly less polished version of cfinal, the parameters were as follows (reduced  $\chi^2 = 0.611$  for this fit to 62 data points):

$$p_3 = (10.86 \pm 0.31) \times 10^{-3}/m_\pi$$

$$\alpha = (1.54 \pm 1.09) \times 10^{-3}/\text{MeV}$$

$$\text{Shift} = +0.43 \pm 0.21\text{MeV}$$

These values agree within error with the cfinal results with shifting permitted, suggesting the software correctly handles a flat background subtraction. By using the c032 data with a physical background subtraction, the value for  $p_3$  dropped about 10%. The reduced  $\chi^2$  for this fit was 0.528 for 62 data points or 59 degrees of freedom.

$$p_3 = (10.38 \pm 0.43) \times 10^{-3}/m_\pi$$

$$\alpha = (2.77 \pm 1.70) \times 10^{-3}/\text{MeV}$$

$$\text{Shift} = +0.27 \pm 0.32\text{MeV}$$

Comparing with the results from runs c1 to c5 below, we see that the old and new data sets produce consistent results for  $p_3$  and  $\alpha$ , once the physical background subtraction is used and the energy shifts are probed more carefully. This chain of reasoning reasonably shows that the old and new results are not in conflict with each other, but rather that the new results are a refinement of the old. These results also show that the use of the larger data sets of c1 to c5 greatly reduces the error bounds on the parameters, by roughly a factor of four for  $p_3$  and  $\alpha$ .

### 4.3 Results of Refined Analysis

The results for the carbon data from the trials c1 to c5 are as follows:

$$p_3 = (10.27 \pm 0.09) \times 10^{-3}/m_\pi$$

$$\alpha = (2.80 \pm 0.44) \times 10^{-3}/\text{MeV}$$

The energy shifts used for each trial are as follows (a negative sign implies that the listed energies were reduced by that amount; i.e. that the cross sections shifted to lower energies):

$$c1 : -1.46 \pm 0.04\text{MeV}$$

$$c2 : -0.90 \pm 0.07\text{MeV}$$

$$c3 : -0.61 \pm 0.06\text{MeV}$$

$$c4 : -1.29 \pm 0.05\text{MeV}$$

$$c5 : -1.05 \pm 0.03\text{MeV}$$

The quality of this fit was quite good, with a reduced  $\chi^2$  of 1.18 for 310 data points or 303 degrees of freedom. The final shifted cross sections and the fit from the model are shown in Fig. 6. Note that the error bars are of the order of size of the data point markers, and so are omitted for clarity.

The helium results were as follows for the data from the trials he1 to he4:

$$p_3 = (10.82 \pm 0.10) \times 10^{-3}/m_\pi$$

$$\alpha = (3.44 \pm 0.45) \times 10^{-3}/\text{MeV}$$

The corresponding energy shifts for each helium trial are:

$$he1 : -0.84 \pm 0.05\text{MeV}$$

$$he2 : -0.90 \pm 0.02\text{MeV}$$

$$he3 : -0.98 \pm 0.04\text{MeV}$$

$$he4 : -0.90 \pm 0.06\text{MeV}$$

The quality of this fit was extremely good, with a reduced  $\chi^2$  of 0.87 for 248 data points or 242 degrees of freedom. The final shifted cross sections and the fit from the model are shown in Fig. 7. Note that the error bars are of the order of size of the data point markers, and so are omitted for clarity.

