Windowed Multipole – An Efficient Doppler Broadening Technique for Monte Carlo

by

Colin Josey

B.S., Nuclear Engineering (2013)
University of New Mexico

Submitted to the Department of Nuclear Science and Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Nuclear Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

Signature of Author .. Signature redacted ....
Department of Nuclear Science and Engineering
May 18, 2015

Certified by... Signature redacted ....
Benoit Forget
Associate Professor of Nuclear Science and Engineering
Thesis Supervisor

Certified by... Signature redacted ....
Kord Smith
KEPCO Professor of the Practice of Nuclear Science and Engineering
Thesis Reader

Accepted by ... Signature redacted ....
Mujid S. Kazimi
TEPCO Professor of Nuclear Engineering
Chair, Department Committee on Graduate Students

Windowed Multipole – An Efficient Doppler Broadening Technique for Monte Carlo

by

Colin Josey

Submitted to the Department of Nuclear Science and Engineering on May 18, 2015, in partial fulfillment of the requirements for the degree of Master of Science in Nuclear Science and Engineering

Abstract

In this thesis, the windowed multipole method of Doppler broadening is developed from the multipole method. The multipole method is a pole and residue reformulation of the cross section data used in nuclear cross section data libraries. This form is advantageous as the Doppler broadened form is analytical, and the library itself is very small. However, multipole is quite slow. By introducing a slight approximation to the data, however, the computational cost can be reduced substantially. The method used to do this is called the windowed multipole method. The conversion of a multipole library to a windowed multipole library is detailed thoroughly. Then, a windowed multipole library is developed and tested on a light water reactor benchmark. In testing, the library outperformed target motion sampling and pseudomaterials in computational time, memory usage, and cache efficiency. Windowed multipole had a factor of 33 fewer cache misses than target motion sampling, and a factor of 80 fewer than pseudomaterials. This reduction in memory transfers makes it a very suitable on-the-fly Doppler broadening algorithm for Monte Carlo simulations on future supercomputer designs.

Thesis Supervisor: Benoit Forget
Title: Associate Professor of Nuclear Science and Engineering

Acknowledgments

This research was funded in part by the Consortium for Advanced Simulation of Light Water Reactors (CASL) under U.S. Department of Energy Contract No. DE-AC05-00OR22725; DOE's Center for Exascale Simulation of Advanced Reactors (CESAR) under U.S. Department of Energy Contract No. DE-AC02-06CH11357; and by a Department of Energy Nuclear Energy University Programs Graduate Fellowship. Without their support, this work would not be possible.

I would like to thank all of those who have taught me so much over the course of this project. Specifically, I would like to thank my advisors, Ben Forget and Kord Smith. Without them to bounce ideas off of, many aspects of this project would not exist.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last few years, intense interest has been focused on simulating nuclear reactors with coupled thermal-hydraulics and neutron transport. Such high fidelity simulations would be useful everywhere from the design stage to investigation of accidents. This sudden interest has come about since recent computers are becoming large enough to tackle the problem with minimal approximation. Thus, a full core could be simulated in a matter of days.

One issue in particular stands in the way of these simulations. In order to couple the temperature field into the neutronics simulation, the neutronics simulation will have to properly handle the change in collision probability that occurs when the target moves due to the temperature of the medium. There are many ways to handle this probability, from the many ways of performing Doppler broadening (performing a convolution integral to create an effective cross section) to sampling the target motion itself. However, the majority of methods in use today either require a great deal of computational time, a great deal of data stored in memory, or both.

With the advent of future exascale computers, it is predicted that the memory bandwidth will not increase proportionally with the floating point performance [2]. Thus, any task that wishes to take full advantage of such a machine would have to reduce memory load as much as possible. With the cross section library being frequently accessed inside of a Monte Carlo code, a substantial reduction in library size would have substantial performance benefits.

This thesis details the implementation of a new, memory efficient algorithm to perform Doppler broadening called the windowed multipole method. This method takes the original resonance data that is available for most isotopes, converts it into a form that can be analytically Doppler broadened, and then performs an optimization to minimize the number of operations required at the expense of an extremely small loss of accuracy. This method requires orders of magnitude less memory bandwidth than other techniques, while also taking less time to evaluate on current computers. As the memory available becomes slower relative to the rest of the machine, the advantage of the technique will only improve.

1.1 Structure of this Document

This thesis is segmented into five chapters. The first is this introductory chapter. Chapter 2 discusses Doppler broadening. It introduces the algorithms currently in use, their advantages, and their weaknesses. Specifically mentioned is the multipole algorithm and its two primary weaknesses. Then, in Chapter 3, these two weaknesses are addressed with a novel algorithm, the windowed multipole method. In Chapter 4, the resulting windowed multipole cross section libraries are tested for accuracy and performance. Direct accuracy comparisons are done between the library and the ENDF-B/VII.1 libraries included with MCNP6 [7]. Timing and memory usage comparisons are performed between windowed multipole and two other temperature-dependent algorithms: target motion sampling and pseudomaterials (both of which are described in Chapter 2). Finally, a brief summary of results are presented in Chapter 5.

1.2 On Nomenclature Usage

One issue with the literature with regards to Doppler broadening and specifically multipole is the inconsistent usage of symbols. In order to ensure that the meaning of each term of each equation is well understood, a nomenclature section is provided at the end of this chapter. In this nomenclature section, important variables used multiple times in the document will be described. Symbols will also be described alongside equations when used for the first time.

1.3 On the Machine Used

All performance testing was performed on the same machine, to ensure consistency and repeatability. The processor of this machine was an Intel®Core i7-970 operating at 3.2 GHz. This is a six core machine with 32 kiB of 8-way associative L1 data cache and 256 kiB of 8-way associative L2 cache per core. 12 MiB of 16-way associative L3 cache is shared between cores.

Nomenclature

$\sigma(v)$	The microscopic cross section at a velocity v, page 21
$\sigma_{\text{eff}}(v, T)$	The effective microscopic cross section at a neutron velocity v and temperature T, page 21
A	The ratio of the mass of the target to the mass of the neutron, page 26
k_b	Boltzmann's constant, taken as 8.617342×10^{-5} eV K^{-1} [9], page 20
M	The target mass, page 20
$P(V_T)$	The probability of a target having a velocity V_T, page 20
p_j	The pole at index j, j being an index to the whole set of poles that create a cross section, page 26
$R(v, T)$	The reaction rate for a neutron at velocity v going through a material at temperature T, page 21
$r_{j,x}$	The residue for reaction x corresponding to pole p_j, page 26
T	Temperature, page 20
u	The square root of the energy, E, page 26
v	The velocity of a neutron, page 21
v'	The relative velocity of a neutron and its target, page 21
V_T	The target velocity, page 20

Chapter 2

Temperature Effects

In order to run a Monte Carlo simulation, the probability of collision must be known in order to randomly sample a distance for a particle to travel. Precisely knowing this probability is essential to correctly simulating a nuclear reactor. These probabilities are available in evaluations such as the ENDF-B/VII.1 [3] library as microscopic cross sections. For a two-body problem in which the target is at rest, the cross section is only a function of projectile energy, so all evaluations are reported in this form.

The issue is, though, that a target moves due to thermal energy. In order to compute the probability of collision, first we must know the distribution of target velocities. A brief introduction to these distributions is shown in Section 2.1. Then, the target motion probability distribution must be used to calculate new collision probabilities. This is done in Section 2.2.

2.1 Target Motion Distributions

There are many possible ways to model the target motion distribution, depending on what material is being interacted with. In the simplest case, the material is a free gas. In a free gas, the targets have no interaction amongst themselves, and their velocity distribution follows that of the Maxwell-Boltzmann distribution [15]. In this particular case, the target motion is only a function of temperature (T) and target mass (M), and is shown in Equation (2.1).

$$P(V_T)dV_T = \frac{4}{\pi}\beta^{3/2}V_T^2 e^{-\beta V_T^2} dV_T \tag{2.1}$$
$$\beta = \frac{M}{2k_bT}$$

In the case of solids, the binding energy between atoms can alter the target motion in an anisotropic way. In these circumstances, the probability distribution is now a function of angle, and far more complex. A particular example for lattices with cubic symmetry (such as those in UO_2) is presented in [18]. There are a wide variety of distributions for different crystal structures.

Although fully representing the target motion is preferable, it is substantially more difficult. Due to this, the majority of cross sections are processed assuming a free gas target distribution. As mentioned in the second paper above, this approximation becomes less accurate at low temperatures and with particularly strongly bound crystals. As most reactors operate at temperatures significantly higher than room temperature, these are minor issues for most materials (with exception to some moderators). Thus, all further analysis in this thesis is done assuming a free gas distribution.

2.2 Doppler Broadening

The process of Doppler broadening fundamentally comes down to conserving reaction rate. For a single neutron, the reaction rate is the product of the velocity and the probability of collision. The process is shown in Equation 2.2. In this form, the reaction rate at a relative velocity v' is integrated over all possible target velocity vectors $\vec{V}_T$. Only those that are physically possible ($v' > 0$) are added to the contribution.

$$R(v,T) = \int_{\text{All } \vec{V}_T : v' > 0} v'\sigma(v')P(\vec{V}_T)d\vec{V}_T \tag{2.2}$$

The reaction rate can also be thought to be the product of the velocity and an effective probability of collision.

$$\begin{aligned} R(v,T) &= v\sigma_{\text{eff}}(v,T) \\ v\sigma_{\text{eff}}(v,T) &= \int_{\text{All } \vec{V}_T : v'>0} v'\sigma(v')P(\vec{V}_T)d\vec{V}_T \end{aligned} \tag{2.3}$$

In this general form, whatever the probability distribution, be it Maxwell-Boltzmann as we will use throughout the rest of this document, or some crystalline lattice probability distribution, Equation (2.3) will hold.

When the Maxwell-Boltzmann distribution from Equation (2.1) is substituted into Equation (2.3), the isotropy allows for the conversion of the complex integral to that of a much simpler form as shown in Equation (2.4).

$$\sigma_{\text{eff}}(v,T) = \sqrt{\frac{\beta}{\pi}} \int_0^\infty \left(\frac{v'}{v}\right)^2 \sigma(v') \left[e^{-\beta(v'-v)^2} - e^{-\beta(v'+v)^2}\right] dv' \tag{2.4}$$

There are a large number of ways to implement this algorithm, with varying strengths and weaknesses. The rest of this chapter will be dedicated to discussing five different algorithms.

The first, known as SIGMA1, is an algorithm to exactly Doppler broaden data available in pointwise form. This algorithm is commonly used in preprocessing tools such as NJOY [17], and will be described in detail in Section 2.2.1. The next algorithm converts the Doppler broadening integral into a target motion sampling problem, and then samples the target motion whenever a cross section is needed. This algorithm will be detailed in Section 2.2.2. The third is to perform a curve fit of the temperature data and store it for later usage and is shown in Section 2.2.3. Fourth, simple interpolation between two different temperatures can be done. This is the basis of the Pseudomaterials technique in Section 2.2.4. The final algorithm performs a partial fraction expansion of the resonance data available with most isotopes and broadens these fractions exactly. This algorithm is presented in Section 2.2.5, and then greatly expanded in Chapter 3.

2.2.1 SIGMA1

The SIGMA1 [4] algorithm is the most commonly used algorithm, as it works on pointwise data, which any other kind of data can be converted into. The idea is to first perform a coordinate transform to Equation 2.4, replacing $\beta v^2 = y^2$ and $\beta v'^2 = x^2$:

$$\begin{aligned}\sigma(y,T) &= \sigma^*(y,T) - \sigma^*(-y,T) \\ \sigma^*(y,T) &= \frac{1}{y^2}\left(\frac{1}{\pi}\right)^{1/2}\int_0^\infty x^2\sigma(x,0)e^{-(x-y)^2}dx\end{aligned}$$

Upon substitution of $\sigma(x,0) = A_k + C_k x^2$ for the region $x \in (x_k, x_{k+1})$, the algorithm simplifies into an analytical form:

$$\begin{aligned}\sigma^*(y,T) = \frac{2}{y^2\sqrt{\pi}}\sum_k C_kH_4 + 4C_kyH_3 + (A_k + 6C_ky^2)H_2 + \\ (2A_ky + 4C_ky^3)H_1 + (A_ky^2 + C_ky^4)H_0\end{aligned}$$

where H_n is short for $H_n(x_{k+1} - y, x_k - y)$, which is:

$$\begin{aligned}H_n(a,b) &= F_n(a) - F_n(b) \\ F_n(a) &= \frac{2}{\sqrt{\pi}}\int_0^a Z^n e^{-z^2}dZ\end{aligned}$$

The set of equations $F_n(a)$ form a recursive set, and an analytical form using the error function is available. Thus, to evaluate a cross section, each point is the sum over each line segment of those 10 terms. While the number of segments can be reduced, as those far away contribute very little, this algorithm is particularly slow. As such, it is used to **preprocess** libraries before use in the Monte Carlo simulation. For each temperature T in the problem, the entire 0K pointwise dataset is broadened using the above equation from 0K and stored pointwise. Used in this way, although pointwise lookup is computationally cheap, it requires the storage of large quantities of data.

2.2.2 Target Motion Sampling

Target motion sampling is the technique used in the Serpent Monte Carlo code [22]. It takes an alternate approach in that, instead of reconstructing an equivalent cross section, it samples the target motion from the function that creates the equivalent cross section.

The technique has two stages. First, the maximum possible cross section given any target velocity (within a reasonable band), called the majorant cross section (Σ_{maj}) is calculated prior to simulation. During the simulation, the neutron is transported stochastically using this cross section. At the destination, the target motion is sampled to calculate the real total cross section at the relative velocity $\Sigma^0_{\text{tot}}(E', x)$. Then, the collision is kept with the probability

$$P = \frac{\Sigma^0_{\text{tot}}(E', x)}{\Sigma_{\text{maj}}}. \tag{2.5}$$

In order to sample the target velocity, it is noted that the effective cross section can be represented in two ways:

$$\sigma_{\text{eff}}(v, T) = \iint \sigma(v')P(V_t, \mu)dV_t d\mu$$
$$\sigma_{\text{eff}}(v, T) = \frac{1}{v}\iint v'\sigma(v')P_{\text{MB}}(V_t)dV_t d\frac{\mu}{2},$$

Thus, the distribution of target velocities given a Maxwell Boltzmann target distribution is given by the following equation.

$$P(V_t, \mu) = \frac{v'}{2v}P_{\text{MB}}(V_t) \tag{2.6}$$

It is not trivial to sample from Equation 2.6. The method used splits the distribution into two components and samples randomly from one or the other. Once a velocity is known, then the angle is sampled from an isotropic distribution. Finally, the relative velocities are compared to ensure that such a collision is physically possible. Once a physically realistic target velocity is sampled from, Equation 2.5 sampled. If Equation 2.5 indicates success, the particle is moved. If not, the process begins again.

This technique has numerous benefits. First, since the Doppler broadening is performed in a general way that is not a function of the structure of the data, it will be correct so long as the Maxwell-Boltzmann assumption is correct. Secondly, the technique requires only one library, which can be at either 0K or at some temperature below that of all expected temperatures in the problem. The latter form has been found to be more computationally efficient [23].

