Gamma Cross Sections

Aman LaChapelle

University of Chicago

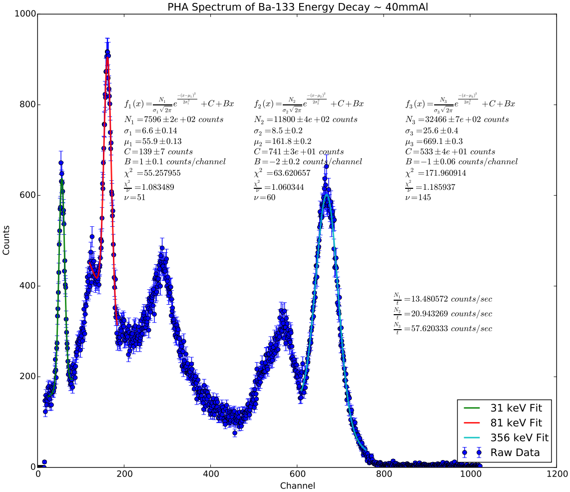
October 21, 2015

Gaussian Fits

$$f(x) = \frac{N}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} + Ax + B$$

We fit all the peaks to this function because we didn't trust the ROI settings and their consistency.

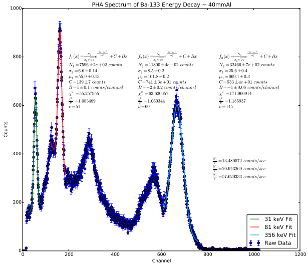
Gaussian Fits



An example fit.

Table of Contents

Distribution Width



The distribution should be a delta function - it isn't because of the detector resolution

Distribution Width

Energy (keV)	Γ (channels)	Γ/E
31	8.36	26.9%
81	10.18	12.6%
356	30.25	8.49%
511	10.68	2.09%
1270	16.69	1.31%

Not perfect resolution - this is our σ (up to $\Gamma/\sqrt{2}\sqrt{ln(2)}$).

Table of Contents

Covariance Matrix

Scipy and Numpy return the covariance matrix, but here is some information on what it is:

$$\Sigma_{ij} = (X_i - \mu_i)(X_j - \mu_j)$$
$$\Sigma_{ij} \equiv \mathrm{cov}(i, j)$$

with each μ_k being what we expect that datum to be assuming it fits perfectly to our model. We can express each X as a vector and have a matrix equation or we can build it up piece-by-piece. In general least-squares optimization we use

$$C(x^*) = (J^T(x)J(x))^{-1}$$

where $J(x)$ is the jacobian of the model function at the point x. x^* is the maximum likelihood estimate, $min||f(x)||^2$. [3]

Error Propagation of the PHA fits

Since we performed fits, the error of the ith parameter is just:

$$\sigma_i = \sqrt{cov(i, i)}$$

Because our largest δt was less than 0.001% the uncertainty in parameter i is just σ_i

Error Propagation of the Countrate Fits

We use the same technique as we did with the PHA spectra to calculate the uncertainty in our parameters. A sample of the data used to calculate the countrates:

Ba-133 Countrates and Uncertainty

Thickness (mm)	N1 (count/s)	N2 (count/s)	N3 (count/s)	δN1	δN2	δN3
56	9.044	9.910	39.361	1.915%	4.832%	1.959%
40	13.480	20.943	57.620	2.632%	3.389%	2.156%
...	...	...	...	...	...	...
2	303.314	145.682	156.056	1.094%	1.302%	2.127%
1	391.492	152.985	155.885	1.032%	1.320%	2.074%

Our error bars in the countrate data are exactly the percent error you see above, and we will see this moving forward.