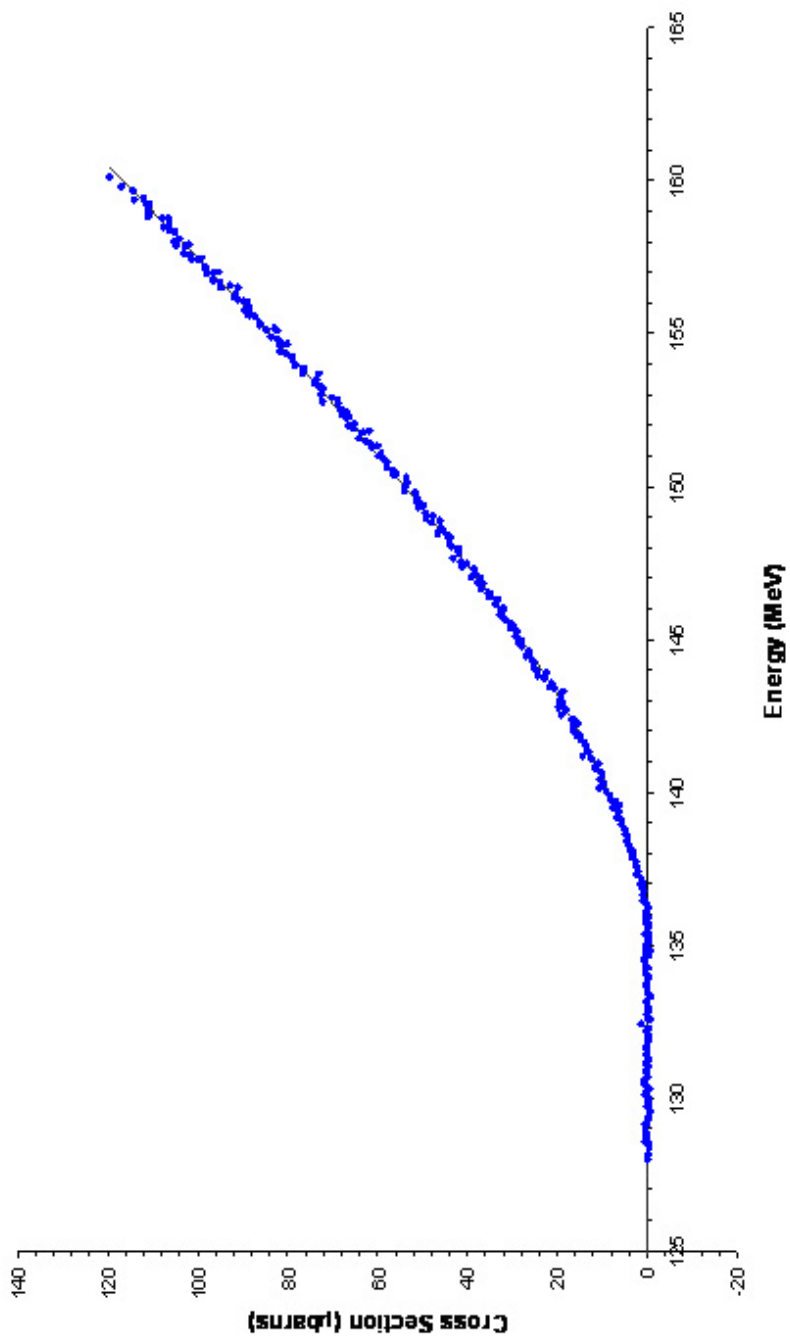


Figure 6: Cross Section vs. Energy for Carbon Target

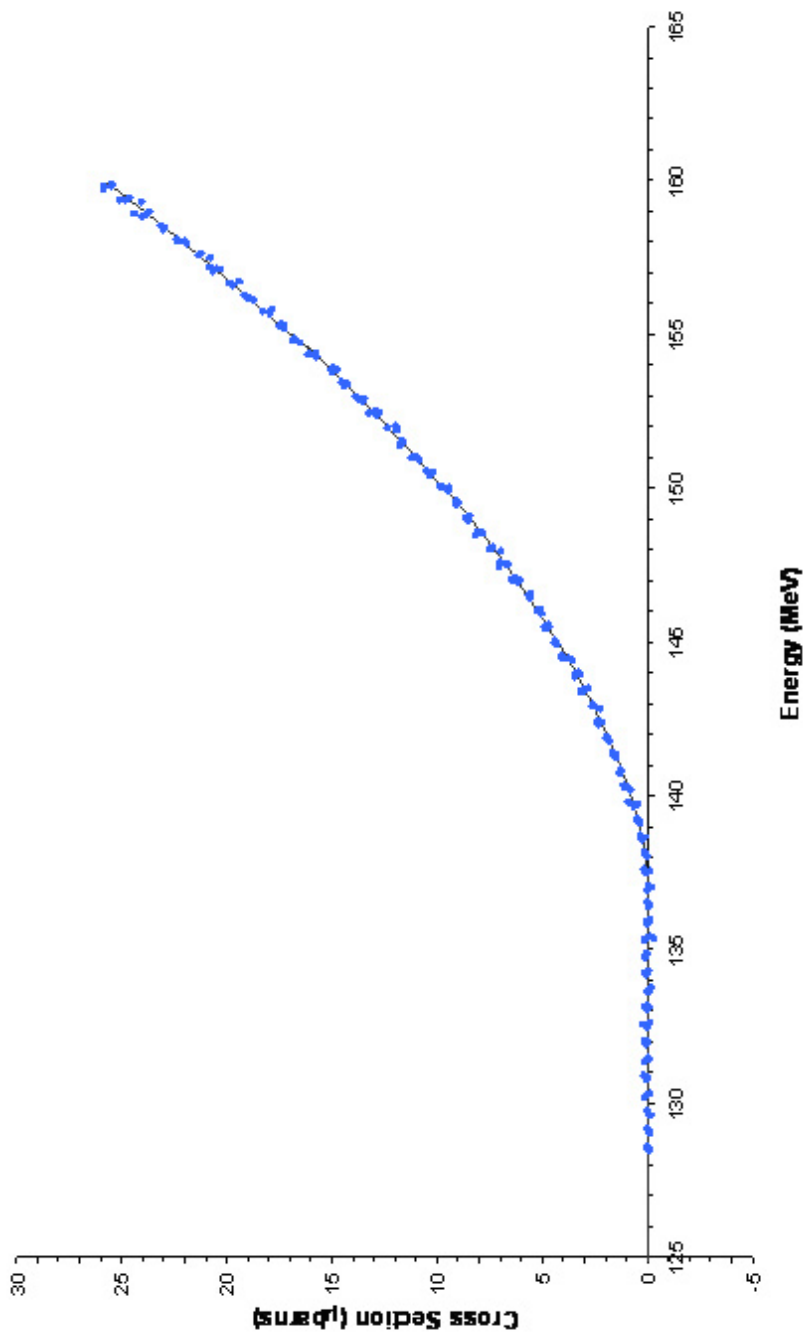


Figure 7: Cross Section vs. Energy for Helium Target



## 5 Conclusions and Discussion

The value for the  $p_3$  amplitudes for pion production from carbon and helium nuclei were shown to be significantly different, with the helium result being slightly larger. The theoretically predicted value for the  $p_3$  amplitude for pion production from a single nucleon is about 11.4, with no precise estimate of its error. This is higher than both results, although the helium result is closer.

It may be hypothesized that there is an inverse relationship between nuclear mass and the  $p_3$  amplitude near threshold. This would explain why the smaller helium nucleus had a  $p_3$  amplitude that was closer to the amplitude for a single nucleon. This hypothesis could be tested by analyzing the experimental data for lithium, nitrogen and oxygen to see whether this trend continues.

The correction term  $\alpha$  was larger than previously reported, which adds credibility to its use in making  $p_3$  weakly energy dependent near the pion production threshold. There is no theoretical prediction for  $\alpha$ , although it is expected to be nonzero even at these low energies.

The energy shifts applied were all of a similar order of magnitude, and none were larger than 1.5 MeV. This is reasonable, as the shifts should act only as a small calibration correction to the standard energy spectrum.

An analysis of the systematic errors present in these result has not been completed. Still, several factors have been observed that should be examined when an attempt to quantify the systematic errors is made.

One source of systematic error is the selection of the cutoff energy for the first pass through the data when the energy shifts are being determined. Varying this cutoff energy from 5 to 11 MeV above threshold produced a small but significant change in the values for  $p_3$  and  $\alpha$ . A first estimate of this effect would be an additional error of  $\pm 0.05$  on the reported value for the  $p_3$  amplitude.

Another possible source of error was the maximum energy allowed for the experimental cross sections. Since low energy approximations were used to simplify the form of the theoretical cross section, higher energy points introduced errors into the fit. Several fits were attempted early in the analysis of this data by cutting 1 to 10 of the highest energy data points. This had almost no effect on the value of  $p_3$ , so it was decided that this was not a significant source of systematic error that merited further examination.

