Methodology For Generating Simplified Cross Section Data Sets For Neutron Transport Calculations

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Thomas Jay Harrison
December 2015

Copyright © 2015 by Thomas Jay Harrison
All rights reserved.

DEDICATION

I dedicate this dissertation to my wife, Colleen, and our children, Cailin, Brendan, Taryn, and Ryanne. Thank you for putting up with me over the last few years while I finished this work. I love you all very much, and I did this for you.

ACKNOWLEDGEMENTS

It would be irresponsible not to acknowledge the people
who helped me get to the end here.

First, I need to acknowledge my parents, Mary and Bob,
who have not stopped pushing me to keep working,
and keep working harder, my entire adult life.
I need to acknowledge my parents (again) and my
mother-in-law, Kathy, for watching the kids when I
needed some peace and quiet. I need to acknowledge my
wife, Colleen, who took the kids to their grandparents'
houses so I could find that peace and quiet.

Outside the family, I need to acknowledge my
boss, Gary, who was supportive of my having a
full-time job and being at times a full-time student.
I also need to acknowledge my previous
boss, Randy, who showed the same support
when I worked for him.

Finally, I need to acknowledge my advisor, Dr. Townsend.
Slowly, and surely, and eventually, I finished my research and dissertation.
He gave me plenty of support, and leeway, and understanding along the way.
And no, I don't know if I'll ever be able to call him anything other than
"Dr. Townsend".

ABSTRACT

Neutron shielding problems involve radiation transport calculations over a wide range of energies. Fission neutrons have initial energy on the order of MeV, fusion neutrons have initial energy on the order of 10s of MeV, and space-origin neutrons have initial energy on the order of 100s of MeV or higher. Shielding calculations must track the neutrons from their initial energies until they are no longer of interest; for deep-penetration neutrons, this final energy can be on the order of eV before the neutron is no longer tracked. Thus, for deep-penetration space radiation shielding problems, the calculation may require tracking the neutron energy through eight orders of magnitude.

The shielding calculations also require the evaluation of the neutron cross section as a function of the neutron energy. However, the cross section value itself may range from 10^{-3} barn (1 mb) to nearly 10^{9} barn (1 Gb), a range of twelve orders of magnitude. Further complicating the cross section analysis is the existence of resonance peaks; these peaks (or valleys) may show a change spanning multiple orders of magnitude in cross section value over less than a 1% change in neutron energy.

The issue of cross section data sets with multiple resonance peaks can be resolved through the use of flux-weighted group cross sections. The most basic group structure is a single cross section; modern analytical codes can use more than 200 groups, or the full cross section data set itself. However, this introduces a tradeoff of efficiency (fewer groups) versus accuracy (more groups), and it also requires an *a priori* knowledge of the flux spectrum.

This research proposes and tests a method to generate group-wise cross section data sets that do not require the *a priori* flux spectrum, which is equivalent to assuming a flat flux spectrum distribution. This method conserves the energy-integrated cross sections, which are an inherent characteristic of an isotope, instead of group-wise reaction rates, which are a function of the overall system. The net result is a reduction in calculation time without a significant loss in neutron survival and penetration results and the transmitted and reflected spectra.

TABLE OF CONTENTS

CHAPTER I Introduction and General Information....1
Background and Problem Statement....1
Research And Initial Results....4
Purpose Of Research....5
CHAPTER II Literature Review....9
Research In Context....9
Evaluation Of Some Existing Shielding Codes And Benchmarks....9
Previous Cross Section Smoothing Work....10
CHAPTER III Materials and Methods....11
Definition Of Problem Solved....11
Cross Section Processing....11
Smoothing Methodology....12
Handling Of Scattering Angles....13
Neutron Tracking In 3-Dimensional Space....14
Values Tracked And Recorded....15
Number Of Scattering Collisions....15
Linear And Radial Distance Traveled....15
Final Energy....15
Deepest Penetration....15
Uncertainty Analysis....15
Transmission And Reflection Probabilities And Spectra....16
CHAPTER IV Results and Discussion....17
Materials Of Interest....17
Carbon (C)....17
Iron (^{56}Fe)....22
Benchmarking Against SCALE....27
Transmitted and reflected flux spectra....27
Group-wise ratio of reflected-to-transmitted flux....28
Benchmarking Results....28
Carbon results....28
Iron results....46
CHAPTER V Conclusions and Recommendations....53
Overall Results, Qualifications, And Caveats....53
Expansion Of The Method And Potential Future Work....53
LIST OF REFERENCES....55
APPENDIX....57
SCALE Benchmark Example Input File....58
238-group Structure....61
Cross Section Conditioning Python Script....63
Shielding Neutron Tracker Python Example Script....66
Shielding Tracker Example Output....83
VITA....85

LIST OF TABLES

Table 1. Carbon cross section set comparison, ENDF/B-VII.1 vs. Smoothed.....22
Table 2. ^{56}Fe cross section set comparison, ENDF/B-VII.1 vs. Smoothed.....27
Table 3. Calculated parameters for different slab thicknesses and k values for carbon.....39
Table 4. Calculated parameters for 20 cm slab and various k values for ^{56}Fe....52

LIST OF FIGURES

Figure 1. Total cross cection (barns) for ^{27}Al, 100 μeV to 150 MeV 2
Figure 2. 5.9 keV resonance peak in ^{27}Al..... 3
Figure 3. Total cross sections for ^{27}Al. 6
Figure 4. Comparison of the full ENDF/B-VII and smoothed total cross section data sets in the resonance range..... 7
Figure 5. Comparison of the full ENDF/B-VII and the stretched (k=5), smoothed total cross section data sets in the resonance range. 8
Figure 6. Cross sections for Elemental C..... 18
Figure 7. Cross sections for Elemental C, 1 MeV – 10 MeV 19
Figure 8. Comparison of ENDF/B-VII and smoothed (k=1) total and elastic cross sections for elemental C through 100 MeV. 20
Figure 9. Comparison of ENDF/B-VII and smoothed (k=1) total (blue and green) and elastic (red and purple) cross sections for the resonance region of elemental C..... 21
Figure 10. Cross sections for ^{56}Fe 23
Figure 11. Comparison of ENDF/B-VII and smoothed (k=1) total and elastic cross sections for ^{56}Fe through 150 MeV 24
Figure 12. Comparison of ENDF/B-VII and smoothed (k=1) total (blue and green) and elastic (red and purple) cross sections for the 1.15 keV resonance of ^{56}Fe. 25
Figure 13. Comparison of ENDF/B-VII and smoothed (k=1) total (blue and green) and elastic (red and purple) cross sections for the 10 keV to 10 MeV resonance region of ^{56}Fe 26
Figure 14. Comparison of reflected spectra for 4.5 MeV neutron incident on a 50 cm carbon slab 30
Figure 15. Comparison of reflected spectra using smoothed and stretched cross sections for 4.5 MeV neutron incident on a 50 cm carbon slab. 31
Figure 16. Comparison of ENDF/B-VII and smoothed total cross sections in the resonance region (1 MeV to 10 MeV) for elemental carbon 32
Figure 17. Comparison of transmitted spectra for 4.5 MeV neutron incident on a 50 cm carbon slab. 33
Figure 18. Comparison of transmitted spectra for 4.5 MeV neutron incident on a 50 cm carbon slab with 50k histories for k=0 and k=1 cross section sets.... 34
Figure 19. Comparison of transmitted spectra for 4.5 MeV neutron incident on a 50 cm carbon slab for 10k, 50k, and 100k histories for k=0 cross section sets 35
Figure 20. Comparison of transmitted spectra for 4.5 MeV neutron incident on a 50 cm carbon slab for 10k, 50k, and 100k histories for k=1 cross section sets. 36
Figure 21. Comparison of reflected spectra for 4.5 MeV neutron incident on a 50 cm carbon slab 37
Figure 22. Comparison of transmitted spectra for 4.5 MeV neutron incident on a 50 cm carbon slab. 38

Figure 23. MAVRIC group flux ratio (reflected/transmitted) for 4.5 MeV neutrons incident on 50 cm carbon slab. 40
Figure 24. Smoothed k=1 group count ratio (reflected/transmitted) for 4.5 MeV neutrons incident on 50 cm carbon slab. 41
Figure 25. Comparison of group ratios for MAVRIC and k=1 for 4.5 MeV neutrons on 50 cm carbon slab. 42
Figure 26. MAVRIC group flux ratio (reflected/transmitted) for 4.5 MeV neutrons incident on 20 cm carbon slab. T. 43
Figure 27. Smoothed k=1 group count ratio (reflected/transmitted) for 4.5 MeV neutrons incident on 20 cm carbon slab. 44
Figure 28. Comparison of group ratios for MAVRIC and k=1 for 4.5 MeV neutrons on 20 cm carbon slab. 45
Figure 29. Comparison of group ratios for MAVRIC, k=1, and k=5 for 4.5 MeV neutrons on 20 cm carbon slab....46
Figure 30. Comparison of reflected spectra for 0.4 MeV neutron incident on a 50 cm iron slab. 48
Figure 31. Comparison of transmitted spectra for 0.4 MeV neutron incident on a 50 cm iron slab. 49
Figure 32. Comparison of transmitted spectra for 0.4 MeV neutron incident on a 20 cm iron slab. 50
Figure 33. Comparison of group ratios for MAVRIC, k=1, and k=5 for 0.4 MeV neutrons on 20 cm ^{56}Fe slab....51

CHAPTER I
INTRODUCTION AND GENERAL INFORMATION

Background and Problem Statement

Neutron shielding problems involve radiation transport calculations over a wide range of energies. Fission neutrons have initial energy on the order of MeV, fusion neutrons have initial energy on the order of 10s of MeV, and space neutrons have initial energy on the order of 100s of MeV or higher. Shielding calculations must track the neutrons from their initial energies until they are no longer of interest. For deep-penetration neutrons, this can be on the order of eV before the neutron is no longer tracked. Thus, for deep-penetration space radiation shielding problems, the calculation may require tracking a neutron through eight orders of magnitude.

The cross section data itself also has a wide range of values. Figure 1 shows the total cross section for ^{27}Al [Reference 1]. The cross section ranges from around 1 b to 100 b, a difference of two orders of magnitude.

Figure 1 also shows the resonance peaks and valleys in the neutron cross section for neutron energies spanning from the 10s of keV to 10s of MeV. For example, there is a large resonance peak around 5.9 keV; the surrounding value is around 1 b, but the peak is at ~100 b. This is shown in Figure 2.

This complicates Monte Carlo-based shielding calculations because the microscopic cross section determines the distance a neutron travels before interacting with the medium. For reference, the average distance traveled by a particle at energy E before interaction is the mean free path. The mean free path for the neutron is proportional to $\frac{1}{\sigma}$ where σ is the microscopic cross section.

Thus, a Monte Carlo-based shielding calculation using a linear-interpolation lookup table could see a factor of 100 variation in the mean free path for a variation of 1.7% (100 eV/5900 eV) in the calculated neutron energy. That is, a 5.8-keV or 6.0-keV neutron on average travels 100x the distance of a 5.9-keV neutron. In Monte Carlo-based calculations, the post-scatter neutron energy is a function of the randomly-chosen scattering angle from a scattering interaction; this implies that based on the series of random numbers generated, a resonance peak may be missed completely or may dominate the calculation, leading to poor statistics and slow convergence. The calculation statistics can be improved by increasing the number of histories, but this leads directly to increased runtime.

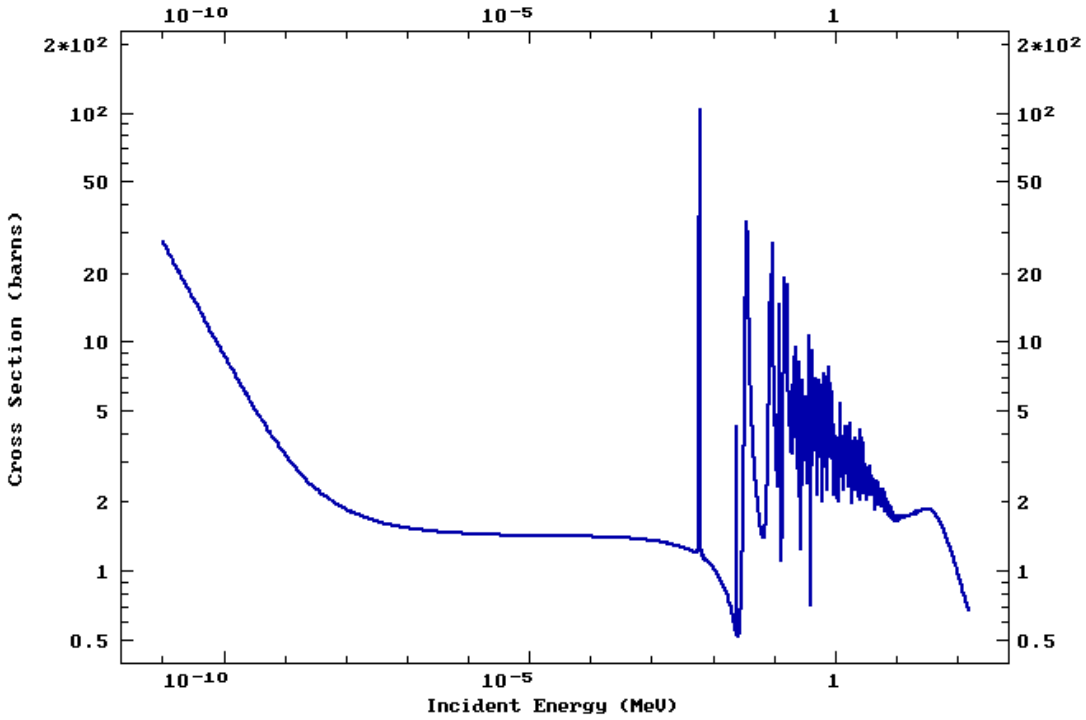

Figure 1. Total cross section (barns) for ^{27}Al, 100 μeV to 150 MeV. Resonances (peaks and valleys) appear in the keV through MeV range.

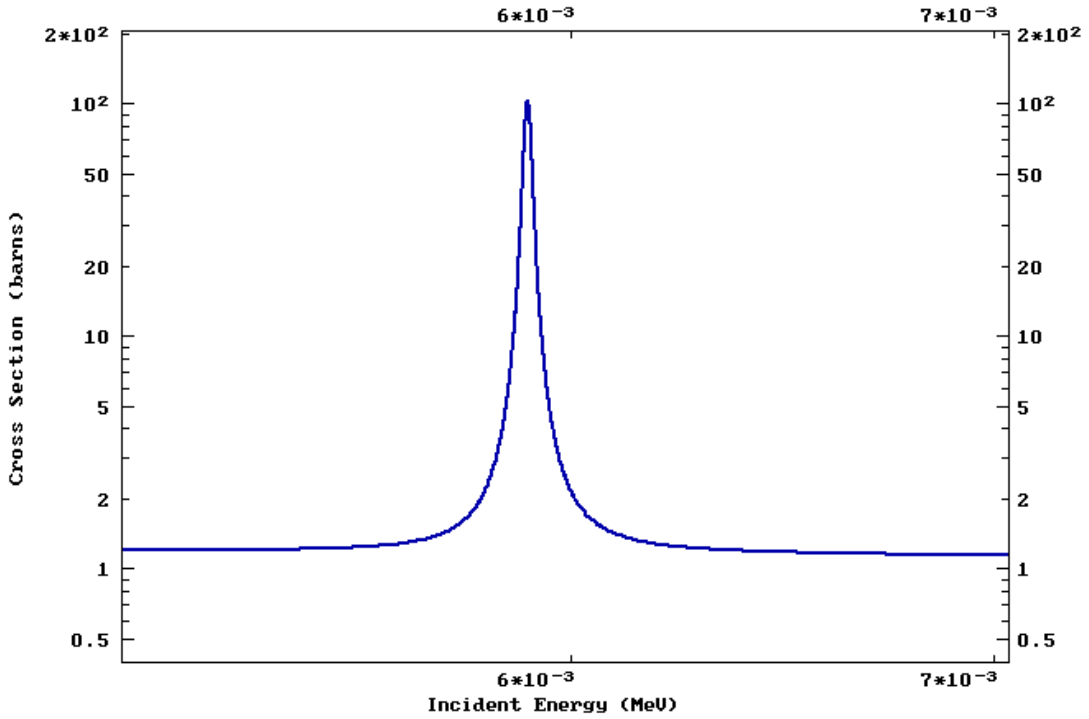

Figure 2. 5.9 keV resonance peak in ^{27}Al. The resonance peak rises to ~100 b from a local representative value of ~1 b.

The "problem" of resonances can be somewhat alleviated by the use of flux-weighted energy group cross sections and group transfer functions [Reference 2]. A flux-weighted group cross section is calculated by:

$$\sigma_g = \frac{\int_{E_{Low}}^{E_{High}} \phi(E)\sigma(E)dE}{\int_{E_{Low}}^{E_{High}} \phi(E)dE} \qquad \text{(Equation 1)}$$

This standard approach seeks to conserve reaction rates within an energy group, essentially smoothing the resonance peaks and valleys throughout the energy group. It also seeks to conserve how neutrons scatter from one energy group to the others. However, this approach requires an *a priori* knowledge of the flux within the medium; this may not be a practical assumption, especially for space radiation events.

Additionally, the use of standardized cross section group structures (such as the common 27 group or 238 group structures) requires each isotope to use the same energy limits for each group. This is likely not the optimal method for

assuring accuracy in calculations, and increasing the number of groups to account for the inherent differences in isotopes simply moves closer to the use of the full continuous energy cross section sets.

Research And Initial Results

This research uses the notion of group cross sections but does not seek to conserve the reaction rates, nor does it seek to conserve the scattering group structure. Rather, these group cross sections conserve the integrated cross sections as a function of energy. Thus, the group cross section is then:

$$\sigma_g = \frac{\int_{E_{Low}}^{E_{High}} \sigma(E)dE}{\int_{E_{Low}}^{E_{High}} dE} \qquad \text{(Equation 2)}$$

The variables that require definition in Equation 1 are then E_{High} and E_{Low}, the limits of integration. These can be defined with the recognition that for elastic scattering, or for inelastic scattering with a definite energy loss, the neutron can only scatter into a defined energy range. For a neutron of initial energy E_i an elastic scatter off a nucleus of mass A will result in final energy E_f with:

$$\left(\frac{A-1}{A+1}\right)^2 E_i = \alpha E_i \leq E_f \leq E_i \qquad \text{(Equation 3)}$$

Thus, the neutron can only scatter into the energy range $\alpha \mathrm{E_i} \leq \mathrm{E_f} \leq \mathrm{E_i}$. This gives the proposed range over which the neutron cross section should be integrated. Since the neutron cross section data is given in point-wise, linear interpolation data tables, this is a straightforward task.

The resulting energy group structure is this: given the initial (maximum) energy E_0 of the cross section data set, the first group has endpoints $(\alpha E_0, E_0)$. The next possible interval for the neutron to scatter into (assuming it scatters to the low end of the interval) is $(\alpha^2 E_0, \alpha E_0)$; the third group is $(\alpha^3 E_0, \alpha^2 E_0)$, and so on.

This creates a problem-specific cross section set based on the raw cross section data. The number of groups n in this cross section set is a function of the maximum energy E_{max}, the minimum energy E_{min}, and the atomic mass. Given these values:

$$\alpha^n E_{max} = E_{min} \qquad \text{(Equation 4)}$$

Rearranging yields:

$$n = \frac{\ln \frac{E_{min}}{E_{max}}}{\ln a} \qquad \text{(Equation 5)}$$

For a space shielding problem with an initial energy of, for example, 150 MeV and a lower-bound cutoff energy of, for example, 10 μeV, there would be $\frac{\ln\frac{1e-5}{1.5e8}}{\ln\alpha}$ intervals. For ^{27}Al (α=0.8622), this yields 204 intervals. For comparison, the ENDF/B-VII cross section data set for ^{27}Al has 3729 data points.

Figure 3 shows the results of generating the "smoothed" cross section in ^{27}Al using this methodology.

This approach exactly conserves cross sections in ranges with little variation, for example less than 1 keV or greater than 10 MeV. Its applicability in the resonance region is of the primary interest. Figure 4 shows the resonance region for ^{27}Al in greater detail.

The two cross section sets in Figure 4 still show good agreement—the smoothed data tends to find the centerline in the resonance peaks around 1 MeV, but they follow exactly on the transition between peaks between 10 keV and 100 keV. It also spreads the peak at 5.9 keV, similar to Doppler broadening.

The next step is to evaluate whether the cross section set can be made smoother. This can be accomplished through the use of a "stretching" factor $k > 1$ applied to the cross section interval. This changes the initial energy range to $\left(\frac{\alpha}{k}E_0, E_0\right)$, the second to $\left(\left(\frac{\alpha}{k}\right)^2 E_0, \frac{\alpha}{k}E_0\right)$, and so on.

This also changes the number of energy groups. Keeping the original ^{27}Al smoothing example as k=1, using k=2 yields 36 groups, k=3 yields 24 groups, and so on, given by $\frac{\ln\frac{1e-5}{1.5e8}}{\ln\frac{\alpha}{k}}$. Figure 5 shows the smoothed and stretched cross sections for ^{27}Al using k=5.

The stretched, smoothed cross section shows good agreement in the 1 MeV resonance area, but the agreement may be questionable in the 10 keV to 100 keV range; it may be too gross of a smoothing. Similarly, the smoothed cross section misses the peak at 5.9 keV.

Purpose Of Research

This research analyzes the application of these smoothed cross section and establishes whether this approach can be used as a viable and practical alternative to standard multigroup or point-wise continuous energy cross section data sets.

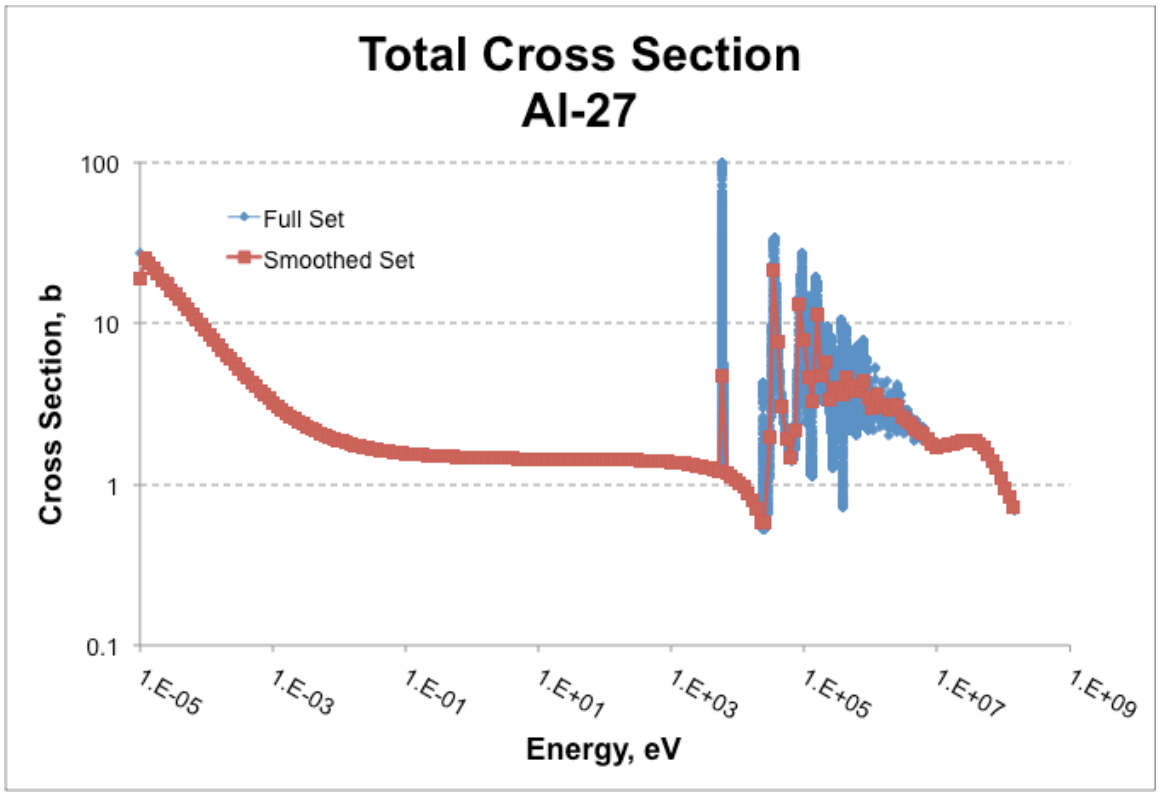

Figure 3. Total cross sections for ^{27}Al. The blue dots show the full ENDF/B-VII data set; the red squares show the data set resulting from the "smoothing" algorithm.[1]

[1] Figures in the remainder of the dissertation will omit the discrete points and show only the connecting lines for clarity.

