Lauren Bains August 27, 2002

Differential Scattering Cross Sections of Photons Scattered from Oxygen Nuclei

Introduction

The project undertaken was the analysis of data obtained from the nuclear Compton scattering of high energy photons off of oxygen nuclei (Saskatchewan Accelerator Laboratory (SAL) Experiment 056). Nuclear Compton scattering occurs when a photon with some initial momentum interacts with a nucleus, and is emitted from the nucleus with some angle θ. When the photon and nucleus interact, the nucleus may be left in an excited state; in this case the scattering is inelastic. Elastic scattering occurs when the nucleus remains in its ground state; the analysis undertaken here focusses entirely on this type of scattering.

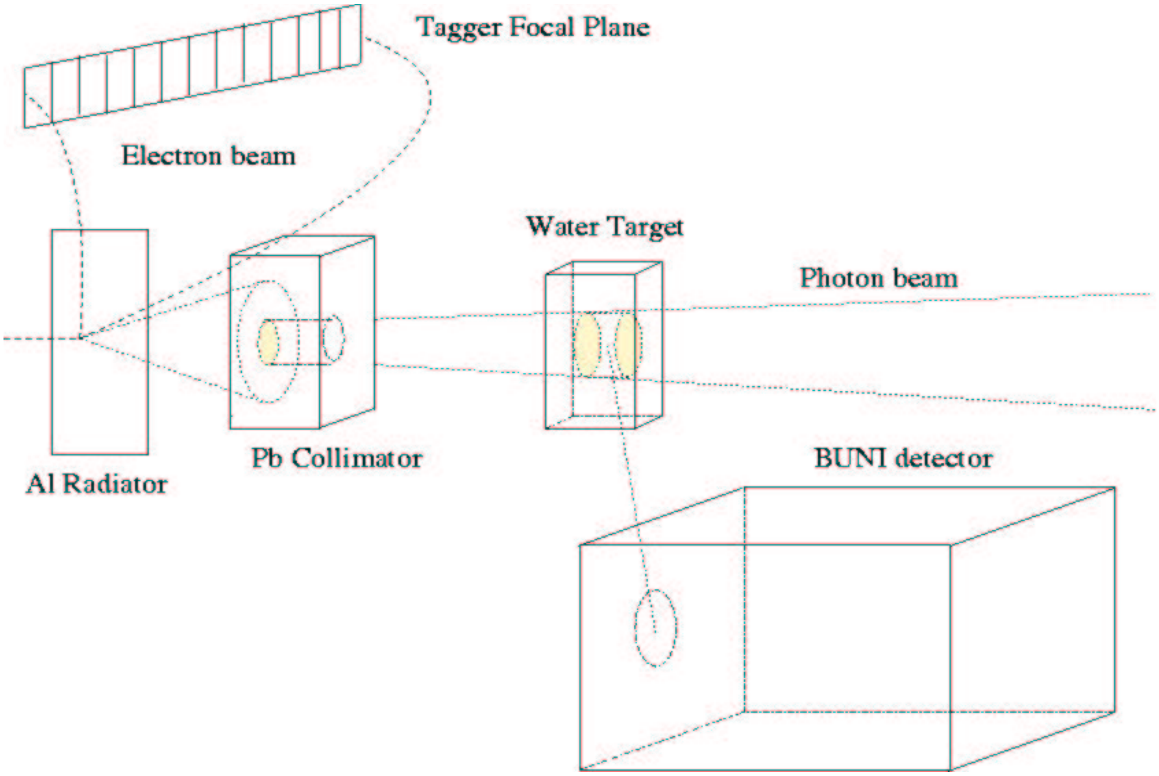


Illustration 1: A simplified diagram of the experimental setup

The goal of the analysis was to calculate the differential scattering cross sections for high energy (46 − 81 MeV) photons scattering off of oxygen nuclei. The differential scattering cross section indicates the probability that a photon hitting a target nucleus will scatter from that nucleus at a specific angle. In this experiment, data was taken for the angles θ = 45°, 60°, 90°, 120°, 135°, and 145°.

In order to obtain photons of the desired energy, a beam of electrons was raised to an appropriate energy in a linear accelerator was focussed on a thin aluminum radiator. Interacting with the atoms of the radiator caused the electrons to accelerate and emit bremsstrahlung ("breaking radiation") photons. The energies of these photons ranged from 0 MeV up to the original energy of the beam. After interacting with the radiator, the electrons proceeded into a tagger. The tagger is an apparatus used to determine the energy of the detected electrons. It consists of a region of magnetic field which swept electrons to the side, where they were detected in a focal plane of 62 scalars (the "tagger focal plane"). The range of energies which the focal plane detects is dependent on the geometry of the plane and the characteristics of the magnetic field applied. As the electrons were being tagged, the photons continued along unperturbed by the magnetic field, and passed through a lead collimator and a water target. The collimator simply served to produce a well defined beam spot on the surface of the target. If a photon Compton scattered from the target at the appropriate angle, it was detected in a large BUNI (Boston University Sodium Iodine (NaI)) detector. At this point, the energy of the photon could be pinpointed by associating it with the energy of the electron detected in the tagger focal plane at such a time as the photon and electron could be said to be related. The energy of a bremsstrahlung photon is related to the energy of the electron producing it through simple kinematics:

$$E_\gamma = E_0 - E_e$$

Where E_γ is the energy of the photon, E_0 is the (monochromatic) energy of the incident electron beam, and E_e is the energy of the tagged electron.
After the photons of this energy were produced at the radiator, their energy was altered by the Compton scattering process:

$$k' = \frac{k*M_A}{M_A + k(1 + \cos\theta)}$$

Where k' is the energy after scattering, k is the energy prior to scattering, θ is the scattering angle, and M_A is the atomic mass of the target nucleus.
As previously stated, the analysis focussed on the examination of those photons which were of an energy such that the probable interaction with the target was elastic.

Differential Scattering Cross Section: $d\sigma/d\Omega$

The differential scattering cross section was calculated according to:

$$\frac{d\sigma}{d\Omega} = \frac{N_c}{\varepsilon_d*N_e*\varepsilon_t*N_t*\Delta\Omega}$$

where N_c is the number of detected Compton scattered particles, ε_d is the detector efficiency, N_e is the number of incident electrons, ε_t is the tagging efficiency, N_t is the number of target particles per unit area, and $\Delta\Omega$ is the solid angle.

Carrying out this calculation required the creation of a number of programs to examine the various components of the equation before they could be assembled for the final calculation.

$\Delta\Omega$

The first component which was examined was the solid angle. In order to calculate the solid angle, a small C program was written which simulated the experimental environment. Note that all measurements that follow were taken from the log book of the experiment.; the diameter of the beam spot was also verified geometrically. A beam of 81 MeV photons was first passed through the 20 mm circular collimator. This beam was dispersed to produce a beam spot with diameter 6.70 cm on the front face of the water target, which was a 16.5 by 17.5 by 8.375 cm box. The target had 100 μm mylar windows on the front face, through which the beam passed. This beam spot had an intensity dictated by a (random) Gaussian distribution over its circular cross section. The Gaussian distribution ($\sigma = 1.402$[1]) was 'cut' at +/− 1cm due to the effect of the collimator on the beamspot (see Ill. 2).