Possibly the most important goal of the methods developed in this analysis was to make them easy to apply to new targets. Data exists from SAL experiments using lithium, nitrogen, oxygen and lead targets, and this code is designed to handle all of them. All that is needed is the target specific calculations needed to generate the theoretical cross sections; the optimization routines need virtually no modification.

Code for the theoretical cross sections for oxygen has in fact already been incorporated into the software. All that is needed is the energy dependent detector efficiency needed to convert the raw yields into useable cross sections.

## References

- [1] Bergstrom, Jack, Ru Igarashi, J. M. Vogt. *Measurement of the  $^{12}\text{C}(\gamma, \pi^0)$  Reaction Near Threshold*. Phys. Rev. C. **55** 2923 (1997).

## Appendix A

```
#  
# Michael Barnett  
#  
# August 3, 2004  
#
```

Documentation for Program mike\_fullfit.sh  
-----

Syntax:  
-----

```
sh mike_fullfit.sh <target_type> <"flat"/"physical"> <low_cutoff>
```

The script finds the p3, alpha and energy shift parameters for a set of files of a given target type. This is done by calling the script "mikefit.sh" twice: once with the <low\_cutoff> as the maximum allowed data point, in order to examine the linear region near threshold where alpha may be neglected in order to find the best shift parameters and a first guess for p3; the second pass uses the entirety of each data file to find p3 and alpha, with the shifts fixed to the values determined in the first pass.

Supported target types:

```
"c" - carbon: c1, c2, c3, c4, c5  
"he" - helium: he1,he2, he3, he4  
"w" - water: not yet fully supported
```

User's Guide  
-----

This script acts as the front end interface for all of the programs used in the determination of the cross section parameters. The only options for the user are the type of target being examined, the style of background subtraction (flat or physical) and the cutoff energy for the first pass through the data.

Typical cutoff energies for the first pass through the data are 5 to 9 MeV above threshold, with the exact cutoff left up to the user's discretion. Recall that the

threshold for carbon is 135.8 MeV and the threshold for helium is 137.4 MeV.

The other option for the user is to specify the type of background subtraction used in the generation of the yield files. A flat background subtraction assumes that the background is energy-independent, and corresponds to the files "yield.flat.<target>" (e.g. yield.flat.c1). A physical background subtraction is more meaningful, assuming some energy dependence in the background spectrum; these correspond to the files "yield.<target>" (e.g. yield.c1). The type of subtraction to be used is left to the user's discretion, although physical backgrounds should be used for the most accurate results.