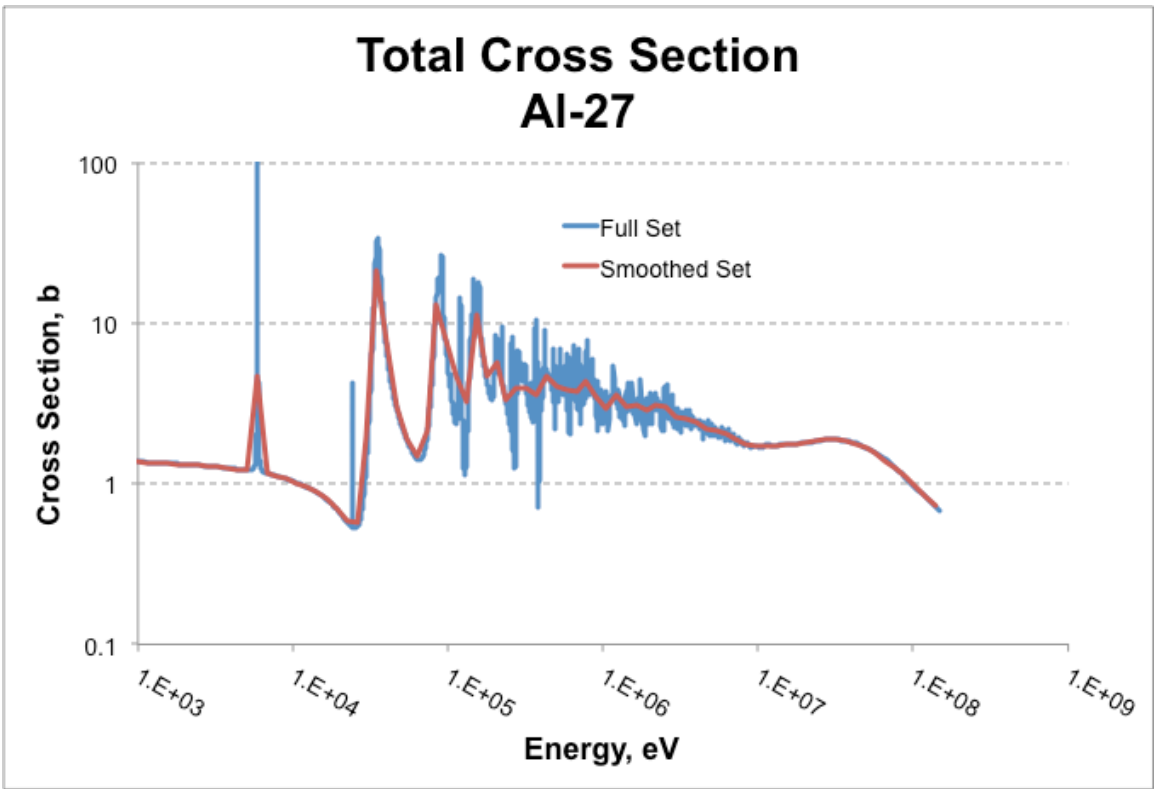

Figure 4. Comparison of the full ENDF/B-VII and smoothed total cross section data sets in the resonance range. The smoothed set (red line) follows the approximate centerline of the full data set (blue line).

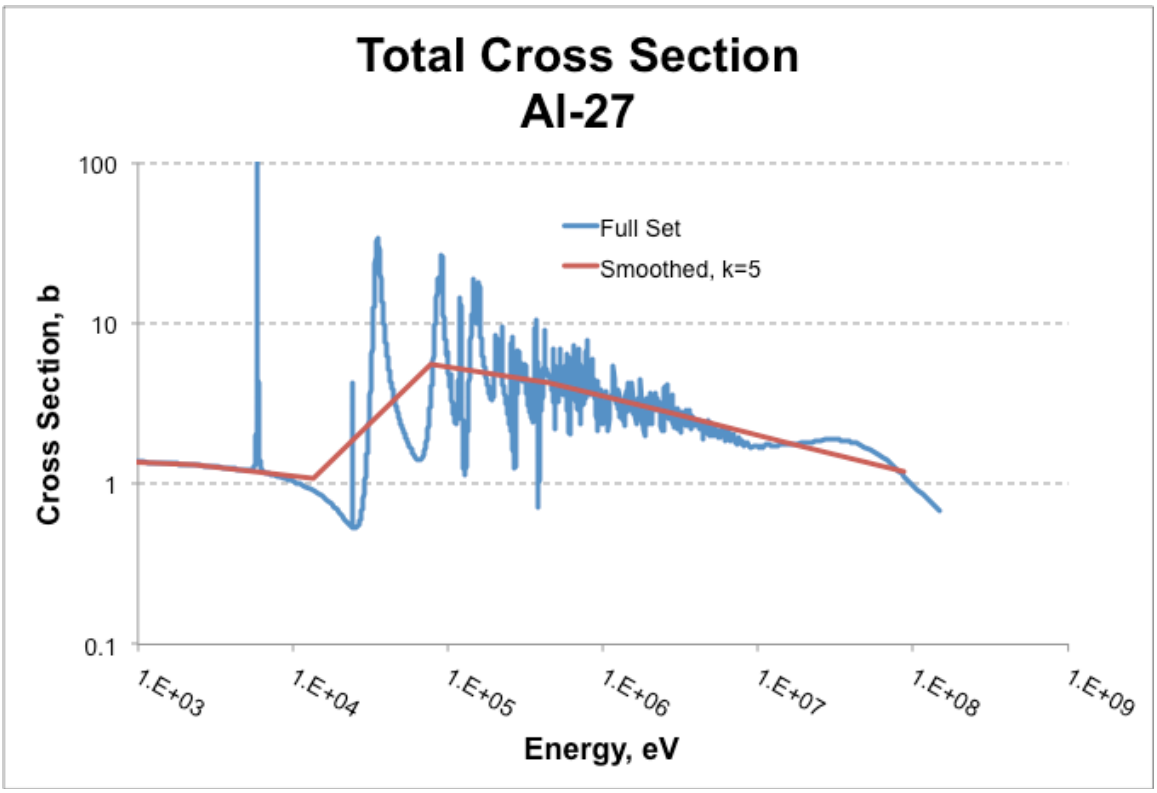

Figure 5. Comparison of the full ENDF/B-VII and the stretched (k=5) smoothed total cross section data sets in the resonance range. The smoothed set (red line) still follows the approximate centerline of the full data set (blue line), but it has more coarse agreement in the 10 keV to 100 keV range.

CHAPTER II
LITERATURE REVIEW

Research In Context

The fidelity and accuracy of any radiation transport code depends significantly on the quality of the data describing the underlying interaction channels, their relevant outcomes, and the probabilities for each event. These data sets typically take the form of cross section data sets for neutron transport calculations.

In many transport calculations, the cross section data sets are not used directly. Rather, they are converted into flux-weighted groups in order to conserve overall reaction rates, typically based on a known benchmark case. The problem of using the "correct" flux distribution has been acknowledged since the earliest reactor physics were performed, especially with respect to hard spectrum, high energy systems, such as fast reactors [Reference 3].

In addition, the problem of resonances in self-shielding requires a detailed analysis of the covariance of the cross section data sets and multigroup approximations [Reference 4].

Evaluation Of Some Existing Shielding Codes And Benchmarks

Two commonly used radiation transport codes used for radiation shielding calculations are the SCALE suite of codes [Reference 5] and the MCNP suite of codes [Reference 6]. These shielding codes have progressed through their continuous development from few-group cross section sets (e.g., 27-group ENDF4) to the current use of continuous energy cross sections (which are typically converted to a group structure for reporting calculation results). These codes typically have maximum neutron energy groups in the 10s of MeV, and as such are most useful for fusion, spallation[2], fission, and neutron source calculations, in order of descending neutron energy. Space radiation transport codes such as GEANT4 [Reference 7] and OLTARIS [Reference 8] are capable of tracking higher-energy neutrons, but establishing a neutron flux spectrum for the generation of flux-weighted group cross sections is problematic.

---

[2] Spallation neutron energies follow a distribution with a very long tail at high energies, up to and including the incident proton energy for single-neutron spallation events. However, the largest part of the distribution falls into the 1s to 10s of MeV range.

Benchmarking and validation efforts for existing codes show the difficulty in performing accurate transport calculations. For example, the shielding benchmark analysis for SCALE4 shows calculations disagreeing with experiments on the order of up to 20% [Reference 9, 10] in the 1990s. Recent benchmarking efforts [Reference 11] show similar results.

Previous Cross Section Smoothing Work

The concept of smoothing cross sections has been previously noted and described by DeLauder, Townsend, *et al* [Reference 12]. That work introduced the problem of attempting to use flux-weighted cross sections to calculate particle transport for space radiation events; since each space radiation event has a unique and unpredictable flux spectrum, using a flux-weighted cross section data set generated to conserve reaction rates on a known and predictable system does not work.

The methodology proposed in Reference 12 uses an inverse weighting approach. That is, the width of the averaging window is inversely proportional to the energy point evaluated. This approach therefore uses a large averaging window for low energies and a small averaging window for high energies. This is a rational approach that recognizes that cross sections tend to be smoother at low energies, and therefore amenable to using a single characteristic value calculated from a large range of energies. Conversely, resonance regions tend to appear at higher energies; a smaller integration window preserves some of the shape of the resonances.

The results of that smoothing methodology were not evaluated in a transport code. A visual comparison of the smoothed cross sections in Reference 12 show that they resemble the smoothed cross sections generated through this research methodology.

CHAPTER III
MATERIALS AND METHODS

Definition Of Problem Solved

This research uses a simple problem as its standard. A monoenergetic beam of neutrons impinges orthogonally on a homogeneous shielding material with a finite thickness in the direction of initial travel and infinite thickness in the other two dimensions. The material has a given density (ρ) and molar mass (A). The direction of initial travel is denoted *x*, while the other directions are *y* and *z*.

The neutron travels until it either interacts with the shielding material or escapes from the front (reflected) or back (transmitted) surface. If there is an interaction, the only reaction options are isotropic and elastic scatter, or absorption.

The underlying cross section data are point-wise ENDF/B-VII total (σ_T) and elastic scatter (σ_E) microscopic cross sections. This research assumes that the absorption (σ_A) cross section is the difference:

$$\sigma_T - \sigma_E = \sigma_A \qquad \text{(Equation 6)}$$

This is obviously a simplifying assumption that introduces error for isotopes with large inelastic scatter (σ_{In}) cross sections; it also does not account for neutron production in (n,2n) or (n,3n) reactions, nor does it account for any secondaries. The macroscopic cross section is:

$$\Sigma = \frac{\rho N_A}{A}\sigma \qquad \text{(Equation 7)}$$

where (N_A) is Avogadro's Number.

When a neutron scatters, its location and initial and final energies are recorded. When a neutron is absorbed, its location and final energy are recorded. When a neutron escapes, its energy and disposition (reflected or transmitted) are recorded.

Cross Section Processing

The neutron cross section data sets are given in point-wise data pairs of energy and cross section (E, σ). However, the energy points provided for total cross section may not match the energy point provided for the elastic scattering cross section. For example, the total cross section data set for ^{27}Al has 9785 energy points, while the elastic scattering cross section has 3729 energy points.

The first step is to “condition” the cross section sets so that the point-wise integration uses consistent endpoints.

The conditioning process reads the data sets and generates an overall set of energy points by taking the union of the total and elastic scatter cross sections. The conditioning process then interpolates on the original cross section data to create a full set of both total and elastic scatter cross sections. The final step is to generate the absorption cross section as the arithmetic difference between the total and elastic scatter cross sections at each energy point.

Smoothing Methodology

The cross section data in the ENDF/B-VII data sets is presented as point-wise values with linear interpolation for both energy and cross section. Thus the interpolated cross section σ_I at an intermediate energy E_I between two adjacent, explicit points $[(E_0, \sigma_0), (E_1, \sigma_1)]$ (with $E_0 < E_1$)[3] can be calculated as being on the line:

$$\sigma_I - \sigma_0 = \left(\frac{\sigma_1 - \sigma_0}{E_1 - E_0}\right)(E_I - E_0) \qquad \text{(Equation 8)}$$

This can be rearranged to yield:

$$\sigma_I = \left(\frac{\sigma_1 - \sigma_0}{E_1 - E_0}\right) E_I - \left(\frac{\sigma_1 - \sigma_0}{E_1 - E_0}\right) E_0 + \sigma_0 \qquad \text{(Equation 9)}$$

Integrating this function over E_I with the limits (E_0, E_1) yields:

$$\int_{E_0}^{E_1} \sigma_I dE_I = \int_{E_0}^{E_1} \left(\left(\frac{\sigma_1 - \sigma_0}{E_1 - E_0}\right) E_I - \left(\frac{\sigma_1 - \sigma_0}{E_1 - E_0}\right) E_0 + \sigma_0 \right) dE_I \qquad \text{(Equation 10)}$$

Solving this integral yields:

$$\int_{E_0}^{E_1} \sigma_I dE_I = (\sigma_1 + \sigma_0)\left(\frac{(E_1 - E_0)}{2}\right) \qquad \text{(Equation 11)}$$

This results in a value describing the energy-integrated cross section (cross section*energy) with units of b*eV.

[3] This contradicts typical convention of $E_1 < E_0$ for downscattering; this is inconsequential for the derivation.

Alternatively, the assumption of linear interpolation between the points immediately yields the average cross section between the two points as:

$$\bar{\sigma} = \frac{\sigma_1+\sigma_0}{2} \quad \text{(Equation 12)}$$

and the integrated (cross section*energy) value (with units of b*eV) of:

$$\bar{\sigma}(E_1 - E_0) = \frac{(\sigma_1+\sigma_0)}{2}(E_1 - E_0) \quad \text{(Equation 13)}$$

This agrees with the integrand in Equation 11.

This implies that a single average cross section value in the interval $[E_0, E_1]$ would conserve the integrated (cross section*energy) value. Expanding this implication and approach to multiple consecutive data points thus provides the basis for the cross section smoothing methodology.

For example, integrating two intervals defined by three data points $[(E_0, \sigma_0), (E_1, \sigma_1), (E_2, \sigma_2)]$ would yield an overall interval average cross section:

$$\bar{\sigma} = \frac{(\sigma_1+\sigma_0)(E_1-E_0)+(\sigma_2+\sigma_1)(E_2-E_1)}{2(E_2-E_0)} \quad \text{(Equation 14)}$$

This is easily extended to any number of data points, including data points that are generated through interpolation of the explicit ENDF/B-VII data.

The cross section is then assigned to the midpoint:

$$E_{mid} = \frac{E_{lower}+E_{upper}}{2} \quad \text{(Equation 15)}$$

of the energy group, yielding the point $(E_{mid}, \bar{\sigma})$. To evaluate cross sections for energy values that do not coincide with energy group midpoints, linear interpolation is used.

A different methodology would use the group cross section $\bar{\sigma}$ for any neutron with an energy in the energy interval; however, the linear interpolation method allows the cross section data to avoid large discontinuities which would approximate resonance behavior, or would mimic the standard flux-weighted group structure.

Handling Of Scattering Angles

This methodology is a proof of principle of the comparability of smoothed cross sections with respect to multigroup or continuous energy cross section

sets. As such, it uses isotropic scattering distributions for all scattering angle calculations. Depending on the target isotope and neutron energy, this assumption may cause divergence relative to existing benchmarked transport codes. This assumption is a recognized problem in calculations involving light nuclei [Reference 13], but for the purposes of this research that issue will be set aside.

Future work can incorporate a methodology to include anisotropic scattering. However, the use of a single group-wise polar scattering angle may not be advised.[4] This contrasts with the desired use of a single group-wise cross section; the probability of interaction defines how deeply into a material the neutron can penetrate, but the depth is along whatever scattering angle the calculation method chooses. For some very narrowly-defined problems, moving to a 1-group or 2-group cross section data set is acceptable [Reference 14], but the ability to "point" the scattered (or generated) neutrons in more than a single direction, even when accounting for an isotropic azimuthal angle, is vital in tracking neutrons through the full problem space; at the very least, backscatter must be accounted for; this will become apparent when examining the benchmark results.

Neutron Tracking In 3-Dimensional Space

The distance the neutron travels (d) is calculated by drawing a random number (r) from a uniform [0,1] distribution and calculating:

$$d = \frac{-\ln r}{\Sigma_T} \qquad \text{(Equation 16)}$$

The interaction type is determined by drawing another random and comparing it to the absorption probability ($\frac{\sigma_A}{\sigma_T}$). If the random number is less than the absorption probability, the neutron is absorbed.

When the neutron scatters, the cosine of the polar scattering angle (μ) is chosen from a uniform distribution between [-1,1]; this assumption has been previously noted as an incorrect and simplifying assumption. The azimuthal scattering angle is chosen from a uniform distribution [0,2π].

For the initial interaction, tracking the neutron in the global (*x, y, z*) coordinate system is straightforward: the polar angle is measured with respect to the *x*-axis. For subsequent reactions, the polar angle is measured with respect to the

---

[4] Average polar scattering angles (typically π/2) are used in calculating moderating ratio, for example.

direction of neutron motion (typically referred to as $\hat{\Omega}$). The distance to interaction is calculated as normal, but the location must be converted from relative position to global coordinates. The conversion to global coordinates is accomplished with a rotation matrix; an example of rotation matrices can be found in Reference 15, and the implementation of the rotation matrix is included in the Appendix.

If the global coordinate places it outside the shielding material, it is recorded as having escaped.

Values Tracked And Recorded

Several values are tracked and recorded in order to provide a methodology for comparing the results from the different cross section data sets.

Number Of Scattering Collisions

The number of scatting collisions gives a measure of whether the absorption cross section is dominating for any of the cross section data sets generated.

Linear And Radial Distance Traveled

The linear distance traveled is the total summation of distance a neutron travels before escaping or being absorbed. This can be used to determine the linear energy loss for the neutrons. The radial distance is the distance from (0, 0, 0) at the final non-escape interaction. This gives some measure of the diffusion (not necessarily in the sense of diffusion theory) of the neutrons through the shielding.

Final Energy

The final energy is the energy at which the neutron is either absorbed or escapes. This is also useful in determining whether an absorption resonance peak appears in one data set exclusively of the others.

Deepest Penetration

The deepest penetration is the maximum x coordinate for any neutron history. This includes any transmitted neutrons.

Uncertainty Analysis

Since the results are tallied over multiple independent runs, they form a distribution of results. The uncertainty is tracked cumulatively and updated with each history.

Transmission And Reflection Probabilities And Spectra

The primary parameters tracked and recorded are the transmission and reflection probabilities and spectra. The probabilities are simply calculated as the number of neutrons that transmit, or reflect, divided by the total histories. The spectra are calculated by binning a histogram of final energies for both cases.

The spectra also serve as the basis for comparison with a SCALE benchmark case.

CHAPTER IV
RESULTS AND DISCUSSION

Materials Of Interest

The demonstration of the method is not intended to calculate and validate a full suite of neutron transport problems. Instead it is intended to demonstrate a consistency of results using the smoothed cross section sets. However, it is useful to compare the results of this methodology to results obtained using existing transport codes for materials of interest.

Some materials of interest are shown below, including the ENDF/B-VII cross sections. The cross sections are:

1) Total cross section; this is the summation of all individual interaction channels
2) Elastic cross section; this is the elastic scattering cross section
3) Non-elastic cross section; this is the summation of all cross sections other than the elastic scattering cross section, including absorption cross sections and scatters that are not elastic
4) Inelastic cross section; this is the summation of all scatters that are not elastic

Cross sections 3) and 4) are provided for reference; only 1) and 2) were used in this methodology. Cross section plots come from [Reference 1].

Carbon (C)

Carbon is a common neutron moderator and is increasingly a material of interest for lightweight, high-strength construction. The isotope ^{12}C is the major isotopic component of natural carbon at 98.9%; however, the ENDF/B-VII data is technically given for elemental carbon. Figure 6 shows the total, elastic, non-elastic, and inelastic cross sections for elemental carbon.

The non-elastic cross sections are dwarfed by the elastic cross sections to nearly 10 MeV. Further, the inelastic scatter cross section data does not appear below ~4.8 MeV. Figure 7 shows more detail for elemental carbon cross sections in the 1-10 MeV range.

Figure 8 shows the comparison of the total and elastic scatter cross sections from the original ENDF/B-VII.1 set along with the smoothed cross sections for k=1.

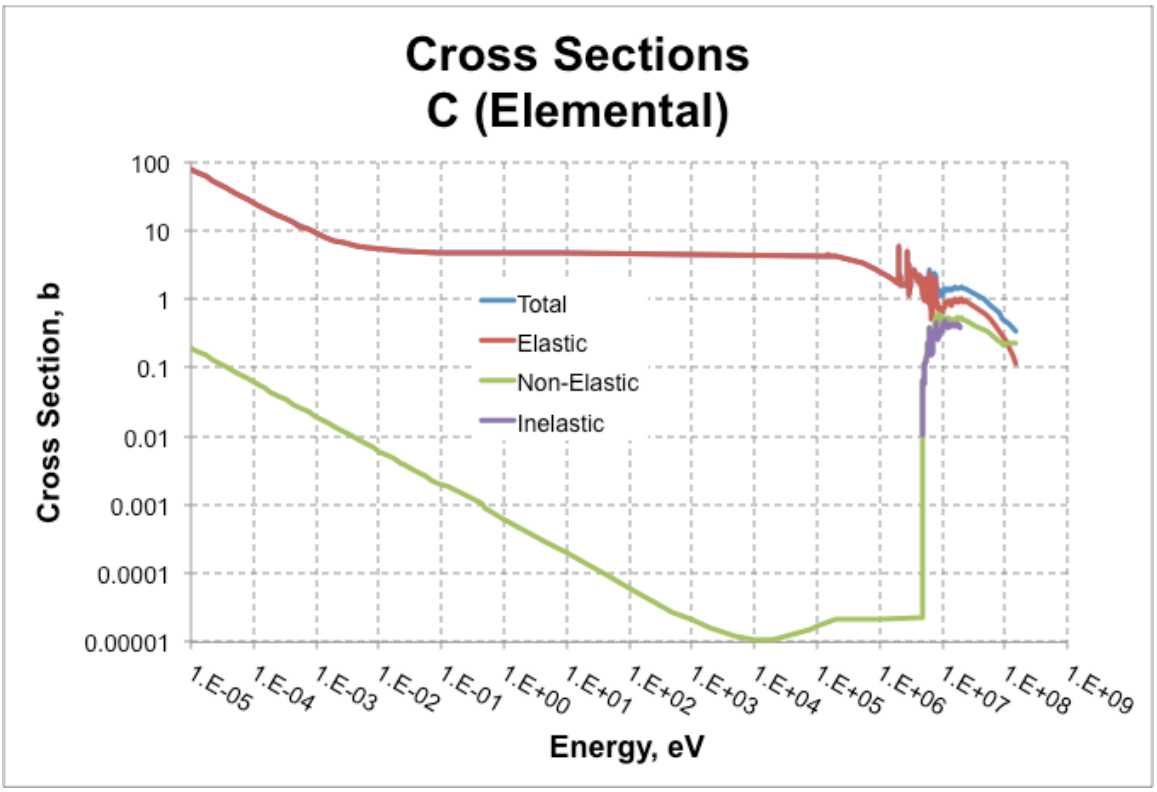

Figure 6. Cross sections for Elemental C. The total cross section (blue line) is the sum of elastic (red) and non-elastic (green) values. The inelastic (purple) is a subset of the non-elastic and implies a scattering event with an energy loss leading to an excited state in the target nucleus. The total cross section is dominated by elastic scattering, but non-elastic collisions become more important in the MeV range. At 100 MeV, the non-elastic cross section surpasses the elastic cross section as the dominant interaction.