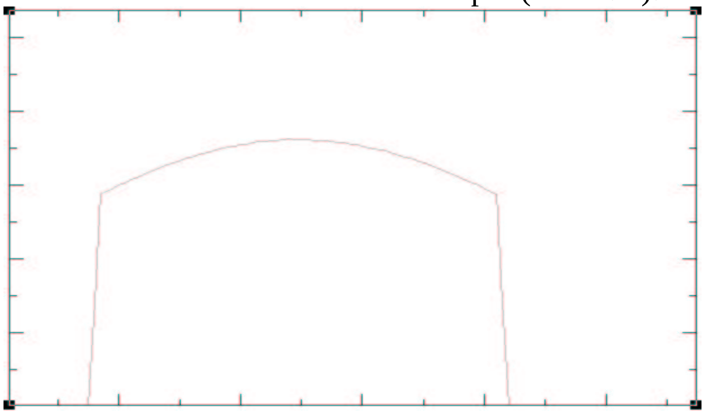


Illustration 2: A histogram of the x values of the photons striking the face of the detector. Note that the values follow a Gaussian distribution which is clipped due to the effects of the collimator.

As the incoming beam traveled through the water, individual photons were randomly absorbed. This random absorption proceeded throughout the time the photons remained in the target according to an exponential attenuation factor:

$$I_i = I_o*\exp(-\mu*\rho*x_i)$$

where I_i is the intensity of the beam at a distance x_i into the target, I_o is the initial intensity of the beam, $\mu = 0.017$ is the mass energy attenuation coefficient (from the NIST web site of physical reference data[2]), and ρ is the density of the target material (in this case, water, $1 g/cm^2$).

After a photon left the target, the point of intersection between its path and the plane of the BUNI detector was determined. In this program, the detector was approximated by a 12.7 cm diameter circular aperture. If the distance from the point of intersection to the centre of the aperture was less than 6.35 cm, the photon was assumed to have been intercepted by the detector.

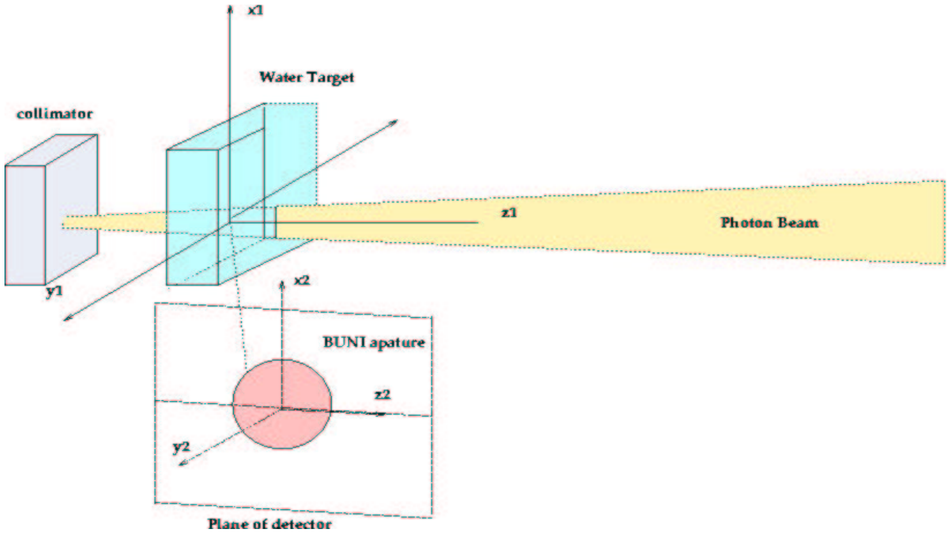


Illustration 3: A diagram of the setup for the solid angle simulation

Due to the

attenuation factors introduced, this was not a pure solid angle, but rather a modified one, taking into account the absorption of photons into the target.

As each stage of a photon's path was randomly determined, several histograms were set up to examine the distribution of photons being assigned a particular parameter in each stage. The histograms confirmed that the random distribution was uniform or not, as expected. For example, the distribution of the z coordinates of the incoming photons had an exponential distribution, which was expected as a random sample of the photons was being absorbed according to an exponential attenuation factor. The distribution of scattering angles, x, y, z, absorbed angles, and incoming x and y were all histogrammed and found to be behaving as expected in each case.

A program was written which compared the effects of different factors on the solid angle. Attenuation applied to the incoming and outgoing photons was found to be 6% and 10% respectively. Variation in the distance between the target and detector of 10 mm caused about 2% variation in solid angle; variation with 5mm was about 1%. Variation of the photon energy along the range allowed by the tagger focal plane (μ = 1.67e−2 to μ = 1.7e−2) resulted in 1% variation in solid angle. As these variations were of little significance, the values stated in the log book were simply taken and used at face value (109.8 mm target detector distance for 145, and 94.3 mm for all other angles).

ε_t

The next component of the differential scattering cross section which was calculated was the tagging efficiency. This number represented the percentage of tagged photons which made it from the radiator to the target; the effects of the lead collimator which lay between the radiator and the target would have had the most significant effect on this efficiency. During the experiment, tagging efficiency runs were performed using a lead glass detector, a different type of detector from the NaI detector used during production runs. This lead glass detector was placed directly in the beam, which was run at a reduced current (fewer electrons per unit time) at these times to protect the detector. Each time this detector was used, two runs were done: one with the aluminum radiator in place, and one with the radiator removed. When the radiator was removed, no bremsstrahlung photons should have been produced. Thus any photons detected in the lead glass detector were purely background. Tagging efficiencies were calculated both with and without taking into account the effects of the background (radiator out) runs. Radiator out contributions were accounted for via:

$$\varepsilon_t = \frac{g_i - g_o*(t_i/t_o)}{e_i - e_o*(t_i/t_o)}$$

While tagging efficiencies which did not take it into account were simply

$$\varepsilon_t = \frac{g_i}{e_i}$$

Where g_i and g_o are the number of photons detected with the radiator in and out respectively, and e_i and e_o are the number of electrons detected in the tagger channel focal plane with the radiator in and out.

When plots of efficiencies vs run number were compared with and without taking the radiator out subtraction into account, a constant offset was noted. The constant difference between efficiencies with and without correcting for background indicated that the beam current remained relatively constant throughout the radiator out runs.

During the experiment, 2 or 3 production runs were typically performed between each set of tagging efficiency runs. To get what may have been a more accurate idea of the tagging efficiency for each production run, an interpolation process was used. A line was drawn between the two tagging efficiency values, and the equation of that line was used along with the position of the production run to determine the appropriate tagging efficiency for that run.