However, the primary disadvantage is the computational time. As mentioned by Viitanen and Leppänen [23], target motion sampling with an elevated basis temperature reduced performance by 46 to 60%, depending on the problem simulated. Secondly, the majorant cross section needs to be tabulated to allow for proper sampling. This additional table adds overhead.

2.2.3 Curve Fitting

One approximate method is to process the cross section data at many temperatures, and then curve fit at each energy point the temperature dimension of the data. For example, at energy E_g, the cross section can be represented as follows:

$$\sigma(T, E_g) \approx \sum_{i=1}^{N} \frac{a_{g,i}}{T^{i/2}} + \sum_{i=1}^{N} b_{g,i} T^{i/2} + c_g \tag{2.7}$$

The values $a_{g,i}$, $b_{g,i}$ and c_g are the curve fitting coefficients at E_g for this reaction. As for the value N, it has been found that a value of 8 allowed for a library to be within 0.1% for all reactions, isotopes, and energies [25]. While Equation (2.7) is very quick to evaluate, a 17 term expansion would require the storage of approximately 17 times as much data as would be required for a single temperature.

2.2.4 Pseudomaterials

Pseudomaterials is arguably the most basic of all of these algorithms. Given a data set at temperature T_1, and another at T_2, all one needs to do to find the cross section at T is to interpolate between them. There are a few different methods to perform this interpolation, such as linear-linear:

$$\sigma(E,T) = \sigma(E,T_1) + [\sigma(E,T_2) - \sigma(E,T_1)]\left[\frac{T-T_1}{T_2-T_1}\right]$$

Or log-log:

$$\sigma(E,T) = \ln(\sigma(E,T_1)) + [\ln(\sigma(E,T_2)) - \ln(\sigma(E,T_1))]\left[\frac{\ln(T)-\ln(T_1)}{\ln(T_2)-\ln(T_1)}\right]$$

While the choice of methods matters, the most important thing is the spacing between temperature datasets. For complicated isotopes, such as ^{238}U, log-log interpolation could not get 0.1% accuracy with 28K steps between temperatures [21]. For most other isotopes, the requirements are less stringent.

However, if 28K is taken as the required spacing, and a range of temperatures from 300K to 3000K is of interest, nearly 100 times as much data will need to be stored compared to the single temperature case.

2.2.5 Multipole

The multipole formalism is, essentially, a different way to store the fundamental resonance information from the cross section libraries. When evaluators run experiments to calculate the cross section of a reaction, the model that the data is fit to is one of several simplifications of R-matrix theory [16]. R-matrix is a two-body model of the quantum-physical interaction between a projectile and a target. However, in its full form, the matrix itself is infinite in size, as there are an infinite number of possible channels in which particles can enter or leave. So, this matrix is simplified in a variety of ways, most often into the Reich-Moore or the multi-level Breit-Wigner (MLBW) forms. The coefficients that go into those models are distributed for many isotopes over their resolved resonance energy range.

The functions used to evaluate the cross sections from these simplifications have a unique property. They can be reconstructed using a partial fraction expansion that, instead of representing a cross section through a series of energies and channel widths, represents a cross section using poles and residues [11]. For example, the cross section for reaction x could take the form shown in Equation (2.8). The term j represents the sum over all quantum numbers, levels, and poles necessary to fully reconstruct the function.

$$\sigma_x(E) = \frac{1}{E}\sum_j \Re\left[\frac{r_{j,x}}{p_j - \sqrt{E}}\right] \tag{2.8}$$

When Equation (2.8) is Doppler broadened, however, the main advantage of the multipole formalism shows itself. The Doppler broadened form is shown in Equation (2.9).

$$\sigma_x(u,T) = \frac{1}{2u^2\sqrt{\xi}}\sum_j \Re\left[i r_{j,x}\sqrt{\pi}W(z) - \frac{r_{j,x}}{\sqrt{\pi}}C\left(\frac{p_j}{\sqrt{\xi}}, \frac{u}{2\sqrt{\xi}}\right)\right] \tag{2.9}$$

Where:

$$u = \sqrt{E}$$
$$\xi = \frac{k_b T}{4A}$$
$$z = \frac{u - p_j}{2\sqrt{\xi}}$$
$$C\left(\frac{p_j}{\sqrt{\xi}}, \frac{u}{2\sqrt{\xi}}\right) = 2p_j \int_0^\infty du' \frac{exp\left[-\frac{(u+u')^2}{4\xi}\right]}{p_j^2 - u'^2}$$

There are two functions of interest. First, W is the Faddeeva function, a function in which numerous high performance evaluators are available [13, 8]. The second function, C, has been found to be so small as to be completely ignored when $E \gg k_b T/A$ [12].

There are two major advantages to this technique. First, the resolved resonance region can be stored as a few thousand complex numbers instead of hundreds of thousand pointwise values. This reduces the memory requirements for a single temperature tremendously. Second, since the Doppler broadening is analytical, more than one temperature need not be stored.

There are three drawbacks. The first drawback is that this only works when the cross section evaluators have provided resonance parameters which can be converted to multipole. Currently, this is limited to isotopes with cross sections not in pointwise format and without any charged particle exit channels. The next drawback is computational cost. Each resonance needs $2(l+1)$ poles and residues to reconstruct it. In the case of ^{238}U as represented in the ENDF-B/VII.1 library, that will yield 11,520 poles. The Faddeeva function must be evaluated at each pole for each cross section evaluation. Thus, multipole can be even slower than SIGMA1. Finally, although the C function is negligible for $E \gg k_bT/A$, k_bT/A for ^{238}U at 300K is 1.1×10^{-4} eV. This is even worse for low mass isotopes at higher temperatures. The last two of these issues are remedied Chapter 3, at the cost of slightly decreased overall accuracy.

2.3 Summary

Now that Doppler broadening has been introduced, along with five possible methods to perform it, it is worth making a table comparing one algorithm to another. First, Table 2.1 compares computational time relative to single temperature table lookup.

Method	Time
SIGMA1 – Preprocessed	Reference
SIGMA1 – On the Fly	Extremely slow
Target Motion Sampling	2× Reference
Multipole	Extremely slow
Curve fitting	Near Reference
Pseudomaterials	2× Reference

Table 2.1: Methods comparison – computational cost

Table 2.2 similarly compares the memory requirements relative to single temperature table lookup.

Method	Memory
SIGMA1 – Preprocessed	One pointwise library per temperature
SIGMA1 – On the Fly	One pointwise library
Target Motion Sampling	One pointwise library + Majorant table
Multipole	$0.01\times$ a pointwise library
Curve fitting	$17\times$ a pointwise library
Pseudomaterials	10-$100\times$ a pointwise library

Table 2.2: Methods comparison – memory cost

Multipole is unique in that it can take substantially less memory than methods currently in use. Even if it were to be optimized to take a factor of ten longer than lookup in a preprocessed library, it would have niche value on a memory-constrained machine. However, Chapter 3 details a novel optimization of multipole called windowed multipole, and Chapter 4 shows that this algorithm can do much better than a factor of ten.

Chapter 3

Windowed Multipole

Although Multipole had major advantages (analytic Doppler broadening, low memory cost), it has two issues. The first is with regards to the C function in Equation (2.9). When $E < kT/A$, either the C function needs to be evaluated, or some replacement must be made. The second is with regards to computational performance. As previously explained, for ^{238}U there are 11,520 poles. For each cross section evaluation, the Faddeeva function would have to be evaluated once per pole to reconstruct the cross section. This typically made Multipole on the order of several hundred times slower than other on-the-fly algorithms. Both issues are solved in the same way.

There are two interesting effects when a pole is taken individually. The first is that, distant from the center of a resonance, the pole is relatively smooth [12]. This allows this region to be replaced with a lower order approximation, such as a curve fit [6]. The second is that the difference between a Doppler broadened pole and a 0K pole approaches zero in these smooth regions. Both of these effects are shown in Figure 3-1.

This understanding forms the basis for windowed multipole. At any given point in energy, the contribution of the majority of poles could be represented by a single curve fit. The goal then is to simultaneously store as few curve fits as possible, evaluate as few poles per cross section evaluation as possible, and compromise accuracy as little as possible.

This leads to the windowing part of windowed multipole. Given a set of poles

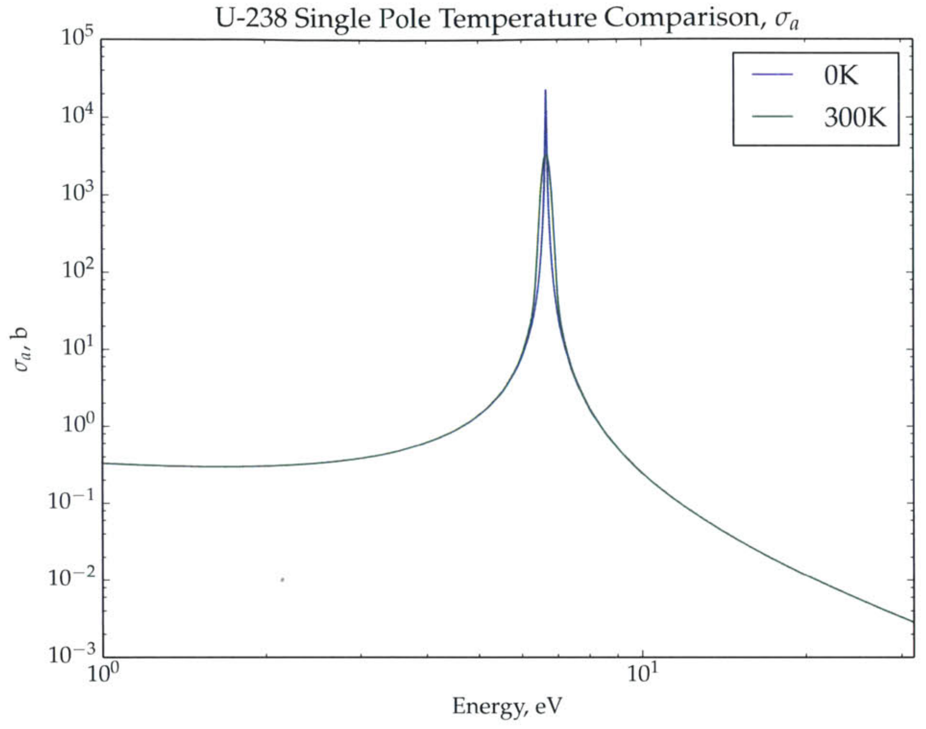

Figure 3-1: A single pole comparison between 0K and 300K for ^{238}U

and residues that represent the cross section, we first segment the energy domain with a grid. Each cell is called an inner window. For each inner window, another window is laid on top, called the outer window. All poles that have a real component lying within the outer windows are exactly evaluated. An optimization routine then identifies the optimal size of these outer windows. The remaining terms are then curve fit. The process is shown in Figure 3-2.

Thus, if one wanted to evaluate a cross section, the process is to find what inner window brackets the energy, look up which poles must be evaluated exactly, evaluate them, and add in the window's curve fit.

As for the C function, the curve fitting can also eliminate it. The C function is only relevant at low energies. These low energies are often very smooth, and can be entirely represented by curve fits. If the curve fit itself can be exactly Doppler broadened with no approximation, the error disappears.

However, there are several unknowns that must be dealt with first to ensure that

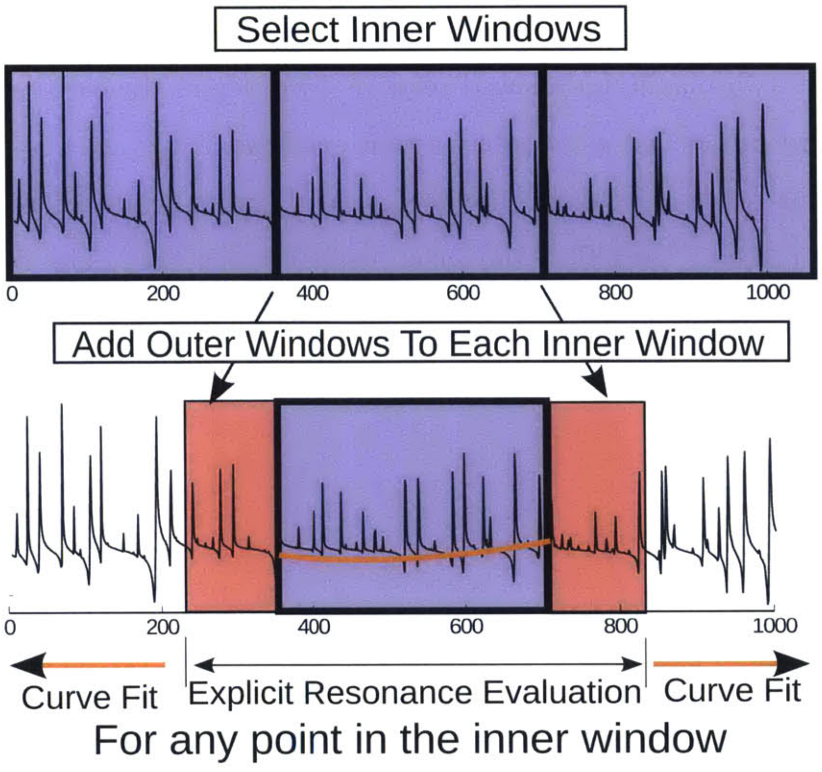
Select Inner Windows
0
200
400
600
800
1000
Add Outer Windows To Each Inner Window
0
200
400
600
800
1000
Curve Fit
Explicit Resonance Evaluation
Curve Fit
For any point in the inner window

Figure 3-2: The windowing process

the algorithm works quite well. The first is the general structure of the optimization process, detailed in Section 3.1. The second one is the choice of Faddeeva function to evaluate, as there are several different algorithms. Three are compared in Section 3.2. Finally, not all data is easy to compress into the windowed multipole library. One particular issue is the existence of "File 3" data in the ENDF-6 format. This is pointwise data that must be summed with the resonance data to reconstruct the cross section. The current method used to eliminate this issue is described in Section 3.3.

3.1 The Optimization Process

The first issue is the optimization process. This process transforms a multipole library (that is precomputed from the resonance parameters) into a windowed multipole library. The goal is to make the actual computation of a cross section as cheap and as efficient as possible. The transformation also introduces minute approximations to the data. Thus, a balance is formed. The more approximations made, the faster the cross section can be evaluated. As such, the goal is to take a user defined tolerance, often measured as a maximum allowed relative error across the entire dataset, and make the library run as fast as possible in that envelope.

To make this process possible, three components must be considered. The first is the choice of energy grid that the inner windows exist on. The second is the curve fitting function to be used. The last component is the structure of the actual optimization algorithm. The rest of this section is dedicated to the analysis of these facets.

3.1.1 Energy Grid

In the original development of windowed multipole, the only requirement set on the energy grid was that it should not require a binary search due to the possible performance repercussions. This eliminates arbitrarily sized windows, but it also leaves available equally spaced windows. It is of course not necessary that the windows be equally spaced along the energy axis. In light of that, three different axes were tested: energy, momentum, and lethargy.

^{238}U was processed into three different windowed multipole libraries. Each of them had the same number of windows, 2000. For each window, the outer window was set to span a set distance past the inner window such that the difference in span between the inner and outer window was a constant. This distance was 50 eV. The only difference was the size of the inner windows of each library: one equally spaced in energy, one equally spaced in momentum, and one equally spaced in lethargy. The value of the 0K σ_a cross section is plotted in Figures 3-3, 3-4, and 3-5, for each of the libraries, with a comparison to the exact 0K value.