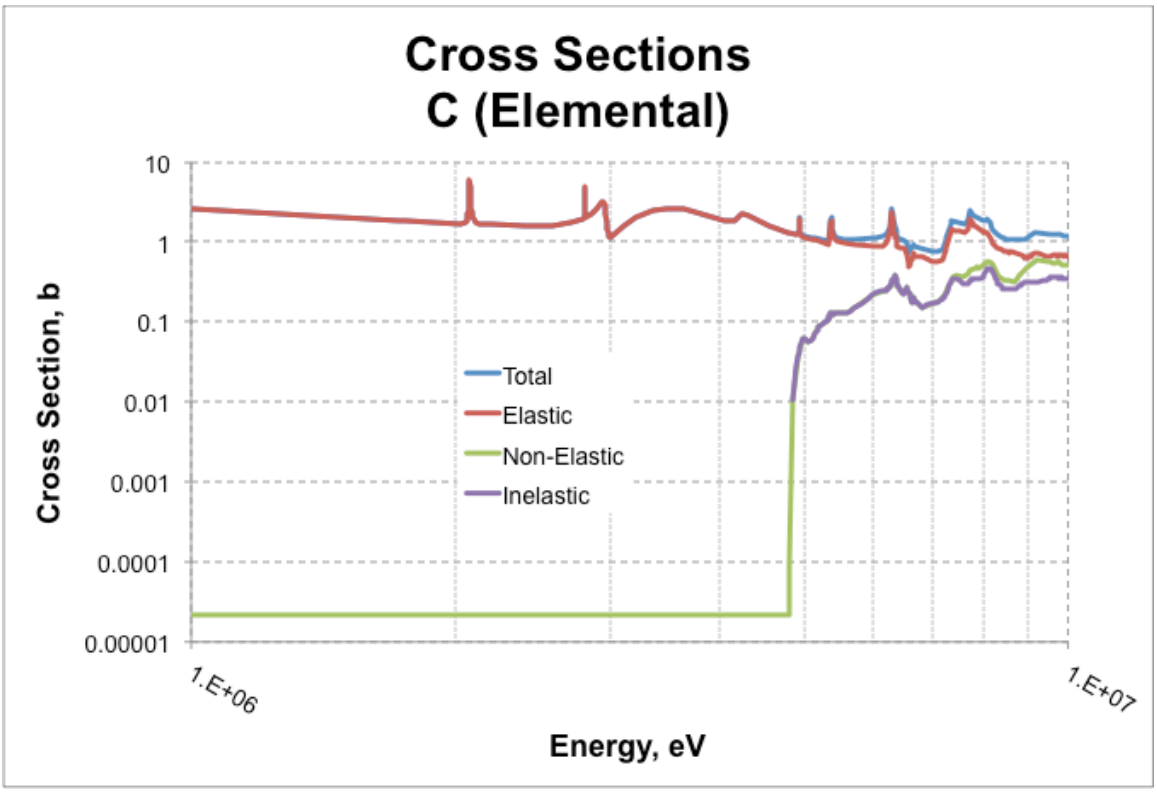

Figure 7. Cross sections for Elemental C, 1 MeV – 10 MeV. The non-elastic cross section is effectively 0 until ~4.5 MeV, and it is dominated by inelastic scattering until ~7.5 MeV. The non-elastic contribution to the total cross section becomes evident around 5 MeV when the total and elastic cross sections show some separation.

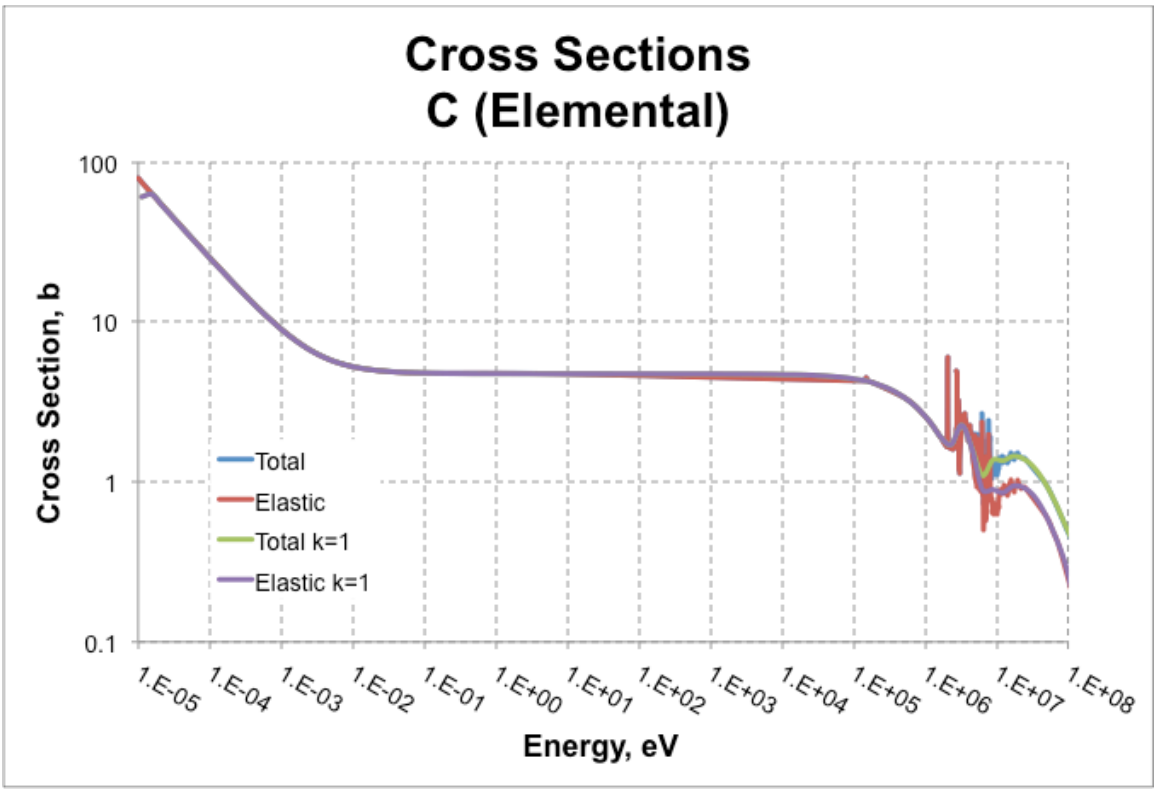

Figure 8. Comparison of ENDF/B-VII and smoothed (k=1) total and elastic cross sections for elemental C through 100 MeV. The two sets coincide into the MeV range and again in the 10 MeV to 100 MeV range; the 1 MeV to 10 MeV range includes a large number of resonances.

The cross sections show perfect agreement from 10 μeV to ~2 MeV. However, this agreement is trivial—there are no resonance peaks in that range. Rather, the resonance region for elemental carbon starts in the MeV range. Figure 9 shows the cross sections in the 1-10 MeV range.

A comparison of the number of data points that describe the original and smoothed data sets shows a large compression of data. Table 1 shows that comparison for k=1, k=2, and k=5.

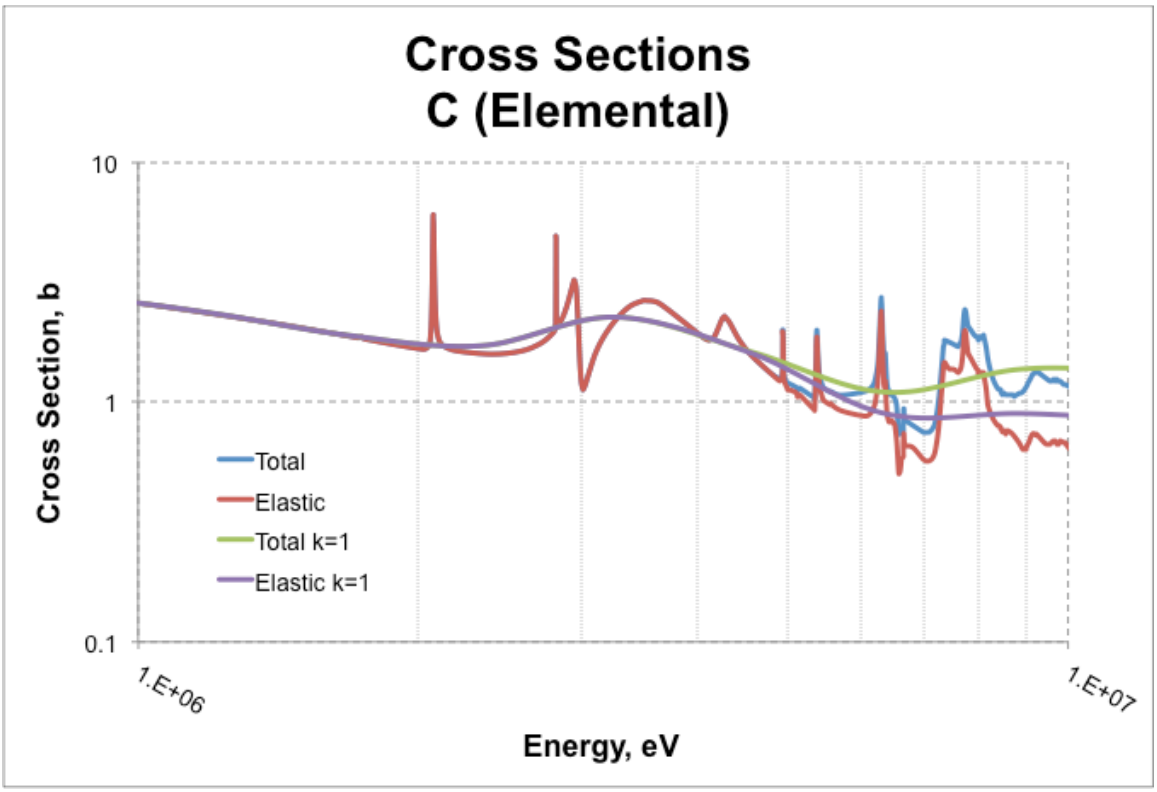

Figure 9. Comparison of ENDF/B-VII and smoothed (k=1) total (blue and green) and elastic (red and purple) cross sections for the resonance region of elemental C. The smoothed cross sections (green and purple) track the divergence in the total and elastic cross sections due to the increasing non-elastic cross sections at ~4.5 MeV.

Table 1. Carbon cross section set comparison, ENDF/B-VII.1 vs. Smoothed

Cross Section Data Set	Cross Section	Number of Points
ENDF/B-VII.1	Total	321
	Elastic Scatter	328
Smoothed, k=1	Total	92
	Elastic Scatter	92
Smoothed, k=2	Total	30
	Elastic Scatter	30
Smoothed, k=5	Total	16
	Elastic Scatter	16

The table shows that the number of data points has been reduced by a factor of ~3.5 in moving from the full set to k=1. Further reductions are achieved by moving to k=2 and k=5.

Iron (^{56}Fe)

Iron is used to make steel, which is a common structural material. ^{56}Fe represents 91.8% of iron. Figure 10 shows its total, elastic, non-elastic, inelastic, and (n,γ) cross sections for ^{56}Fe. The resonance region for ^{56}Fe starts at a lower energy (10s of keV) compared to the carbon cross sections, and there are far more resonance peaks to account for as well.

Figure 11 shows the comparison between the point-wise ENDF/B-VII cross sections and the smoothed k=1 cross sections. The smoothed cross sections follow the midline of the resonance region, but closer examination in Figures 12 and 13 show that some of the detail from the resonances may not be captured in the smoothed cross sections.

Figure 12 shows the cross section comparison at the resonance peak at 1.15 keV. The total and elastic cross sections from the ENDF/B-VII data separate at that resonance peak; the (n,γ) peak at that energy in Figure 11 explains that difference. The k=1 cross sections show divergence as well, but the magnitude of the cross section and the magnitude of the separation does not match the ENDF/B-VII data.

Figure 13 shows the cross section comparison in the resonance region from 10 keV to 10 MeV. The k=1 cross sections follow the midline well, but they do not necessarily capture the sharp depths of the resonance valleys, nor the sharp heights of the resonance peaks.

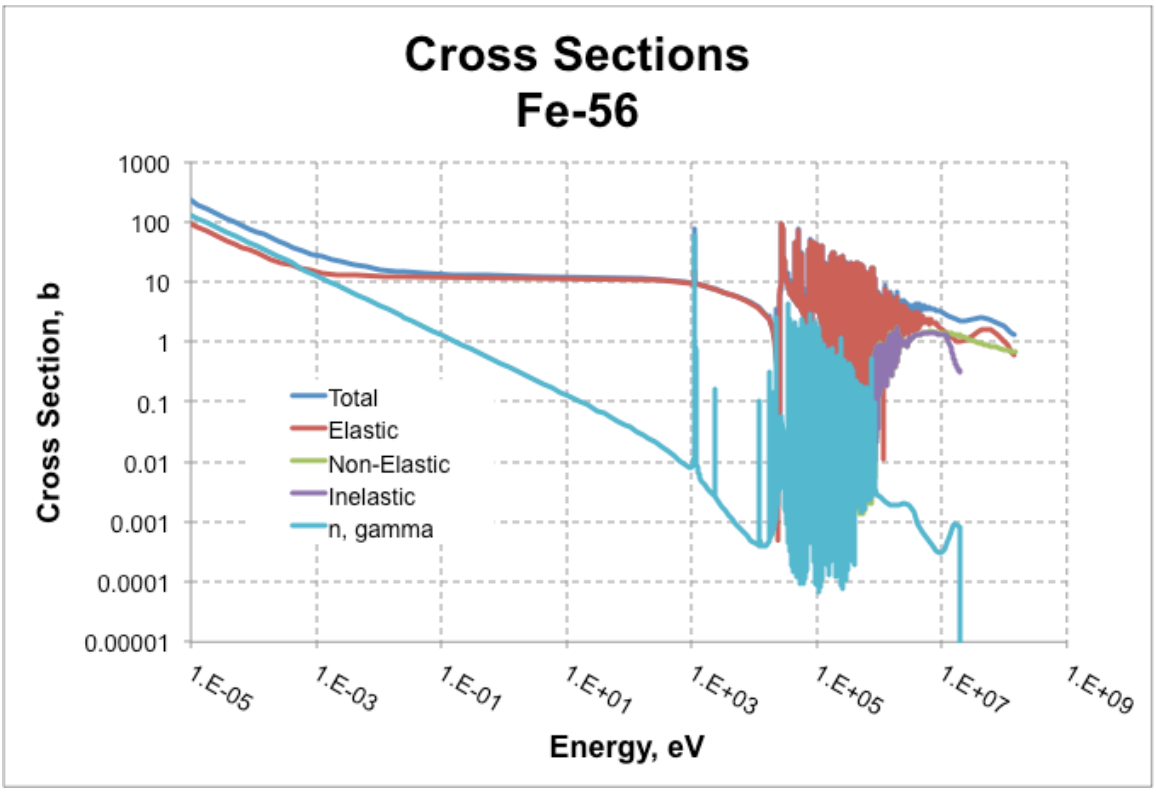

Figure 10. Cross sections for ^{56}Fe. The total cross section (dark blue line) is the sum of elastic (red) and non-elastic (green) values. The inelastic (purple) is a subset of the non-elastic and implies a scattering event with an energy loss leading to an excited state in the target nucleus. The total cross section is dominated by elastic scattering in the 0.1 eV to 1 MeV range. At < 0.1 eV, the difference between the total and elastic cross section is via (n,γ) (light blue).

Table 2 shows the data compression for ^{56}Fe. Moving from the ENDF/B-VII data to the k=1 data shows a compression of ~20x, and moving from k=1 to k=5 shows an additional compression of ~20x.

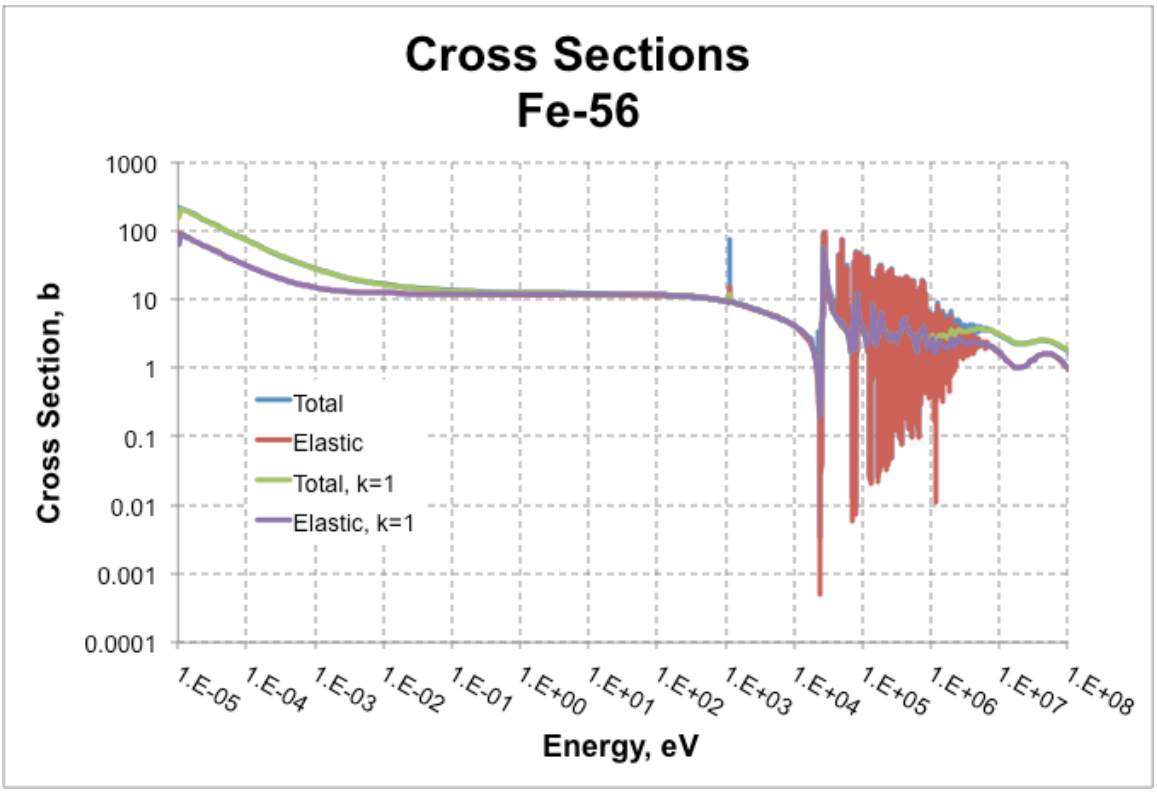

Figure 11. Comparison of ENDF/B-VII and smoothed (k=1) total and elastic cross sections for ^{56}Fe through 150 MeV. The two sets coincide into the 10 keV range and again in the 5 MeV to 100 MeV range; the 10 keV to 5 MeV range includes a large number of resonances.

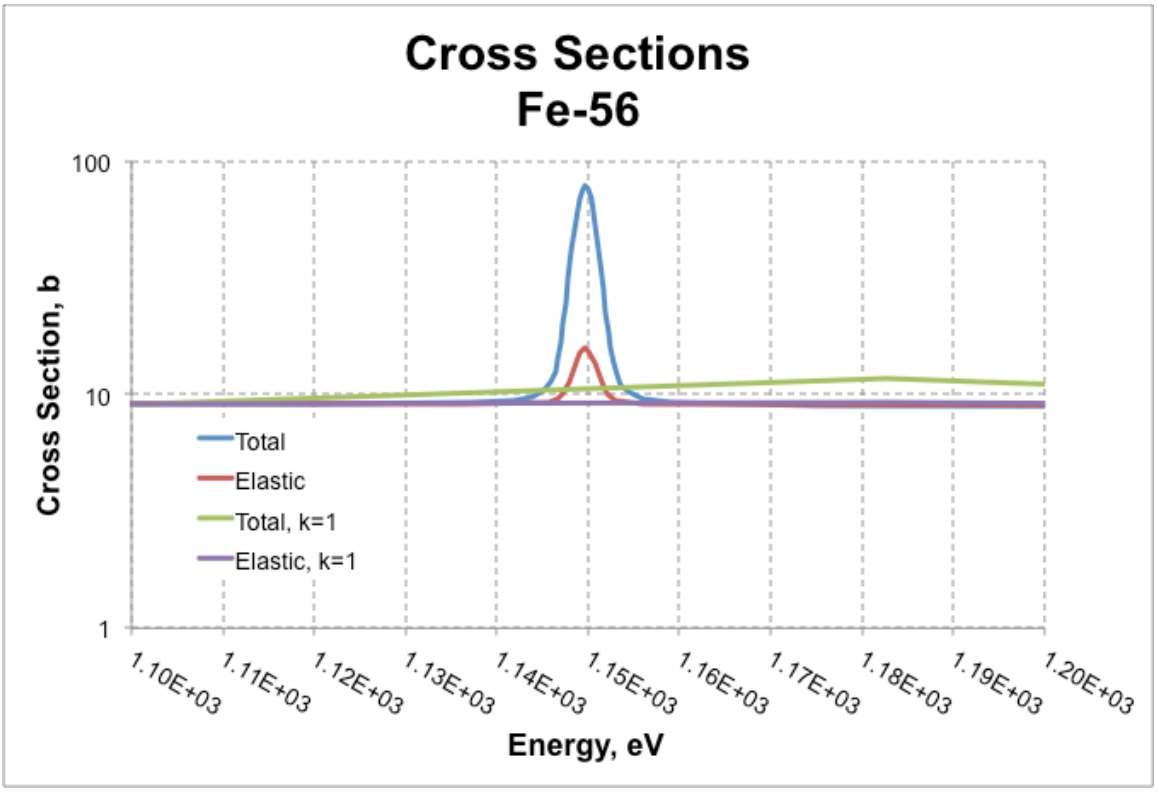

Figure 12. Comparison of ENDF/B-VII and smoothed (k=1) total (blue and green) and elastic (red and purple) cross sections for the 1.15 keV resonance of ^{56}Fe. The smoothed cross sections (green and purple) do not capture the resonance peak as a discrete point at 1.15 keV but instead broaden it with a smaller peak at 1.18 keV.

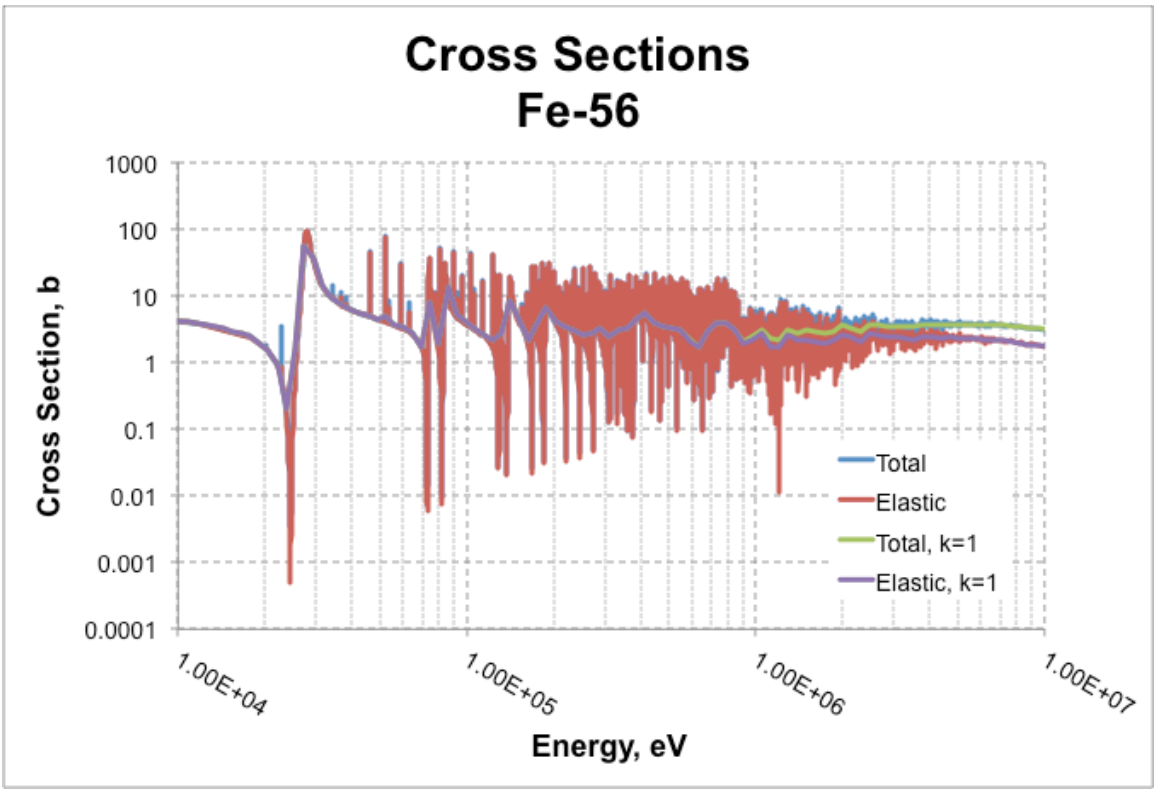

Figure 13. Comparison of ENDF/B-VII and smoothed (k=1) total (blue and green) and elastic (red and purple) cross sections for the 10 keV to 10 MeV resonance region of ^{56}Fe. The smoothed cross sections (green and purple) follow the general shape of the resonances, but they do not necessarily have the same magnitude, such as at the resonance valley at ~25 keV.