The differences produced in the overall cross sections by using an interpolated vs. a non interpolated tagging efficiency were quite small, on the order of 0.6% (cross section error bars are on the order of 6%). These differences produced a small improvement in the overall χ^2. However, despite the improvement there was some uncertainty as to whether the error propagation methods being used to calculate interpolated errors adequately reflected the errors introduced. Considering these two facts, it was decided that the corrected, non−interpolated efficiencies should be used.

N_C

The number of detected Compton photons was calculated using histograms produced by LUCID[i], the program used during data acquisition. The histograms of detected photons as well as the histograms of background radiation were used. In the histograms shown below, the energy at which most of the photons appeared was 81.5807 MeV. This is simply the highest energy for photons coincident with electrons in the tagger focal plane (for all runs in this experiment except the second set of 90°). A peak appears here because this was chosen as the point towards which LUCID shifted the lower energy peaks. These peaks would have otherwise appeared at a distribution of different energies due to the distribution of channels in the tagger focal plane to this energy. By shifting all of the peaks to the same energy, the process of peak integration was simplified.

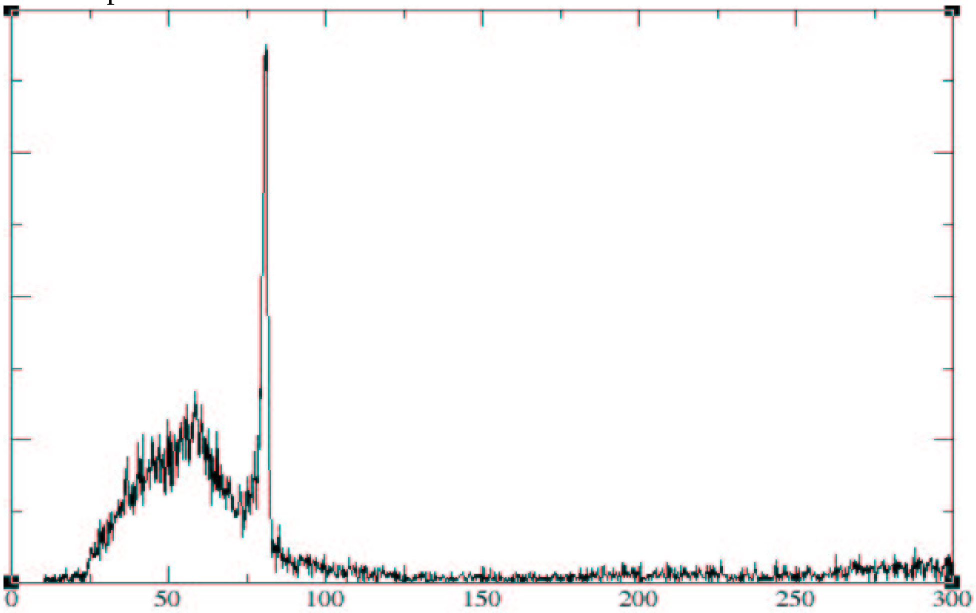


Illustration 4: A histogram of all photons detected by the BUNI at 135

[i] LUCID is a software package developed at SAL used in acquiring, saving, and organizing data. It can be used to produce histograms and make adjustments to the data.

Around this large peak on the histograms, a bin was selected which extended 6 MeV down and 4MeV up from this energy. 6MeV was chosen as the lower limit of the bin as this is the highest point at which photons experiencing inelastic scattering would appear. As the analysis focussed on elastic scattering, it was desirable to exclude these inelastic photons from analysis, and so the bin was chosen to end just prior to their appearance. Using this 10 MeV wide bin, the peak within that bin was integrated to find the total number of photons detected in that energy range. This exercise was performed both on the histograms of detected photons and the histograms of background. In order to find the total number of Compton scattered photons, background or "accidental" counts had to be subtracted from the total counts.

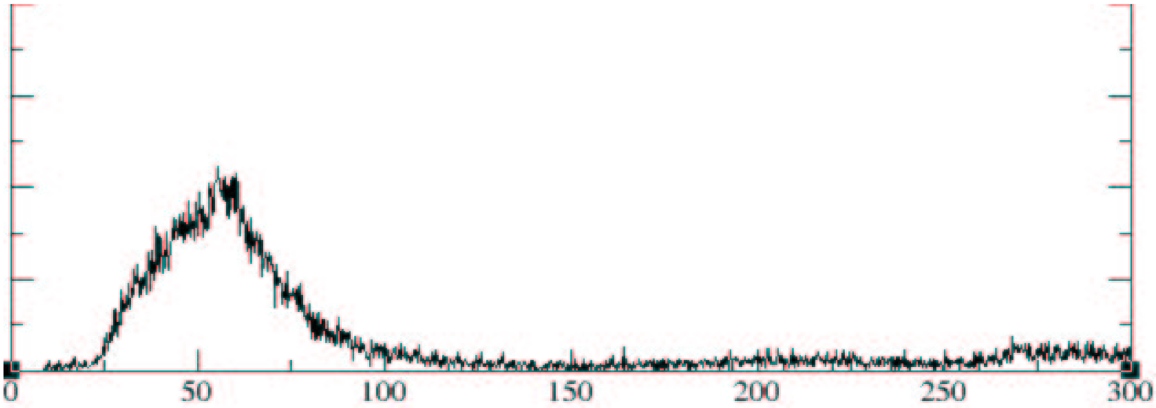


Illustration 5: A histogram of detected photons which were not the result of a Compton scattering ("accidentally" detected photons)

After the subtraction, almost all Compton photons were found in a sharp peak around 81 MeV. There was also some background, but it is consistent with zero − the fluctuations average zero.

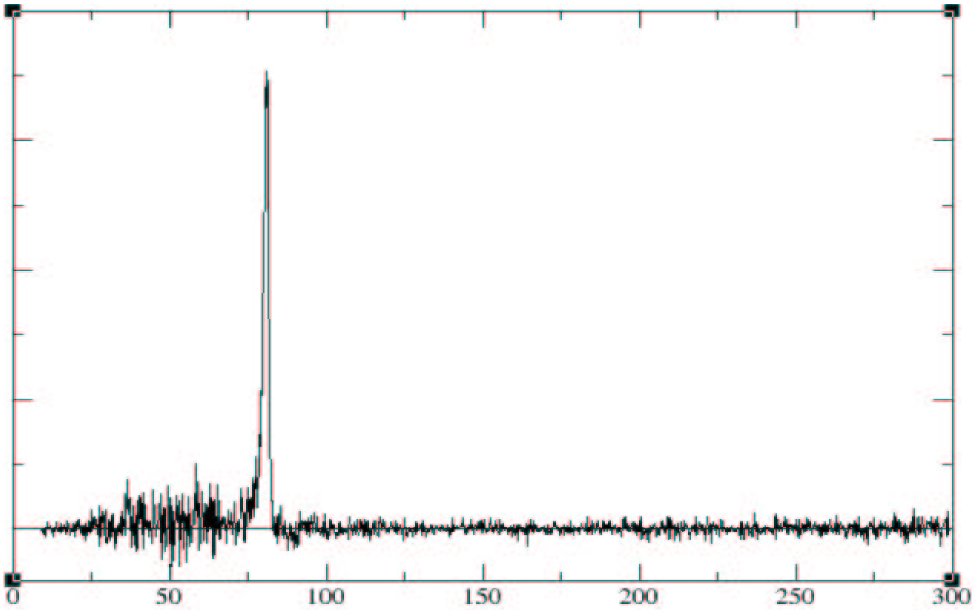


Illustration 6: This histogram is the result of subtracting the histogram seen in Ill. 5 from Ill. 4