In essence, each different mode has a different compromise. The linear in energy window is very accurate at high energy, and not very accurate at low energy. The opposite can be said of the equally spaced in lethargy. The equally spaced in momentum version is roughly balanced between the two. The choice between the spacing methods depends on whether or not it is easier to recover accuracy at lower energies or higher energies by increasing the outer window size.

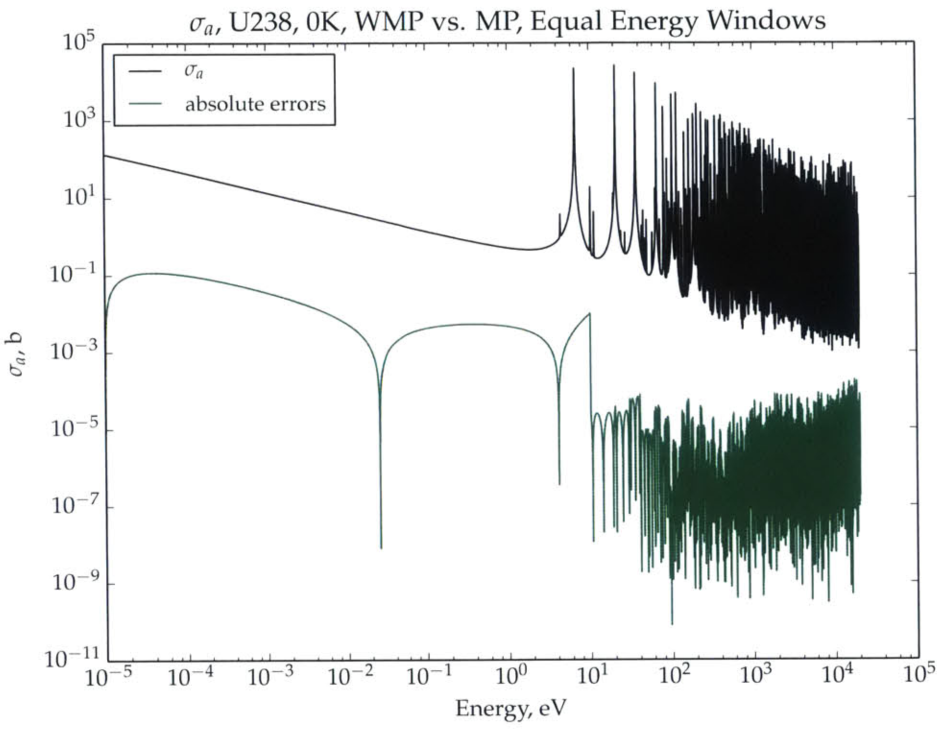

Figure 3-3: Using equal in energy windows

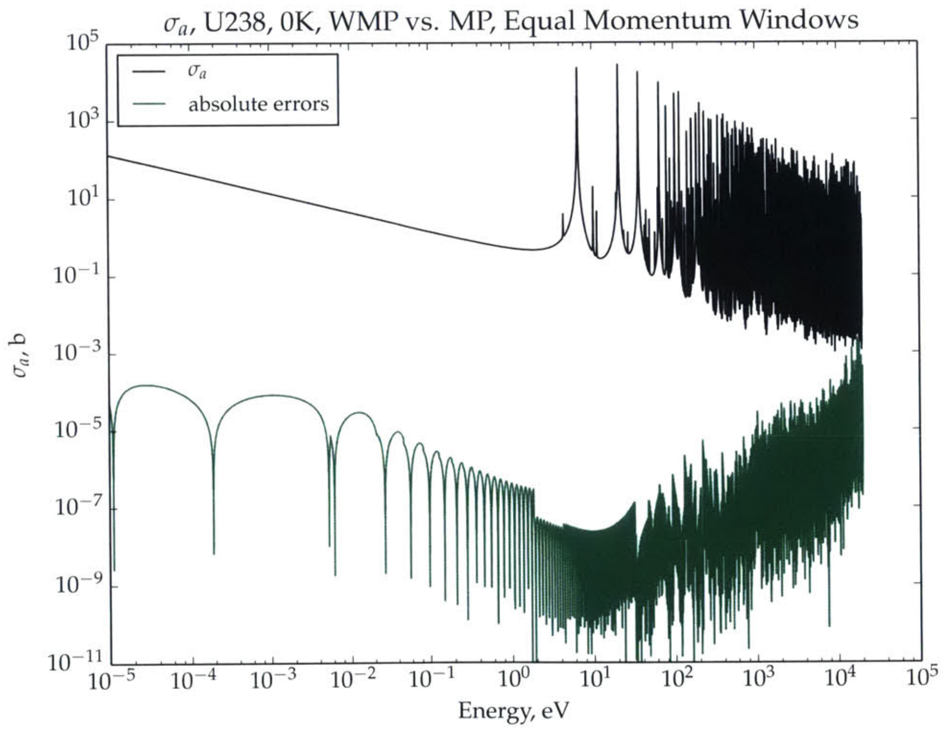

Figure 3-4: Using equal in momentum windows

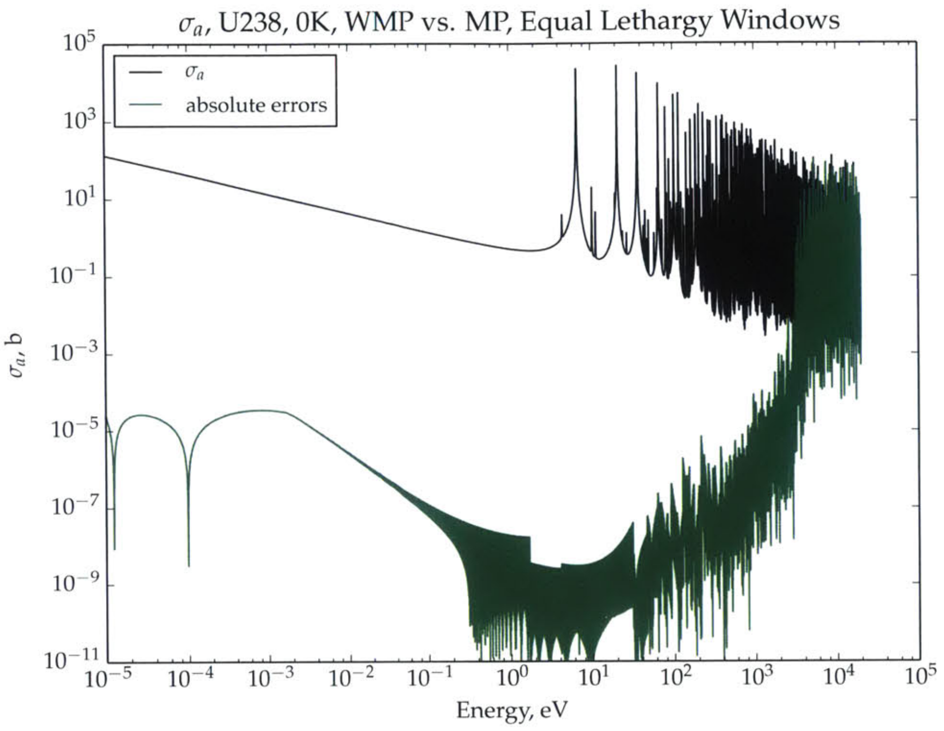

Figure 3-5: Using equal in lethargy windows

The 6.67 eV resonance of ^{238}U is shown in Figure 3-6. In this, the positive real pole, the negative real pole in which the sign is flipped to make it plottable, and their sum are shown. It is clear that either pole has little to no impact on the high energy. However, both have a large impact at low energy. All of the σ_a pole and residue pairs have similar $1/v$ components at low energy.

Essentially, all poles have an impact at low energy, but few have an impact at high energy. Thus, to reduce the curve fit error, more poles would have to be considered per unit error at low energies than at high energies. Furthermore, since nuclear reactors often have the majority of their neutrons at thermal energy, full reactor analysis is most sensitive to increases in computational cost at low energy. Both combined indicate that anything that adds low energy errors should be avoided at all costs. This eliminates the equally spaced in energy method.

Finally, it is noted that going from equally spaced in momentum bins to equally spaced in lethargy only slightly decreases low energy error and massively increases

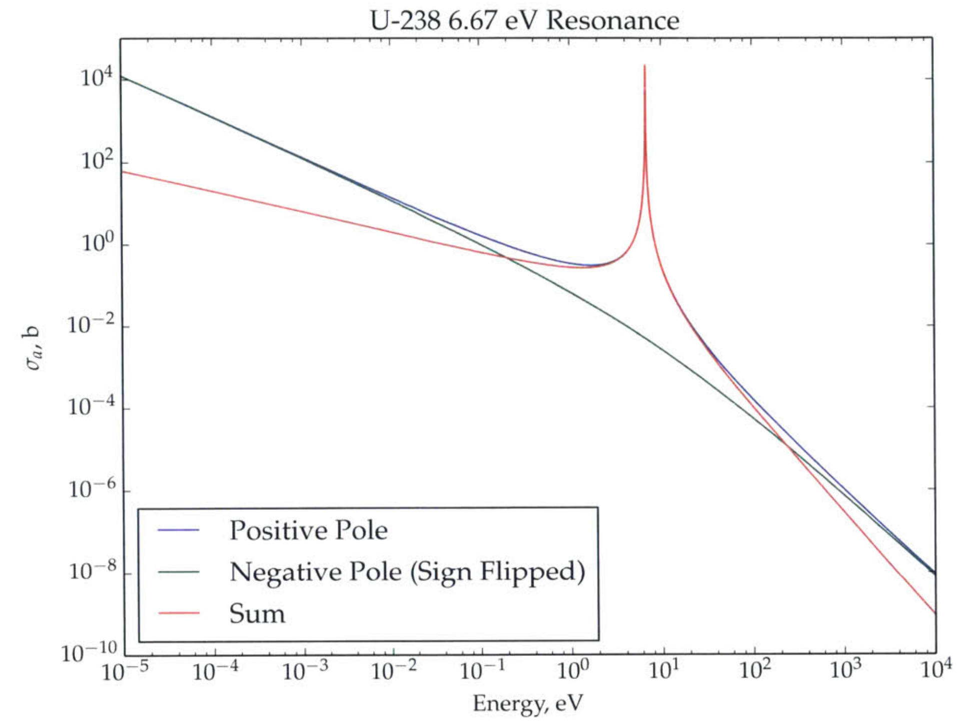

Figure 3-6: Pole reconstruction of the 6.67 eV resonance of ^{238}U

high energy error. From this, equally spaced in momentum is the preferred spacing, and will be used throughout the rest of this document.

3.1.2 Curve Fits

The curve fit needs to meet two criteria, accuracy and Doppler broadenability. Accuracy is important for obvious reasons. If we could not accurately reconstruct the data with the curve fit, it serves no purpose.

In order to construct an accurate curve fitting scheme, the curve fit must reconstruct exactly as many cases of interest as possible. There are three functions that are particularly important. The first is that of a constant. Not only is a constant generally useful, but it also is the low-energy asymptotic representation of the scattering cross section at 0K. The second is that of $1/v$, as it is the asymptotic form of absorption cross sections. The final one of interest is $1/E$, but for less obvious reasons. When a resonance is expanded into poles and residues, as shown in Figure 3-6, the individual poles take a $1/E$ form in the asymptotic region. This can also be found in the equations, as $1/E$ is the leading term in Equation (2.8). Thus, if a single pole is being curve fitted without its counterpart, $1/E$ becomes essential.

The second criteria, that the curve fit must be Doppler broadenable, is required for two reasons. The first is due to the inaccuracy at low energies of the multipole formalism when the C function of Equation (2.9) is ignored, as explained in this chapter's introduction. The second is that, also at low energies, the cross section is often quite smooth and resonance free. However, it is composed of thousands of the tail ends of resonances. In order to get Doppler broadening at low energies, the choice is then between broadening possibly thousands of poles, or broadening a single curve fit. For computational efficiency reasons, the latter choice is obvious.

Several polynomials were tested, but only one met all the above criteria. It takes the form of Equation (3.1).

$$\sigma(E) = \sum_{n=0}^{N} a_n E^{n/2-1}. \tag{3.1}$$

In this form, $1/E$, $1/v$, and a constant could all simultaneously be represented exactly. Furthermore, Equation (3.1) can itself be Doppler broadened in a recursive way [14].

$$\begin{aligned}
\sigma(E,T) &= \sum_{n=0}^{N} a_n \mathfrak{D}_n(E,T) \\
\mathfrak{D}_0(E,T) &= \frac{1}{E}\operatorname{erf}(\sqrt{\alpha E}) \\
\mathfrak{D}_1(E,T) &= \frac{1}{\sqrt{E}} \\
\mathfrak{D}_2(E,T) &= \left[\frac{1}{2\alpha} + E\right]\mathfrak{D}_0(E,T) + \frac{e^{-\alpha E}}{\sqrt{\alpha\pi E}} \\
\forall n > 0, \mathfrak{D}_{n+2}(E,T) &= \left[\frac{2n+1}{2\alpha} + E\right]\mathfrak{D}_n(E,T) - \frac{n(n-1)}{4\alpha^2}\mathfrak{D}_{n-2}(E,T) \\
\alpha &= \frac{A}{k_b T}
\end{aligned}$$

In this form, only a single error function and a single exponential need to be evaluated. The rest is simple. Thus, the Doppler broadening of a curve fit can be performed quickly with minimal overhead. There was also another accidental advantage to this fit. When one takes Equation (2.8) and performs the Laurent series expansion on it at $u = 0$, it takes the form of Equation (3.2), which is structurally identical to Equation (3.1).

$$\frac{1}{u^2}\left[\frac{r_{j,x}}{p_j - u}\right] = \sum_{n=0}^{\infty} \frac{r_{j,x}}{p_j^{n+1}} u^{n-2} \tag{3.2}$$

This converges so long as $|p| > |u|$. This implies that the curve fit can always exactly reconstruct any pole when near zero energy given enough terms. As mentioned many times prior, low energies are the most important region for curve fitting. This polynomial is used as the curve fit of choice for the rest of this document.

3.1.3 The Optimization Algorithm

The optimization algorithm is designed to minimize computational time. As with many optimization algorithms, this one is a bit convoluted. In order to keep the algorithms reasonably brief, Table 3.1 shows some symbolic abbreviations used throughout this section.