Table 2. ^{56}Fe cross section set comparison, ENDF/B-VII.1 vs. Smoothed

Cross Section Data Set	Cross Section	Number of Points
ENDF/B-VII.1	Total	8571
	Elastic Scatter	8822
Smoothed, k=1	Total	425
	Elastic Scatter	425
Smoothed, k=5	Total	19
	Elastic Scatter	19

Benchmarking Against SCALE

The benchmark cases use the MAVRIC sequence in SCALE [Reference 5]. The problem modeled in that case is a nearly monoenergetic, narrow-angle beam impinging upon a homogeneous target of finite dimension. The *y*- and *z*-dimensions are set at 1000 cm, making them essentially infinite in those directions. The transmitted and reflected flux spectra are collected on the far end and near end, respectively. The calculations use the 238-group ENDF-7 cross section library in SCALE, and they perform 10 batches of 13000 histories each. The sample input file appears in the Appendix. The 238-group structure also appears in the Appendix.

For comparison, this research methodology uses the smoothed cross section set and runs 10000 total histories. The neutron transport calculations are performed by a Python script, which also appears in the Appendix.

The benchmarking effort uses two metrics described below. Both are necessary to validate the smoothing methodology.

Transmitted and reflected flux spectra

The immediate application for space radiation shielding is to calculate survival probabilities for incident neutrons [Reference 12]. Thus, the calculation methodology as implemented explicitly tracks the survival (transmission) probability for any neutron that enters impinges on the shield material. Rigorously speaking, and defining “survival” as any outcome other than absorption, the true survival probability must include reflection as well. The implemented methodology tracks each neutron as transmitted, reflected, or absorbed, as well as the energy when each it is transmitted, reflected, or absorbed. Thus the transmitted spectrum is an energy histogram of all neutrons that pass through the shield without being absorbed, and the reflected spectrum is an energy histogram of all neutrons that leave the shield on the incident face.

The SCALE calculation does not treat the problem the same way; rather, it calculates group-wise flux in defined regions of the modeled space. For comparison with this research methodology, the flux is calculated in a region that includes the transmitting (back) face of the shield, as well as in a region that includes the reflecting (front) face of the shield.

The units for the research methodology are not typical units of flux, usually given in neutrons per unit area per unit time. Rather, they are simply recorded as a particle that survived. In order to compare the SCALE flux to the methodology survivors, the survivors are binned using intervals corresponding to the 238 group cross section set, then normalized based on the count for the most populous group. The same algorithm is used for the SCALE flux.

When plotted together, the transmitted spectra can be compared between the flux-weighted cross section results (from SCALE) and the smoothed cross sections results (from the research methodology).

However, this metric does not provide the full information needed to validate the method. While the metric provides information about the shape of the spectra, it does not provide information about the relative probabilities for transmission and reflection; the second metric will provide that.

Group-wise ratio of reflected-to-transmitted flux

The previous metric compared the reflected and transmitted spectra for the two different cross section sets. This metric compares the reflected and transmitted spectra within the same cross section set.

The motivation for this comparison is to estimate a measure of the relative probabilities for reflection and transmission. Instead of attempting to integrate flux to calculate a single value, the group-wise comparison of fluxes will stand in as a surrogate. The ratio of the reflected flux to the transmitted flux in each energy group provides some measure of the relative probabilities of a neutron, after having reached some energy, escaping the shield from the front face (reflection) or the back face (transmission).

The result is a spectral ratio that can be compared between the two results.

Benchmarking Results

Carbon results

The first benchmark case uses a monoenergetic beam of 4.5 MeV neutrons impinging on a target of carbon with varying thicknesses. A 4.5 MeV neutron energy was chosen to be less than the rising non-elastic and inelastic cross

sections, yet be in a region that experiences resonance activity (Figure 7). Further, 4.5 MeV exceeds the typical fission neutron energy of ~2 MeV, meaning that this test case would have direct fission reactor applications.

At 4.5 MeV, the total cross section is ~1.6 b. With a density of 2.25 g/cc and a molar mass of 12 g/mol, the total macroscopic cross section Σ_T=0.18 cm^{-1}, yielding a mean free path of ~5.54 cm.

Figure 14 shows the comparison of the MAVRIC reflected spectrum output with the research methodology k=0 (no smoothing; full cross section data set) and k=1 output for a 50 cm slab, which is ~9 mean free paths in thickness[5]. There is good agreement on the transmitted spectrum shape, except for a divergence at the 10-100 meV range.

Figure 15 shows the reflected spectra for k=0, k=1, k=2, and k=5. Moving from the explicit point-wise data for k=0 to the highly stretched and smoothed data in k=5 shows essentially no impact on the reflected spectra, which were shown to agree well with the MAVRIC results. However, there are some differences.

Figure 16 shows the total cross sections that correspond to the different k values in elemental carbon to attempt to explain those differences. Figure 16 shows, as expected, that increasing the stretching parameter k flattens the cross section set and loses some detail, which will cause some loss of accuracy. However, the reflected spectra in Figures 14 and 15 show good consistency in their shapes.

Figure 17 shows a comparison of the transmitted spectra for the 50 cm carbon slab for MAVRIC, k=0 data, and k=1 data. Figure 17 shows an agreed transmitted peak in the 10-100 meV range, but for k=0, and more so for k=1, the Python script predicts much higher transmission of keV to MeV neutrons. The k=0 spectrum is closer to the MAVRIC spectrum than the k=1 spectrum. However, the MAVRIC spectrum is the result of 130000 histories while the Python spectrum is the result of 10000 histories.

Figure 18 shows the transmitted spectrum for MAVRIC, k=0 with 50000 histories, and k=1 with 50000 histories. The figure shows the Python output for k=1 starting to converge with the MAVRIC output. This indicates that for relatively deep penetration problems, the Python script requires comparable histories to calculate comparable results.

---

[5] A shield of this thickness means there should be sufficient interactions within the shield to achieve good statistical results.

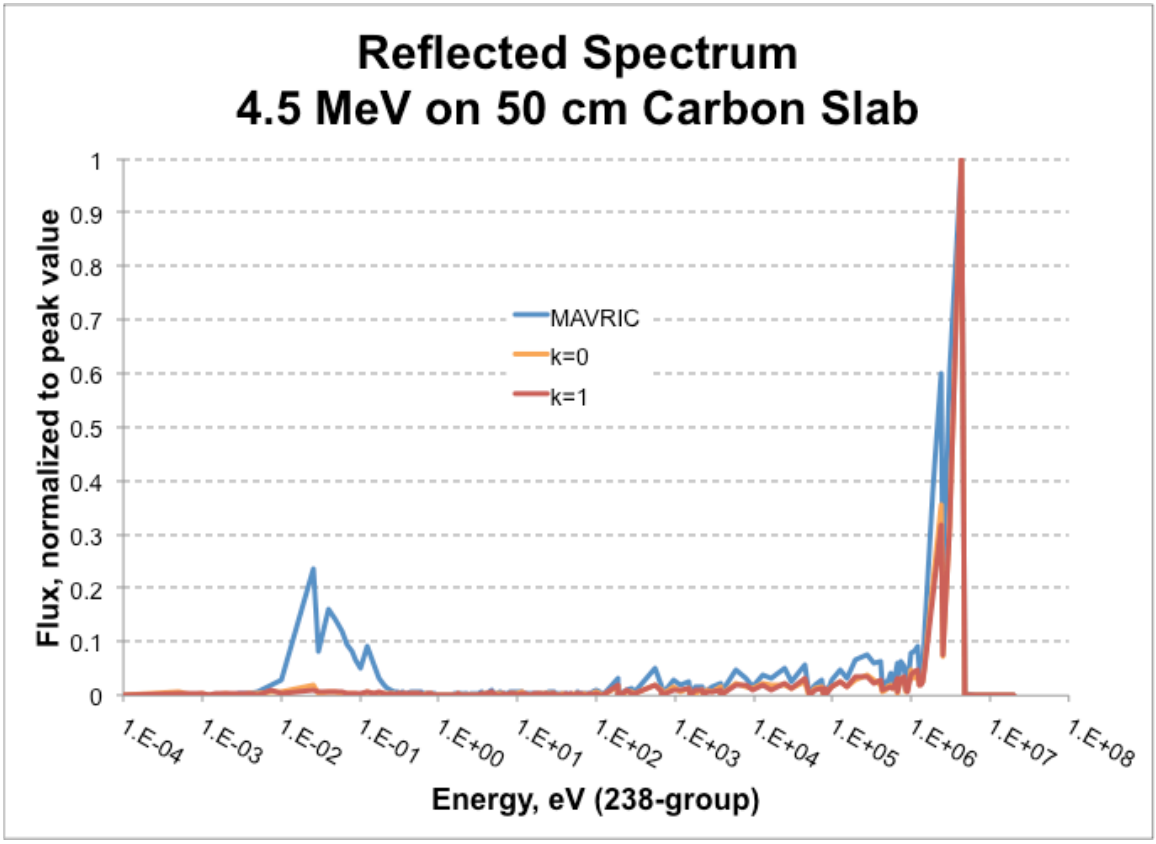

Figure 14. Comparison of reflected spectra for 4.5 MeV neutron incident on a 50 cm carbon slab. Both the MAVRIC (blue) and Python script calculations (orange and red) show a peak in Group 11 (3.0 to 4.3 MeV); this group includes the direct backscatter energy of 3.22 MeV. There is a divergence in the results in Groups 13 and 12 (1.85 to 2.48 MeV), corresponding to a resonance peak insufficiently treated at ~2 MeV. There is good agreement in the shape of the spectrum (with some minor offset) down to the 0.1 eV range; the MAVRIC results show a peak in the 1 meV to 100 meV range. The “k=0” data uses the full ENDF/B-VII cross section data, while “k=1” uses a smoothed data set.

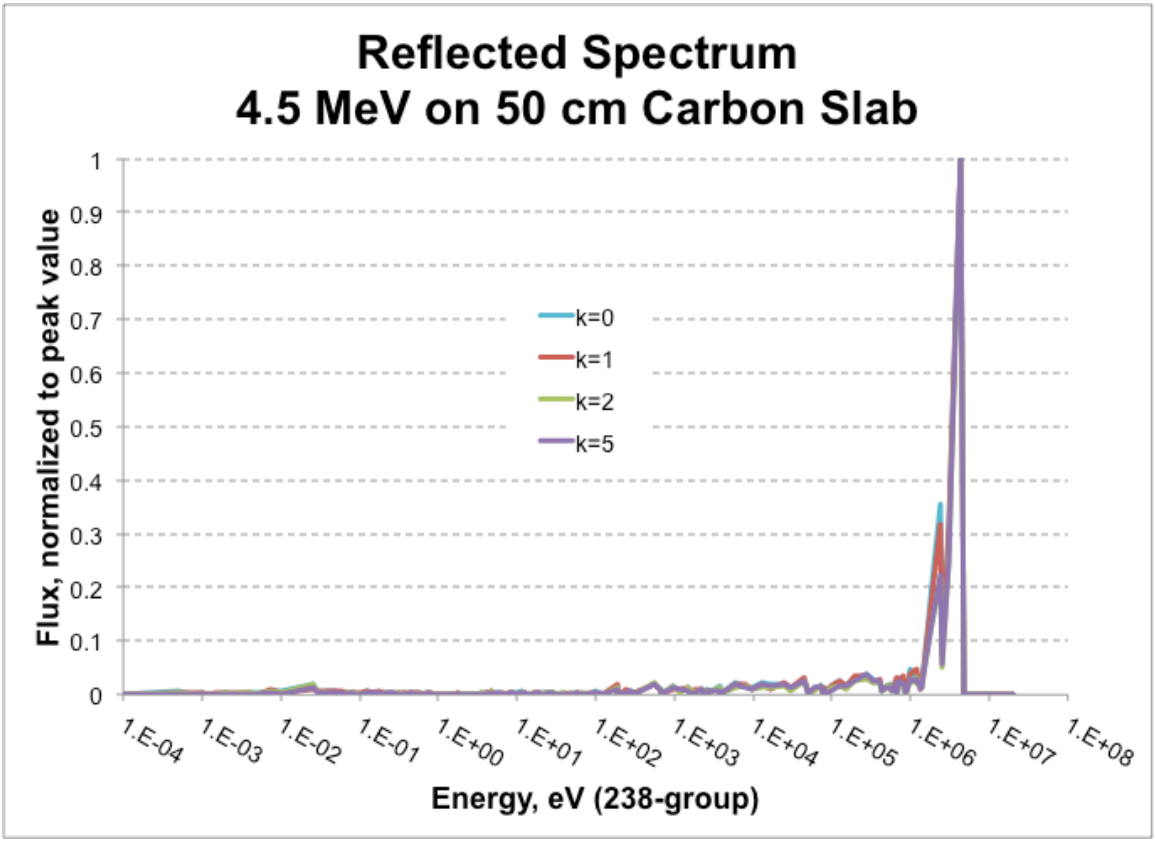

Figure 15. Comparison of reflected spectra using smoothed and stretched cross sections for 4.5 MeV neutron incident on a 50 cm carbon slab. All the Python script results are coincident with each other, with some minor variation between in Groups 13 and 14 (1.85 to 2.48 MeV); this corresponds to differing treatments of the resonance at ~2 MeV.

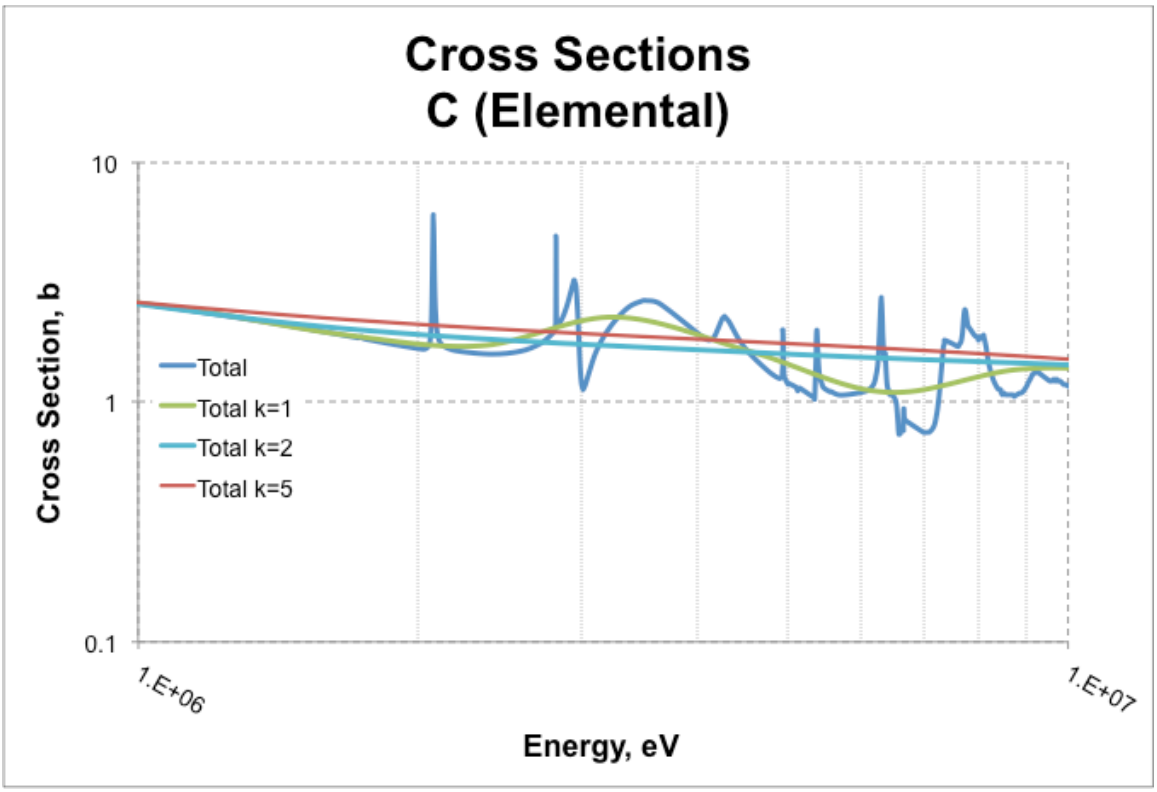

Figure 16. Comparison of ENDF/B-VII and smoothed total cross sections in the resonance region (1 MeV to 10 MeV) for elemental carbon. The k=1 (green) cross sections follow the overall contours of the full set (dark blue); the k=2 (light blue) and k=5 (red) cross sections show the effect of increasing the integration energy interval. As an example, an integration energy interval with upper energy 10 MeV integrates through 7.16 MeV with k=1; through 3.58 MeV with k=2; and through 1.43 MeV with k=5.

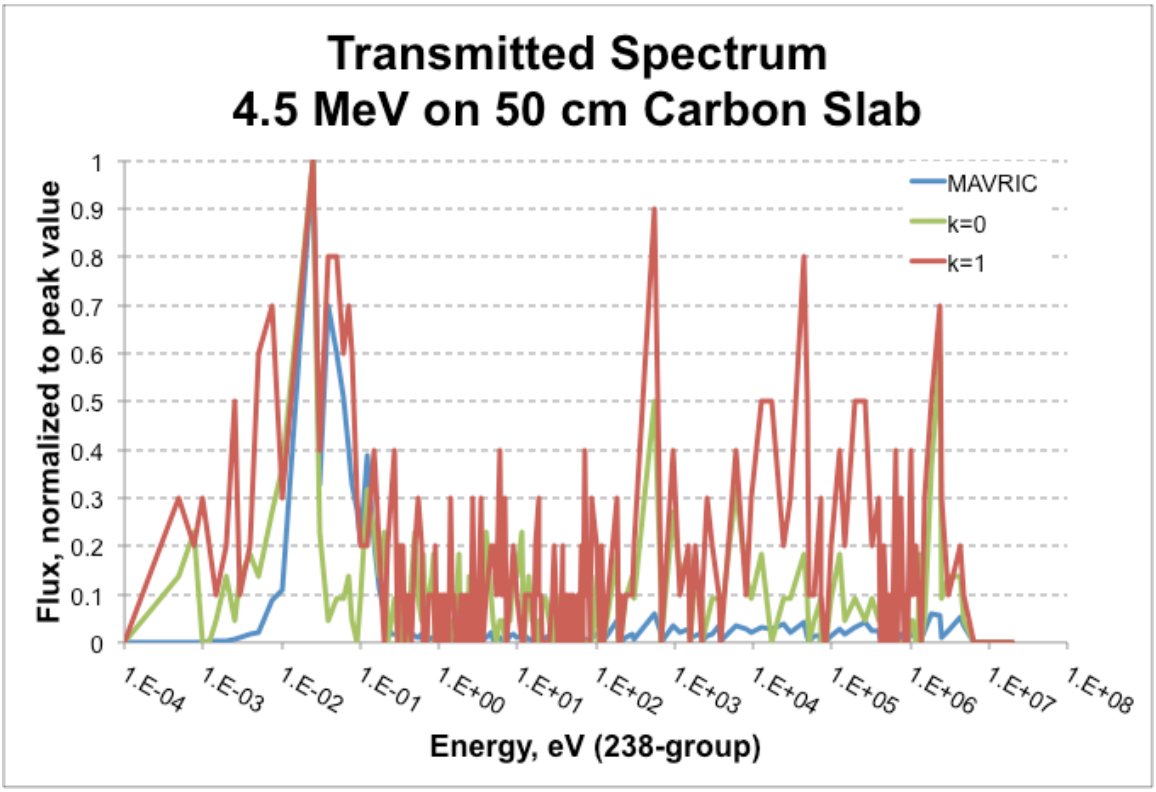

Figure 17. Comparison of transmitted spectra for 4.5 MeV neutron incident on a 50 cm carbon slab. Both the MAVRIC (dark blue) and Python script calculations (green and red) show a peak in Group 225 (0.01 to 0.0253 eV); this group includes neutrons thermalized to the temperature of 300 K, a result consistent with multiple scatters through a moderating medium like carbon. The k=0 results show moderate agreement up to the incident energy of 4.5 MeV, but the k=1 results have additional peaks in the keV to MeV range.

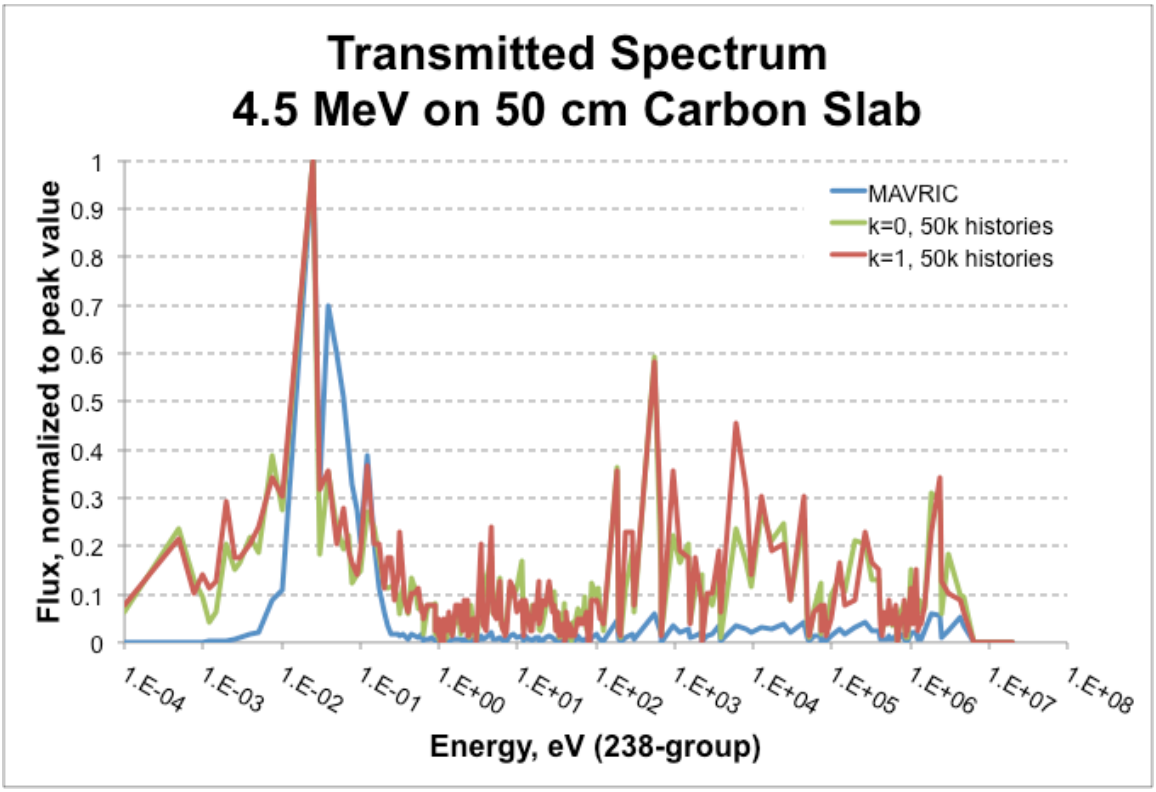

Figure 18. Comparison of transmitted spectra for 4.5 MeV neutron incident on a 50 cm carbon slab with 50k histories for k=0 and k=1 cross section sets. For context, the MAVRIC results are based on 130k histories. The spectra still share the Group 225 (0.01 to 0.0253 eV) peak, and the higher energy peaks in the keV to MeV range have decreased in height relative to the 10k histories results.

Figure 19 shows the transmitted spectrum for MAVRIC, and k=0 with 10000, 50000, and 100000 histories. All the Python script results agree well with each other, but even increasing the histories to 100000 is insufficient to bring them into agreement with MAVRIC.

Figure 20 shows the transmitted spectrum for MAVRIC, and k=1 with 10000, 50000, and 100000 histories. For k=1, increasing the number of histories does bring the results closer to agreement with MAVRIC, but the results are still more comparable to the k=0 results from Figure 19.

The results from benchmarking the 50 cm carbon target were somewhat mixed—the reflected spectra show good agreement, but the transmitted spectra show some rather significant differences. However, in context, a 50 cm carbon target is probably thicker than would be used.