Some 'raw' data was examined, to ensure that the adjustments being made to the data prior to this process (in LUCID) were not discarding necessary data. However, the cross sections produced using this raw data tracked the other cross sections fairly well (the graph shape was the same with a constant added in the y−direction), and the plots agreed within error. Thus it was decided it was adequate to use the adjusted data. This was fortunate, as there were many adjustments made in LUCID − everything from discarding cosmic rays and other very−high−energy particles to shifting the peaks.

ε_d

The detector efficiency is another important measurement. At one point during the experiment, a run was done to determine the efficiency of the detector. This run was similar to the tagging efficiency runs; the detector was moved directly into the beamline, and the beam current was reduced. However, unlike the tagging efficiency runs, during the detector efficiency run the BUNI was used instead of the lead−glass detector. This allowed the measurement of the detector response function to the incident photon beam, which is an important part of determining the percentage of tagged photons coming from the collimator which were detected in the BUNI in the energy range described in the N_c section above.

The equation used to determine detector efficiency was:

$$\varepsilon_d = \frac{N_c}{N_e * \varepsilon_t}$$

Here N_c is the number of Compton photons detected during the run. This number is found and corrected for accidentals as described above in the N_c section. N_e is the number of electrons detected in the focal plane, and ε_t is the tagging efficiency for the run, which was calculated as described in the ε_t section.

N_e

The number of detected incoming electrons, when combined with the tagging efficiency, gave the incident photon flux. During the experiment, whenever a production run was performed with the radiator in place, the next run was performed without the radiator. This allowed the number of incident electrons to be calculated both with and without taking the radiator−out contributions into account:

$$N_e = [r_i - r_o*(t_i / t_o)]* \varepsilon_t$$

or

$$N_e = r_i * \varepsilon_t$$

Where r_i is the radiator−in number of electrons, r_o is the radiator−out number of electrons, ε_t is the tagging efficiency, and t_i and t_o are the live times with the radiator in and out, respectively. When the radiator−out runs were not considered, the corresponding tagging efficiency was used, which also did not take into account radiator out contributions. Similarly, if the number of electrons calculated did take into account the radiator out runs, so did the tagging efficiency used. Not correcting for the radiator out contributions resulted in a cross section which was slightly lower than the corrected one (on the order of 0.4%, where the cross sections have error bars on the order of 6%).

The comparison of the reduced χ^2 for the corrected and uncorrected cross sections revealed that including or excluding the radiator out contributions produced a very small difference in consistency for the cross sections; the χ^2 improved for about half of the detector angles and worsened for the other half. However, all differences in cross sections were well within error. Overall, the differences in corrected and uncorrected cross sections indicated that the beam current during radiator out runs was not drifting, but simply varying randomly. Therefore, it was decided that the radiator out contributions should be accounted for.

N_t

The remaining component of the cross section equation, the number of particles in the target, was determined via a simple calculation:

$$N_t = \frac{x*\rho*N_A}{A}$$

where x is the target thickness, r is the density of the target material (here, $1_{g/cm}2$), N_A is Avogadro’s number, and A is the atomic mass of the target.

Conclusion

Once each of these factors had been examined and calculated, the final cross sections were calculated. These cross sections were then compared by angle and by energy.

The plot of cross sections vs. energy of incident electrons shows two 45° plots. During the experiment, two sets of runs were done at 45°: one near the beginning of the experiment, and one near the end. The plot below shows that the cross sections for the two sets agree within error. This serves as a consistency check, and verifies that no undetected problems arose during the experiment. At 90°, one set of runs was done at a lower energy than the other runs in this experiment. On the cross section vs. energy plot, it can be seen that this lower energy plot overlaps with the higher energy 90° plot for the energies which the two sets share. This also serves as a consistency check; the overlap is reassuring.

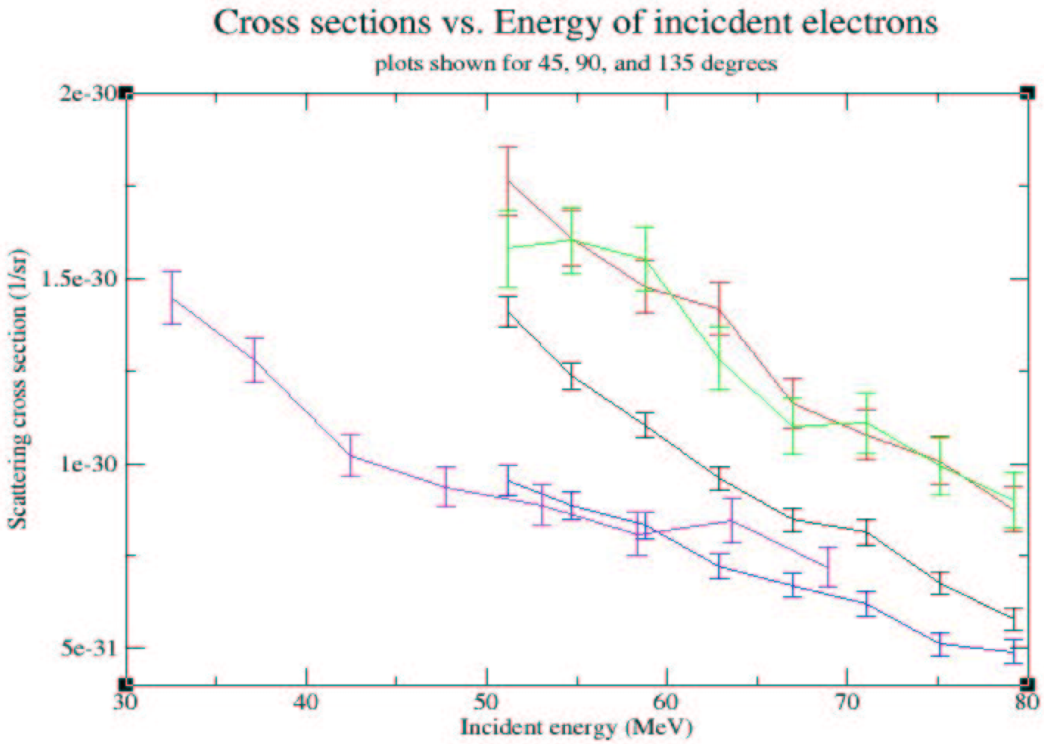


Illustration 7: Plot showing cross sections vs. incident energy for 45 (the two highest), 90 (the two lowest, which overlap), and 135 degrees.