Symbol	Meaning
$\mathbf{P}$	the set of poles for the window
$\mathbf{C}$	the set of curve fit terms for the window
$\sigma_e(E,T)$	the exact cross section operator for this window
$\sigma_w(E,T)$	the WMP cross section operator for this window
$C_b(E,T,\mathbf{C})$	the broadened curve fit evaluation operator at T, E
$C_u(E,\mathbf{C})$	the unbroadened curve fit evaluation operator at T, E
$P(E,T,\mathbf{P})$	the pole evaluation operator at T, E
$\mathbf{T}$	the span of all temperatures of interest
$\mathbf{E}$	the span of all energies within this window
$fit(x(E))$	the curve fit coefficient generator on some function $x(E)$

Table 3.1: Algorithm notation table

Using this notation, the overall optimization algorithm for a single window is presented in Algorithm 1. The rationale for this algorithm is that it is always cheaper to not broaden a curve fit than to broaden one, and that even a broadened curve fit should be cheaper than evaluating several poles and residues exactly. This algorithm

Algorithm 1 Optimization Process of a Single Window

```
$\mathbf{P} \leftarrow \varnothing$
$\mathbf{C} \leftarrow fit(\sigma_e(\mathbf{E}, 0))$
if the maximum error in the domain of $\mathbf{T}$, $\mathbf{E}$ is less than the tolerance without broadening the curve fits then
    $\sigma_w(E,T) \leftarrow C_u(E, \mathbf{C})$
    Go on to the next window
else if the maximum error in the domain of $\mathbf{T}$, $\mathbf{E}$ is less than the tolerance while broadening the curve fits then
    $\sigma_w(E,T) \leftarrow C_b(E, T, \mathbf{C})$
    Go on to the next window
end if
$\mathbf{P} \leftarrow$ an optimal set of poles using Algorithm 2
$\mathbf{C} \leftarrow fit(\sigma_e(\mathbf{E}, 0) - P(E, 0, \mathbf{P}))$
if there exists a $\mathbf{P}$ such that the maximum error in the domain of $\mathbf{T}$, $\mathbf{E}$ is less than the tolerance without broadening the curve fits then
    $\sigma_w(E,T) \leftarrow C_u(E, \mathbf{C}) + P(E, T, \mathbf{P})$
    Go on to the next window
else if there exists a $\mathbf{P}$ such that the maximum error in the domain of $\mathbf{T}$, $\mathbf{E}$ is less than the tolerance while broadening the curve fits then
    $\sigma_w(E,T) \leftarrow C_b(E, T, \mathbf{C}) + P(E, T, \mathbf{P})$
    Go on to the next window
end if
If no setting has been found, fail and alert the user that the curve fit order is too small or that $\mathbf{P}$ may not be allowed to grow enough
```

raises several interesting problems to implement. What is σ_e defined as? How does one efficiently find if the error is less than a tolerance for all temperature and energy? And finally, how does one choose an optimal set of poles, $\mathbf{P}$?

The Exact Cross Section

As explained earlier, multipole itself cannot be used as the reference cross section, as it is inexact at low energies. SIGMA1, detailed in Section 2.2.1, is exact for a pointwise dataset. Given a sufficiently fine pointwise set, it can be used as a reference solution. However, there is a bit of an issue. If we need to know the value of the cross section at a variety of energies and temperatures that are not known prior, SIGMA1 will be very inefficient. As such, a crossover was implemented. Below this threshold,

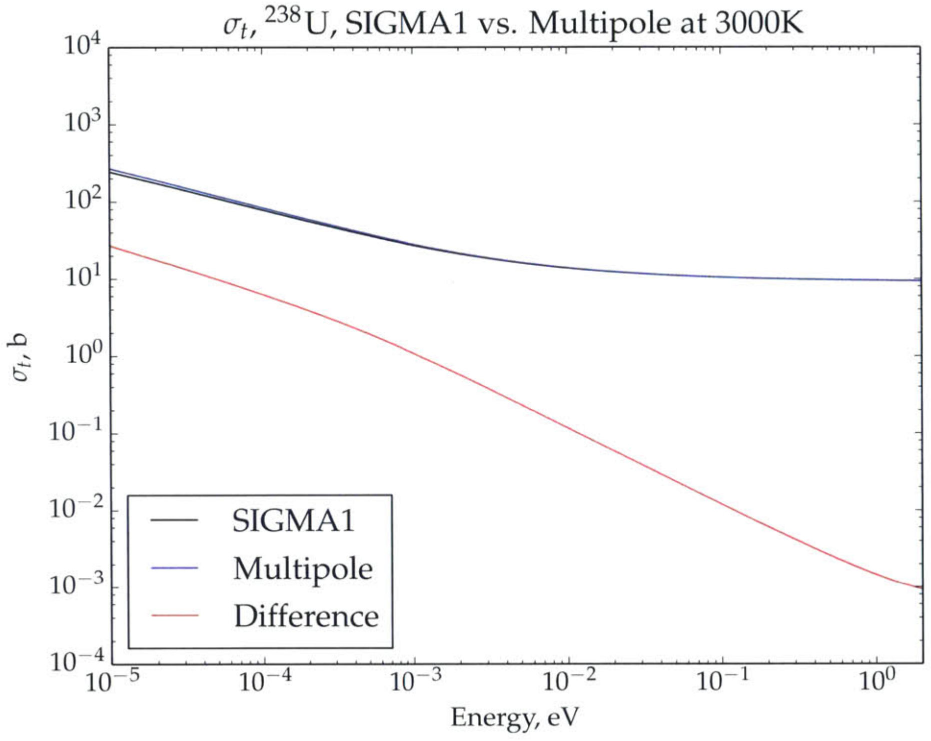

Figure 3-7: Difference between normal multipole and SIGMA1 at 3000K for ^{238}U

SIGMA1 is used to generate the reference solution. Above this value, multipole is used. A graph of the difference between multipole and SIGMA1 for ^{238}U is shown in Figure 3-7. The discrepancy drops below 0.1% at around 1 eV.

Similarly, the transition point for a lower mass isotope, say, ^{27}Al, is around 20 eV, as shown in Figure 3-8. As this cutoff should be roughly proportional to atomic weight ratio and little else, a simple 2-point correlation can be made. In the automation step, this correlation was used to predict the cutoff for other isotopes.

Finding if the Error Criterion is Met

It would be useful to define a maximum allowed relative error and use that as the benchmark in which success is measured. This makes the error checking portion of the code quite difficult. There are two approaches here. The first is to create a function f, defined in Equation 3.3, and search for the existence of a zero in the two-dimensional

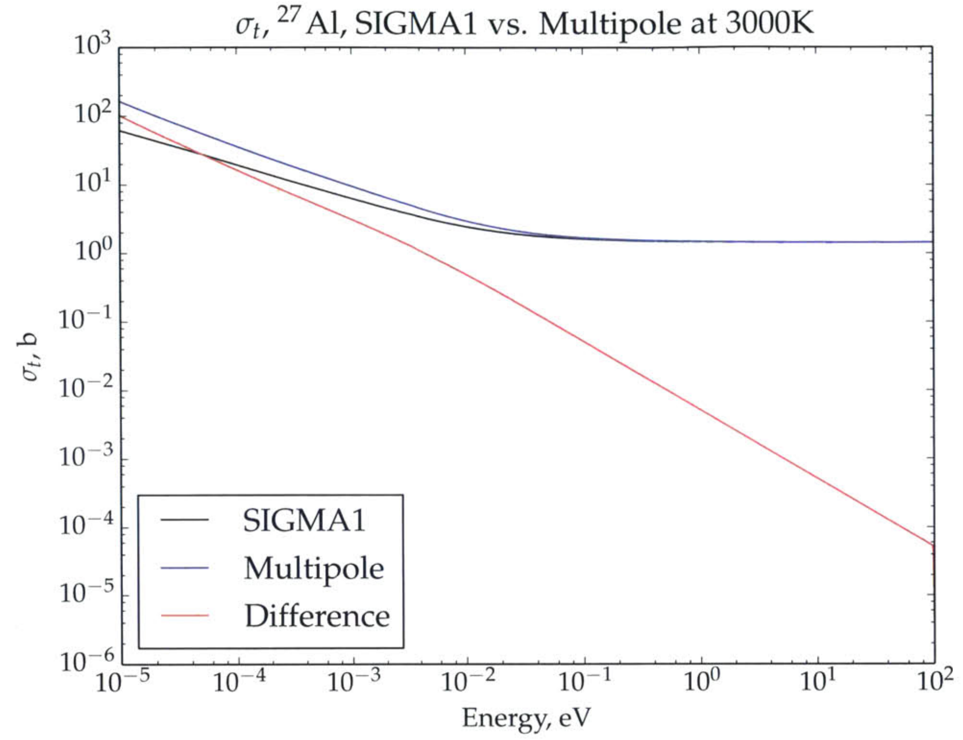

Figure 3-8: Difference between normal multipole and SIGMA1 at 3000K for ^{27}Al

energy-temperature span.

$$f(E,T) = \frac{|\sigma_w(E,T) - \sigma_e(E,T)|}{\sigma_e(E,T)} - \epsilon \tag{3.3}$$

Unfortunately, not finding a zero of this function gives no evidence of the success or failure in meeting the error criterion. It very well may be that the zero finder keeps missing its intended target. Furthermore, there are no guarantees that Equation 3.3 is smooth, which makes the search for zeros even more difficult. The other approach is to formulate Equation 3.4, and find the global maximum of f.

$$f(E,T) = \frac{|\sigma_w(E,T) - \sigma_e(E,T)|}{\sigma_e(E,T)} \tag{3.4}$$

This is itself not trivial, and there is no guarantees with any algorithm of success. However, only an approximate maximum is of interest, for if any point exceeds tolerance, the process can be safely terminated with a failure. The algorithm currently implemented to search for the global maximum is the differential evolution technique [20]. In the code, it is configured with the settings listed in Table 3.2. These values were found entirely by adjusting and checking, and likely do not actually resemble the optimal settings.

Variable	Value
Differential Weight	0.5
Crossover Probability	0.9
Population	100

Table 3.2: Differential evolution settings

Finding an Optimal Set of Poles

The actual pole finding process is also a bit convoluted. In order to reduce the dimension of the problem, instead of scanning the computational cost of each pole relative to the error introduced, only the outer window span is varied. Furthermore, simple bisection searching is performed to find the optimal window size. The general process is shown in Algorithm 2. There are several possible improvements to this algorithm, such as treating j_{begin} and j_{end} as the optimization targets instead of S_{ow}, but these have not been attempted.

3.2 Faddeeva Functions

Now we consider components that do not change the structure of the library. The choice of Faddeeva function is a relatively important one, as the performance and accuracy of this function have a very strong bearing on the performance and accuracy of the final result. As such, three different Faddeeva functions were tested. The first is the Faddeeva Package by Steven G. Johnson [13]. This algorithm has been tested by its author and it was found "that the accuracy is typically at least 13 significant digits in both the real and imaginary parts." This algorithm was considered the reference solution. It will be known as "MIT W" throught the rest of this document. The second algorithm is the one that came with the original WHOPPER code [11], and

Algorithm 2 Outer Window Optimization

```
$M \leftarrow$ midpoint of the inner window
$S_{low} \leftarrow$ the initial guess for the minimum window size (0)
$S_{high} \leftarrow$ the initial guess for the maximum window size
$S_{ow} \leftarrow S_{high}$
repeat
    $j_{\text{begin}} \leftarrow$ index of first pole with real component $> M - S_{ow}/2$
    $j_{\text{end}} \leftarrow$ index of last pole with real component $< M + S_{ow}/2$
    $\mathbf{P} \leftarrow [j_{\text{begin}} : j_{\text{end}}]$
    $\mathbf{C} \leftarrow fit\,(\sigma_e(\mathbf{E}, 0) - P(E, 0, \mathbf{P}))$
    if the maximum error < tolerance then
        $S_{high} = S_{ow}$
        $S_{ow} = (S_{high} + S_{low})/2$
    else if iteration $\neq 1$ then
        $S_{low} = S_{ow}$
        $S_{ow} = (S_{high} + S_{low})/2$
    else
        Fail and alert the user.
    end if
until Converged
```

will be called "WHOPPER W" throughout the rest of this document. The third one is from the $\text{MC}^2 2$ code [8], and will be called "QUICK W" throughout this document. There are other algorithms that were not thoroughly tested. For example, there is a Fourier transform method whose advantage is that it is fully vectorizable [1]. Such an algorithm may be advantageous on future architectures, such as GPUs.

Both MIT and WHOPPER W have similar structures. Each separates the complex domain into regions in which differing expansions are used. The primary difference between them is the shape of the domain for each method, as well as how many terms are used. The QUICK W algorithm takes a different approach. In the region $|z| < 6$ (where z is the input variable to the Faddeeva function), an interpolation table is used. Outside this region, a three-term expansion is used.

In order to properly scale just which part of each algorithm is important, a basic hydrogen scattering simulation was run. At each collision, the cross section of ^{238}U as represented in WMP format was evaluated. At each evaluation of the Faddeeva function, the value of z was tallied. This is shown in Figure 3-9. From this, it is clear

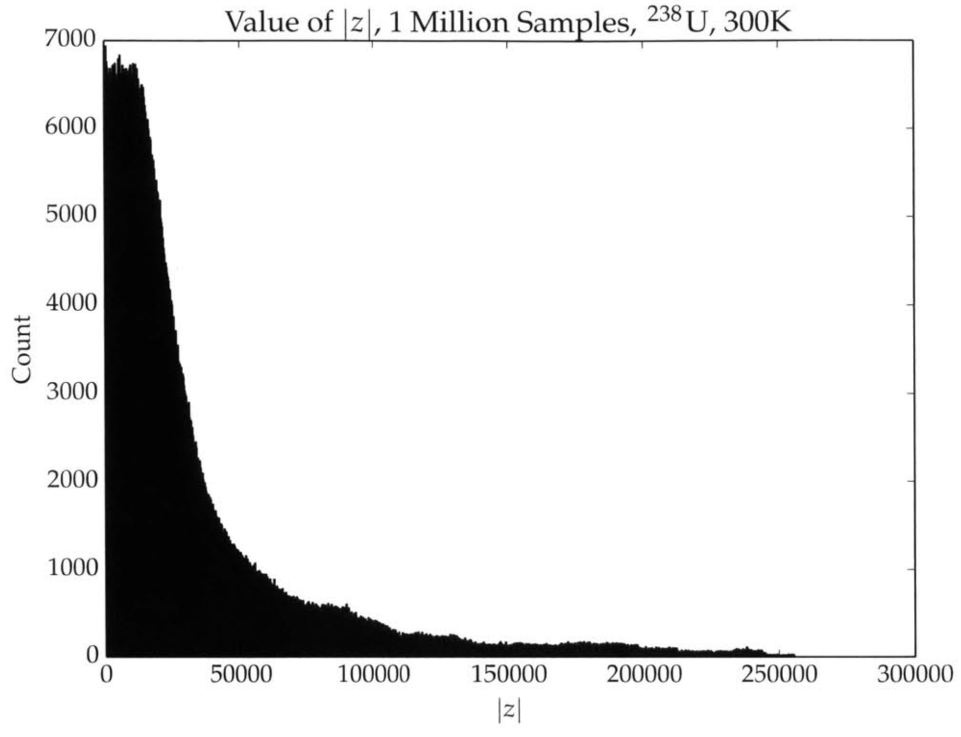

Figure 3-9: $|z|$ histogram for a simple hydrogen scattering simulation

that the majority of evaluations are with values of $|z|$ in excess of 10 thousand. This is good, as most of these algorithms use fewer terms as $|z| \to \infty$. For example, when $\Re z + \Im z > 4000$, MIT W uses only two terms in its expansion.