#### Algorithm

-----

The first case statement in the program makes sure that the target type entered as an argument is supported by the software, and exits if the target is unsupported. The script mikefit.sh is then called, using the cutoff energy given in the argument list as the cutoff argument for mikefit.sh. The set of shift values produced by this call to mikefit.sh are then copied into the file list that will be used by mikefit.sh the next time it is called. The script mikefit.sh is called again, with no upper threshold given; this causes the script to do a fit on p3 and alpha while leaving the energy shifts with the values found on the first pass through the script.

#### Maintenance Manual

-----

The main concern for this script will likely be to extend it to work for other target types. This may be done by adding another option into the case statement. This should have exactly the same form as the options currently in the statement; its purpose is only to ensure that the target type is recognized by the software as now being supported.

#### Known Problems

-----

The script runs as expected and appears to have no errors. There is no error checking performed in this script; this is left up to the user.

## Appendix B

```
#  
# Michael Barnett  
#  
# August 3, 2004  
#
```

Documentation for program mikefit.sh

Syntax:

```
sh mikefit.sh <target_type> <"flat"/"physical"> [<max_energy>]
```

This program prepares a set of cross section files using `yield_to_xs`, and then uses `dwba_mike` to find an optimal set of parameters to fit these cross sections to the theoretical values. If `<max_energy>` is specified, the fit uses only `p3` and energy shifts. If `<max_energy>` is not specified, then the fit uses `p3` and `alpha`, with the energy shifts fixed to the values contained in the file `"xsec_list/xsec_list.<target_type>all"`. The second argument specifies whether the yield files should be based on a flat or physical background subtraction.

Supported target types:    c: carbon  
                          he: helium  
                          w: water (not fully supported yet)

User's Guide

The `target_type` argument specifies a material, as well as the data sets associated with that material. The flat background subtraction argument causes the program to use yield files of the form `"yield.flat.<cur_run>"`, which use an energy-independent background subtraction procedure. The physical background subtraction argument causes the program to use yield files of the form `"yield.<cur_run>"`, which use an energy-dependent background subtraction.

If `<max_energy>` is specified, the `dwba_mike` program will fit using `p3` and shifts only, and will not use data points above the `max_energy` value. If it is not specified, the maximum energy is set in the script (typically to 163 MeV), and the fit uses `p3` and `alpha` only. The shifts are held fixed to the

values specified in the file "xsec\_list/xsec\_list.<target\_type>all", which is written by mike\_fullfit.sh after calling mikefit.sh with a specified max\_energy. A typical fitting procedure is then to call mikefit.sh twice consecutively: once with a max\_energy value specified to determine the shifts, and again without a max\_energy specified to find alpha and p3 given the determined shifts.

This script is non-interactive while it is running. The primary way to change its behaviour is to change the values of the parameters passed into dwba\_mike, as described in the Maintenance Manual below.

#### Algorithm

-----

The script functions by calling yield\_to\_xs and dwba\_mike after properly preparing parameter sets for them. The set of yield files to be used is decided in the first case statement, based on the input argument from the user for the target type. The <physical/flat> argument decides on the type of yield files used: yield.<run> for a physical background, and yield.flat.<run> for a flat background.

To call yield\_to\_xs, a fourth order polynomial fit is performed on the efficiency data for the specified target type, using the polyfit program. The results are filtered through an awk script to give the five coefficients of the fit, which are placed into the file "efficiency\_file". This file is passed into yield\_to\_xs along with several other parameters. This program converts yield files into cross section files using these parameters, and stores the results in the xs\_files directory.

Finally, a list of all cross section files to be used is prepared, and dwba\_mike is called. The interactive input for this program is handled by a here document, so that the script is totally automated once it is called. The parameters used in the call are described internally within the script in the commented section preceding their assignments.

The results of dwba\_mike are passed to standard output, except for the data normally stored in files by that program.

#### Maintenance Manual

-----

Two issues likely to come up with this script are the need to handle new target types and the need to change the parameters sent into yield\_to\_xs or dwba\_mike.

To handle a new target type, a new entry is needed in the first case

statement, which determines the effective target type and what data files to use for a given first argument. For example, to add water as a new target type, an entry "w) all\_runs="w1 w2 w3..."; eff\_target="w"; " would be needed. The all\_runs variable contains all data files pertaining to the desired target. The eff\_target keeps track of the actual type of the target, which is useful if there are more than one set of files of a given material (e.g. c1,c2,c3... vs. c032).

Additionally, if the eff\_target is of a type not listed in the second case statement, an entry will have to be added. This must contain the pion production threshold in MeV for the material (threshold), the molar mass of the material (mmass) and the scaling factor for the cross section data (scale). The scale variable should always be 1.0 unless there is a compelling reason to change it.

If any of the data files listed in all\_runs are not included in the third case statement (contained inside the 'for cur\_run in \$all\_runs' loop), an entry must be added. This must contain the density of the target in grams per square centimetre (rhot).

To change the parameters for the yield\_to\_xs or dwba\_mike programs, the appropriate variables must be changed in the script. The variables for the yield\_to\_xs program are all determined by the target type and thickness, and should not be changed except when new data files are introduced. The variables for dwba\_mike are assigned just before the program is called at the end of the script. Each variable is described in the comment block preceding the assignment section, and they may be adjusted as desired.