In the plot of cross section versus detector angle, there are eight plots shown. These correspond to 8 groups into which the 62 tagger channels were divided (0−8, 9−16, ..). As the span of the focal plane is approximately 30 MeV, each tagger channel group spans about 3.8 MeV. From this plot, it is apparent that the cross section values increase as the energy of the incident photons decreases. As well, higher energy plots seem to have higher cross sections at the forward angles (angles less than 90°), while cross sections at the backward angles rise as the energy drops. This may be partially due to the kinematics − at higher energies, photons have more forward momentum and are thus less inclined to be scattered to backward angles. However, other physical processes can reverse this trend. It can also be noted at all eight plots are continuous, with no large leaps in cross section values.

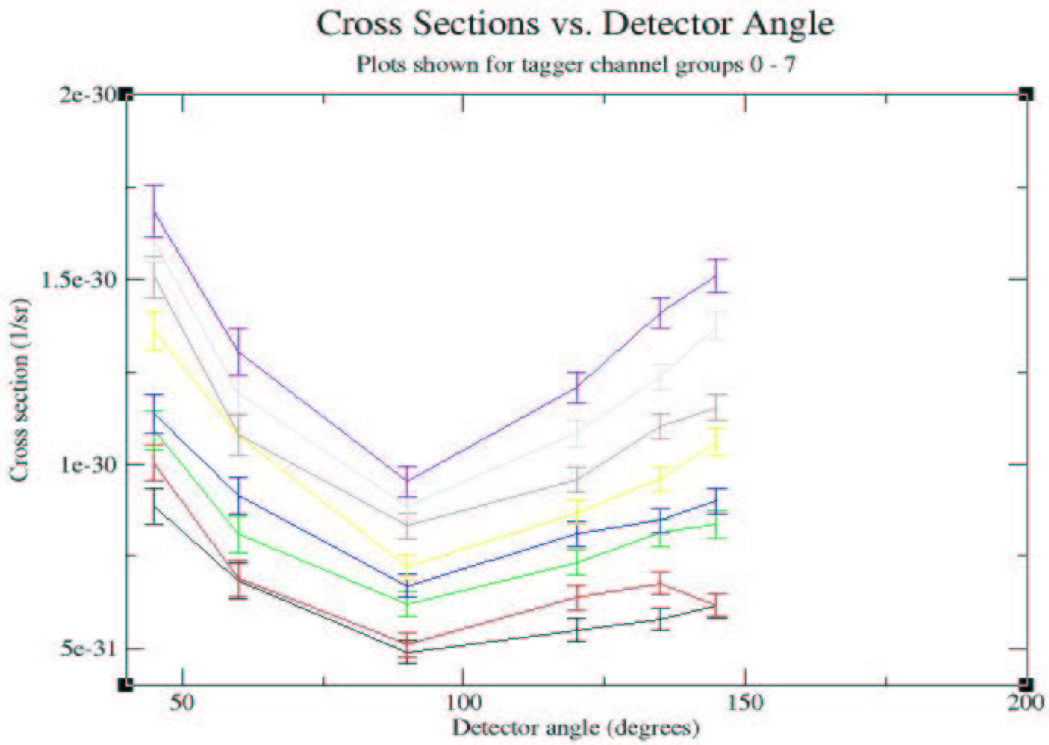
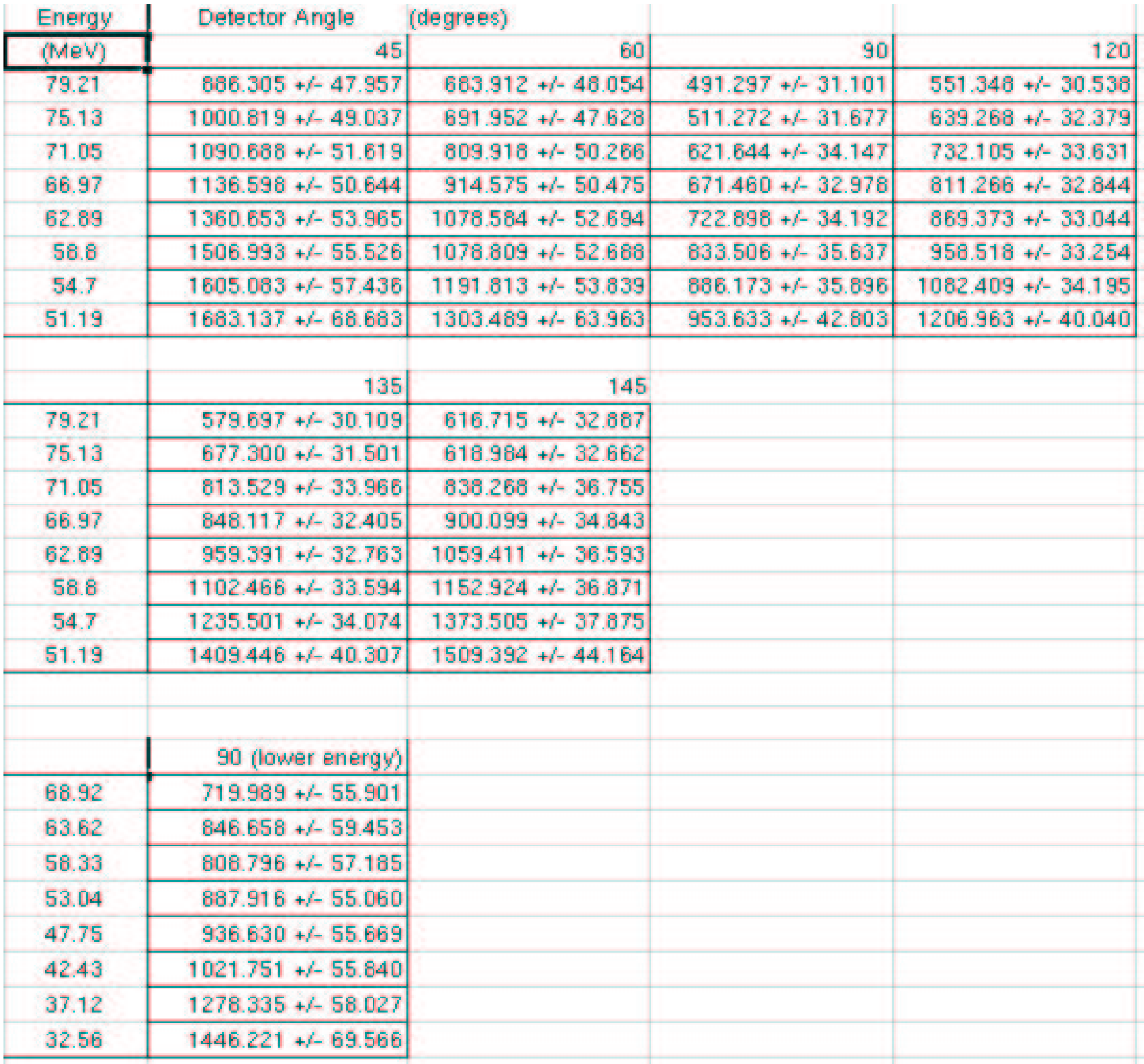


Illustration 8: Tagger channel groups 0−7 representing energies of 81 − 46 MeV. Cross sections increase with decreasing energy.