However, it is also worth considering when low-$|z|$ is important. As the value of $z = (u - p_j)/(2\sqrt{\xi})$, this value will approach zero in two circumstances. The first is when $u \approx p_j$, as is the case at the peak of a resonance. The second is when $\sqrt{\xi} \to \infty$, which occurs when $T \to \infty$. As such, it is most difficult to evaluate the shape of the peak of a resonance at high temperatures.

Plotted in Figure 3-10 and Figure 3-11 are comparisons of WHOPPER W and QUICK W to MIT W respectively. Relative to one another, it is clear that the WHOPPER W is much more accurate near the peaks. Even then, fairly substantial errors in the wings of the resonance make usage of WHOPPER W unfavorable. Table 3.3 details the time cost per cross section used in the making of these two graphs.

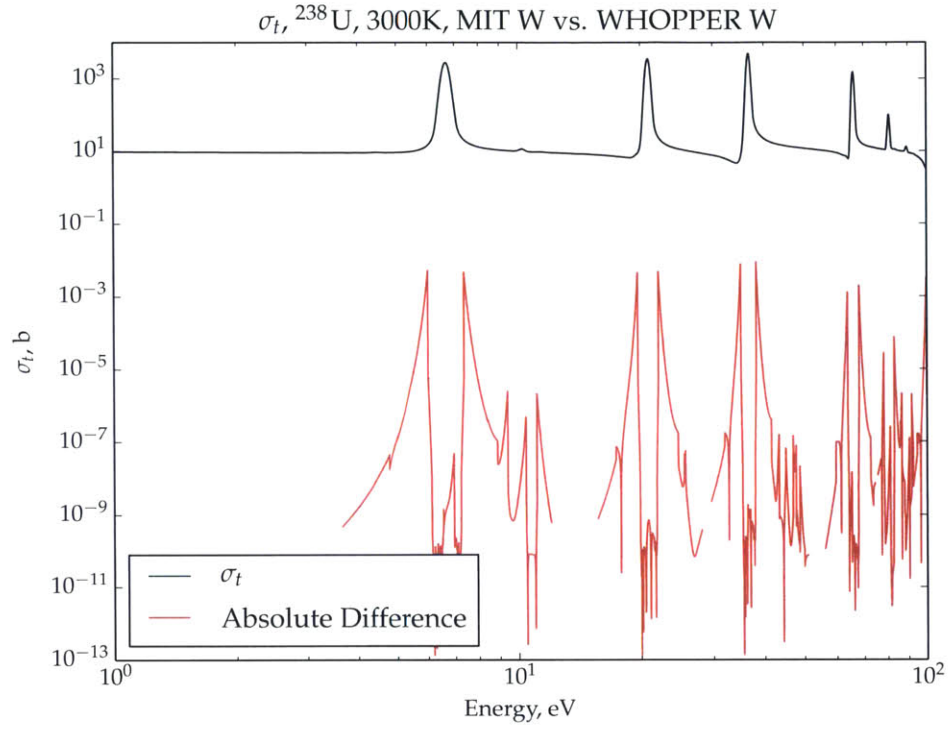

Figure 3-10: ^{238}U σ_t MIT W to WHOPPER W comparison

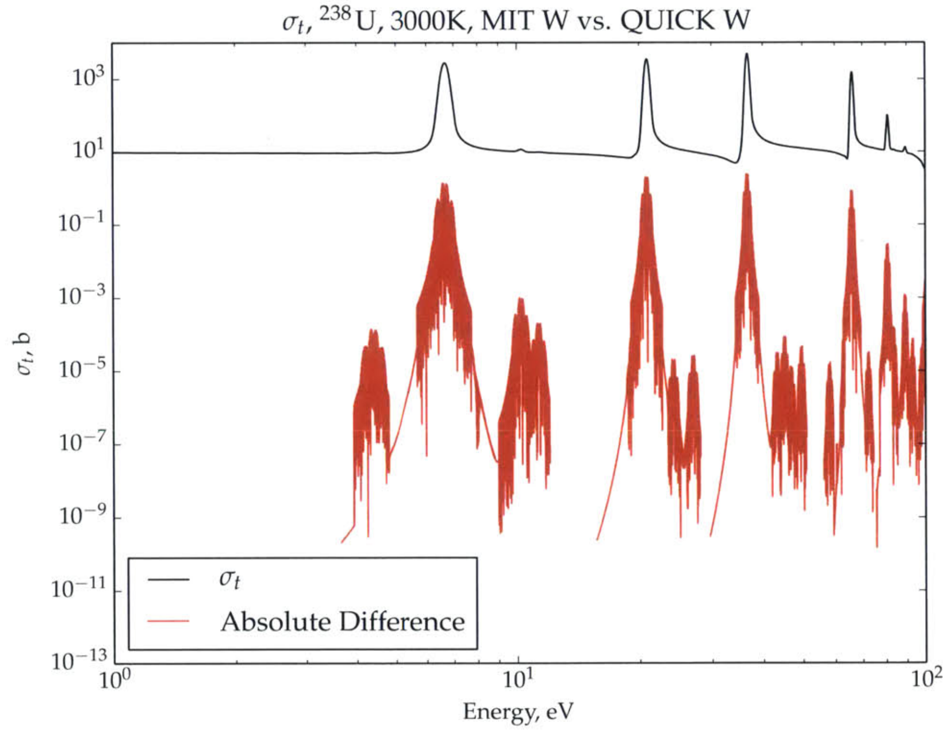

Figure 3-11: ^{238}U σ_t MIT W to QUICK W comparison

Algorithm	Time Per, μs
MIT W	0.740
WHOPPER W	0.498
QUICK W	0.413

Table 3.3: Time cost of differing Faddeeva functions

Although alternative algorithms could in principle provide a large improvement in performance in exchange for accuracy, this has not yet been investigated on a fully integrated problem. In order to ensure the accuracy of, or at least the predictability of the algorithm, the MIT W Faddeeva function was used throughout the rest of this document. It will be noted in Section 4.4.1 that the Faddeeva function is itself not a major contributor to the evaluation time of windowed multipole.

3.3 Dealing With File 3

The "File 3" data, so called because it is File 3 in the ENDF-6 manual [9], is a set of pointwise data that must be summed into the cross section to properly calculate it. There are two components of interest, those that affect the primary cross sections σ_t, σ_a, σ_s, and σ_f, and those that do not (known as secondary reactions).

Take, for example, ^{238}U. The resolved resonance region extends from 10^{-5} eV to 20 keV. The File 3 data is shown in Figure 3-12 for σ_t. It is convenient that the value is zero for all the components of the resolved resonant region. Furthermore, when Doppler broadened, as done in Figure 3-13 to 3000K, the error introduced is extremely minor. For this isotope, the pointwise 0K data is merely saved alongside and loaded into the simulation for use as the fast cross section. For similar reasons, the secondary reactions, which are often at energies above the resolved resonance region, are also just saved as pointwise and not treated with any Doppler broadening in the simulation.

However, there are isotopes in which this is not even close to a valid solution. For example, there is ^{23}Na. The resonant region spans 600 eV to 500 keV. The File 3 data is plotted in Figure 3-14.

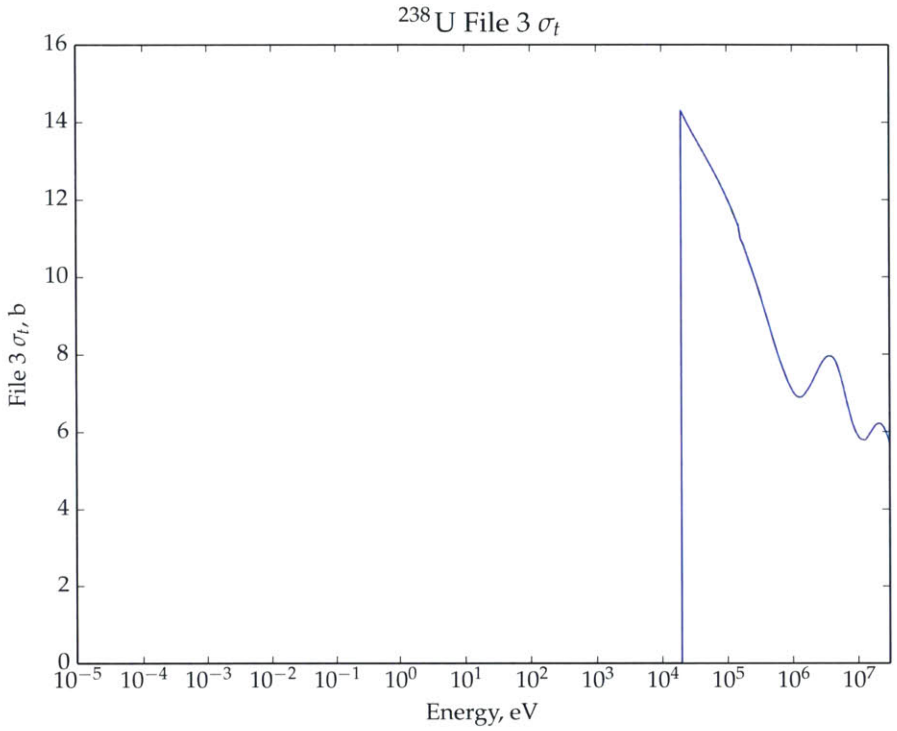

Figure 3-12: ^{238}U File 3 for σ_t

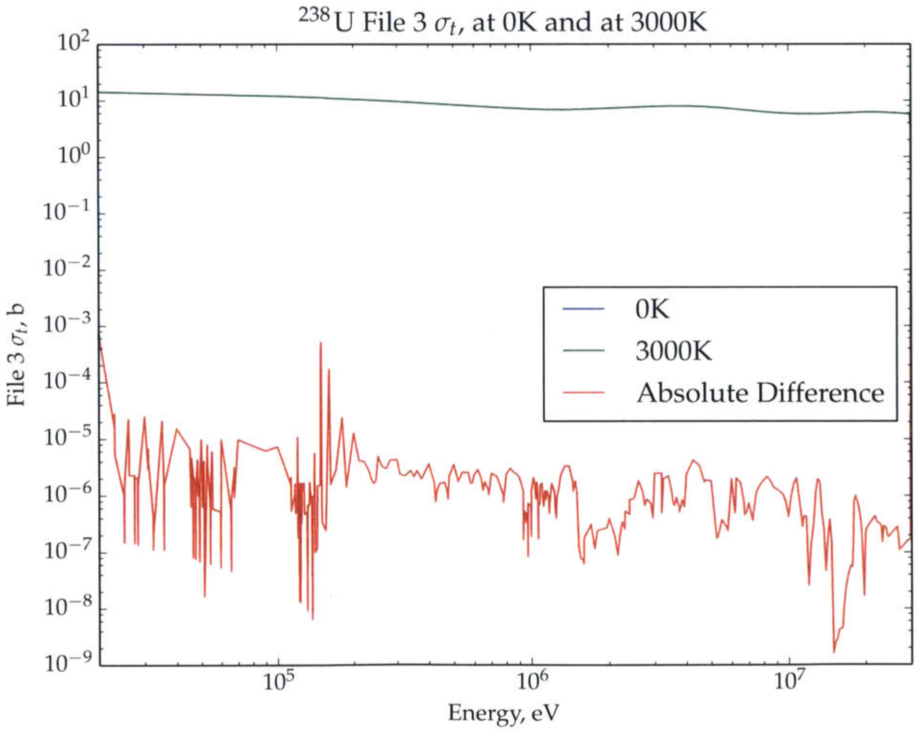

Figure 3-13: ^{238}U File 3 for σ_t Doppler broadened

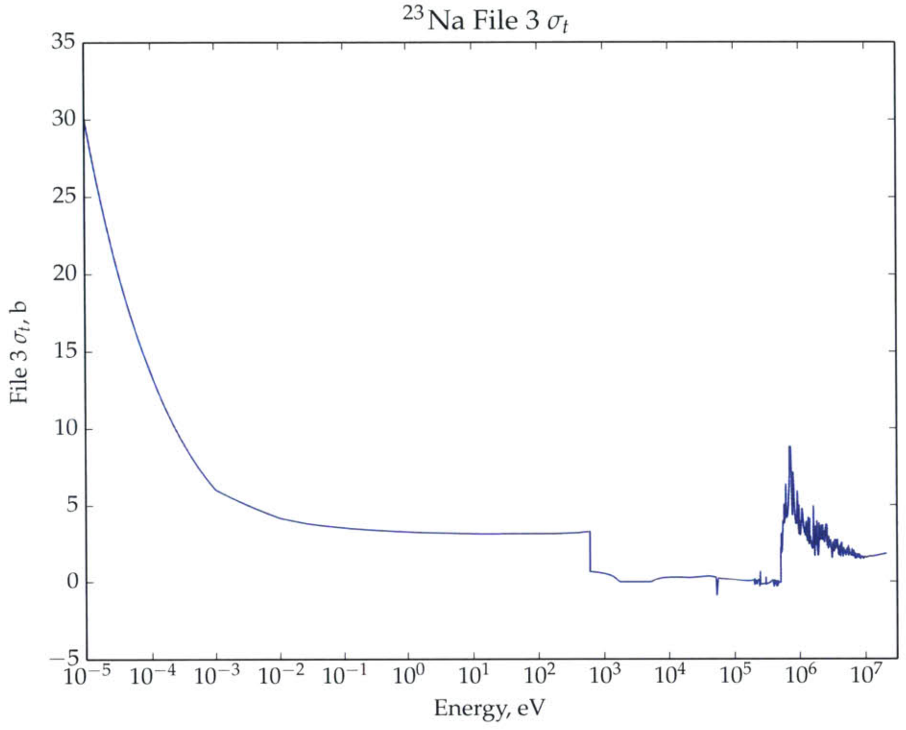

Figure 3-14: ^{23}Na File 3 for σ_t

In this particular case, there are two issues. First, the File 3 data is non-zero in the resonance region. Second, as portrayed in Figure 3-15, the difference between Doppler broadening and not is quite significant. For this particular isotope, the windowed multipole library generation algorithm has been slightly modified. Instead of spanning just the resolved resonance range, the windowed multipole data always spans from 10^{-5} eV to the end of the resolved resonance range. Whenever the resonance data is not applicable, no poles get introduced into the window. Then, whenever File 3 data is encountered, it is summed into the curve fitted portion of the data. Finally, all pointwise data beyond the end of the resolved resonance range is stored pointwise for use in the Monte Carlo simulation. Along with the Doppler broadening algorithm described in Section 3.1.2, this mostly mitigates the issues of File 3 data.