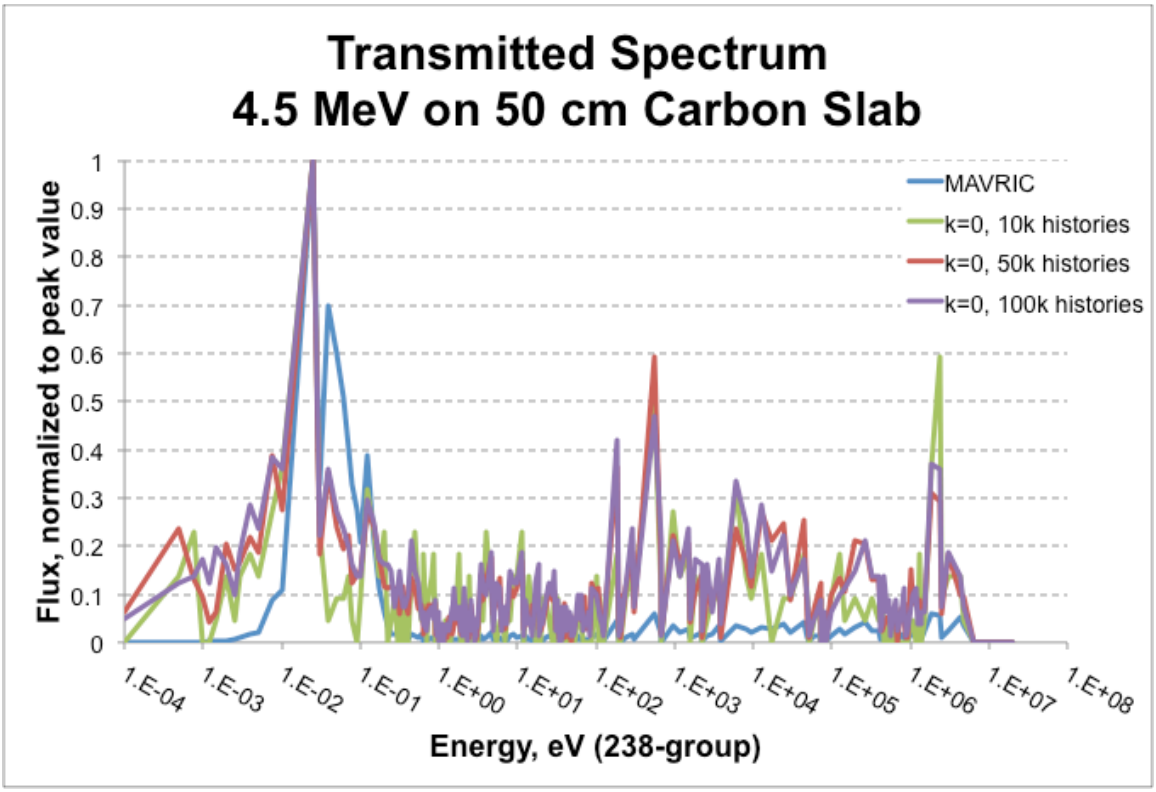

Figure 19. Comparison of transmitted spectra for 4.5 MeV neutron incident on a 50 cm carbon slab for 10k, 50k, and 100k histories for k=0 cross section sets. For context, the MAVRIC results are based on 130k histories. Increasing the number of histories for k=0 reduces the height of the MeV peak, but there is little impact at the other energies. All the results share the Group 225 peak.

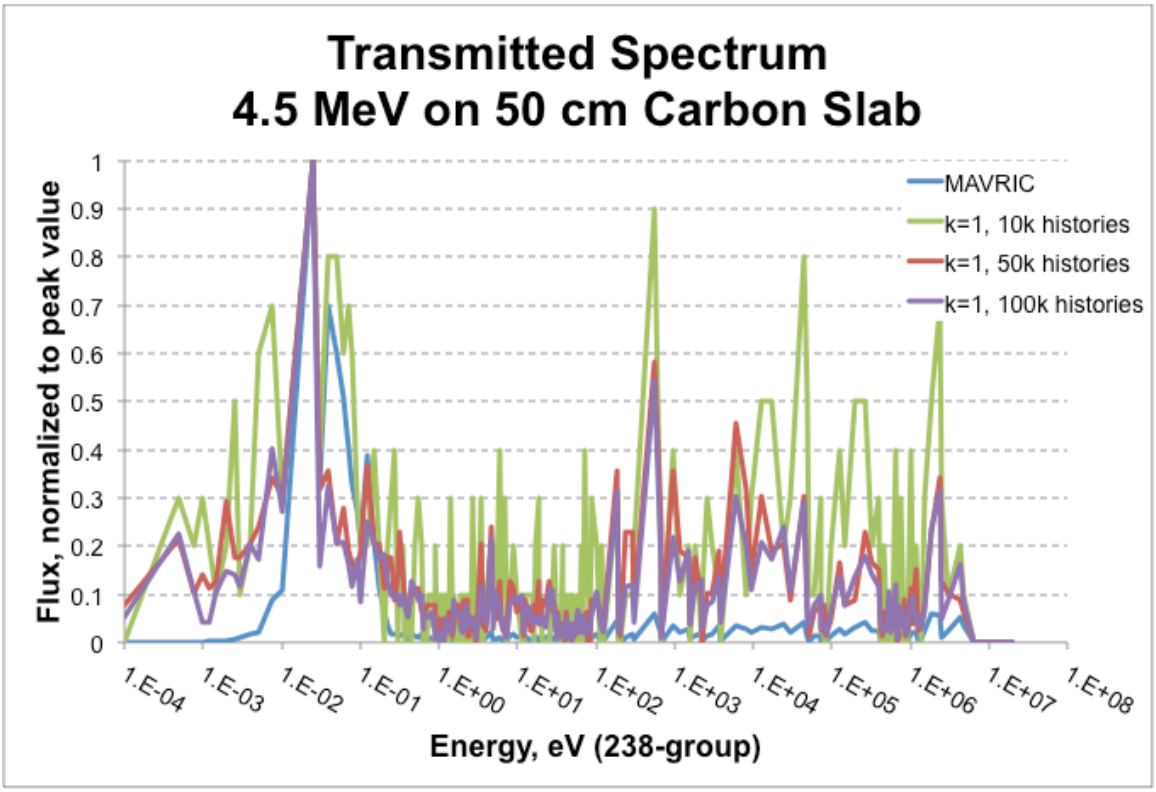

Figure 20. Comparison of transmitted spectra for 4.5 MeV neutron incident on a 50 cm carbon slab for 10k, 50k, and 100k histories for k=1 cross section sets. For context, the MAVRIC results are based on 130k histories. Increasing the number of histories for k=1 reduces the height of the keV through MeV peaks, with high impact throughout. All the results share the Group 225 peak.

Benchmarking with a 4.5 MeV neutron on a 20 cm carbon slab (~3.6 mean free paths) shows more immediate convergence. Figure 21 shows the reflection spectra for the MAVRIC, k=1, k=2, and k=5 cases. The reflected spectra lie on top of one another for the full data case (MAVRIC), and the k=1, k=2, and k=5 cases; the k=0 case is omitted. This is expected since the 50 cm slab showed good agreement on the reflected spectra, and a thinner slab provides less opportunity for backscatter after multiple internal scatters.

The 10-100 meV hump in the MAVRIC results from Figure 14 is absent, and thus the reflected spectra show immediate convergence.

Figure 22 shows the corresponding reflection spectrum data. Only the k=5 case shows some appreciable difference in the transmitted spectrum. Otherwise, the spectra are indistinguishable. Again, the k=0 spectrum is omitted.

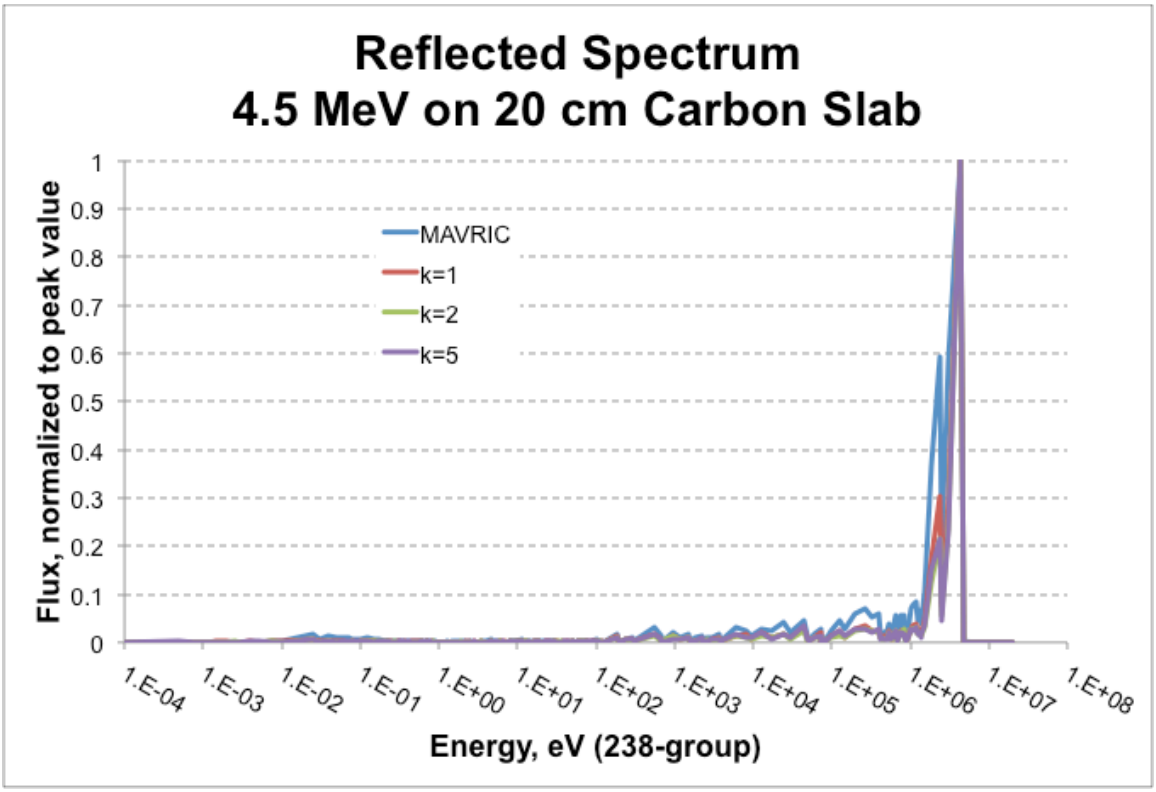

Figure 21. Comparison of reflected spectra for 4.5 MeV neutron incident on a 50 cm carbon slab. Both the MAVRIC (dark blue) and Python script calculations (red, green, and purple) show a peak in Group 11 (3.0 to 4.3 MeV); this group includes the direct backscatter energy of 3.22 MeV. There is a divergence in the results in Groups 13 and 12 (1.85 to 2.48 MeV), corresponding to a resonance peak insufficiently treated at ~2 MeV. There is good agreement in the shape of the spectrum (with some minor offset), including the 0.1 eV range; the MAVRIC results show a smaller peak in the 1 meV to 100 meV range than it did in the 50 cm slab case.

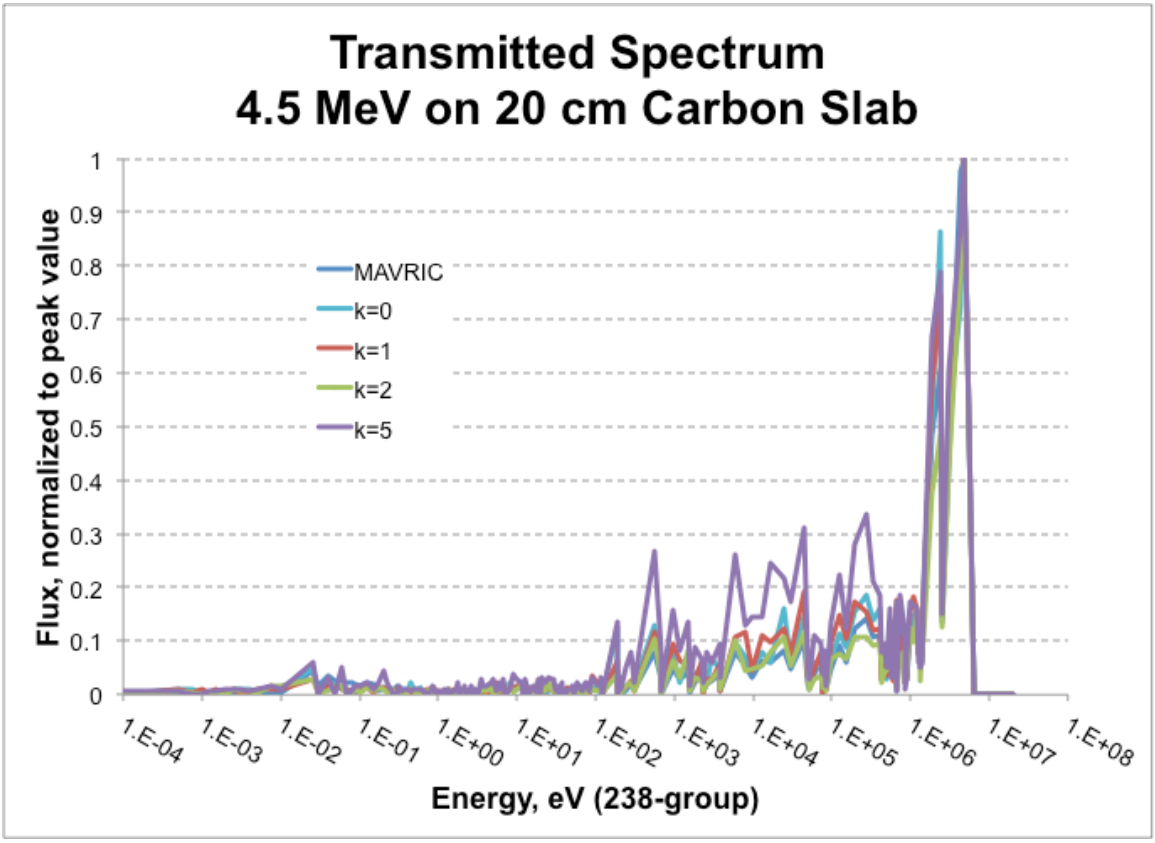

Figure 22. Comparison of transmitted spectra for 4.5 MeV neutron incident on a 50 cm carbon slab. All the results share the peak at Group 10 (4.3 to 4.8 MeV), corresponding to uncollided transmission of the incident beam. The k=0, k=1, and k=2 results show excellent agreement with the MAVRIC results, but the k=5 results show some disagreement between 0.1 keV and 1.0 MeV.

The benchmarks above demonstrate that, depending on the shield thickness, smoothing and stretching the cross sections generates the same spectrum. Table 3 below lists some the collected data as described above.

Table 3. Calculated parameters for different slab thicknesses and k values for carbon

	20 cm carbon slab				50 cm carbon slab			
	k=0	k=1	k=2	k=5	k=0	k=1	k=2	k=5
Mean linear distance (cm)	57.6	57.5	52.5	51.3	121.0	120.5	106.7	92.5
Mean radial distance (cm)	0.1	0.1	0.9	1.0	0.9	1.0	1.9	1.8
Mean deepest penetration (cm)	11.7	11.8	11.2	10.0	15.4	15.5	14.4	12.4
Mean number of scatters	19.0	19.3	17.2	18.8	57.1	56.8	50.0	44.4
Std Dev linear distance (cm)	61.4	61.6	60.7	61.0	122.0	122.8	119.2	111.1
Std Dev radial distance (cm)	1.6	1.7	3.5	3.5	6.1	6.1	6.9	6.4
Std Dev deepest penetration (cm)	8.7	8.8	8.7	8.2	14.4	14.6	14.0	12.8
Std Dev number of scatters	31.5	31.8	31.4	32.3	69.5	69.5	67.7	63.8
Elapsed runtime (s)	50.3	19.1	14.3	15.0	139.1	49.8	37.1	32.0
k=0 time/k=n time	1	2.63	3.52	3.36	1	2.79	3.75	4.35
Transmission per 10000	3103	3179	2748	2345	344	353	305	180
Reflection per 10000	6720	6497	5989	6167	7192	7102	6393	6564

Table 2 shows that for the 20 cm and 50 cm slab cases, the collected statistics agree well for all values of k. Further, using k>0 can speed up the calculation by a factor of ~3; this factor may be greater (or less) in other materials.

Figure 23 shows the calculated MAVRIC fluxes and ratio for the 50 cm slab case as a function of energy group; energy decreases as the group increases. The figure shows that the reflected spectrum is much harder (skewed to higher energy) than the transmitted spectrum. This is expected since transmitted neutrons are more likely to have undergone multiple scatters, while the reflected neutrons have the initial backscatter as their primary contributor.

Figure 24 shows the calculated Python counts and ratio for the 50 cm slab case as a function of energy group; energy decreases as the group increases. The figure shows behavior similar to that of Figure 23.

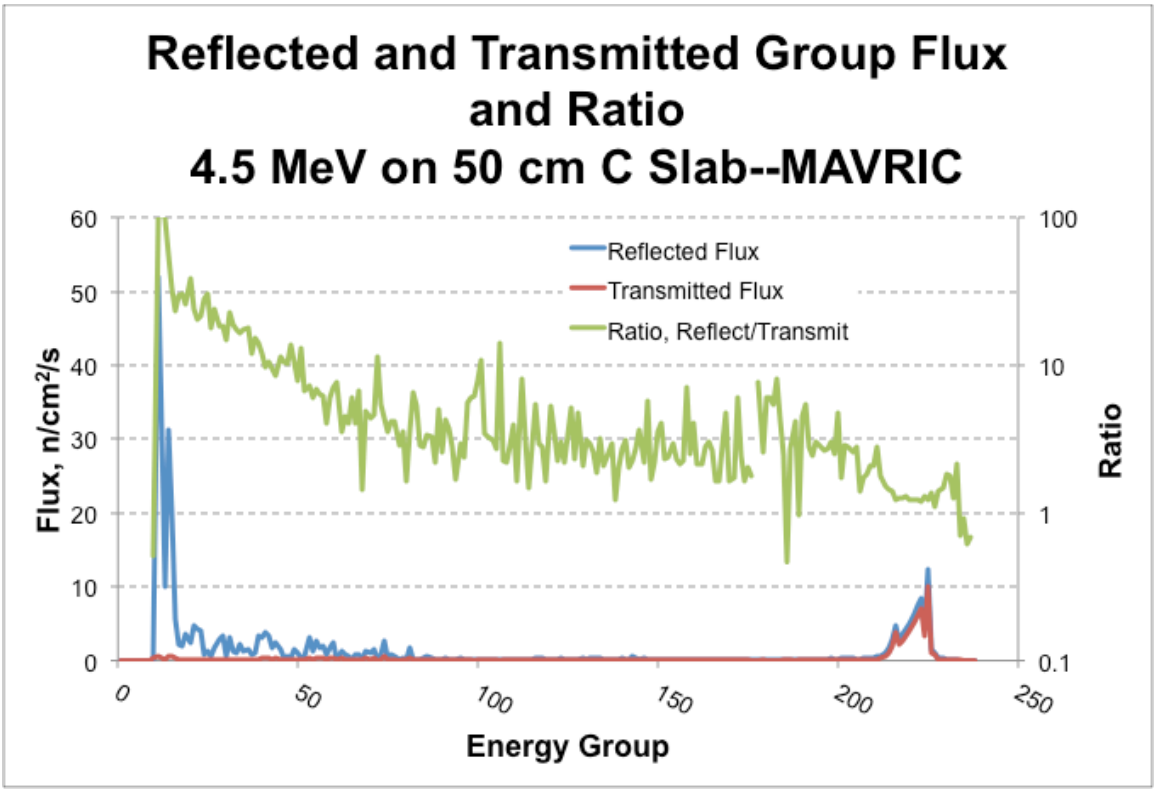

Figure 23. MAVRIC group flux ratio (reflected/transmitted) for 4.5 MeV neutrons incident on 50 cm carbon slab. The ratio (green line) shows that the reflected low group (high energy) flux is around 100x that for transmitted flux, while the flux ratio starting around Group 75 (305 eV) through Group 200 (0.625 eV) has a characteristic value of ~3.2 (halfway between 1 and 10 on a logarithmic scale). This shows that the reflected flux is a harder spectrum, and the relative probability of reflection is greater than the probability of transmission.

Figure 25 plots the two 50 cm spectral ratios together. The shapes of the spectral ratios are similar, but the magnitudes of the peaks disagree. This is a result of the mismatched transmission spectra in Figure 17.

Figure 26 shows the calculated MAVRIC fluxes and ratio for the 20 cm slab case as a function of energy group; energy decreases as the group increases. The figure shows that the reflected spectrum is somewhat harder (skewed to higher energy) than the transmitted spectrum. While the transmitted neutrons have still had opportunity to downscatter, the total number of scatters is fewer for a thinner shield.

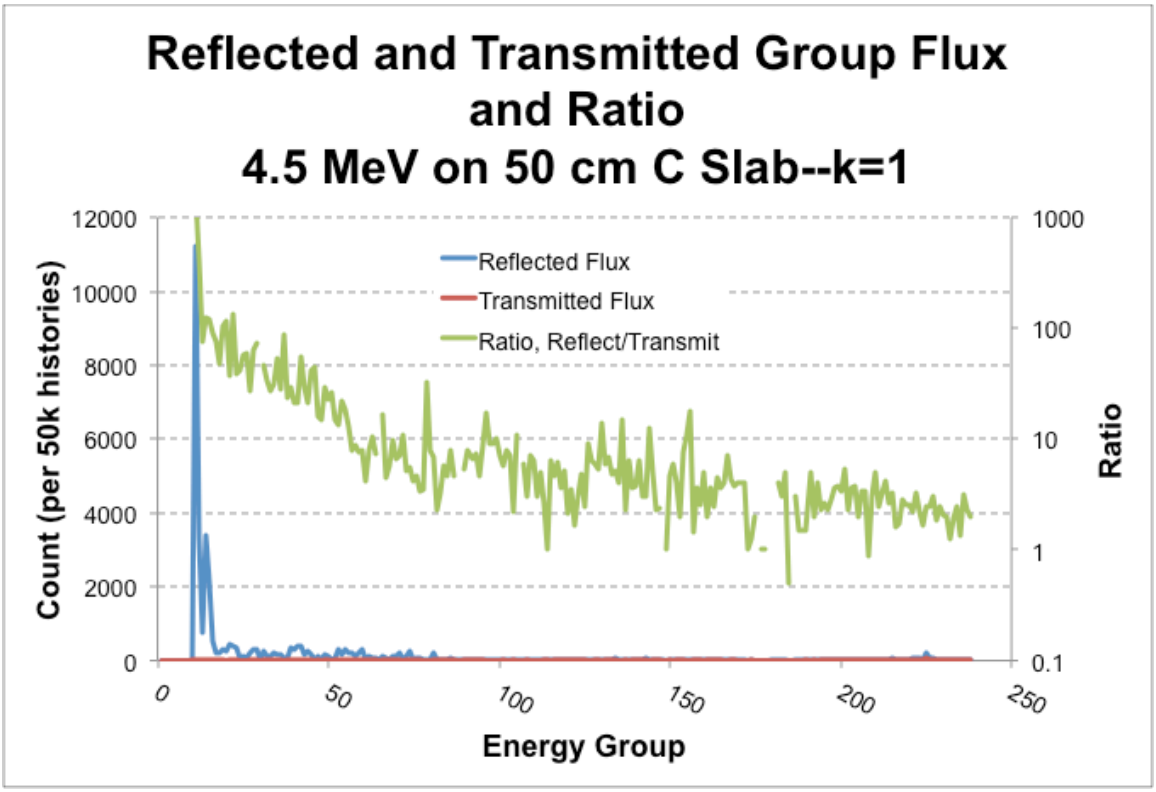

Figure 24. Smoothed k=1 group count ratio (reflected/transmitted) for 4.5 MeV neutrons incident on 50 cm carbon slab. The ratio (green line) shows that the reflected low group (high energy) flux is around 1000x that for transmitted flux, while the flux ratio starting around Group 75 (305 eV) through Group 200 (0.625 eV) has a characteristic value between 4.65 (two-thirds between 1 and 10 on a logarithmic scale) and 2.15 (one-third between 1 and 10 on a logarithmic scale). This shows that the reflected flux is a harder spectrum, and the relative probability of reflection is greater than the probability of transmission.