Below is a table of cross section values for each tagger channel group (these values lie at the core of the preceding plots). Note that there are two entries for 90° − one at the same energy as the other angles, and a separate entry for the lower energy 90°. There is only one entry for 45°. This is the weighted average of the two separate 45° sets which were performed at different times (and are consistent with one another, as seen in Illustration 7).

At the time of this paper, there is still much to be gained from further analysis of the data from Experiment 056. The use of the cross sections found here in the extraction of the electric and magnetic polarizabilities of the oxygen nucleus is one avenue which merits further exploration.



Energy	Detector Angle	(degrees)		
(MeV)	45	60	90	120
79.21	686.305 +/- 47.957	683.912 +/- 48.054	491.297 +/- 31.101	551.348 +/- 30.538
75.13	1000.819 +/- 49.037	691.952 +/- 47.628	511.272 +/- 31.677	639.268 +/- 32.379
71.05	1090.688 +/- 51.619	809.918 +/- 50.266	621.644 +/- 34.147	732.105 +/- 33.631
66.97	1136.598 +/- 50.644	914.575 +/- 50.475	671.460 +/- 32.978	811.266 +/- 32.844
62.89	1360.653 +/- 53.965	1078.584 +/- 52.694	722.898 +/- 34.192	869.373 +/- 33.044
58.8	1506.993 +/- 55.526	1078.809 +/- 52.688	833.506 +/- 35.637	958.518 +/- 33.254
54.7	1605.083 +/- 57.436	1191.813 +/- 53.839	886.173 +/- 35.896	1082.409 +/- 34.195
51.19	1683.137 +/- 68.683	1303.489 +/- 63.963	953.633 +/- 42.803	1206.963 +/- 40.040

	135	145
79.21	579.697 +/- 30.109	616.715 +/- 32.887
75.13	677.300 +/- 31.501	618.984 +/- 32.662
71.05	813.529 +/- 33.966	838.268 +/- 36.755
66.97	848.117 +/- 32.405	900.099 +/- 34.843
62.89	959.391 +/- 32.763	1059.411 +/- 36.593
58.8	1102.466 +/- 33.594	1152.924 +/- 36.871
54.7	1235.501 +/- 34.074	1373.505 +/- 37.875
51.19	1409.446 +/- 40.307	1509.392 +/- 44.164

	90 (lower energy)
68.92	719.989 +/- 55.901
63.62	846.658 +/- 59.453
58.33	808.796 +/- 57.185
53.04	887.916 +/- 55.060
47.75	936.630 +/- 55.669
42.43	1021.751 +/- 55.840
37.12	1278.335 +/- 58.027
32.56	1446.221 +/- 69.566

Table 1: The values of the differential cross sections are tabulated here according to detector angle and energy of incident photons. The entry for 45 is the weighted average of the cross sections from the two separate 45 sets which were run. All values for cross sections are in nanobarns (x 1e−33)

References

1. This sigma was found via:

   σ = dist*sqrt($\theta_0^2 + \theta_r^2$)

   where $\theta_r = \frac{m_e * c^2}{E}$

   (where is the mass of the electron, c is the speed of light, and E is the energy of the particle), and

   $\theta_0 = \frac{13.6 \text{ MeV}}{\beta cp}$*z*sqrt(x/X0)*(1 + 0.038*ln(x/X0))

   (see The Review of Particle Physics at http://pdg.lbl.gov/index.html for further details)

2. NIST data: http://physics.nist.gov/PhysRefData/contents.html

Program Appendix:

Notes:

− All relevant programs are written in C and are currently stored in /home/lauren/experiments/

−Throughout many of the programs, there are switches (rawSwitch = sr, correctedSwitch = sc, interpolateSwitch = si), enabling different functions. Many of these switches need to be coordinated. Check for any switches before you run something; they are generally located in main(), and are amongst the first statements after the declarations.

To calculate solid angles:

Currently, I'm using solidAngle2esTest.c to run solid angle stuff. It goes through each angle once, and the output is a bit simpler. solidAngle2es.c goes through each angle multiple times. Note that when the outgoing attenuation is calculated for the photons, the case where the photons go out through the top or bottom of the target doesn't work properly. However, this should not make a difference to the final solid angles being calculated. If a photon is exiting through the top or bottom of the target, it is too far above the y−z plane to ever hit the detector. But for the sake of completeness, this is something that I would have liked to correct, if I had the time.

− input is from user

− to change the number of incoming photons, change the i<# at the beginning of the big for loop in solidAngleCalculation() (ie for(i=0; i< 1000000; i++) change 1000000 to 1e8, or whatever. Run times right now are ~ 25 minutes for 8000000, ~ 3.5 hours for 1e8

− output files:
/home/lauren/exp056/oxygen/xsections/solidAngle.txt
/home/lauren/exp056/oxygen/xsections/plots/solidAngle
/home/lauren/exp056/oxygen/xsections/plots/solidAngle.hist

(The first file contains the important stuff that's put to screen: the solid angle and error. The last file contains the histograms. The middle file is used in solidAngleCompare.c, for comparing the solid angles just produced to some others in a specified file (see notes on solidAngleCompare.c for details))

In any case, right now the solid angles are hard−coded into the cross section programs. To change these values, go into sAngle() in crossSection.c or crossSectionsCumulative.c and change the values by hand.