#### Known Problems

-----

The program runs as expected and appears to have no errors. There is no error checking performed within the script; this is left up to the user.

## Appendix C

Michael Barnett

July 27, 2004

Documentation for yield\_to\_xs program  
-----

Syntax:  
-----

```
yield_to_xs <cur_target> <fit_file> <rhot> <mmass> <scale>
```

The program takes in a yield file `"/home/ru/exp062/yields/yield.<cur_target>"` and produces a cross section file `"xs_files/xs.<cur_target>"`. The file `"./<fit_file>"` contains a fourth order polynomial fit of the energy dependence of the efficiencies for the target type. `<rhot>` is the thickness of the target in  $\text{g/cm}^2$ . `<mmass>` is the molar mass of the target. `<scale>` is an arbitrary scaling factor for the cross sections, which should always be set to 1.0 unless a scaling is absolutely necessary.

User's Guide  
-----

The program is totally automated once the initial arguments are given. The resultant cross section file is `"xs_files/xs.<cur_target>"`, and will overwrite any file with the same name. The input yield file is always located at `"/home/ru/exp062/yields/yield.<cur_target>"`. If no file of that name exists, the program will exit with error and will not produce a cross section file.

Algorithm  
-----

Read in efficiency file to get five coefficients for polynomial fit of efficiency vs. energy curve; place into array "param" in reverse order so that `param[i]` is the 'i'th order coefficient.

For each data point in the yield file:



Calculate the efficiency factor using the point's energy and the coefficients of the polynomial fit.

Calculate the cross section and error:

```
cross section = yield * scale * molar mass
               /(thickness(g/cm^2) * Avagadro's Number * efficiency factor )
cross section error= yield error * scale * molar mass
                  /(thickness(g/cm^2) * Avagadro's Number * efficiency factor )
```

A factor barns2mbarns = 1e30 is also included to change the units of the cross sections into microbarns.

Print results into a file of the form "xs\_files/xs.<target>".

#### Maintenance Manual

-----

The major issue which may arise with this program is the need to change the pathnames for the yield or cross section file. These pathnames are both defined by macros located at the beginning of the source file "yield\_to\_xs.c". To change the pathnames, simply change the strings in the two macros and recompile the program.

#### Known Problems

-----

The program runs as expected and appears to have no errors. The argument list is checked for the correct number of arguments, but their types are not checked; this is left up to the user.

## Appendix D

Michael Barnett

July 27, 2004

Documentation for dwba\_mike program  
-----

Syntax:  
-----

dwba\_mike [target\_type]

target\_type supported values: "c" (Carbon)  
                                  "he" (Helium)  
                                  "w" (Water) (not fully supported yet)

This program performs a full analysis on a set of data files corresponding to the stated target\_type. The program collects the necessary input parameters through interactive input/output once the program is running. The output of the program is displayed on standard output; copies of the chosen energy shift values and the cross section residuals are saved to the files "shift\_values.txt" and "residuals.dat" respectively. The results of the program are the optimal p3 amplitude, the optimal corrective parameter alpha and the optimal energy shift values for each input file.

User's Guide  
-----

Once the program has been called with the target type specified, the program prompts the user to input several parameters into the program:

"File with list of xsection files"

- A filename relative to the executable's directory that contains a list of the filenames of each cross section file to be used in the analysis. The file should list each filename, followed by the first guess for the energy shift to be applied to the file's energy spectrum. If no first guess is known, a zero should still be entered after the filename.

"MAXIMUM KLAB"

- The maximum energy of a point that will be used in the fit of the

data in the cross section files to the theoretical model. A typical value is 163 MeV. For fits of shifts and p3 only (alpha set to 0), more typical values are 6 to 12 MeV above the pion production threshold for the material.

"Lower Energy Threshold"

- This is the pion production threshold for the material used in the simulation.

" PCT SLOPE OF p3(+) RE THRESHOLD (0-100): first guess"

- PCT is  $\alpha \cdot 2500$ , so typical values are 6 to 10. This is the median value for pct about which the search range will be. If alpha is not to be used in the fit, this value should be zero.

" Width of PCT search range, centred on first guess"

- This is the width of the search range, so that the minimal pct value probed will be the first guess chosen above minus half this value and the maximum pct probed will be the first guess plus half this value. A typical value is 20; zero should be used if alpha is not being used in the fit.

" p3: first\_guess"

- Specifies the median for the region in which p3 values will be probed. Typical values are 10 to 11.5.

" Width of p3 search range, centred on first guess"

- Exactly like the pct search range, but for the p3 parameter. Typical value is 10.

" DWBA CORRECTION? (1=Yes, 0=No)"

- For carbon, oxygen and some other target nuclei, it may be necessary to use the Distorted Wave Born Approximation in calculating the theoretical cross sections. This specifies whether to use this correction, and is only relevant if the theoretical lookup table is being generated.