There are a few circumstances in which even this will fail, and that is when the data is particularly non-smooth. As a final example, Figure 3-16 presents a particularly egregious issue. ^{36}Ar has a very complex File 3, and a method to handle such data

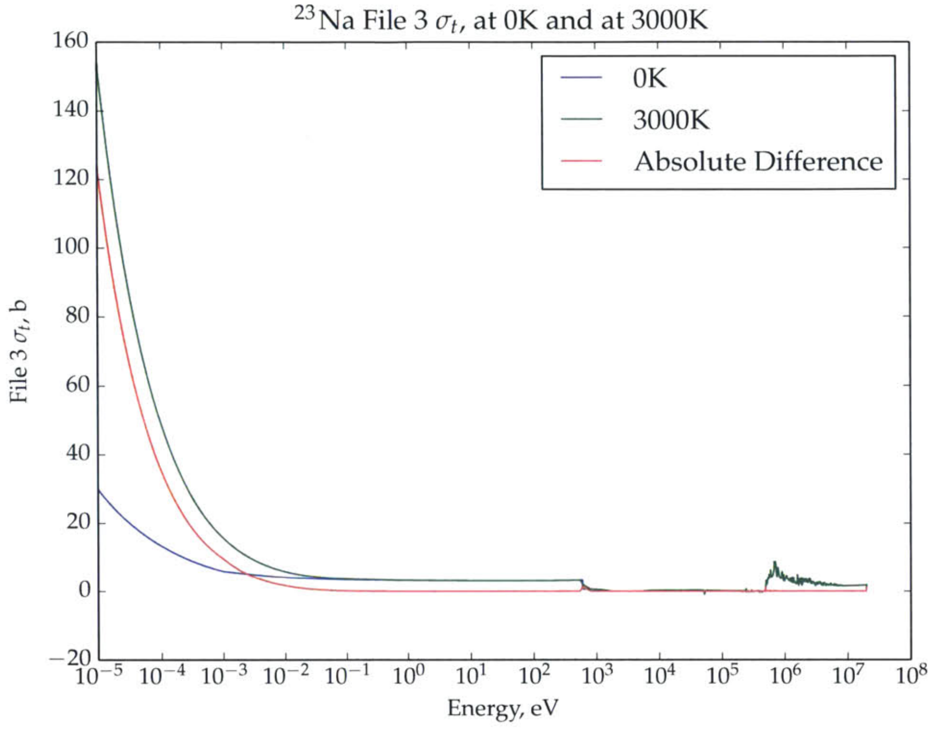

Figure 3-15: ^{23}Na File 3 for σ_t Doppler broadened

has yet to be developed. Several other isotopes exhibit this or similar problems. For minor issues, such as those of ^{56}Fe, the tolerance can be slightly loosened in proximity to the issue.

3.4 Summary of the Windowed Multipole Algorithm

The windowed multipole generation process is designed to mitigate the computational cost of the process of calculating cross sections. It does this by separating the energy into separate domains, and for each domain evaluating some poles exacly and the rest with Doppler broadenable curve fits.

In this chapter, the energy grid used to separate the energy into domains is selected. The equal-in-momentum bins were chosen due to their ability to minimize errors across the energy domain in a roughly equal way. Further, as the bins were equal along an axis, no binary searching would have to be performed.

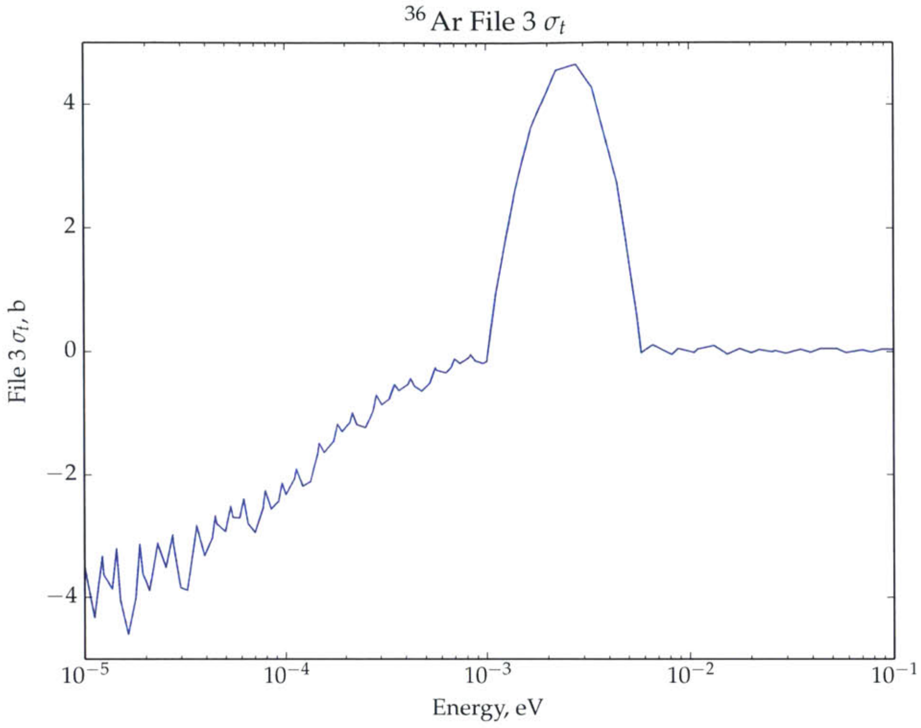

Figure 3-16: ^{36}Ar File 3 for σ_t

The next component decided upon was the form of the curve fit algorithm. The function $\sigma(E) = \sum_{n=0}^{N} a_n E^{n/2-1}$ was decided upon as it contained many favorable properties. It was able to exactly represent $1/E$, $1/v$, and a constant, all of which appear in some way with the data. It was also Doppler broadenable with a highly efficient recursive algorithm. Finally, the Laurent expansion at low energies indicates that this curve fit will converge when fitting a pole given an infinite number of terms.

The optimization algorithm and its components are then detailed. First it scans if a curve fit could suffice. If not, another algorithm selects the optimal number of poles to exactly evaluate for a given window using a bisection search routine. The maximum error is stochastically searched for using a differential evolution algorithm.

The Faddeeva function is then chosen. The input to the Faddeeva functions are analyzed to investigate how often certain portions of each algorithm are triggered. The ability to evaluate peaks of resonances are then compared. The "MIT W" function is selected as it has high accuracy and decent performance. It is also noted that some performance can be gained by switching algorithms, but as will be noted in Section 4.4.1, the overall performance is loosely dependent on the Faddeeva function anyways.

Finally, since the input data is not always clean, a method for handling the extra pointwise data some cross section libraries have is explained. Pointwise data, so long as it is mostly smooth, can be subsumed into the curve fit for the primary reactions. For the fast region and for secondary reactions, analysis notes that the error introduced is quite small.

Chapter 4

Integrated Testing

Now that the windowed multipole method is built up, a detailed analysis of its capabilities relative to other methods must be performed. Of interest would be the accuracy, the clock cycle cost per evaluation, the memory requirements, and the memory bandwidth requirements.

In order to run this evaluation, first a reference problem to simulate must be generated. The light water reactor benchmark used for this purpose is described in Section 4.1. Then, using the isotopes in that model, a library must be constructed. The structure and the constraints of the library are presented in Section 4.2.

Once those are set, two separate sets of simulations will be run. The first set is an uninstrumented large-scale run to assess relative accuracy and quality. The results of these runs will be in Section 4.3. Then, far more small scale instrumented tests will be run to compare the relative performance and memory usage of other Doppler broadening techniques relative to windowed multipole. These will be performed in Section 4.4.

4.1 BEAVRS Benchmark

The benchmark that was run is the Benchmark for Evaluation and Validation of Reactor Simulations, or BEAVRS for short [10]. This model is a pressurized light water reactor based as closely as possible on an operating design. A core layout

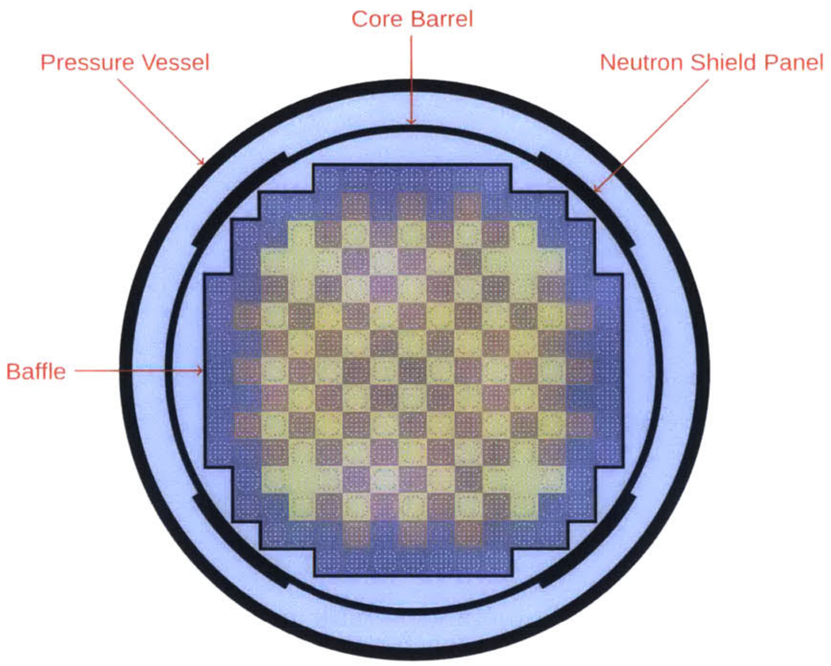

Figure 4-1: BEAVRS core diagram, image from [10]

diagram is shown in Figure 4-1

In the hot zero-power version of this core, there are 90 isotopes. The entire list is detailed in Table 4.1. As many of these as possible were converted into windowed multipole format before simulation.

4.2 The Library

The library was generated using the resonance information from the ENDF-B/VII.1 cross section library [3]. Due to the limitations of the basic multipole method as described in Section 2.2.5, some isotopes could not be processed into libraries as they exist only in pointwise form. Furthermore, some failed to be converted due to some peculiarity of their data. The ones not processed are listed in Table 4.2, with the reason for failure enumerated.

The isotopes in pointwise format have an obvious reason for failure, in that multipole only works on resonance data. The ones with "File 3" listed as a mode of failure failed because of the addition of overly complex pointwise data to the Reich-

Element	Isotope List
Hydrogen	1, 2
Helium	3, 4
Boron	10, 11
Carbon	Natural
Nitrogen	14, 15
Oxygen	16, 17
Aluminium	27
Silicon	28, 29, 30
Phosphorus	31
Sulfur	32, 33, 34, 36
Argon	36, 38, 40
Calcium	40, 42, 43, 44, 46, 48
Titanium	46, 47, 48, 49, 50
Vanadium	50, 51
Chromium	50, 52, 53, 54
Manganese	55
Iron	54, 56, 57, 58
Nickel	58, 60, 61, 62, 64
Copper	63, 65
Zirconium	90, 91, 92, 94, 96
Niobium	93
Molybdenum	92, 94, 95, 96, 97, 98, 100
Silver	107, 109
Cadmium	106, 108, 110, 111, 112, 113, 114, 116
Indium	113, 115
Tin	112, 114, 115, 116, 117, 118, 119, 120, 122, 124
Uranium	234, 235, 238

Table 4.1: Isotopes in the BEAVRS core

Moore/MLBW data. As explained in Section 3.3, smooth pointwise data can be curve fit and subsumed into the windowed multipole format. For the listed isotopes, the data was particularly non-smooth, and library generation was not successful.

As for those isotopes that were processed, the target accuracy was a 0.1% maximum relative error, with a few exceptions. If the difference between inexact and exact was less than 10^{-5}b, this discrepancy was ignored. All isotopes were allowed a 1% error between 10^{-5} and 10^{-4} eV, as the scattering cross section in this region is the difference between two very large numbers. This was not expected to impact results, as few neutrons reach such energies and it does not affect the total cross section. For

Isotope	Reason
H-1	Pointwise
H-2	Pointwise
He-3	Pointwise
He-4	Pointwise
B-10	Pointwise
B-11	Pointwise
Carbon (natural)	Pointwise
N-14	Pointwise
N-15	Pointwise
O-16	Pointwise
O-17	Pointwise
P-31	Pointwise
S-36	Pointwise
Ar-36	File 3
Ar-38	File 3
Ar-40	File 3
Ca-46	Pointwise
Nb-93	File 3

Table 4.2: Isotopes not in windowed multipole format

^{56}Fe, the errors in the absorption cross sections were ignored within 1.5 keV of 400.5, 450.6, 500.5, 550.5, 600.5, 650.5, 700.5, 750.5, and 800.5 keV. The reason for this is that there is a slight square wave in the File 3 data. The same is done for ^{63}Cu and ^{65}Cu at 55 keV and 60 keV, respectively, as there is a sharp discontinuity at both points. As these errors are in narrow bands at high energies and only in the relatively small aborption reaction, it is not expected that these will affect the results much either.

4.3 Accuracy Testing

The BEAVRS model and the WMP libraries were then loaded into OpenMC [19], which had been modified to support the libraries. Two major runs were performed. Both runs had the coolant at 600K, and everything else at 900K. The first run used WMP data wherever possible. The second one used MCNP6-sourced data. These were run on a single assembly, with 3.1% enriched fuel with no instrument tubes or burnable absorbers. To ensure high accuracy on the differential tallies, a large number of neutrons were run in the problem. The simulation parameters are listed in Table 4.3.

Parameter	Value
Inactive Batches	250
Active Batches	450
Neutrons Per Batch	20 million

Table 4.3: Accuracy run configuration

For each simulation, several components were tallied: the eigenvalue, the total absorption rate in ^{238}U and ^{235}U, the total fission rate in ^{235}U, and the flux. The tallies were performed per-pin, as well as over 16 energy bins. The energy bins are shown in Table 4.4.

Bin Index	Bottom of Bin	Top of Bin
1	10^{-5} eV	3×10^{-2} eV
2	3×10^{-2} eV	0.1 eV
3	0.1 eV	0.3 eV
4	0.3 eV	0.625 eV
5	0.625 eV	1 eV
6	1 eV	4 eV
7	4 eV	6 eV
8	6 eV	10 eV
9	10 eV	25 eV
10	25 eV	50 eV
11	50 eV	100 eV
12	100 eV	1 keV
13	1 keV	10 keV
14	10 keV	100 keV
15	100 keV	1 MeV
16	1 MeV	20 MeV

Table 4.4: Energy Bins Used For All Tallies

First of all, the eigenvalues are listed in Table 4.5. The eigenvalues are separated by around 8 pcm. While the difference exists and is well outside of statistics, the eigenvalue difference is quite small. More important is the difference on tallies, however.

Run	Eigenvalue
MCNP6 Data	1.2034302 ± 0.0000095
Multipole Data	1.2035119 ± 0.0000094

Table 4.5: Eigenvalues

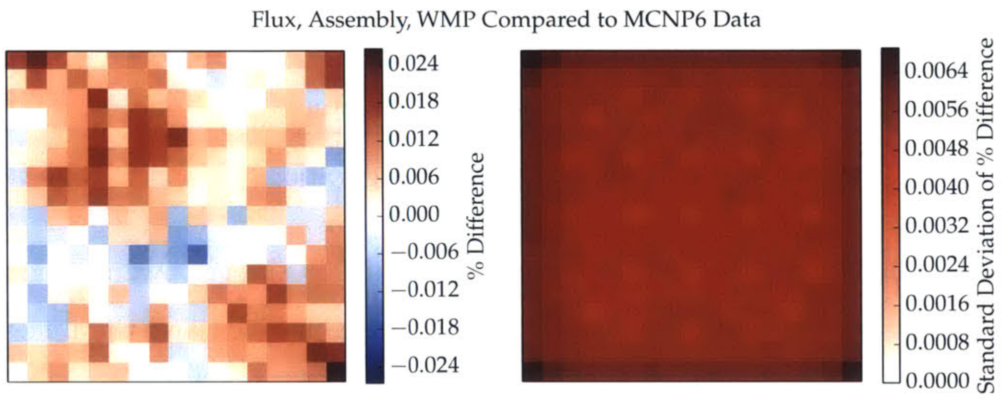

Figure 4-2: Relative difference, flux, by pin, WMP vs. MCNP6-sourced data

4.3.1 Flux

The first set of plots are for the flux for the problem. Starting first with the spatial graph in Figure 4-2, of the 289 pins, 58 are greater than two standard deviations above zero and 2 are greater than two standard deviations below zero. There is a weak spatial pattern, possibly indicating a slight neutron clustering bias [5].