To calculate cross sections run−by−run:

compile peakFind.c as 'pf' (check sr, file names)

− input files required (these are dump files produced by LUCID):
/home/lauren/exp056/oxygen/dumpsavefiles/run<run#>/alt_shift_prompt.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<run#>/alt_shift_acc.gz

−output files produced: (NOTE: check these names, the may be set to cumulative names, in which case just change them back)
/home/lauren/exp056/oxygen/intResults/subt.<run#>
/home/lauren/exp056/oxygen/intResults/<run#>.int
or

/home/lauren/exp056/oxygen/intResults/subt.<run#>.raw
/home/lauren/exp056/oxygen/intResults/<run#>raw.int
(output depends on status of sr)

compile tageffCheck.c as ’tec’ (check resolution, note that both corrected and uncorrected will be calculated for resolution = 8)
− input files
/home/lauren/exp056/oxygen/control_file
/home/lauren/exp056/oxygen/effres/dumpsavefiles/run<radInRun#>/efficiency.gz
/home/lauren/exp056/oxygen/effres/dumpsavefiles/run<radInRun#>/channel.scalers.gz
/home/lauren/exp056/oxygen/effres/dumpsavefiles/run<radInRun#>/run.info.gz
/home/lauren/exp056/oxygen/effres/dumpsavefiles/run<radOutRun#>/efficiency.gz
/home/lauren/exp056/oxygen/effres/dumpsavefiles/run<radOutRun#>/channel.scalers.gz
/home/lauren/exp056/oxygen/effres/dumpsavefiles/run<radOutRun#>/run.info.gz
(NOTE all of these except the control_file are dump files produced by LUCID)

−output files
/home/lauren/exp056/oxygen/tageffResultsNew/<eff#>.eff
/home/lauren/exp056/oxygen/tageffResultsNew/uncorrected<eff#>.eff
or
/home/lauren/exp056/oxygen/tageffResultsNew/Lg<eff#>.eff
(output depends on resolution status)

compile and run crossSection.c as ’cs’ (check sr, si, sc)
(NOTE: the reason that pf and tec were compiled as such is that there are system calls in cs to ensure that they have been run for all the appropriate runs before cross sections are calculated)
− input files required:
/home/lauren/exp056/oxygen/tageffResultsNew/<tagging eff #>.eff
/home/lauren/exp056/oxygen/intResults/<run#>.int
/home/lauren/exp056/oxygen/dumpsavefiles/run<radInRun#>/tagTotals.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radInRun#>/run.info.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radOutProd#>/tagTotals.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radOutProd#>/run.info.gz
/home/lauren/exp056/oxygen/detectorEff/eff
/home/lauren/exp056/oxygen/control_file
(NOTE: the first two are files produced by pf and tec, the next 4 are dump files produced by LUCID, the last two are files which should just be there. /detectorEff/eff has previously been calculated and left unchanged. To change this file, run calcDetEff() in crossSection.c (the call is currently commented out and replaced by a file read−in))

− output files produced:
/home/lauren/exp056/oxygen/xsections/<run#>.xs
or /home/lauren/exp056/oxygen/xsections/<run#>raw.xs
or /home/lauren/exp056/oxygen/xsections/uncorrected<run#>.xs
or /home/lauren/exp056/oxygen/xsections/interpolated<run#>.xs
(the output produced depends on the status of the switches)

NOTE that crossSection.c must be run using scripts (or by hand, run by run)
the scripts used previously are crossSection45.sh, crossSection60.sh, ... etc. for all angles

To calculate cumulative cross−sections:

compile tageffCheck.c as ’tec’(check resolution)
input and output files are the same as above.

compile and run crossSectionIncident.c Note that this currently calculates corrected, non−interpolated efficiencies and incident electrons (using calcCorr()). There are other methods within crossSection.c for performing other functions (doing uncorrected, doing interpolated), but these are not fully tested; use at your own risk.
− input files:
/home/lauren/exp056/oxygen/control_file
/home/lauren/exp056/oxygen/dumpsavefiles/run<radInRun#>/tagTotals.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radInRun#>/run.info.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radOutRun#>/tagTotals.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radOutRun#>/run.info.gz
/home/lauren/exp056/oxygen/tageffResultsNew/<tageff#>.eff
(the first 4 are LUCID dump files, the last one is a file produced by tec)
− output files:
/home/lauren/exp056/oxygen/xsections/incident<angle>.xs

compile and run peakFind.c as ’pf’ (check sr, file names)
− input files required (these are dump files produced by LUCID):
/home/lauren/exp056/oxygen/dumpsavefiles/run<run#>/alt_shift_prompt.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<run#>/alt_shift_acc.gz
− output files produced: (NOTE: check these names, they may be set to the run by run names, in which case just change them back)
/home/lauren/exp056/oxygen/intResults/Cumulsubt.<run#>
/home/lauren/exp056/oxygen/intResults/Cumul<run#>.int
(NOTE that pf is not system−called in csc. So you’ll have to run it some other way if you’ve changed something (probably by running one of the crossSection<angle>.sh scripts))

compile crossSectionsCumulative.c
− input files:
/home/lauren/exp056/oxygen/detectorEff/eff
/home/lauren/exp056/oxygen/intResults/Cumul<run#>.int
/home/lauren/exp056/oxygen/xsections/incident<angle>.xs
(detectorEff/eff is as before, to change it you must run crossSection.c. The other two files are from running tec, csi, and pf)
− output files:
/home/lauren/exp056/oxygen/xsections/cumul<angle>.xs
NOTE that csc is easier to run than cs; it no longer requires scripts, and can be run angle by angle using user input.

So now the cross sections have been calculated. There are a million things to do with them; here are some of the programs I've used to do them. All of these programs have internal documentation, however, with the time I have left to finish this I've chosen to update the documentation here rather than in the programs themselves. Thus some of the internal documentation will not include all functionality of the program (it may be incomplete, but it will rarely be totally wrong)

avgPlotMain.c

− has sr, sc, si. Check all.
− as a general input, the control file is needed
− before the program begins, you will be asked to choose either the tagging efficiency option or the cross section option.

− tagging efficiency option ('e'):
Finds the average tagging efficiency per bin for each run.
in:
/home/lauren/exp056/oxygen/tageffResultsNew/<eff#>.eff
or /home/lauren/exp056/oxygen/tageffResultsNew/uncorrected<eff#>.eff
(both files from tageffCheck.c)
out:
/home/lauren/exp056/oxygen/tageffPlots/ave<angle>.eff
or /home/lauren/exp056/oxygen/tageffPlots/aveUncorrected<angle>.eff

− cross section option('c'):
Finds the average cross section per bin for each run. Also calculates the overall reduced chi^2 using these values.
in:
/home/lauren/exp056/oxygen/xsections/<run#>.xs
or interpolated<run#>.xs or uncorrected<run#>.xs or <run#>raw.xs
/home/lauren/exp056/oxygen/xsections/plots/ave<angle>.xs
out:
/home/lauren/exp056/oxygen/xsections/plots/ave<angle>.xs
or aveUncorrected<angle>.xs or ave<angle>raw.xs or aveInterpolated<angle>.xs
tmp<angle>.xs (can be used as backup, just a tmp storage)
/home/lauren/exp056/oxygen/xsections/plots/avgPerAngle.xs
(NOTE: prints the reduced chi^2 to the screen, not to an output file)

avePlotAccs.c

− this program does a number of things, all of which occur on every run through of the program.
− the average number of Compton scattered photons per bin for every run is calculated (corrected for radiator out contributions)
in:

/home/lauren/exp056/oxygen/intResults/<run#>.int
out:
/home/lauren/exp056/oxygen/xsections/plots/ComptPeak<angle>.ave