" Number of sig. figs. in final result:"

- Specifies how many decimal places of precision should be used in probing the parameter space, and thus how many decimal places should be present in the final results for p3 and alpha. Note that the shift parameters cannot be probed beyond a 0.01 MeV accuracy since the input data files only report their energies to two decimal places.

" Use shifts in optimization?: (1=Yes, 0=No)"

- If "No" is selected, the shifts are fixed to the values specified in the file containing the list of input cross section files and their initial

shifts, and the fit is restricted to p3 and alpha.

If "Yes" is selected, the shifts are probed to +/- 4 MeV from their initial guesses in the file containing the list of cross section files.

" 4.43 CROSS-SECTION RENORM FACTOR for Carbon runs (typ 1.75)"

- For carbon (and oxygen and some other nuclei), an excited state may be involved in the theoretical cross section, affecting its value. For carbon, this effect should have a renormalization factor of 1.77, as specified in Bergstrom, Igarashi and Vogt, Phys. Rev. C, page 2923 Volume 55, June 1997.

" NORMAL OR REDUCED CROSS SECTION for He runs (0 or 1)"

- Specifies whether the momentum of the pion should be divided out of the cross section to create a reduced cross section. This should NOT be done, as this program only deals with normal cross sections. This option is a relic ported over from the previous Fortran code for helium cross sections, and is left in to avoid modifying (and inevitably breaking) code that already works.

" Generate lookup table for theoretical cross sections?"

" 1 -> Generate new table"

" 2 -> Table exists in file xs\_files/xs.theor.<target>"

- If a cross section lookup table does not already exist for the target nucleus of interest, the program can generate one. This calculation is time intensive, especially for more complicated nuclei like carbon and oxygen, and may take several hours to produce. This table is used by the program to allow rapid and precise probing of different values for the energy shifts for each of the files used, and greatly reduces the program's running time.

Once all of the parameters have been entered, the program will search for the optimal set of parameters based on the input ranges and values. The results are printed to standard output, along with several debugging statements. The final output is the p3 and alpha values with their associated errors, and a list of the shift parameters with their errors, as well as the  $\chi^2$  value for the fit to the data. These are the critical outputs of the program, along with two data files: "residuals.dat" and "shift\_values.txt".

"residuals.dat" contains 5 columns of data concerning the fit and its quality. For each data point in each input file, a line is written containing the new shifted energy, the experimental cross section, the theoretical cross section, the residual (experimental - theoretical) cross section, and the error in the experimental cross section, in that order.

"shift\_values.txt" contains each of the input files and the optimal value for

their energy shifts, in exactly the same form as the file the user specified to list the cross section files at the start of the program. This is done so that this file itself can be used as input to another call to the dwba\_mike program without any modification.

### Algorithm

-----

The program is divided into several routines contained in the following files:

#### dwba\_mike.c:

Contains the main() function where execution begins, as well as a routine for applying energy shifts to data sets and a routine for reading in the experimental cross section data files.

#### dwba\_error\_functions:

Contains functions for calculating a total  $\chi^2$  value for all data files simultaneously, and routines for finding the errors in parameter values through probing of  $\chi^2$  values.

#### make\_theor\_xsections:

Contains routines for generating a lookup table file for theoretical cross sections, reading in a lookup table file into an array, looking up a set of theoretical cross sections corresponding to the energy values in a data file structure, and getting a single theoretical cross section given a specific energy value.

#### dwba\_carbon:

#### dwba\_he:

#### dwba\_oxygen:

For each type of target supported by the program, a source file is needed which has at least two routines: one for calculating the  $\chi^2$  value for a data file of that target type (this involves using a p3 and alpha pair to "finish" a theoretical cross section calculation taken from the lookup table), and one for generating a lookup table of theoretical cross sections for that target type. Other routines are typically also included to organize the lengthy calculations needed to compute the theoretical cross sections.

In the source file dwba\_mike.c:

main():

The program first checks that the target type specified as an argument is supported by the program; that is that the lookup table for the theoretical cross sections can be generated.

The next part of the program is a sequence of requests for the user to input various parameters using standard input, as described in the user's guide above.

The program then calls the `get_xsections()` function to read in each of the cross section files specified in the file given by the user and place their information into structures. The theoretical cross sections are generated if the user has requested it using the `make_lookup_table()` function. The lookup table is then read into the global array `xsection_table`.

The search increment sizes are chosen for  $p_3$ ,  $\alpha$  and the shifts based on the precision specified by the user, with the shift resolution limited to 0.01 MeV accuracy. The size of the search ranges are then compared to the increment size to determine the number of iterations for each search loop.

The four nested loops perform the actual search over the parameters. The outer 2 loops iterate over choices for  $p_3$  and  $\alpha$  (if  $\alpha$  is to be included in the search). The third loop iterates over all files, while the innermost loop iterates over all choices of energy shift for a single file. Once the innermost loop is complete, the best shift for a single file given the current  $p_3/\alpha$  pair is known. Once the second loop is complete, the best set of energy shifts for the current  $p_3/\alpha$  pair is known. "Best" is defined as the choice of parameter which gives the lowest  $\chi^2$  value. The outer 2 loops then attempt all choices of  $p_3$  and  $\alpha$ , and the best possible choice for the parameters is determined by those which give the lowest total  $\chi^2$  for the data files.