Table of Contents

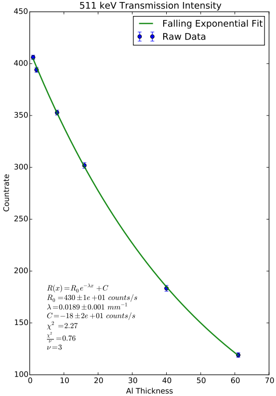
Fitting to find λ

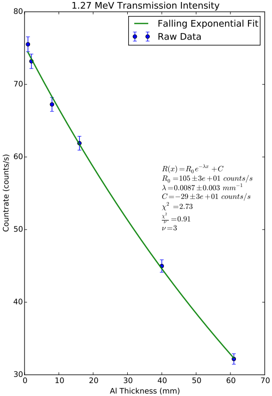
We fit to the function

$$R(x) = R_0 e^{-\lambda x} + C$$

to extract our values for λ

Fitting to find λ

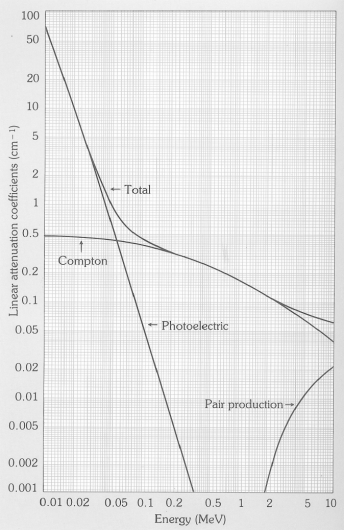




Energy	λ	$\delta\lambda$
511 keV	$0.19\ cm^{-1}$	$0.03\ cm^{-1}$
1.27 MeV	$0.09\ cm^{-1}$	$0.05\ cm^{-1}$

Table of Contents

Linear Attenuation Coefficient Character



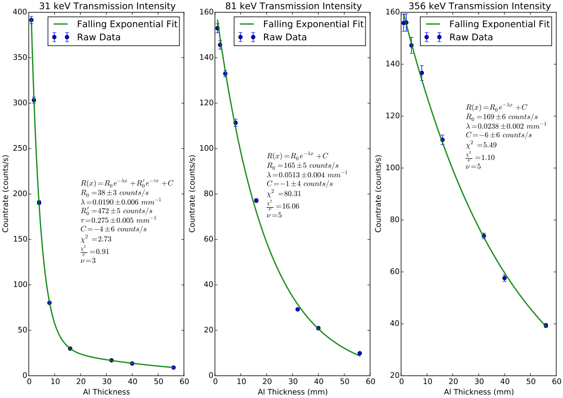
Fitting to find λ

We fit to the function

$$R(x) = R_0 e^{-\lambda x} + R_0' e^{-\tau x} + C$$

with λ being the compton-like attenuation and τ being the photoelectric-like attenuation.

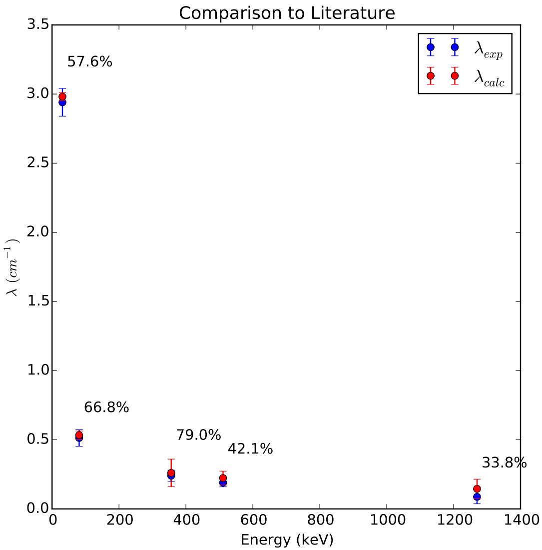
Fitting to find λ



Energy	λ	$\delta\lambda$
31 keV	$2.9\ cm^{-1}$	$0.1\ cm^{-1}$
81 keV	$0.513\ cm^{-1}$	$0.06\ cm^{-1}$
356 keV	$0.24\ cm^{-1}$	$0.04\ cm^{-1}$

Table of Contents

Comparison to Literature



Comparison to Literature

Energy	$\lambda \pm \delta\lambda\,(cm^{-1})$	$\lambda_{calc} \pm \delta\lambda_{calc}(\,cm^{-1})$	Confidence Level
31 keV	2.9 ± 0.1	2.98 ± 0.03	57.6%
81 keV	0.513 ± 0.06	0.533 ± 0.03	66.8%
356 keV	0.24 ± 0.04	0.3 ± 0.1	79.0%
511 keV	0.19 ± 0.03	0.22 ± 0.05	42.1%
1.27 MeV	0.09 ± 0.05	0.15 ± 0.07	33.8%

The NIST values have uncertainties calculated based on how far the tabulated energy was from our energy and estimating the slope of the area immediately around the point of interest.

Table of Contents

Systematic Uncertainty

We noticed that our data (for λ) was consistently lower than the NIST values, so we tried to fit it and the NIST values to:*

$$f(x) = Ae^{Bx} + Ce^{Dx} + E.$$

We found that the difference between our values and the NIST values was in E - it was just an offset. Furthermore we got a value for E of 0.020 ± 0.003 and so we added that 0.02 to all our uncertainties throughout for λ.

* not shown

References

- Physics.NIST.gov - Table of XRay Mass Attenuation Coefficients
  http://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTab/z13.html
- University of Chicago, PHYS 211 Lab Manual - P211 Wiki
- Ceres Solver, an Open-Source C++ optimization library
  http://ceres-solver.org
- An Introduction to Error Analysis - John Taylor