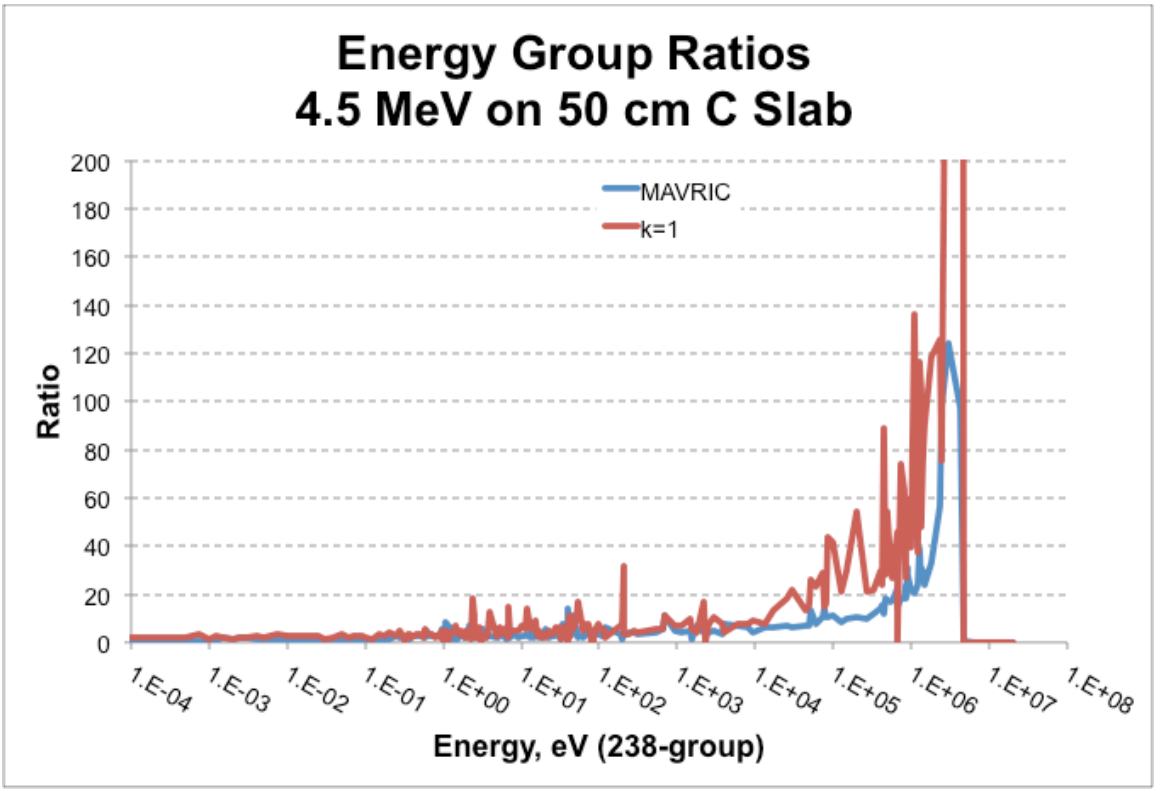

Figure 25. Comparison of group ratios for MAVRIC and k=1 for 4.5 MeV neutrons on 50 cm carbon slab. The two ratio spectra show that the k=1 smoothed cross section generally agrees with the MAVRIC calculation when comparing the probabilities for reflection and transmission. The thickness of the target and the number of histories will impact the flux spectrum; these results use 50k histories for the k=1 cross sections.

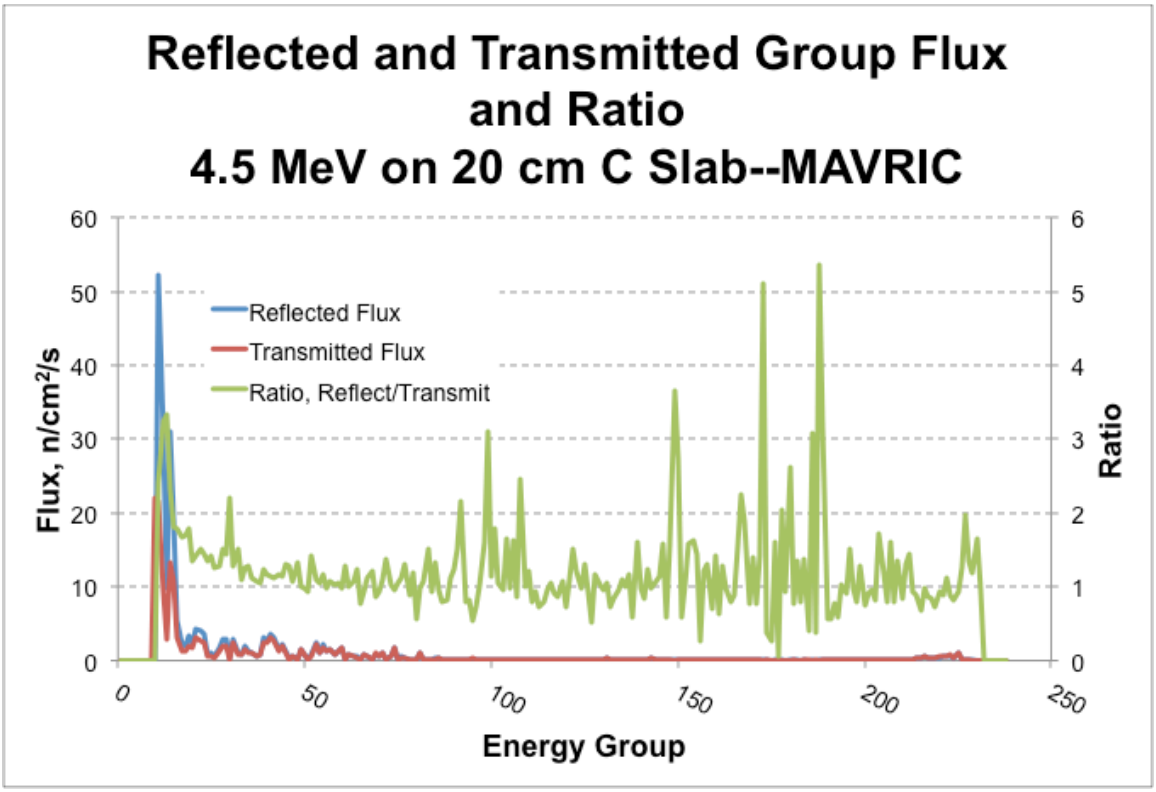

Figure 26. MAVRIC group flux ratio (reflected/transmitted) for 4.5 MeV neutrons incident on 20 cm carbon slab. The ratio (green line) shows that the reflected low group (high energy) flux is around 3x that for transmitted flux, while the flux ratio starting around Group 75 (305 eV) has a characteristic value of ~1. This shows that the reflected flux has a higher backscatter peak compared to uncollided transmission, and the relative probability of reflection is greater than the probability of transmission.

Figure 27 shows the calculated Python counts and ratio for the 20 cm slab case as a function of energy group; energy decreases as the group increases. The figure shows behavior similar to that of Figure 26.

Figure 28 plots the two 20 cm spectral ratios together. The shapes of the spectral ratios show much better agreement than the 50 cm spectral ratios in Figure 25. The k=1 results still show higher overall ratios, but the ratios in the keV to MeV range show excellent agreement.

Figure 29 plots the two 20 cm spectral ratios together along with the k=5 results. The k=5 results have a spurious peak in the MeV range, but otherwise show good agreement with the MAVRIC and k=1 results.

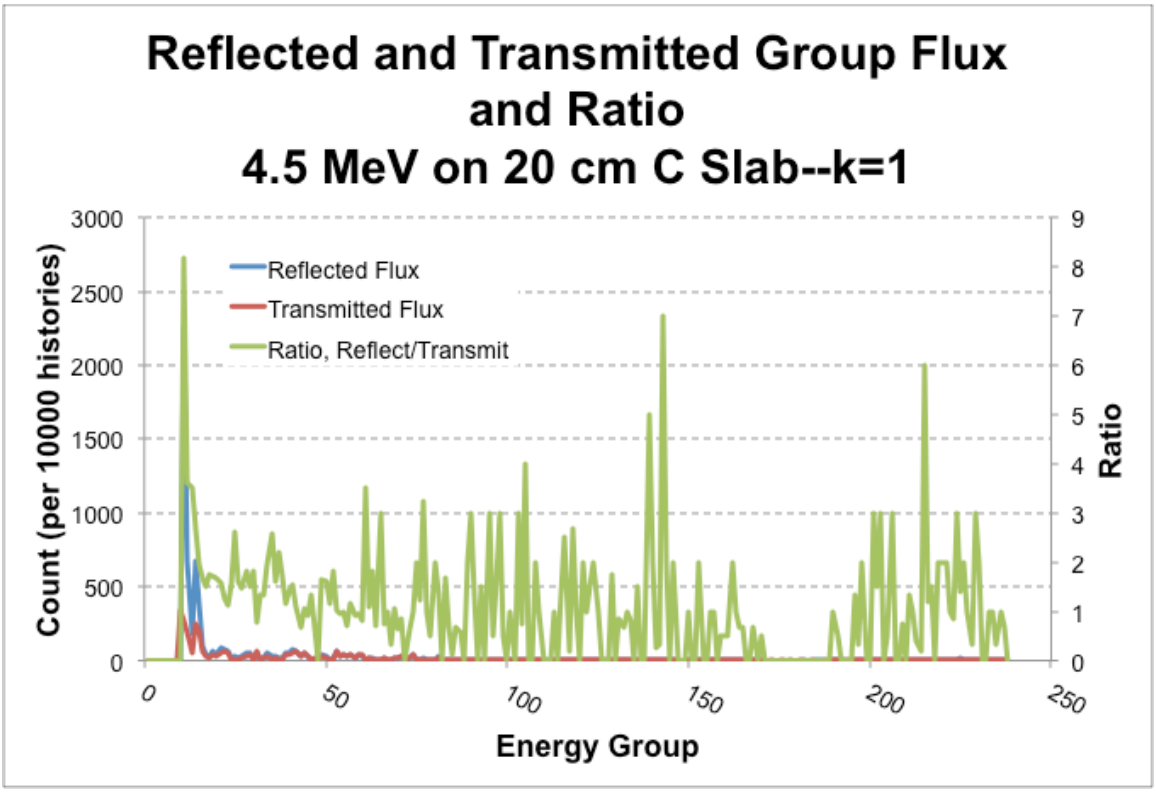

Figure 27. Smoothed k=1 group count ratio (reflected/transmitted) for 4.5 MeV neutrons incident on 20 cm carbon slab. The ratio (green line) shows that the reflected low group (high energy) flux is around 8x that for transmitted flux, while the flux ratio starting around Group 75 (305 eV) has a characteristic value between 1 and 3. This shows that the reflected flux is a harder spectrum, and the relative probability of reflection is greater than the probability of transmission.

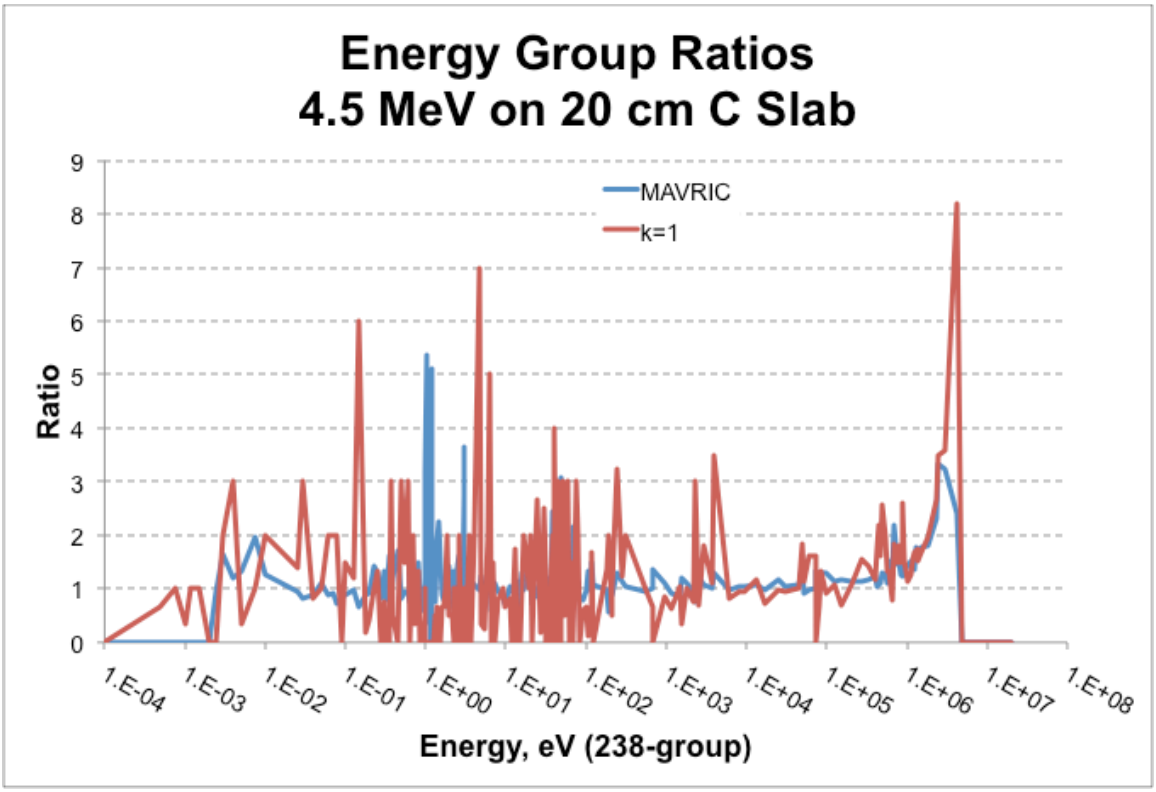

Figure 28. Comparison of group ratios for MAVRIC and k=1 for 4.5 MeV neutrons on 20 cm carbon slab. The two ratio spectra show that the k=1 smoothed cross section generally agrees with the MAVRIC calculation when comparing the probabilities for reflection and transmission, especially in the MeV and keV ranges; these are important for shielding applications.

Overall, the method shows promising results for MeV range neutrons impinging on carbon slabs of 20 cm thickness, and acceptable results for MeV range neutrons impinging on thicker slabs of 50 cm thickness.

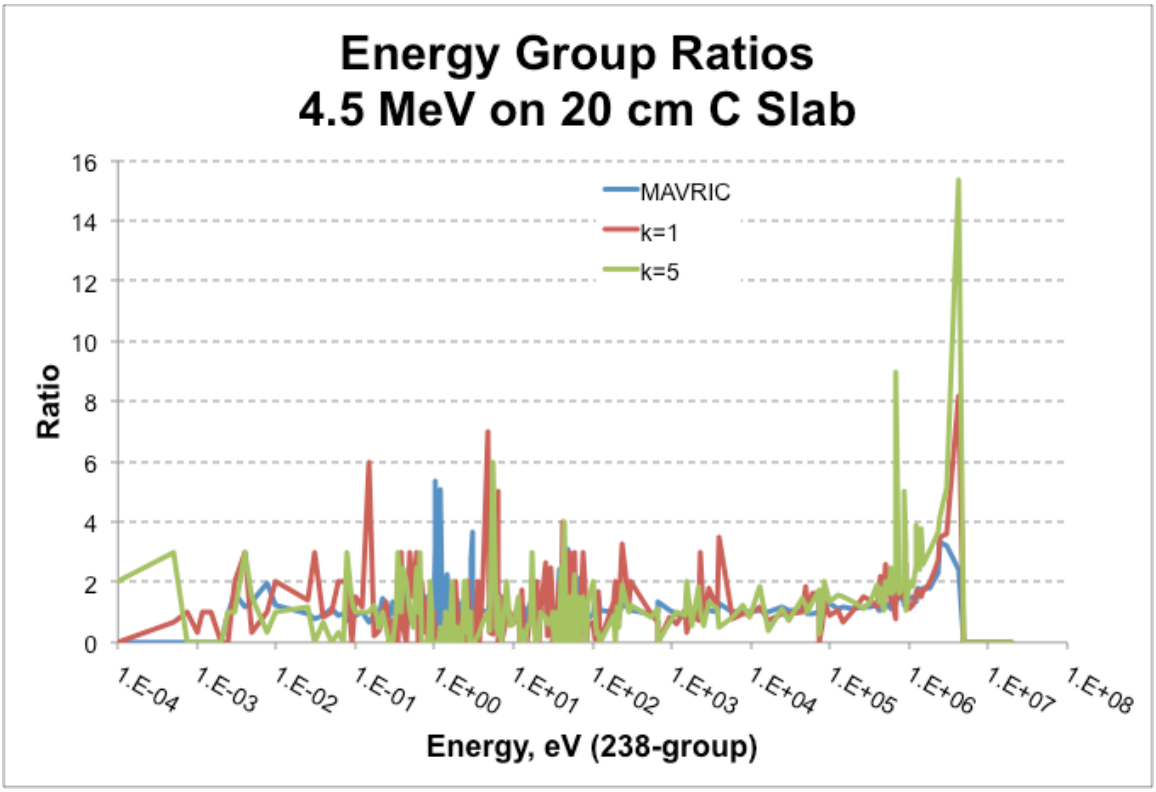

Figure 29. Comparison of group ratios for MAVRIC, k=1, and k=5 for 4.5 MeV neutrons on 20 cm carbon slab. The three ratio spectra show that smoothing the cross sections from k=1 to k=5 does not significantly impact the shape of the spectral ratios, which in turn adds confidence that the relative probabilities for transmission and reflection are conserved through the use of simplified cross section data sets.

Iron results

The next step in the benchmarking efforts uses a similar approach for transport in iron. The first benchmark case uses a monoenergetic beam of 0.4 MeV neutrons impinging on a target of iron with varying thicknesses. A 0.4 MeV neutron energy was chosen to be less than the rising non-elastic and inelastic cross sections, yet be in a region that experiences resonance activity (Figure 11). While 0.4 MeV is less than the typical fission neutron energy of ~2 MeV, it is still in the range at which this test case would have direct fission reactor applications.

At 0.4 MeV, the total cross section is ~5 b. With a density of 7.87 g/cc and a molar mass of 56 g/mol, the total macroscopic cross section Σ_T=0.42 cm^{-1}, yielding a mean free path of ~2.36 cm.

Figure 30 shows the comparison of the MAVRIC output reflected spectrum with the research methodology k=1 output and k=5 output for a 50 cm slab, which is ~21 mean free paths in thickness.

The reflected spectra show excellent agreement, but the spectrum itself is not particularly interesting; there is a large peak at the direct backscatter group with little other detail.

Figure 31 shows the transmission spectrum for MAVRIC, k=0, and k=1. That figure shows good agreement between the MAVRIC and k=0 results, but the k=1 results do not agree with them. There is significant spectral softening moving from k=0 to k=1. This disagreement may be a result of the thickness of the shield, or it may be a result of the cross section smoothing.

To provide insight, Figure 32 shows the transmitted spectra for a 20 cm slab (~8.5 mean free paths) for MAVRIC, k=1, and k=5 (k=0 is omitted). Decreasing the shield thickness brought the k=1 results closer to the MAVRIC results. Further, the k=5 results agree well with both the k=1 and the MAVRIC results.

Figure 33 shows the group reflected-to-transmitted flux ratios for 20 cm iron for MAVRIC, k=1, and k=5. The agreement in flux ratios decreases as k increases, but the shape of the ratio curve is constant among them.

Table 4 shows some calculational results for iron. For the 20 cm iron slab, moving from k=0 (full data set) to k=1 yields a speedup of a factor of ~34, but it does not impact the accuracy with respect to comparisons with MAVRIC. Moving to k=5 yields a total speedup to a factor of ~74, again with little loss in spectral or transmission probability accuracy.

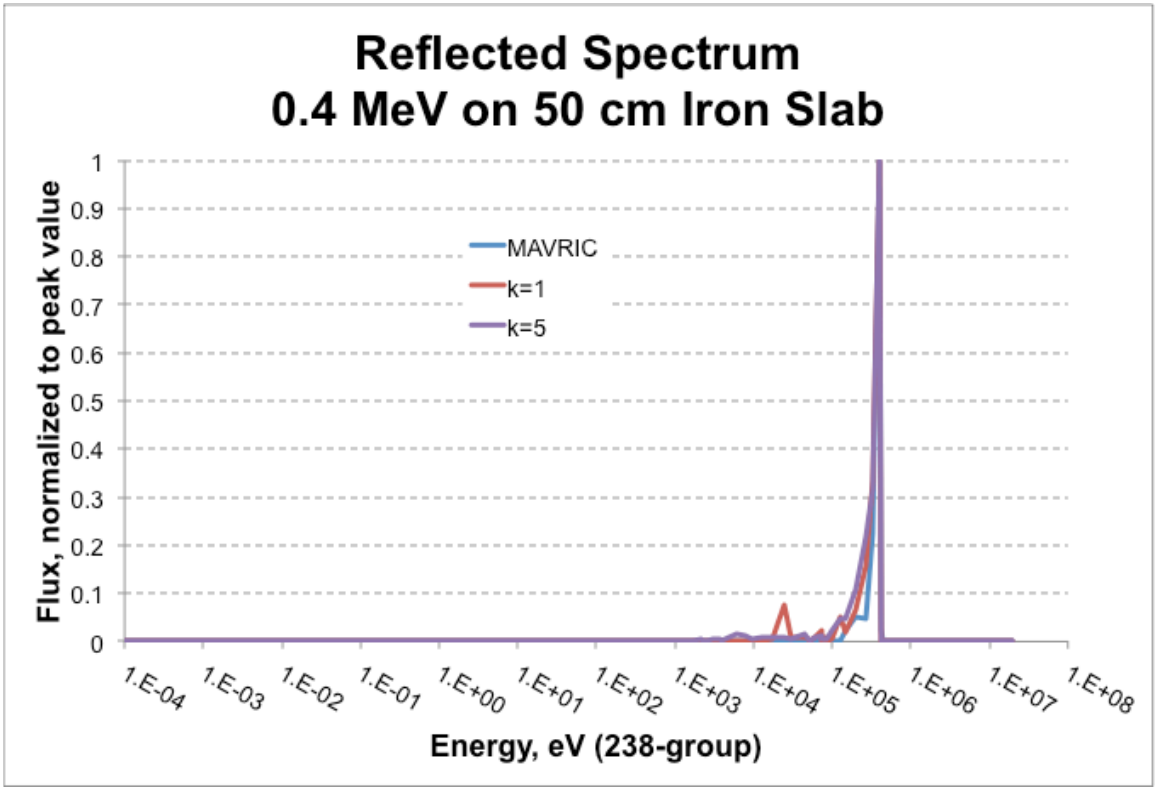

Figure 30. Comparison of reflected spectra for 0.4 MeV neutron incident on a 50 cm iron slab. Both the MAVRIC (dark blue) and Python script calculations (red and purple) show a peak in Group 39 (330 keV to 400 keV); this group includes the direct backscatter energy of 372 keV. All three sets of results show good agreement in the reflected spectra.

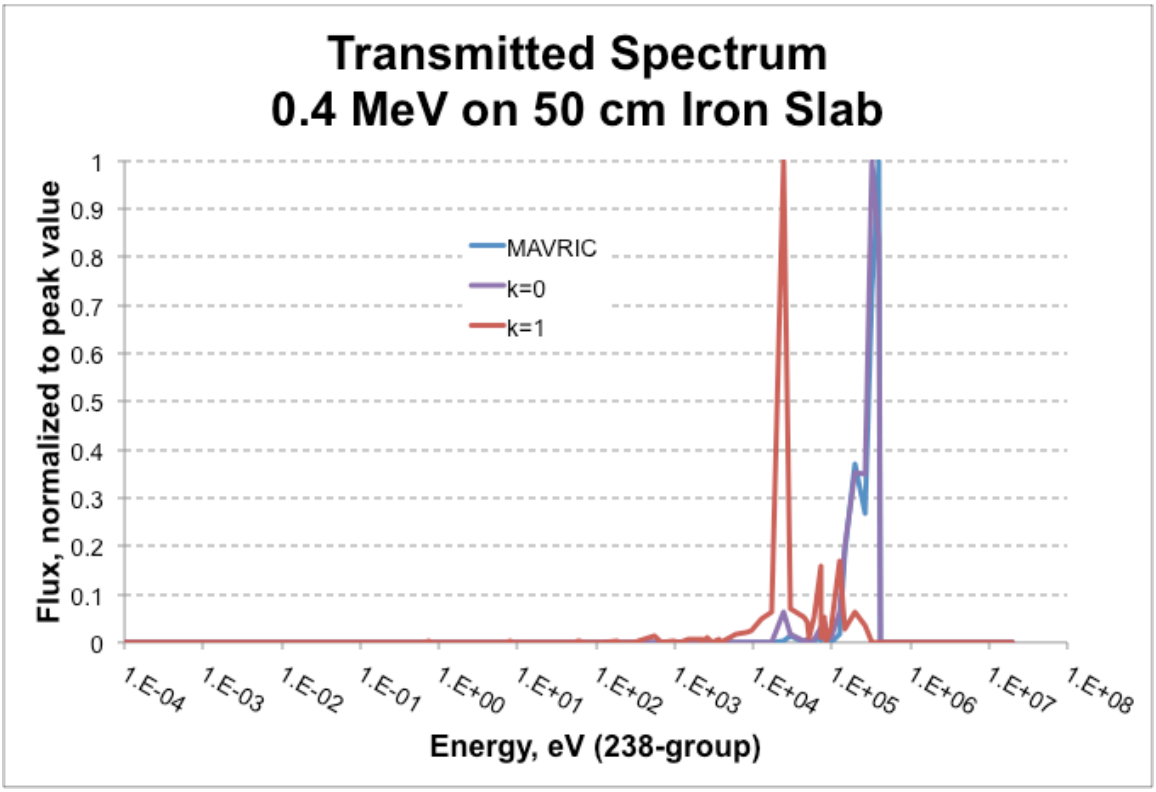

Figure 31. Comparison of transmitted spectra for 0.4 MeV neutron incident on a 50 cm iron slab. The MAVRIC results show a peak at Group 39, corresponding to uncollided transmission (or potentially consecutive backscatters). The k=0 (full data set) shows a peak at Group 40 (0.27 to 0.33 MeV) and otherwise excellent agreement with the MAVRIC results. However, the k=1 data shows a peak at Group 55 (25 to 30 keV).