Once the best parameters are known, the values are printed to standard output as well as to the file "shift\_values.txt". The residuals are also calculated for the best fitting parameters, and these are printed to the file "residuals.dat". The errors in the parameters are found by calling the appropriate function from the `dwba_error_functions.c` source file. The program then frees any memory that was dynamically allocated and exits.

`get_xsections()`:

The routine scans through the file containing the list of all cross section

files twice: the first pass counts the number of entries in the file so that an array of cross section file structures can be dynamically allocated, while the second pass actually reads in the filenames and the first guesses for their energy shifts.

For each cross section filename:

The corresponding file is opened and scanned twice: once to count the number of data points in the file so data arrays can be dynamically allocated, and again to read in the data in the form energy, cross section, cross section error.

Once all of the data have been read in, the program echoes the list of cross section files to standard output to let the user confirm that the correct data sets were loaded.

apply\_shift():

The routine adds the energy specified as an argument to each raw energy value for a data file structure, and places these shifted energies into the energy[] array in that data file structure. Raw energy refers to the actual energy values that were read from the file; they are NOT shifted by any amount, not even the first guess shift specified in the file listing all of the cross section files.

In the source file dwba\_error\_functions.c:

find\_ellipse():

This function is no longer used in the program. It worked by finding the error in each parameter and the covariances between them, but did not accurately produce maximal error bars. A more exhaustive routine is now used, in the function search\_all\_parameters().

search\_errors():

This routine is also no longer used; this is where the algorithm used in search\_all\_parameters() was developed.

search\_errors\_with\_shifts():

search\_all\_parameters():

These two routines use similar algorithms, but search\_errors\_with\_shifts does not include alpha in its searching. The general idea is to probe out in every possible direction to find the maximum change allowed in each

parameter before the total  $\chi^2$  increases by one. The `search_all_parameters` routine searches three variables at a time: `p3`, `alpha`, and one of the shift values.

The program iterates one of the variables until  $\chi^2$  is one greater than its minimal value. It then attempts to iterate the other two variables (up or down) once. If this succeeds, it checks to see whether the first variable can now be incremented further. This is repeated until none of the variables can be incremented without raising  $\chi^2$  more than one above the minimum. This is repeated using another variable as the first choice, and then the third. The three variables are then used again, but are now incremented in the other direction. The maximum and minimum values obtained for each of the three variables are then subtracted and divided by two to find the mean maximum error allowed in each parameter. The progression array and sign variable are used to make the complicated iterations as compact in code as possible.

```
move():  
move_with_shifts():
```

These two routines increment one of the parameters being searched for error ranges by passing the arguments by reference. `move_with_shifts` checks which variable is to be incremented, and checks the sign of that incrementation from the arguments supplied and applies the move.

```
calc_full_chi2():
```

This routine calculates the  $\chi^2$  value for each data file, using the set of shift parameters supplied by the fifth argument (a pointer). The sum of the  $\chi^2$  values for all of the files is then returned. This function is used by the error searching routines to see the effect of varying single parameters on the total  $\chi^2$  value.

In the source file `make_theor_xsections.c`:

```
make_lookup_table():
```

The routine takes in a set of theoretical cross sections that has been generated already and stored in an array and writes this to a lookup file in the form energy, theoretical cross section.

```
read_lookup_table():
```

The routine opens the lookup file corresponding to the target type specified



in the argument to the `dwba_mike` program and reads in the theoretical cross sections into an array. The array is global to the program, so that it may be accessed by any routines that need it, such as the error calculation routines and the  $\chi^2$  calculation routines.

`lookup_xsections()`:

The routine takes in a file structure and a shift index (old code relic, should always be set to zero). The routine places the partial theoretical cross section for each energy listed in the file structure into the `theor_xsection` array in the file structure.

`get_xsection()`:

The routine returns the partial theoretical cross section at the energy specified in the argument.

In the source files `dwba_carbon.c`:

`dwba_he.c`:

`dwba_oxygen.c`:

Each of these files, one per target type, contains two major routines, along with helper routines unique to each target type. The helper routines were ported directly from Fortran code written by Jack Bergstrom and Ru Igarashi. As such, their inner workings have become mysterious over time, and are not documented here. The routines should never need to be modified, and so this lack of external documentation should not be a problem.

`calc_xsections_<target>()`:

This routine generates the partial theoretical cross sections for the target type, which differ from the complete cross sections only through the removal of  $p_3$  and  $\alpha$ . The logic for the actual computation is taken from the Fortran code originally written by Ru Igarashi and Jack Bergstrom, and was not modified except to leave out the  $p_3$  and  $\alpha$  dependence, which is inserted later in the program logic.