In the case of the energy variable as shown in Figure 4-3, the reason for the discrepancy is slightly more apparent. For almost every bin below 1 keV, the flux is higher in the WMP case than in the MCNP6-sourced data case. This indicates that the absorption rate inside of the core is slightly lower in the WMP case in the resolved resonance regime. Despite these discrepancies, the worst bin is only off by 0.02%, which is well below the target accuracy of both libraries.

4.3.2 Fission

The fission rate for ^{235}U was tallied throughout the simulation as well. This spatial tally is shown in Figure 4-4. As compared to the flux error, the differences across the spectrum are more pronounced, as high as $0.06 \pm 0.02\%$.

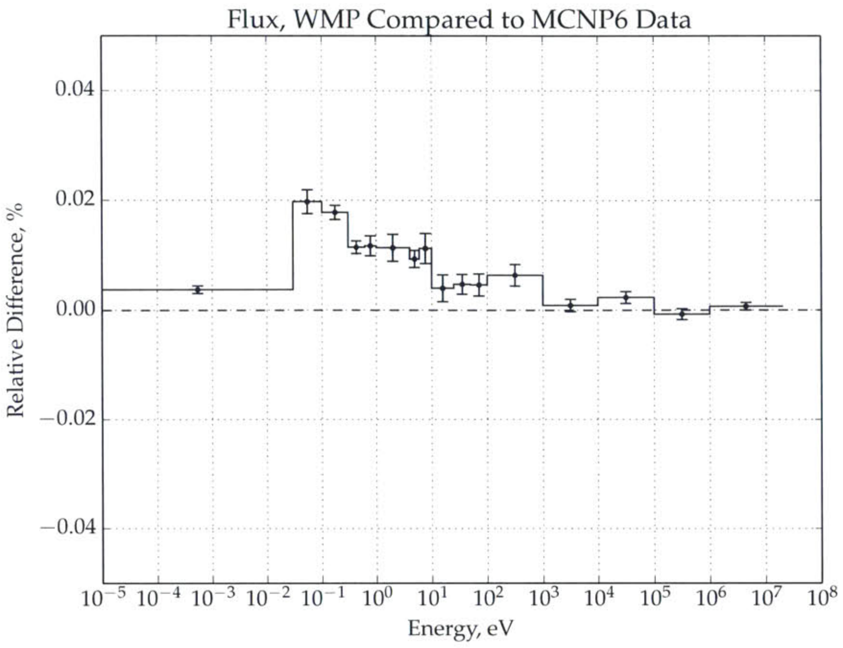

Figure 4-3: Relative difference, flux, over energy, WMP vs. MCNP6-sourced data

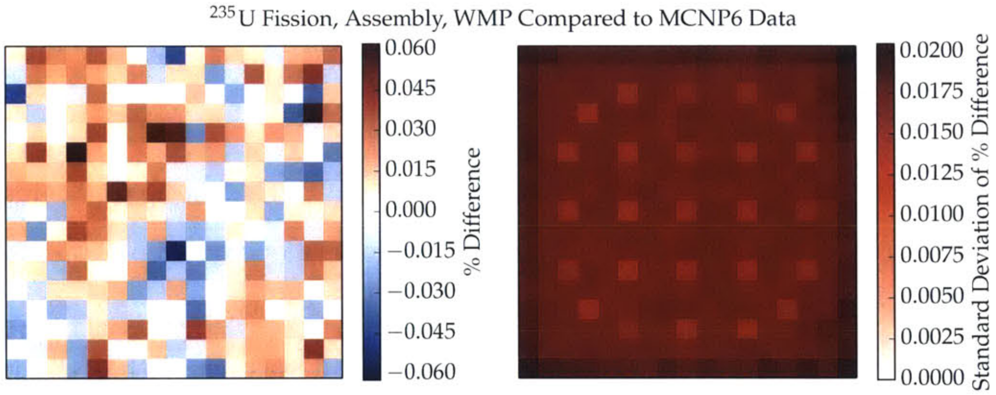

Figure 4-4: Relative difference, fission, ^{235}U, by pin, WMP vs. MCNP6-sourced data

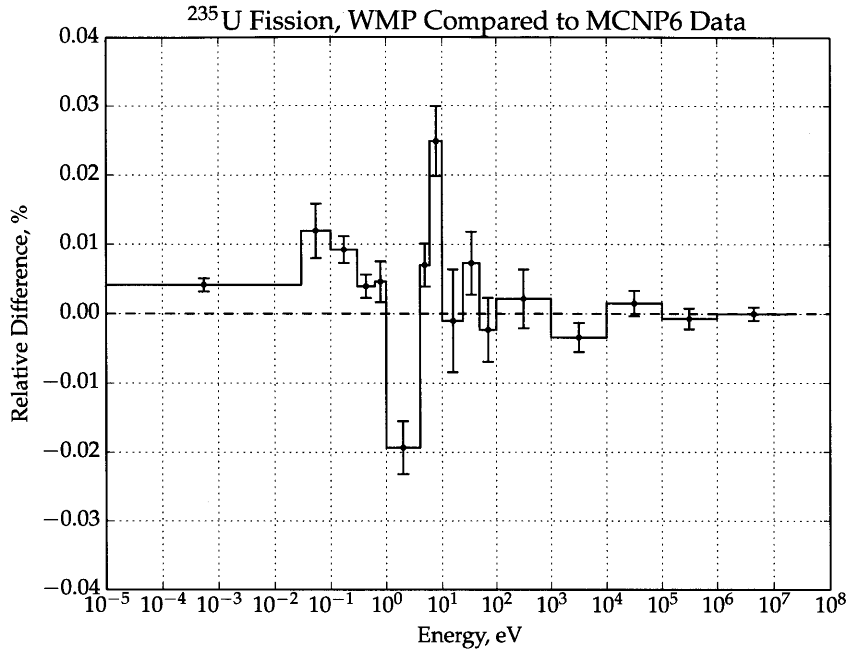

Figure 4-5: Relative difference, fission, ^{235}U, over energy, WMP vs. MCNP6-sourced data

The fission rate as a function of energy was also compared in Figure 4-5. Two interesting features appear. The first is that the 1 to 4 eV bin is nearly 0.02% reduced, whereas the 6 to 10 eV bin is 0.025% increased. These bins correspond to important resonances in ^{235}U (at 1.13 eV) and ^{235}U (at 6.67 eV) and this will be further analyzed in the absorption section.

4.3.3 Absorption

Finally, the absorption rate was tallied for both ^{235}U and ^{238}U. The spatial tallies are presented in Figure 4-6 for ^{235}U. The percent difference is a maximum of $0.06 \pm 0.02\%$. For ^{238}U, in Figure 4-7, the maximum difference is $0.08 \pm 0.03\%$.

More interesting is the energy component. Earlier, it was noted that the fission in ^{235}U was low in the 1 to 4 eV band, and high in the 6 to 10 eV band. When examining the absorption rate for ^{235}U in Figure 4-8, much the same appears, as the absorption rate is mostly composed of fission. However, when examining the ^{238}U graph in Figure 4-9, the entire range from 1 eV to 10 eV is depressed by 0.05%. This

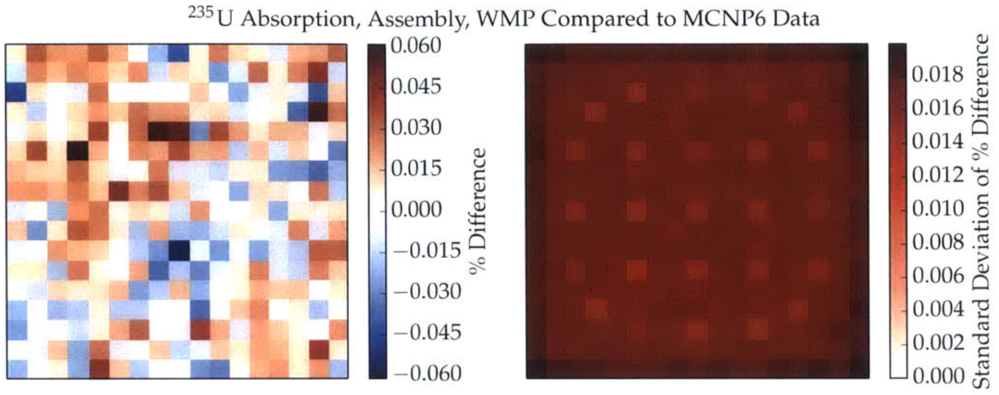

Figure 4-6: Relative difference, absorption, ^{235}U, by pin, WMP vs. MCNP6-sourced data

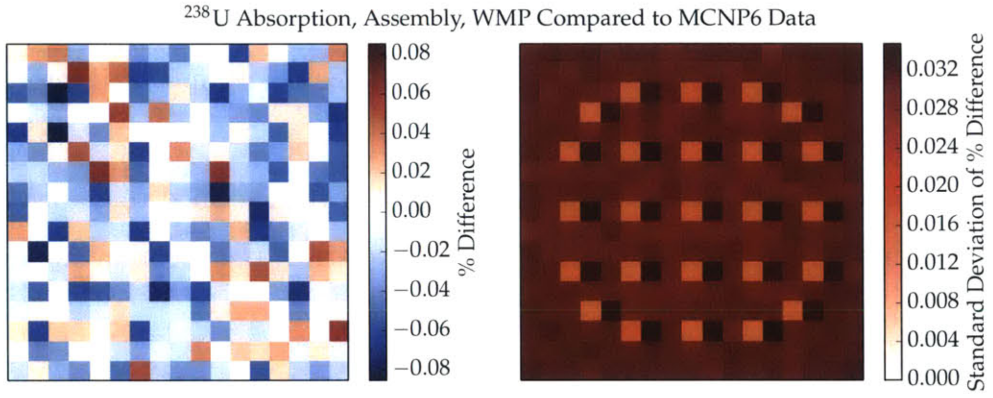

Figure 4-7: Relative difference, absorption, ^{238}U, by pin, WMP vs. MCNP6-sourced data

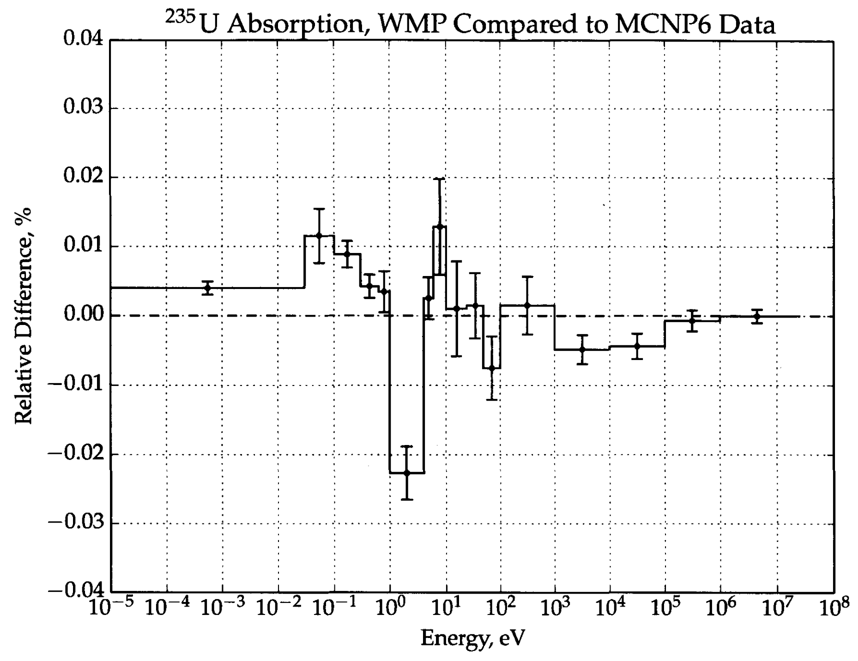

Figure 4-8: Relative difference, absorption, ^{235}U, over energy, WMP vs. MCNP6-sourced data

would increase the number of neutrons available in that region, and would increase the fission rate.

The fairly large discrepancy at high energies for ^{238}U is also interesting. For both simulations, the same unresolved resonance data was used and the flux stayed roughly constant for the bins of interest. As such, this may indicate a complicated interaction between other isotopes.

Overall, the accuracy on each tally was well within the specified tolerance of the library. With no tally exceeding 0.08±0.03% relative error, the accuracy appears to be well beyond what is needed for any practical calculation.

4.4 Performance Testing

OpenMC was then run inside of Callgrind, a module of Valgrind designed to measure the number of function evaluations as well as their clock cycle cost [24]. For these runs, the cache performance metrics were also enabled so that cache misses could be

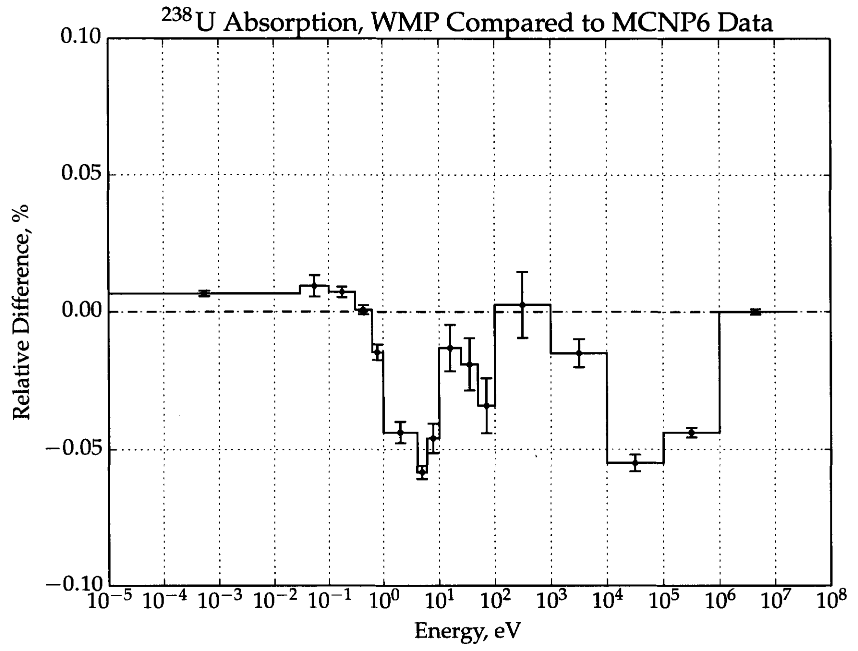

Figure 4-9: Relative difference, absorption, ^{238}U, over energy, WMP vs. MCNP6-sourced data

monitored. As running inside of Callgrind is significantly slower (typically by orders of magnitude) than on the hardware directly, the problem size was shrunk. The parameters used are listed in Table 4.6.