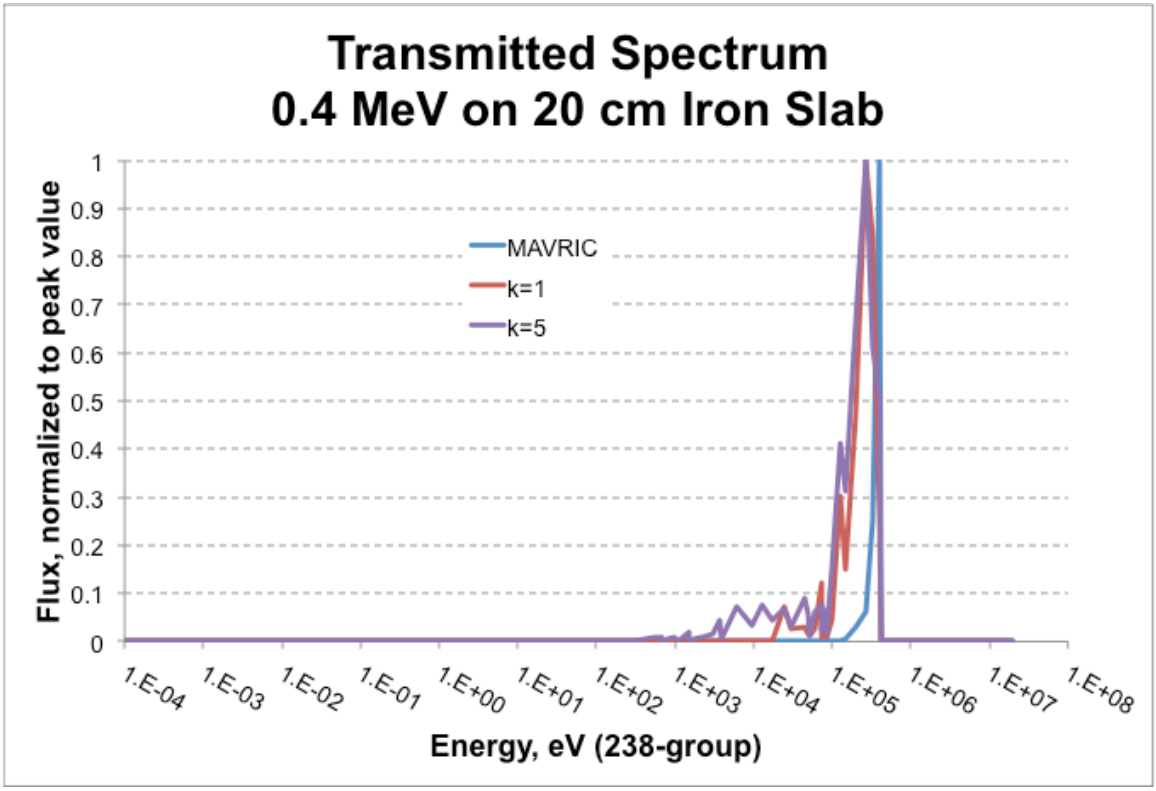

Figure 32. Comparison of transmitted spectra for 0.4 MeV neutron incident on a 20 cm iron slab. The MAVRIC results show a peak at Group 39, corresponding to uncollided transmission (or potentially consecutive backscatters). The k=1 and k=5 data sets show a peak at Group 41 (0.20 to 0.27 MeV). The MAVRIC spectrum softens rapidly, with essentially only the uncollided peak in the spectrum. In contrast, the k=1 and k=5 results have harder spectra with a slower descent through the keV range.

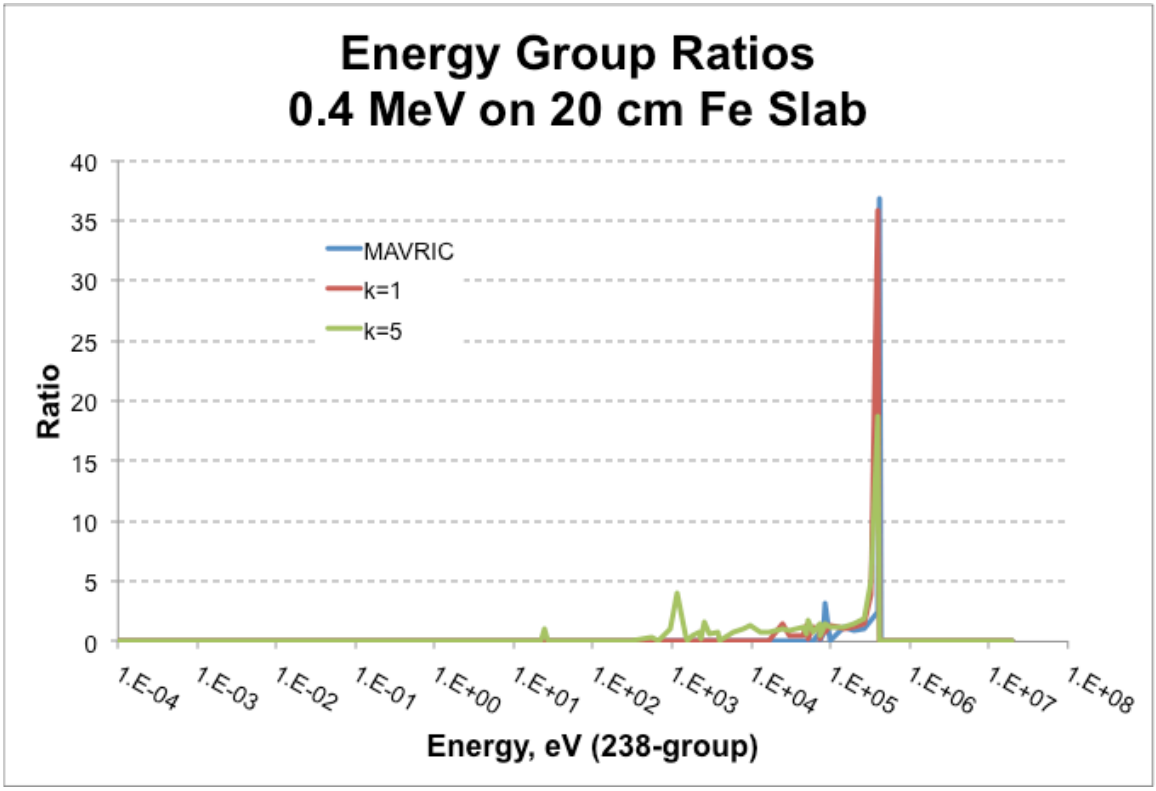

Figure 33. Comparison of group ratios for MAVRIC, k=1, and k=5 for 0.4 MeV neutrons on 20 cm ^{56}Fe slab. The three ratio spectra show that smoothing the cross sections from k=1 to k=5 does not significantly impact the shape of the spectral ratios, which in turn adds confidence that the relative probabilities for transmission and reflection are conserved through the use of simplified cross section data sets.

Table 4. Calculated parameters for 20 cm slab and various k values for ^{56}Fe

	20 cm iron slab		
	k=0	k=1	k=5
Mean linear distance (cm)	67.4	40.4	49.4
Mean radial distance (cm)	0.1	0.3	1.1
Mean deepest penetration (cm)	14.5	8.0	8.9
Mean number of scatters	4.4	10.6	18.4
Std Dev linear distance (cm)	88.5	47.6	60.7
Std Dev radial distance (cm)	2.0	2.5	4.7
Std Dev deepest penetration (cm)	32.3	8.6	7.9
Std Dev number of scatters	4.8	13.8	32.2
Elapsed runtime (s)	1134.9	33.7	15.4
k=0 time/k=n time	1	33.68	73.69
Transmission per 10000	2365	1763	1938
Reflection per 10000	7582	8062	7294

CHAPTER V
CONCLUSIONS AND RECOMMENDATIONS

Overall Results, Qualifications, And Caveats

The overall results show good agreement in calculation results with cross sections of varying degrees of “smoothed” or “stretched” values. The calculations show good agreement, not only with each other, but also with established production-level shielding codes.

The primary qualification and caveat applied to this demonstration is that it does not explicitly account for anything other than isotropic elastic scatters. Accordingly, the demonstration cases were intentionally designed to prevent that shortcoming in the cross section data set from affecting the comparison between the Python transport code and MAVRIC. A more complete implementation of the methodology would extend to other interaction cross sections.

The main benefits of the smoothing and stretching method is that it decreases the total computational burden for shielding calculations caused by treating continuous energy cross sections as a lookup and interpolation table. Further, it generates a group-wise cross section set that does not require flux-weighting, which means it has application beyond the problem-specific reaction rate-conserving standard group-wise cross section sets.

Expansion Of The Method And Potential Future Work

The method can be expanded to include other cross section channels besides elastic scattering, as well as to incorporate scattering cross sections with angular distributions. The channel expansion would require the addition of each potential interaction channel to the transport calculation code, as well as the relevant additional information.

For example, the code currently draws a random number from a uniform distribution to determine whether the interaction is a scatter or a capture; the inclusion of anisotropic scatters in competition with captures and elastic scatters would use the same binning logic to determine whether an interaction is an elastic scatter, an inelastic scatter, or a capture, based on the relative probabilities. If the interaction were an anisotropic elastic scatter, the scattering angle would then be drawn from the appropriate distribution rather than a uniform distribution in polar cosine.

In the event of an inelastic scatter in competition with other interactions, the Q value of the interaction would be necessary with any relevant scattering

distribution to determine the final energy of the neutron. For scattering media, there is no obstacle to implementing this method for any available cross section.

For multiplying media, including media with significant (n,2n) or higher production terms, the current implementation of the transport code is incapable of tracking the secondary neutrons. The methodology for cross section processing is valid for those purposes, but the Python transport code would require expansion to record the location, direction, and energy of the produced particles. This can be accomplished, but it is outside the scope of this work.

Another further expansion would use a polynomial fit to generate a cross section functional form, rather than a tabular form.

Further future work would use machine learning, such as neural networks or fuzzy logic, to determine the minimum cross section data set to maintain a desired level of accuracy.

LIST OF REFERENCES

1. http://www.nndc.bnl.gov/
2. "Generation of Multi-group Cross Sections and Scattering Matrices with the Monte Carlo Code MCNP5", J. Hoogenboom, *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications,* meeting proceedings, 2007
3. *Fast Reactor Cross-Section Processing Codes—Is There A Dollars Worth Of Difference Between Them?*, Weisbin, C. R, *et al*, 1974
4. "Resonance self-shielding effect in covariance processing", G. Chiba, *Perspectives on Nuclear Data for the Next Decade,* workshop proceedings, 2005
5. http://scale.ornl.gov
6. https://mcnp.lanl.gov
7. https://geant4.web.cern.ch/geant4/
8. https://oltaris.nasa.gov
9. *Guide to Verification and Validation of the SCALE-4 Radiation Shielding Software*, NUREG/CR-6484, ORNL/TM-13277, 1996
10. *MCNP: Neutron Benchmark Problems*, Whalen, D. J., *et al*, LA-12212, DE92 004710
11. *Development and Testing of ENDF/B-VI.8 and ENDF/B-VII.0 Coupled Neutron-Gamma Libraries for SCALE 6*, NUREG/CR-6990, ORNL/TM-2008/047, 2008
12. "Averaging of Neutron Reaction Cross Sections in the Resonance Region", DeLauder, N.P., Townsend, L.W., *et al*, 2011 Accelerator Applications Topical Meeting of the Accelerator Division of the ANS, 2011
13. "The Effect of Linearly Anisotropic Neutron Scattering on Disadvantage Factor Calculations", G. R. Bond and C. E. Siewert, *Nuclear Science and Engineering:* 35, 277-282 (1969)
14. *Analytical Benchmark Test Set for Criticality Code Verification*, LA-13511, 1999
15. http://mathworld.wolfram.com/RotationMatrix.html

APPENDIX

SCALE Benchmark Example Input File

```
'******************************************************************************
'
' Benchmark case
' 4.5 MeV incident neutron beam
' 5 cm carbon @ 2.25 g/cc
' 1000 cm y-z extent (essentially infinite)
' 100 cm flux integration zone
' source strength 1e9
'
'******************************************************************************


'============================================================================
==================
' Basic MAVRIC Sequence items: title and SCALE Library
'============================================================================
==================
=mavric
Monaco/MAVRIC Benchmark Work
v7-238

'------------------------------------------------------------------------------
' Composition Block - standard SCALE input
'------------------------------------------------------------------------------
read composition
   carbon       1 den=2.25 1.0 300.0 end
end composition

'------------------------------------------------------------------------------
' Geometry Block - SCALE standard geometry package (SGGP)
'------------------------------------------------------------------------------
read geometry
   global unit 1
      cuboid  1   5.0 0  1000.0 -1000.0   1000.0 -1000.0
      cuboid 98  105.0 0  1000 -1000 1000 -1000
      cuboid 99  105 -100 1000 -1000 1000 -1000
      media 1 1  1
      media 0 1  98 -1
      media 0 1  99 -98 -1
   boundary 99
end geometry

'------------------------------------------------------------------------------
```

```
' Sources Block
'   Monoenergetic collimated point source
'-------------------------------------------------------------------------------
read sources
   src 1
      title="Monoenergetic collimated point source"
      strength=1e9
      sphere 0
      spectrumDist 1 end
      neutronBounds 4.501e6 4.499e6 end
      direction 1.0 0.0 0.0
      angleCosines -1.0  0.99999 1.0 end
      angularDist 0.0 1.0 end
      angularSample 0.001 1.0 end
   end src
end sources

'-------------------------------------------------------------------------------
' Tallies Block
'-------------------------------------------------------------------------------
read tallies
   regionTally 1
      title="Flux in region 2 of unit 1"
      unit=1
      region=2
   end regionTally
   regionTally 2
      title="Flux in region 3 of unit 1"
      unit=1
      region=3
   end regionTally
end tallies

read volume
type=random
nrays=1000000
xp=105 xm=-100 yp=1000 ym=-1000 zp=1000 zp=-1000
end volume

'-------------------------------------------------------------------------------
' Parameters Block
'-------------------------------------------------------------------------------
read parameters
   randomSeed=00003ecd7b4e3e8b
   perBatch=13000 batches=10
```

```
    noFissions
    noSecondaries
end parameters

'-------------------------------------------------------------------------------
' Biasing Block - use region based weight windows
'-------------------------------------------------------------------------------

end data
end
```

238-group Structure

Group	Upper Energy	Group	Upper Energy	Group	Upper Energy
1	2.00E+07	42	2.00E+05	83	1.19E+02
2	1.73E+07	43	1.50E+05	84	1.15E+02
3	1.57E+07	44	1.28E+05	85	1.08E+02
4	1.46E+07	45	1.00E+05	86	1.00E+02
5	1.38E+07	46	8.50E+04	87	9.00E+01
6	1.28E+07	47	8.20E+04	88	8.20E+01
7	1.00E+07	48	7.50E+04	89	8.00E+01
8	8.19E+06	49	7.30E+04	90	7.60E+01
9	6.43E+06	50	6.00E+04	91	7.20E+01
10	4.80E+06	51	5.20E+04	92	6.75E+01
11	4.30E+06	52	5.00E+04	93	6.50E+01
12	3.00E+06	53	4.50E+04	94	6.10E+01
13	2.48E+06	54	3.00E+04	95	5.90E+01
14	2.35E+06	55	2.50E+04	96	5.34E+01
15	1.85E+06	56	1.70E+04	97	5.20E+01
16	1.50E+06	57	1.30E+04	98	5.06E+01
17	1.40E+06	58	9.50E+03	99	4.92E+01
18	1.36E+06	59	8.03E+03	100	4.83E+01
19	1.32E+06	60	6.00E+03	101	4.70E+01
20	1.25E+06	61	3.90E+03	102	4.52E+01
21	1.20E+06	62	3.74E+03	103	4.40E+01
22	1.10E+06	63	3.00E+03	104	4.24E+01
23	1.01E+06	64	2.58E+03	105	4.10E+01
24	9.20E+05	65	2.29E+03	106	3.96E+01
25	9.00E+05	66	2.20E+03	107	3.91E+01
26	8.75E+05	67	1.80E+03	108	3.80E+01
27	8.61E+05	68	1.55E+03	109	3.70E+01
28	8.20E+05	69	1.50E+03	110	3.55E+01
29	7.50E+05	70	1.15E+03	111	3.46E+01
30	6.79E+05	71	9.50E+02	112	3.38E+01
31	6.70E+05	72	6.83E+02	113	3.33E+01
32	6.00E+05	73	6.70E+02	114	3.18E+01
33	5.73E+05	74	5.50E+02	115	3.13E+01
34	5.50E+05	75	3.05E+02	116	3.00E+01
35	5.00E+05	76	2.85E+02	117	2.75E+01
36	4.70E+05	77	2.40E+02	118	2.50E+01
37	4.40E+05	78	2.10E+02	119	2.25E+01
38	4.20E+05	79	2.08E+02	120	2.10E+01
39	4.00E+05	80	1.93E+02	121	2.00E+01
40	3.30E+05	81	1.86E+02	122	1.90E+01
41	2.70E+05	82	1.22E+02	123	1.85E+01

Group	Upper Energy	Group	Upper Energy	Group	Upper Energy
124	1.70E+01	163	1.77E+00	202	5.50E-01
125	1.60E+01	164	1.68E+00	203	5.00E-01
126	1.51E+01	165	1.59E+00	204	4.50E-01
127	1.44E+01	166	1.50E+00	205	4.00E-01
128	1.38E+01	167	1.45E+00	206	3.75E-01
129	1.29E+01	168	1.40E+00	207	3.50E-01
130	1.19E+01	169	1.35E+00	208	3.25E-01
131	1.15E+01	170	1.30E+00	209	3.00E-01
132	1.00E+01	171	1.25E+00	210	2.75E-01
133	9.10E+00	172	1.22E+00	211	2.50E-01
134	8.10E+00	173	1.20E+00	212	2.25E-01
135	7.15E+00	174	1.17E+00	213	2.00E-01
136	7.00E+00	175	1.15E+00	214	1.75E-01
137	6.75E+00	176	1.14E+00	215	1.50E-01
138	6.50E+00	177	1.13E+00	216	1.25E-01
139	6.25E+00	178	1.12E+00	217	1.00E-01
140	6.00E+00	179	1.11E+00	218	9.00E-02
141	5.40E+00	180	1.10E+00	219	8.00E-02
142	5.00E+00	181	1.09E+00	220	7.00E-02
143	4.75E+00	182	1.08E+00	221	6.00E-02
144	4.00E+00	183	1.07E+00	222	5.00E-02
145	3.73E+00	184	1.06E+00	223	4.00E-02
146	3.50E+00	185	1.05E+00	224	3.00E-02
147	3.15E+00	186	1.04E+00	225	2.53E-02
148	3.05E+00	187	1.03E+00	226	1.00E-02
149	3.00E+00	188	1.02E+00	227	7.50E-03
150	2.97E+00	189	1.01E+00	228	5.00E-03
151	2.87E+00	190	1.00E+00	229	4.00E-03
152	2.77E+00	191	9.75E-01	230	3.00E-03
153	2.67E+00	192	9.50E-01	231	2.50E-03
154	2.57E+00	193	9.25E-01	232	2.00E-03
155	2.47E+00	194	9.00E-01	233	1.50E-03
156	2.38E+00	195	8.50E-01	234	1.20E-03
157	2.30E+00	196	8.00E-01	235	1.00E-03
158	2.21E+00	197	7.50E-01	236	7.50E-04
159	2.12E+00	198	7.00E-01	237	5.00E-04
160	2.00E+00	199	6.50E-01	238	1.00E-04
161	1.94E+00	200	6.25E-01		
162	1.86E+00	201	6.00E-01		

Cross Section Conditioning Python Script

```
import csv
import math
import numpy

TotEnergy = []
ElEnergy = []
TotCrossSection = []
ElCrossSection = []
AllEnergy = []
TotEnergyF = []
ElEnergyF = []
TotCrossSectionF = []
ElCrossSectionF = []
AbsEnergyF = []
AbsCrossSectionF = []

isotope = "56Fe"
totalXSfile = "sigTot" + isotope + ".csv"
elasticXSfile = "sigEl" + isotope + ".csv"


# Generate working total cross section library
with open(totalXSfile, 'rb') as f:
    reader = csv.reader(f)
    x = 0
    for row in reader:
        if x == 0:
            EnCheck = 0
        else:
            EnCheck = TotEnergy[x-1]
        EnCheck2 = float(row[0])
        if EnCheck2 != EnCheck:
            TotEnergy.append(float(row[0]))
            AllEnergy.append(float(row[0]))
            TotCrossSection.append(float(row[1]))
            x += 1

print(len(TotEnergy))
print(len(AllEnergy))
print(min(TotEnergy))
print(min(AllEnergy))
print(max(TotEnergy))
print(max(AllEnergy))
```

```
# Generate working elastic scattering cross section library
with open(elasticXSfile, 'rb') as f:
    reader = csv.reader(f)
    y = 0
    for row in reader:
        if y == 0:
            EnCheck = 0
        else:
            EnCheck = ElEnergy[y-1]
        EnCheck2 = float(row[0])
        if EnCheck2 != EnCheck:
            ElEnergy.append(float(row[0]))
            ElCrossSection.append(float(row[1]))
            if EnCheck2 not in AllEnergy:
                AllEnergy.append(EnCheck2)
            y += 1

print(len(ElEnergy))
print(len(AllEnergy))
print(min(ElEnergy))
print(min(AllEnergy))
print(max(ElEnergy))
print(max(AllEnergy))

p = len(AllEnergy)

AllEnergy.sort()

print(AllEnergy[0])
print(AllEnergy[p-1])

# Generate full data point list
for q in range(0, len(AllEnergy)):
    TotEnergyF.append(AllEnergy[q])
    ElEnergyF.append(AllEnergy[q])
    AbsEnergyF.append(AllEnergy[q])
    TXS = numpy.interp(AllEnergy[q], TotEnergy, TotCrossSection)
    EXS = min(TXS, numpy.interp(AllEnergy[q], ElEnergy, ElCrossSection))
    AXS = TXS - EXS
    TotCrossSectionF.append(TXS)
    ElCrossSectionF.append(EXS)
    AbsCrossSectionF.append(AXS)
    print(q)
```

```
print("Done\n")

XSTotCSVname = "sigTot" + isotope + "_cond.csv"

with open(XSTotCSVname, 'wb') as csvfile:
    spamwriter = csv.writer(csvfile)
    for k in range(0, len(TotEnergyF)):
        spamwriter.writerow([TotEnergyF[k]] + [TotCrossSectionF[k]])

XSElCSVname = "sigEl" + isotope + "_cond.csv"

with open(XSElCSVname, 'wb') as csvfile:
    spamwriter = csv.writer(csvfile)
    for m in range(0, len(ElEnergyF)):
        spamwriter.writerow([ElEnergyF[m]] + [ElCrossSectionF[m]])

XSAbsCSVname = "sigAbs" + isotope + "_cond.csv"

with open(XSAbsCSVname, 'wb') as csvfile:
    spamwriter = csv.writer(csvfile)
    for n in range(0, len(AbsEnergyF)):
        spamwriter.writerow([AbsEnergyF[n]] + [AbsCrossSectionF[n]])

print("All Done\n")
```