− the average flux per bin for each run (corrected for background) is calculated
in:
/home/lauren/exp056/oxygen/tageffResultsNew/<tagEff#>.eff
/home/lauren/exp056/oxygen/dumpsavefiles/run<radInRun#>/tagTotals.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radInRun#>/run.info.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radOutRun#>/tagTotals.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radOutRun#>/run.info.gz
out:
/home/lauren/exp056/oxygen/xsections/plots/FluxPeak<angle>.ave

− the ratio of radOut/radIn is calculated for the flux (enables checking of consistency of background contributions (note, this option hasn't been looked at for a while..)
out:
/home/lauren/exp056/oxygen/xsections/plots/ratio<angle>.ave

− tagging efficiencies are calculated (it can be easily changed to allow some runs to have straight efficiencies, some to be interpolated, and some to be the average for the angle, but right now all are interpolated)
in:
/home/lauren/exp056/oxygen/tageffPlots/ave<angle>.eff
out:
/home/lauren/exp056/oxygen/tageffResultsNew/runEffGuide<angle>.eff

binPlots.c

− like avgPlotAccs.c, this also does many things each time it's run.
− run this twice before you trust the chi^2 values, some info from old runs is used.
− output is generally in the form of one file (for each function) for each angle, which is separated into 8 part, one part for each bin, each part listing every relevant run.
− general input: /home/lauren/exp056/oxygen/control_file

− finds the ratio of radOut/radIn for the flux (this hasn't been checked in a while..)
in:
/home/lauren/exp056/oxygen/tageffResultsNew/<eff#>.eff
/home/lauren/exp056/oxygen/dumpsavefiles/run<radInRun#>/tagTotals.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radInRun#>/run.info.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radOutRun#>tagTotals.gz
/home/lauren/exp056/oxygen/dumpsavefiles/run<radOutRun#>/run.info.gz
out:
/home/lauren/exp056/oxygen/xsections/plots/ratio<angle>.bins

− finds the cross section for each angle, each bin
in:
/home/lauren/exp056/oxygen/xsections/<run#>.xs
out:

/home/lauren/exp056/oxygen/xsections/plots/bins<angle>.xs

− using the tagging efficiency results calculated for each bin by tageffCheck.c, reorganizes these to produce bins<angle>.eff files
in:
/home/lauren/exp056/oxygen/tageffResultsNew/<eff#>.eff
out:
/home/lauren/exp056/oxygen/tageffPlots/bins<angle>.eff

− finds the interpolated tagging efficiency for each bin
in:
/home/lauren/exp056/oxygen/tageffPlots/bin<in#>.eff
out:
/home/lauren/exp056/oxygen/tageffPlots/binsInterpolated<angle>.eff

− finds the weighted mean energy for each tagger channel group per angle. This is based on the flux in each bin for an angle (to be used when graphing average cross section versus energy).
in:
/home/lauren/exp056/oxygen/dumpsavefiles/run<run#>/channel.scalers.gz
out:
/home/lauren/exp056/oxygen/xsections/energies<angle>.xs

− finds the reduced chi^2 for each bin.
in:
/home/lauren/exp056/oxygen/xsections/plots/modelXSbins<angle>.xs
output is to screen.

− there is also a method (coreZeroRatio()) for checking the ratio of the core 0 counter to the other core counters. This is no longer in use, as it was only meant to check a few runs. But it can be easily re−inserted into the program (just insert a call and an output file), if desired.

rebin.c

− this sort of goes along with binPlots.c. It can be used to reorganize the plots produced there into a format which may be better suited to some applications (if you just want to view all the angles at one energy, view bin0 instead of opening all 8 bin<run#>.xs files and trying to figure out which is the bin0 for each).
It takes the data produced by xsmethod and tagEff method in binPlots.c and reorganizes it. ie:
binPlots produces 8 files, one for each angle, with all 8 bins in each file (bins<angle>)
rebin.c produces 8 files, one for each bin, with all 8 angles in each file (bin<#>.xs and bin<#>.eff)
in:
/home/lauren/exp056/oxygen/xsections/plots/bins<angle>.xs
/home/lauren/exp056/oxygen/tageffPlots/bins<angle>.eff
out:

/home/lauren/exp056/oxygen/xsections/plots/bin<bin#>.xs
home/lauren/exp056/oxygen/tageffPlots/bin<bin#>.eff

xsVSbin.c

− also does some reorganization from binPlots.c. This produces files which can be viewed as plots of average cross section versus bin, using the information produced by binPlots.c. Right now the program is set to deal with the cumulative data (most of the other programs are not, right now). It can easily be set to deal with non−cumulative, as the non−cumulative version of the program is at the bottom of the file.

In:
/home/lauren/exp056/oxygen/xsections/cumul<angle>.xs
/home/lauren/exp056/oxygen/xsections/energies<angle>.xs

out:
/home/lauren/exp056/oxygen/xsections/plots/cumulxsVSenergies

solidAngleCompare.c

− This program allows you to compare the results of running the solidAngleCalculator using different parameters (ie compare the solid angles obtained using 30 MeV photons and 80 MeV photons)
To compare: paste the solid angle results of the run to be compared to into
/home/lauren/exp056/oxygen/xsections/solidAngleComparisonData.txt
(note that only the first 6 non−comment lines of that file are read). Then run something like solidAngle2esTest.c which writes its results to
/home/lauren/exp056/oxygen/xsections/plots/solidAngle
Then run this program, the comparison will be outputted to
/home/lauren/exp056/oxygen/xsections/solidAngleAverage

Any other programs are either small (made to do one specific task which could probably be done more easily some other way, like comparing two sets of chi^2 data) or outdated, and have been replaced by the programs mentioned here. All programs are currently stored in ~/experiments
Data files are stored in various places, as seen by the references above. But, in general, efficiency data will be in ~/exp056/oxygen/tageffResultsNew, efficiency files which can be viewed as plots will be in ~/exp056/oxygen/tageffPlots. Similarly, xsection data will be in ~/exp056/oxygen/xsections, and xsection related files which can be viewed as plots will be in ~/exp056/oxygen/xsections/plots. LUCID dump files for production runs are in ~/exp056/oxygen/dumpsavefiles, and dump files for efficiency runs are in ~/exp056/oxygen/effres/dumpsavefiles.

Typical values:
This is a general guide so that you know what’s appropriate when running these programs. These values are what I have obtained using the programs listed above. Changes made to the programs may change these values. The values below are taken

from output files.
Tagging efficiencies should be between 0.49 and 0.52
These are the solid angles obtained for each detector angle, with error, at 1e8 incoming photons:

45	0.011865	0.000027
60	0.011638	0.000027
90	0.011353	0.000027
120	0.011635	0.000027
135	0.011900	0.000027
145	0.008878	0.000024

Detector efficiencies should be between 0.89 and 0.93
Number of target particles should be 2.7942x10^23