The routine takes in a file structure and dynamically allocates memory in that structure to hold the partial cross sections. The second argument is the pion production threshold for the target, specified during the initial stages of the `dwba_mike` program. The third argument is an artifact from earlier versions of the software, and should always be set to zero. In earlier versions of the code, multiple sets of theoretical cross sections were generated for each file, and this argument specified which set of cross sections was being generated.

calc\_chi\_2\_<target>():

This routine inserts the p3 and alpha dependence into the partial cross sections to produce a final set of theoretical cross sections.

The energy spectrum of the input file is shifted by the amount specified in the argument list by calling apply\_shift().

For each energy in the newly shifted spectrum:

The appropriate partial cross section is obtained from get\_xsection(), and the p3 and alpha dependence are inserted.

The chi<sup>2</sup> value for that cross section compared with the actual cross section in the file with the same energy is computed and added to the total chi<sup>2</sup> for the file.

The routine returns the total chi<sup>2</sup> value for the fit to the file using the input choices of p3, alpha and energy shift.

#### Maintenance Manual

-----

The most likely change that will be needed for this program is the addition of new target types. To do this, several steps are needed:

1. A new source file should be created, probably with the name "dwba\_<target>.c" that will contain the calculations needed to generate theoretical cross sections and to calculate chi<sup>2</sup> values.

The routine which calculates the theoretical cross sections should have a similar argument list and structure as the routines already written for carbon, helium and oxygen targets. The results of this routine should NOT be the actual cross sections, but rather should be missing the contribution from p3 and alpha, which should be added in during the calculation of chi<sup>2</sup>. This is how a lookup table can be generated that still gives the freedom to probe different values of p3 and alpha without having to recalculate the entire cross section.

2. The routines defined in the new source file should be declared in the header file "dwba\_mike.h", and this header should be included in all source files added to the program. This ensures that the program is aware of the new routines when it becomes necessary to call them.
3. Several switch statements in the program need to have a new case added corresponding to the new target type. They are:

In dwba\_mike.h:

A `#define` should be used to map the target name (e.g. "LITHIUM\_TARGET") to an integer. This is done to make the target type easy to refer to in code, and to maintain readability of the code.

In `dwba_mike.c`:

Just after the variable declarations in `main()`, there is an if-else chain which verifies that the target type given as an argument is valid, and exits the program if it is not.

Nested in the innermost loop of the calculations in `main()` is a switch statement used to calculate the  $\chi^2$  for the given target type. An option should be added to call the  $\chi^2$  calculation routine that has been written for the new target.

After writing the results to `stdout` in `main()`, there is a switch statement in the loop which writes the residuals to a file. A new case should be added, and should apply the best choice for `p3` and `alpha` to the theoretical cross sections in the lookup table to generate the final theoretical cross section data.

In `make_theor_xsections.c`:

In `make_lookup_table()`, a switch is used to call the correct routine for calculating the partial theoretical cross sections. A new case should be added which calls the routine for the new target, and the threshold for the target should be used in the argument call.

Immediately following the last switch statement is one for opening a file to save the lookup table to. A new case should be added which opens a file of the form `"xs_files/xs.theor.<target_type>"`.

In the routine `read_lookup_table` is a switch statement which selects the correct lookup table file to open based on the target type. A new case should be added which opens a file with the name `"xs_files/xs.theor.<target_type>"`.

In `dwba_error_functions.c`:

In the routine `calc_full_chi2()` near the end of the source file, there is a switch statement that chooses the correct routine to calculate  $\chi^2$  for a single file of the current target type. A new case should be added that refers to the new routine written for the new target type.

4. The Makefile must be extended to include the new source file. It should include a rule for generating an object file from the new source file, following the example of `"dwba_carbon.c"`. The object, which should have the name `"obj/dwba_<target>.o"`, should be added into the variable at the top of the Makefile containing a list of all object files that need to be linked to form the complete `dwba_mike` program.

The steps are not lengthy to accomplish, with the possible exception of

writing the code to generate the theoretical cross sections, so a new target should be able to be incorporated into the program in only a few minutes once the code exists to calculate the theoretical cross sections.

Known Problems

-----

The program runs as expected and appears to have no errors.

The input to the program must be of the correct form; no error checking is performed on the validity of user input, except for filenames.

The program may crash while searching for the errors in shift values. This is caused by the limits of the lookup table, which only includes points between 120 and 170 MeV. This error should never occur for reasonably well behaved data sets.

## Appendix E

Michael Barnett

July 27, 2004

Documentation for fitscript.sh program  
-----

Syntax:  
-----

```
sh fitscript.sh
```

The program is a simple script, which can be used to repeatedly call `mike_fullfit.sh` with varying arguments.

User's Guide  
-----

This is a template of a script, which is available for the user to modify as desired in order to run repeated trials of `mike_fullfit.sh` with some degree of automation.

Maintenance Manual  
-----

Maintenance of this script is totally dependent on how the user modifies it. The original script is effectively a blank slate.

Known Problems  
-----

The script does not seem to like the use of I/O redirection to save copies of the output, although this is likely a symptom of a problem in `mikefit.sh` or `mike_fullfit.sh`. Specifically, the program hangs after printing the name of the `xsec_list` file containing the list of all data files to be used in the fit procedure.