Shielding Neutron Tracker Python Example Script

```
import csv
import math
import time
import numpy

# Isotope format is 56Fe, 27Al, etc.  Carbon is just C.
isotope = "56Fe"

# Material density in g/cc
Mat_den = 7.87

# Molar mass in g/mol
Mol_mass = 56.0

# The alpha divisor is used to stretch or compress the integration window
alphadiv = 0

#InitEnergy = float(raw_input("Enter neutron energy:  "))
InitEnergy = 2e7

NumHist = 10000
SetVerbose = 0

# Slab thickness in cm
slabmin = 0
slabmax = 5

#================================================
```

```
# This is the cross section "collapse" function

def xscoll(isotope, alphadiv, mol_mass):
    TotEnergy = []
    ElEnergy = []
    AbsEnergy = []
    TotCrossSection = []
    ElCrossSection = []
    AbsCrossSection = []
    TotXSInt = []
    ElXSInt = []
    AbsXSInt = []

    totalXSfile = "sigTot" + isotope + "_cond.csv"
    elasticXSfile = "sigEl" + isotope + "_cond.csv"
    absorptionXSfile = "sigAbs" + isotope + "_cond.csv"

    # Generate working total cross section library
    with open(totalXSfile, 'rb') as f:
        reader = csv.reader(f)
        x = 0
        for row in reader:
            TotEnergy.append(float(row[0]))
            TotCrossSection.append(float(row[1]))
            if x < 1:
                TotXSInt.append(float(row[1]))
            else:
                DeltaEnergy = TotEnergy[x] - TotEnergy[x-1]
                AvgXS = (TotCrossSection[x] + TotCrossSection[x-1]) / 2
                EXSInt = DeltaEnergy * AvgXS
                NewSum = EXSInt + TotXSInt[x-1]
```

```
            TotXSInt.append(NewSum)
        x += 1

# Generate working elastic scattering cross section library
with open(elasticXSfile, 'rb') as f:
    reader = csv.reader(f)
    x = 0
    for row in reader:
        ElEnergy.append(float(row[0]))
        ElCrossSection.append(float(row[1]))
        if x < 1:
            ElXSInt.append(float(row[1]))
        else:
            DeltaEnergy = ElEnergy[x] - ElEnergy[x-1]
            AvgXS = (ElCrossSection[x] + ElCrossSection[x-1]) / 2
            EXSInt = DeltaEnergy * AvgXS
            NewSum = EXSInt + ElXSInt[x-1]
            ElXSInt.append(NewSum)
        x += 1

# Generate working absorption cross section library
with open(absorptionXSfile, 'rb') as f:
    reader = csv.reader(f)
    x = 0
    for row in reader:
        AbsEnergy.append(float(row[0]))
        AbsCrossSection.append(float(row[1]))
        if x < 1:
            AbsXSInt.append(float(row[1]))
        else:
            DeltaEnergy = AbsEnergy[x] - AbsEnergy[x-1]
```

```
     AvgXS = (AbsCrossSection[x] + AbsCrossSection[x-1]) / 2
     EXSInt = DeltaEnergy * AvgXS
          NewSum = EXSInt + TotXSInt[x-1]
          AbsXSInt.append(NewSum)
    x += 1

lowcutoff = max(min(ElEnergy), min(TotEnergy))

highcutoff = min(max(ElEnergy), max(TotEnergy))

GroTopEn = highcutoff



# Alpha value for scattering
if Mol_mass > 1:
    alpha = ((Mol_mass - 1) / float((Mol_mass + 1)))**2
else:
    alpha = 0.5

if alphadiv == 0:
          alphaprime = alpha
else:
          alphaprime = alpha / alphadiv

CollEn = []
CollTotXS = []
CollElXS = []
CollAbsXS = []

XSInfoFileName = isotope + str(alphadiv) + "XSInfo.txt"
```

```
f = open(XSInfoFileName, 'w')

num_group = 0
while GroTopEn > lowcutoff:
        num_group += 1
        GroBotEn = GroTopEn * alphaprime
        DeltaEnergy = GroTopEn - GroBotEn
        GroEn = (GroTopEn + GroBotEn) / 2
        GroIntEnTotXSTop = numpy.interp(GroTopEn, TotEnergy, TotXSInt)
        GroIntEnTotXSBot = numpy.interp(GroBotEn, TotEnergy, TotXSInt)
        GroIntEnTotXS = GroIntEnTotXSTop - GroIntEnTotXSBot
        GroTotXS = round(GroIntEnTotXS / DeltaEnergy, 5)
        GroIntEnElXSTop = numpy.interp(GroTopEn, ElEnergy, ElXSInt)
        GroIntEnElXSBot = numpy.interp(GroBotEn, ElEnergy, ElXSInt)
        GroIntEnElXS = GroIntEnElXSTop - GroIntEnElXSBot
        GroElXS = min(round(GroIntEnElXS / DeltaEnergy, 5), GroTotXS)
        GroIntEnAbsXSTop = numpy.interp(GroTopEn, AbsEnergy, AbsXSInt)
        GroIntEnAbsXSBot = numpy.interp(GroBotEn, AbsEnergy, AbsXSInt)
        GroIntEnAbsXS = GroIntEnAbsXSTop - GroIntEnAbsXSBot
        GroAbsXS = min(round(GroIntEnAbsXS / DeltaEnergy, 5), GroTotXS - GroElXS)
        CollEn.append(GroEn)
        CollTotXS.append(GroTotXS)
        CollElXS.append(GroElXS)
        CollAbsXS.append(GroAbsXS)
        GroTopEn = GroBotEn
        GroAbsXS2 = GroTotXS - GroElXS
        f.write("\n%6.5e\t%6.5e\t%6.5e" %(GroEn, GroTotXS, GroElXS))
        f.write("\t%6.5e\t%6.5e\t%0.3f" %(GroAbsXS, GroAbsXS2, GroAbsXS2 / GroTotXS))

f.close()
```

```
        XSTotCSVname = "sigTot" + isotope + "_" + str(alphadiv) + ".csv"

        with open(XSTotCSVname, 'wb') as csvfile:
                spamwriter = csv.writer(csvfile)
                for x in range(0, len(CollEn)):
                                spamwriter.writerow([CollEn[num_group - x - 1]] + [CollTotXS[num_group - x - 1]])

        XSElCSVname = "sigEl" + isotope + "_" + str(alphadiv) + ".csv"

        with open(XSElCSVname, 'wb') as csvfile:
                spamwriter = csv.writer(csvfile)
                for x in range(0, len(CollEn)):
                                spamwriter.writerow([CollEn[num_group - x - 1]] + [CollElXS[num_group - x - 1]])

        XSAbsCSVname = "sigAbs" + isotope + "_" + str(alphadiv) + ".csv"

        with open(XSAbsCSVname, 'wb') as csvfile:
                spamwriter = csv.writer(csvfile)
                for x in range(0, len(CollEn)):
                                spamwriter.writerow([CollEn[num_group - x - 1]] + [CollAbsXS[num_group - x - 1]])

#================================================

#====================================
# This is the vector rotation function

def vecrot(omegahat, mu, phi, nu, DistTrav):
    # This rotates a vector based on the previous direction and the post-scatter data

    # This creates a vector based on the difference between the last neutron location
    # and the current neutron location
```

```
# a = dX
# b = dY
# c = dZ

a = omegahat[0]
b = omegahat[1]
c = omegahat[2]

# This prevents an arithmetic error in the event of a delta = 0

if a == 0:
    a = 1e-10
if b == 0:
    b = 1e-10
if c == 0:
    c = 1e-10

# The vecnorms are variables that appear in multiple entries in the rotation matrix

vecnorm1 = math.sqrt(a**2 + b**2 + c**2)
vecnorm2 = (b**2 + c**2)
vecnorm3 = vecnorm1*vecnorm2

# M is the rotation matrix

M = [[a/vecnorm1, -b/vecnorm1, -c/vecnorm1],
[b/vecnorm1, a*b**2/vecnorm3 + c**2/vecnorm2, a*b*c/vecnorm3 - b*c/vecnorm2],
[c/vecnorm1, a*b*c/vecnorm3 - b*c/vecnorm2, a*c**2/vecnorm3 + b**2/vecnorm2]]

# The new interaction point is chosen relative to the vector [1, 0, 0]
# intervec is the interaction vector using mu, phi, nu, and DistTrav
```

```
    intervec = [DistTrav*mu, DistTrav*nu*math.cos(phi), DistTrav*nu*math.sin(phi)]

    K = numpy.dot(M,numpy.transpose(intervec))

    return(K)

#=====================================

start_time = time.time()

TotEnergy = []
ElEnergy = []
AbsEnergy = []
TotCrossSection = []
ElCrossSection = []
AbsCrossSection = []



# Alpha value for scattering
alpha = ((Mol_mass - 1) / float((Mol_mass + 1)))**2

# Avogadro's number in at/mol
Avog_num = 6.022e23

# Number density in at/b*cm
NumDen = Mat_den * Avog_num / float(Mol_mass) * 1e-24

xscoll(isotope, alphadiv, Mol_mass)
```

```
# Cross Section Library
if alphadiv < 0.5:
    XSTotLib = "sigTot" + isotope + "_cond.csv"
    XSElLib = "sigEl" + isotope + "_cond.csv"
    XSAbsLib = "sigAbs" + isotope + "_cond.csv"
else:
    XSTotLib = "sigTot" + isotope + "_" + str(alphadiv) + ".csv"
    XSElLib = "sigEl" + isotope + "_" + str(alphadiv) + ".csv"
    XSAbsLib = "sigAbs" + isotope + "_" + str(alphadiv) + ".csv"

# Generate working total cross section library
with open(XSTotLib, 'rb') as f:
    reader = csv.reader(f)
    for row in reader:
        TotEnergy.append(float(row[0]))
        TotCrossSection.append(float(row[1]))

# Generate working elastic scattering cross section library
with open(XSElLib, 'rb') as f:
    reader = csv.reader(f)
    for row in reader:
        ElEnergy.append(float(row[0]))
        ElCrossSection.append(float(row[1]))

# Generate working absorption cross section library
with open(XSAbsLib, 'rb') as f:
    reader = csv.reader(f)
    for row in reader:
        AbsEnergy.append(float(row[0]))
        AbsCrossSection.append(float(row[1]))
```

```
ECutoff = min(min(TotEnergy), min(ElEnergy))



print("The initial neutron energy is %g eV." %InitEnergy)
print("================")
print("================")

AvgParms = numpy.array([0.0, 0.0, 0.0, 0.0, 0.0])
MinParms = numpy.array([1.0e10, 1.0e10, 1.0e10, 1e10, 1e10])
MaxParms = numpy.array([0.0, 0.0, 0.0, 0.0, 0.0])
Parms2 = numpy.array([0.0, 0.0, 0.0, 0.0, 0.0])
AvgParms_Alt = numpy.array([0.0, 0.0, 0.0, 0.0, 0.0])
M2Parms = numpy.array([0.0, 0.0, 0.0, 0.0, 0.0])
FinEngBuckets = numpy.zeros(15)
TransEngBuckets = numpy.zeros(15)
ReflEngBuckets = numpy.zeros(15)


RecordFileName = "slab" + isotope + str(alphadiv) + str(InitEnergy) + str(slabmax) + ".txt"

f = open(RecordFileName, 'w')

NeutRefl = []
NeutTrans = []

for x in range(0, int(NumHist)):

    NeutEnergy = InitEnergy

    TotDist = 0
```

```
NumScat = 0
FinDist = 0

# LastPoint is given in [x, y, z] format
LastPoint = [0.0, 0.0, 0.0]
xmax = 0.0

f.write("\n%g\nB" %x)

mu = 1

while ( NeutEnergy >= ECutoff):

    totXS = numpy.interp(NeutEnergy, TotEnergy, TotCrossSection)
    elXS = min(totXS, numpy.interp(NeutEnergy, ElEnergy, ElCrossSection))
    absXS = min(totXS - elXS, numpy.interp(NeutEnergy, AbsEnergy, AbsCrossSection))
    raddist = math.sqrt(LastPoint[0]**2 + LastPoint[1]**2 + LastPoint[2]**2)

    f.write("\t%g" %NeutEnergy)
    f.write("\t%g\t%g\t%g" %(totXS, elXS, absXS))
    f.write("\t%g\t%g\t%g" %(LastPoint[0], LastPoint[1], LastPoint[2]))

    AbsProb = absXS / totXS
    micXS = numpy.array([totXS, elXS, absXS])
    MacXS = micXS * NumDen
    TotXS = MacXS[0]
    ElXS = MacXS[1]
    AbsXS = MacXS[2]
    MeanFreePath = 1/MacXS[0]

    # Draw a random number for distance traveled
```

```
DistTrav = -math.log(numpy.random.random()) / TotXS
TotDist += DistTrav

# Draw a random number for the cosine of polar scattering angle
# The initial polar scattering angle is 0 for a perpendicular incident beam

  nu = math.sin(math.acos(mu))
# Draw a random number for the azimuthal scattering angle
phi = 2 * math.pi * numpy.random.random()

# IntPoint is the interaction point

if NeutEnergy == InitEnergy:
    IntPoint = [DistTrav, 0, 0]
    omegahat = IntPoint
else:
    IntPoint = LastPoint + vecrot(omegahat, mu, phi, nu, DistTrav)
    omegahat = IntPoint - LastPoint

LastPoint = IntPoint
f.write("\t%g\t%g\t%g" %(LastPoint[0], LastPoint[1], LastPoint[2]))
xcoord = LastPoint[0]
xmax = max(xcoord,xmax)

if xcoord <= slabmin:
  NeutRefl.append(NeutEnergy)
  NeutEnergy = 0
  FinEnergy = 0
  f.write("\nR")
elif xcoord >= slabmax:
  NeutTrans.append(NeutEnergy)
```

```
        NeutEnergy = 0
        FinEnergy = 0
        f.write("\nT")
    else:
        RandForInt = numpy.random.random()
        if RandForInt < AbsProb:
                FinEnergy = NeutEnergy
                NeutEnergy = 0
                dtr = LastPoint
                FinDist = math.sqrt(dtr[0]**2 + dtr[1]**2 + dtr[2]**2)
                f.write("\nA")
        else:
                mu = 2 * numpy.random.random() - 1
                NeutEnergy1 = ((1 + alpha) + (1 - alpha) * mu)/2 * NeutEnergy
                DeltaEnergy = NeutEnergy - NeutEnergy1
                NeutEnergy = NeutEnergy1
                FinEnergy = NeutEnergy
                NumScat += 1
                f.write("\tS\t%g\t%g\nS" %(mu, NeutEnergy1))


ResultArray = numpy.array([NumScat, FinEnergy, TotDist, FinDist, xmax])
if FinEnergy > 1e8:
    FinEngBuckets[14] += 1
elif FinEnergy > 1e7:
    FinEngBuckets[13] += 1
elif FinEnergy > 1e6:
    FinEngBuckets[12] += 1
elif FinEnergy > 1e5:
    FinEngBuckets[11] += 1
elif FinEnergy > 1e4:
```

```
    FinEngBuckets[10] += 1
elif FinEnergy > 1e3:
    FinEngBuckets[9] += 1
elif FinEnergy > 1e2:
    FinEngBuckets[8] += 1
elif FinEnergy > 1e1:
    FinEngBuckets[7] += 1
elif FinEnergy > 1e0:
    FinEngBuckets[6] += 1
elif FinEnergy > 1e-1:
    FinEngBuckets[5] += 1
elif FinEnergy > 1e-2:
    FinEngBuckets[4] += 1
elif FinEnergy > 1e-3:
    FinEngBuckets[3] += 1
elif FinEnergy > 1e-4:
    FinEngBuckets[2] += 1
elif FinEnergy > 1e-5:
    FinEngBuckets[1] += 1
else:
    FinEngBuckets[0] += 1

print("History number %g results:" %(x+1))
print("Total linear distance traveled:  %g cm." %TotDist)
print("Number of scatters:  %g." %NumScat)
print("Final energy:  %g eV." %FinEnergy)
print("Radial distance traveled:  %g cm." %FinDist)
print("X distance traveled:  %g cm." %xcoord)
print("Deepest penetration:  %g cm." %xmax)
print("================")
```

```
    Parms2 = Parms2 + ResultArray * ResultArray
    DeltaParms = ResultArray - AvgParms_Alt
    AvgParms_Alt = AvgParms_Alt + DeltaParms / (float(x) + 1)
    M2Parms = M2Parms + DeltaParms * (ResultArray - AvgParms_Alt)
    AvgParms = ((AvgParms * float(x)) + ResultArray) / (float(x) + 1)
    MinParms = numpy.minimum(MinParms, ResultArray)
    MaxParms = numpy.maximum(MaxParms, ResultArray)

TrProb = len(NeutTrans)/float(NumHist)
ReProb = len(NeutRefl)/float(NumHist)

print("Transmission probability:  %g" %TrProb)
print("Reflection probability:  %g" %ReProb)

VarParms = Parms2 / (float(x) + 1) - AvgParms * AvgParms
VarParms_Alt = M2Parms / (float(x))
f.close()

FinalResultsFile = "slab" + isotope + str(alphadiv) + str(InitEnergy) + str(slabmax) + "final.txt"

g = open(FinalResultsFile, 'w')

g.write("================")
g.write("\nThe initial neutron energy was:  %g eV." %InitEnergy)
g.write("\nAfter %g histories, the calculated parameters are:  " %NumHist)
g.write("\nMean linear distance traveled:  %g cm." %AvgParms[2])
g.write("\nMean radial distance traveled:  %g cm." %AvgParms[3])
g.write("\nMean deepest penetration:  %g cm." %AvgParms[4])
g.write("\nMean number of scatters:  %g." %AvgParms[0])
g.write("\nMean final energy:  %g eV." %AvgParms[1])
g.write("\nMinimum linear distance traveled:  %g cm." %MinParms[2])
```

```
g.write("\nMinimum radial distance traveled:  %g cm." %MinParms[3])
g.write("\nMinimum deepest penetration:  %g cm." %MinParms[4])
g.write("\nMinimum number of scatters:  %g." %MinParms[0])
g.write("\nMinimum final energy:  %g eV." %MinParms[1])
g.write("\nMaximum linear distance traveled:  %g cm." %MaxParms[2])
g.write("\nMaximum radial distance traveled:  %g cm." %MaxParms[3])
g.write("\nMaximum deepest penetration:  %g cm." %MaxParms[4])
g.write("\nMaximum number of scatters:  %g." %MaxParms[0])
g.write("\nMaximum final energy:  %g eV." %MaxParms[1])
g.write("\nLinear distance population standard deviation:  %g cm." %math.sqrt(VarParms[2]))
g.write("\nRadial distance population standard deviation:  %g cm." %math.sqrt(VarParms[3]))
g.write("\nDeepest penetration population standard deviation:  %g cm." %math.sqrt(VarParms[4]))
g.write("\nScatter population standard deviation:  %g." %math.sqrt(VarParms[0]))
g.write("\nFinal energy population standard deviation:  %g eV." %math.sqrt(VarParms[1]))
g.write("\nLinear distance sample standard deviation:  %g cm." %math.sqrt(VarParms_Alt[2]))
g.write("\nRadial distance sample standard deviation:  %g cm." %math.sqrt(VarParms_Alt[3]))
g.write("\nDeepest penetration sample standard deviation:  %g cm." %math.sqrt(VarParms_Alt[4]))
g.write("\nScatter sample standard deviation:  %g." %math.sqrt(VarParms_Alt[0]))
g.write("\nFinal energy sample standard deviation:  %g eV." %math.sqrt(VarParms_Alt[1]))
for x in range(0, len(FinEngBuckets)):

    g.write("\n10**%g eV to 10**%g eV\t%g" %(x-6, x-5, FinEngBuckets[x]))

elapsed_time = time.time() - start_time
g.write("\n\nElapsed Time:  %g s" %elapsed_time)
g.close()

TransmissionFile = "slab" + isotope + str(alphadiv) + str(InitEnergy) + str(slabmax) + "Trans.txt"

k = open(TransmissionFile, 'w')
for x in range(0, len(NeutTrans)):
```

```
        k.write("\n%g" %NeutTrans[x])
k.close()

ReflectionFile = "slab" + isotope + str(alphadiv) + str(InitEnergy) + str(slabmax) + "Refl.txt"

k = open(ReflectionFile, 'w')
for x in range(0, len(NeutRefl)):
        k.write("\n%g" %NeutRefl[x])
k.close()
```

Shielding Tracker Example Output

```
================
The initial neutron energy was:  1e+06 eV.
After 10000 histories, the calculated parameters are:
Mean linear distance traveled:  90.9749 cm.
Mean radial distance traveled:  0.654629 cm.
Mean deepest penetration:  11.1076 cm.
Mean number of scatters:  49.0712.
Mean final energy:  65.0913 eV.
Minimum linear distance traveled:  0.0882298 cm.
Minimum radial distance traveled:  0 cm.
Minimum deepest penetration:  0.00063832 cm.
Minimum number of scatters:  1.
Minimum final energy:  0 eV.
Maximum linear distance traveled:  359.761 cm.
Maximum radial distance traveled:  56.1016 cm.
Maximum deepest penetration:  57.1685 cm.
Maximum number of scatters:  193.
Maximum final energy:  646244 eV.
Linear distance population standard deviation:  101.805 cm.
Radial distance population standard deviation:  4.71823 cm.
Deepest penetration population standard deviation:  11.7399 cm.
Scatter population standard deviation:  62.0911.
Final energy population standard deviation:  6462.25 eV.
Linear distance sample standard deviation:  101.811 cm.
Radial distance sample standard deviation:  4.71846 cm.
Deepest penetration sample standard deviation:  11.7405 cm.
Scatter sample standard deviation:  62.0942.
Final energy sample standard deviation:  6462.57 eV.
10**-6 eV to 10**-5 eV      8688
10**-5 eV to 10**-4 eV      1157
10**-4 eV to 10**-3 eV      67
10**-3 eV to 10**-2 eV      57
10**-2 eV to 10**-1 eV      19
10**-1 eV to 10**0 eV       8
10**0 eV to 10**1 eV        1
10**1 eV to 10**2 eV        0
10**2 eV to 10**3 eV        1
10**3 eV to 10**4 eV        1
10**4 eV to 10**5 eV        0
10**5 eV to 10**6 eV        1
10**6 eV to 10**7 eV        0
```

10**7 eV to 10**8 eV 0
10**8 eV to 10**9 eV 0

Elapsed Time: 44.0846 s

VITA

Thomas Jay Harrison was born in Johnson City, Tennessee, so long ago that his children cannot comprehend the year. He attended Sulphur Springs Elementary School in Jonesborough, and Daniel Boone High School in Gray. After graduating as one of the valedictorians at Daniel Boone, he accepted an appointment to the United States Naval Academy in Annapolis, Maryland. He spent a year and a half at Navy before transferring back home to East Tennessee State University in Johnson City. After three semesters at ETSU, he finally came to his senses and came to the University of Tennessee.

He majored in mechanical engineering for a semester before switching to nuclear engineering after a chat with Dr. Lee Dodds, then the head of the nuclear engineering department. He finally finished his BS degree in nuclear engineering (*summa cum laude*) in 2002.

After graduating, he couldn't bring himself to leave. He worked with Dr. Wes Hines on his thesis and graduated with his MS degree in nuclear engineering in 2004. His original intent was to continue, but after deciding that he and Ms. Colleen McInturf should be married, he instead joined the work force.

He then moved to Pittsburgh, Pennsylvania, to work for the Bettis Atomic Power Laboratory. He worked in reactor design and analysis, and he really can't talk about his work beyond that.

After living and working in Pittsburgh for nearly two years, he kept his promise to Mrs. Colleen Harrison to move back home. He began work at Nuclear Fuel Services in Erwin, Tennessee, where he was a nuclear criticality safety engineer. He can talk about some of his work there, but has found that not many are interested in it.

While at Nuclear Fuel Services, he passed the Professional Engineer exam for nuclear engineering, and he has a state-issued license to practice nuclear engineering. He also began his work on his PhD in nuclear engineering, mainly so he could keep up with his wife, who already had her doctorate in occupational therapy.

Also while at Nuclear Fuel Services, he and Dr. Harrison had two children, Cailin and Brendan. As the first grandchildren on either side, they were immediately spoiled.

After three and a half years at Nuclear Fuel Services, he decided to move on to something else, and won a position as a research engineer at Oak Ridge National Laboratory in Oak Ridge, Tennessee. The family moved to Dandridge, Tennessee, halfway between family and his work.

After a year at Oak Ridge, the family welcomed another child, Taryn. She was immediately spoiled, as was the family's custom.

Finally, after nearly five years at Oak Ridge, and more than ten years after joining the work force, he has completed his quest for a PhD, working with his advisor, Dr. Larry Townsend, under the new department head, Dr. Wes Hines. And it comes just after the addition of a fourth child, Ryanne. She will likely be spoiled as well.