Parameter	Value
Inactive Batches	250
Active Batches	450
Neutrons Per Batch	100

Table 4.6: Performance run configuration

The full core was simulated, and the temperature of everything but the coolant was varied. Three different techniques were tested. The first, of course, is the windowed multipole method using the library described in Section 4.2. For this technique, the non-coolant temperatures were 600, 900, 1200, 1500, 1800, 2100, and 2500K. The second one is the target motion sampling algorithm described in Section 2.2.2[1]. The initial data to sample from was 600K, which was broadened to 900K. Finally, a single

[1]This implementation was performed by Tuomas Viitanen independently of this project.

pseudomaterials run was completed, in which a "750K" reactor was run with 50% of the atoms at 600K and 50% of the atoms at 900K. Although this would not give a very accurate simulation, the interest was only in measuring how the additional libraries impacted performance.

4.4.1 Clock Cycle Costs

The first number of interest is the estimated cross section calculation cycle cost per collision. This unit of measurement was chosen due to the fact that, due to the stochastic properties of the simulations, the collision count would vary. Furthermore, the per-material lookup time would also not be relevant, as target motion sampling performs several before a target velocity is sampled. The specific routine tracked is the `calculate_xs` routine, used to calculate σ_t, σ_a, σ_f, and σ_s. This does not include the `sample_reaction` routine, which identifies which reaction occurs at a collision, and requires the calculation of $\sigma_{n,2n}$, $\sigma_{n,n'}$, etc., which are still pointwise in the WMP format.

The clock cycle costs per collision are listed in Table 4.7 for the windowed multipole 900K run, the target motion sampling 900K run, and the pseudomaterials "750K" run. In it we find that TMS takes 2.8 times as long to run as WMP, and pseudomaterials double lookup took 1.2 times as long.

Run	Clock Cycles
WMP 900K	31587
TMS 900K	89440
Pseudomaterials "750K"	38415

Table 4.7: Clock cycles per collision, cross section lookup

The time cost of WMP was found to be slowly varying in temperature. This is shown in Figure 4-10. This is anticipated. As explained in Section 3.2, as temperature increases, the cycle cost for the Faddeeva function increases. In these runs, the Faddeeva function averaged only 15% of the total cost to evaluate a cross section, explaining the slight trend. Since the Faddeeva function is such a small fraction of

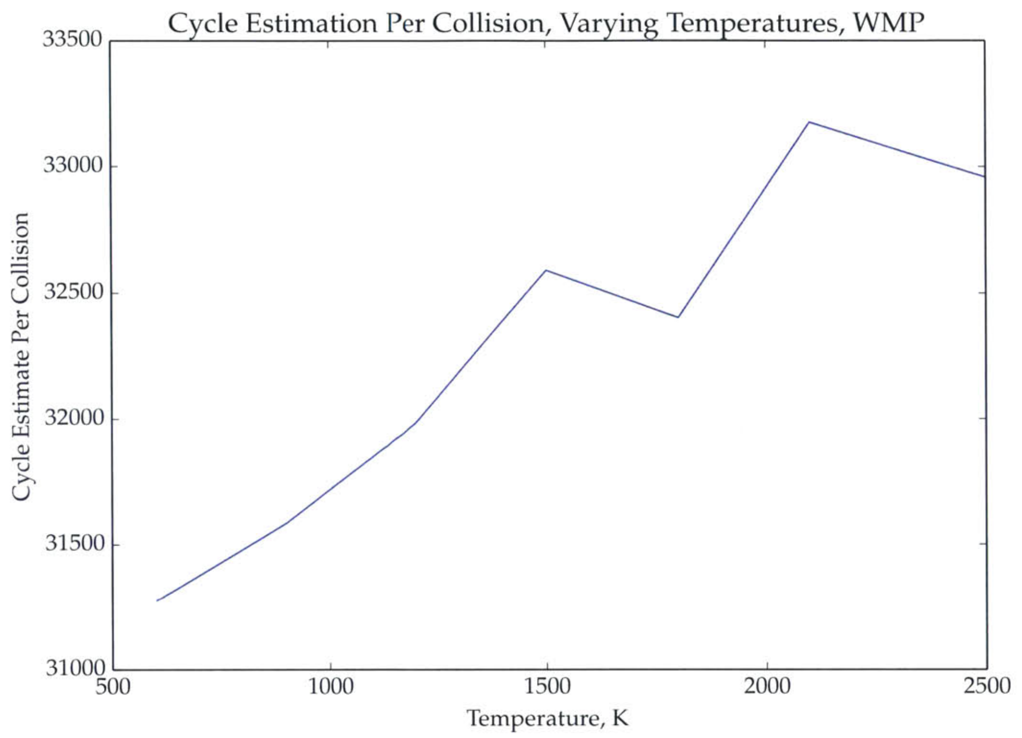

Figure 4-10: Clock cycle cost as a function of temperature

total cost, little effort was expended comparing the tradeoffs between a high accuracy but slow Faddeeva function and vice versa.

4.4.2 Cache Misses

The next set of numbers of interest are the cache misses per collision in the cross section calculator. There are two numbers here. The "L1 miss" is a counter of the number of times the CPU could not find data in the fastest memory close to the CPU and had to look elsewhere. The next, more important number is the "LL miss" rate, or the last-level cache miss rate. This counter indicates the number of times the CPU could not find data anywhere on chip and has to request data from main memory. Such a miss takes substantial time. The values for each of these counters are listed in Table 4.8.

Although there are fairly large differences in first-level cache misses, WMP's major advantage is in last-level misses. Using the WMP libraries effectively eliminated any

Run	L1 Misses	LL Misses
WMP 900K	554	0.072
TMS 900K	1394	5.83
Pseudomaterials “750K”	736	14.4

Table 4.8: Cache misses per collision, cross section lookup

last-level misses during the cross section lookup step. To put this in perspective for the whole problem, the total last-level misses for the entire eigenvalue calculation routine were also recorded. These are shown in Table 4.9. There was a factor of 33 fewer last-level misses compared to target motion sampling, and a factor of 80 fewer misses compared to pseudomaterials. The memory bandwidth benefits are readily apparent.

Run	Total LL Misses
WMP 900K	0.197
TMS 900K	6.50
Pseudomaterials “750K”	15.7

Table 4.9: Cache misses per collision, full problem

4.4.3 Memory Utilization

At the other end, it is useful to know how much memory is being utilized in each simulation. A similar Valgrind module, Massif, was used to investigate how much memory was being used for specific tasks. Each simulation was run until the initiation of the first eigenvalue calculation cycle. These runs are summarized in Table 4.10.

Overall, WMP has substantial benefits in total memory usage relative to the other two methods. There are a few points worth expanding on, however. First, WMP does not completely eliminate the ACE primary cross section data, as explained in Section 4.2. Furthermore, as explained in Section 3.3, the secondary distributions and the fast regimes use pointwise data. The memory cost of the fast regime is listed as “WMP Pointwise” to keep it separate from ACE data. Overall, the primary cross sections in WMP take only 8.6% of the total resident set size of the simulation.

Data	WMP	TMS	Pseudo
ACE Primary	3.5	61.8	121.2
ACE Secondary	39.1	39.1	77.2
ACE Secondary Energy Dist.	21.5	21.5	43.0
ACE URR	0.5	0.5	0.9
ACE Angular Dist.	16.9	16.9	33.6
$S(\alpha, \beta)$	9.3	9.3	9.3
TMS Majorant Tables	—	10.3	—
WMP Tables	5.3	—	—
WMP Pointwise	4.2	—	—
Other	9.3	9.1	12.5
Total	109.6	168.5	297.7

Table 4.10: Memory consumption by component, in MiB

It is also worth mentioning that the ACE secondary distributions and angular distributions are not broadened, so although the memory consumption for these terms are doubled in the pseudomaterials problem, the data loaded in from the files are essentially identical at both temperatures.

4.5 Summary of Integrated Testing

There are several important results from the integrated testing. The first one is of accuracy. As the MCNP6-sourced data itself was pointwise, approximations are made to make sure that the library is not too large. Thus, a discrepancy could mean that the library is more accurate, less accurate, or something orthogonal to both. However, the WMP discrepancies never exceeded 0.1% for any tally recorded. The flux was much closer at a maximum of 0.02%. The eigenvalue was discrepant by about 8 pcm. So, regardless of which direction the discrepancy points, the discrepancy is very small, indicating a high quality library.

The next important result is performance. Windowed multipole was faster, required less memory, and had fewer cache misses than target motion sampling and pseudomaterials on the sample problem. Furthermore, this is on a relatively middle-end consumer machine. On a supercomputer, where the memory bandwidth is far more constrained, WMP should significantly outperform the other techniques.

Chapter 5

Summary and Future Work

In this document, the windowed multipole Doppler broadening method was developed and tested on a real world benchmark. Each component of the technique was described, and the choices made were studied and justified. Then, an entire cross section library was generated and tested in an integrated benchmark.

Overall, the on-the-fly WMP algorithm has three key advantages. It is very quick. The WMP library outperformed pseudomaterials and target motion sampling in computation time by a factor of 1.2 and 2.8 respectively. It has very few last-level cache misses. WMP required a factor of 33 fewer last-level misses than target motion sampling, and a factor of 80 fewer misses than pseudomaterials. Finally, it requires less memory in general. On a 90 isotope simulation, WMP required 58.9 MiB less memory than TMS and 188.1 MiB less memory than pseudomaterials. However, the pseudomaterial simulation was only with two libraries. As the number of libraries loaded increases to increase accuracy and temperature range, the disparity will only increase.

These advantages are tempered by the inflexibility. WMP is only able to operate on resonance data, so if an isotope does not have published resonance data the conversion is impossible. A few isotopes have extra pointwise data that is sufficiently non-smooth as to prevent curve fitting. And finally, the underlying multipole method has yet to be developed to handle charged particle channels.

This then leads to possible future work. Although it would be advantageous if the cross section libraries were more consistent and used fewer non-physical structures (discontinuities, etc.) in their data, it would also be worthwhile to investigate the improvement of the flexibility of WMP to handle such data. Furthermore, published resonance data for isotopes such as ^{16}O would be advantageous. Finally, the addition of charged particle channels to multipole would be beneficial.

Bibliography

[1] S.M. Abrarov and Brendan M. Quine. On the Fourier expansion method for highly accurate computation of the Voigt/complex error function in a rapid algorithm. *arXiv preprint arXiv:1205.1768*, 2012.

[2] Keren Bergman, Shekhar Borkar, Dan Campbell, William Carlson, William Dally, Monty Denneau, Paul Franzon, William Harrod, Kerry Hill, Jon Hiller, et al. Exascale computing study: Technology challenges in achieving exascale systems. *Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep*, 15, 2008.

[3] M.B. Chadwick, M. Herman, P. Obložinský, Michael E. Dunn, Y. Danon, A.C. Kahler, Donald L. Smith, B. Pritychenko, Goran Arbanas, R. Arcilla, et al. ENDF/B-VII.1 nuclear data for science and technology: cross sections, covariances, fission product yields and decay data. *Nuclear Data Sheets*, 112(12):2887–2996, 2011.

[4] Dermott E. Cullen and Charles R. Weisbin. Exact Doppler broadening of tabulated cross sections. *Nuclear Science and Engineering*, 60(3), 1976.

[5] Eric Dumonteil, Fausto Malvagi, Andrea Zoia, Alain Mazzolo, Davide Artusio, Cyril Dieudonné, and Clélia De Mulatier. Particle clustering in Monte Carlo criticality simulations. *Annals of Nuclear Energy*, 63(0):612 – 618, 2014.

[6] Benoit Forget, Sheng Xu, and Kord Smith. Direct Doppler broadening in Monte Carlo simulations using the multipole representation. *Annals of Nuclear Energy*, 64:78–85, 2014.

[7] John T. Goorley, Michael R. James, Thomas E. Booth, Forrest B. Brown, Jeffrey S. Bull, Lawrence J. Cox, Joe W. Durkee Jr., Jay S. Elson, Michael Lorne Fensin, Robert A. Forster III, et al. Initial MCNP6 release overview-MCNP6 version 1.0. Technical report, Los Alamos National Laboratory (LANL), 2013.

[8] H. Henryson II, B. J. Toppel, and C. G. Stenberg. MC^{2}2: A code to calculate fast neutron spectra and multigroup cross sections. Technical Report Argonne-8144, Argonne National Laboratory, 1976.

[9] M. Herman and A. Trkov. ENDF-6 formats manual. *Brookhaven National Laboratory, Brookhaven National Laboratory, Upton, NY*, 2005.

[10] Nicholas Horelik, Bryan Herman, Benoit Forget, and Kord Smith. MIT BEAVRS: Benchmark for evaluation and validation of reactor simulations. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, Sun Valley, ID*, 2013.

[11] Richard N. Hwang. A rigorous pole representation of multilevel cross sections and its practical applications. *Nuclear Science and Engineering*, 96:192–209, 1987.

[12] Richard N. Hwang. An extension of the rigorous pole representation of cross sections for reactor applications. *Nuclear Science and Engineering*, 111(2):113–131, 1992.

[13] Steven G. Johnson. Faddeeva package - AbInitio. pages http://ab-initio.mit.edu/wiki/index.php/Faddeeva_Package, December 2013.

[14] Colin Josey, Pablo Ducru, Benoit Forget, and Kord Smith. Windowed multipole for cross section Doppler broadening. Submitted.

[15] Willis E. Lamb. Capture of neutrons by atoms in a crystal. *Physical Review*, 55:190–197, Jan 1939.

[16] Nancy M. Larson. Updated user's guide for SAMMY: Multilevel R-matrix fits to neutron data using Bayes' equation. *Oak Ridge National Lab., TN (United States)*, 1996.

[17] R.E. MacFarlane and D.W. Muir. The NJOY nuclear data processing system version 91. Technical report, Los Alamos National Lab., NM (United States). Funding organisation: USDOE, Washington, DC (United States), 1994.

[18] Dimitri G Naberejnev, Claude Mounier, and Richard Sanchez. The influence of crystalline binding on resonant absorption and reaction rates. *Nuclear Science and Engineering*, 131(2):222–229, 1999.

[19] Paul Romano and Benoit Forget. The OpenMC monte carlo particle transport code. *Annals of Nuclear Energy*, 51(0):274 – 281, 2013.

[20] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[21] T.H. Trumbull. Treatment of Nuclear Data for Transport Problems Containing Detailed Temperature Distributions. *Nuclear Technology*, 156:75–86, 2006.

[22] T. Viitanen and J. Leppänen. Explicit treatment of thermal motion in continuous-energy Monte Carlo tracking routines. *Nuclear Science and Engineering*, 171:165–173, 2012.

[23] T. Viitanen and J. Leppänen. Effect of the target motion sampling temperature treatment method on the statistics and performance. *Proceedings of SNA+MC 2013 Paris*, 2013.

[24] Josef Weidendorfer, Markus Kowarschik, and Carsten Trinitis. A tool suite for simulation based analysis of memory access behavior. In *Computational Science-ICCS 2004*, pages 440–447. Springer, 2004.

[25] Gokhan Yesilyurt, William R. Martin, and Forrest B. Brown. On-the-fly Doppler broadening for Monte Carlo codes. *Nuclear Science and Engineering*, 171(3):239–257, 2012.