Uncertainty Quantification of a Continuous Temperature Treatment of Thermal Neutron Scattering

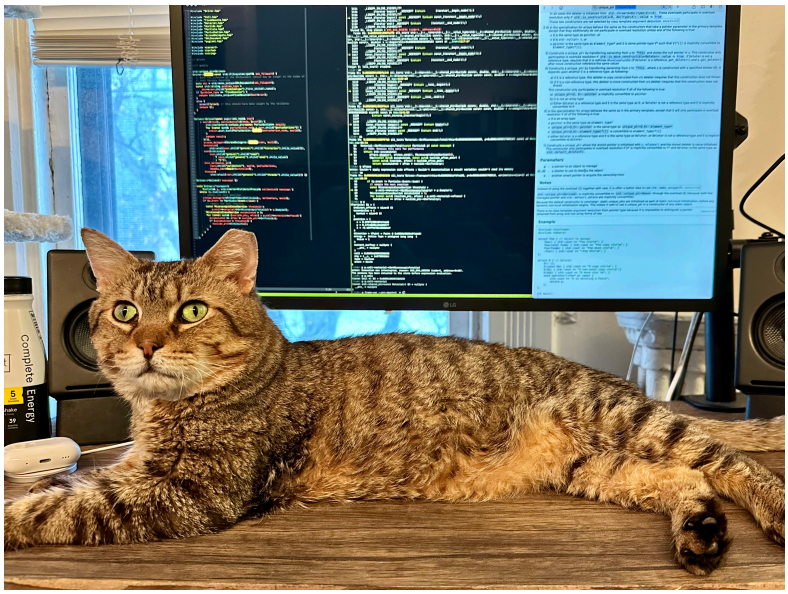
by

Aaron Gaspar Tumulak

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in the University of Michigan
2023

Doctoral Committee:

Associate Professor Brian C. Kiedrowski, Chair
Professor Alfred Hero
Dr. Andrew Holcomb, Oak Ridge National Laboratory
Professor Won Sik Yang



Good job catching that bug, Sweet Cheeks!!!

Aaron Gaspar Tumulak

atumulak@umich.edu

ORCID iD: 0000-0003-2557-429X

© Aaron Gaspar Tumulak 2023

Dedicated to my grandmother, Lola Rosa

Acknowledgments

There are many people who I would like to thank. Some of them are listed here.

I begin with my committee chair and academic advisor, Professor Brian Kiedrowski. Your caricatures of graduate school hell did not at all match the patience and compassion that I experienced under your guidance. My committee members, Professor Alfred Hero, Dr. Andrew Holcomb, and Professor Won Sik Yang provided invaluable feedback and were kind to accommodate me as unexpected issues arose over the course of this program. The mentorship from Dr. Dorothea Wiarda and Professor Vladimir Sobes made a significant portion of the work on covariance formats possible. Thanks Dr. Garnette Roberts for always looking out for my best interests and helping me navigate my way through this doctoral program.

Thanks to all the friends I made while here at Michigan. To my academic siblings, Evan Gonzalez, Emily Vu, Kyle Beyer, and Lincoln Johnston; your insights on everything from running, weightlifting, espresso, physics and all manner of offbeat topics were much appreciated. I am fortunate to have had you as my colleagues: "A+++ would work with again." Thanks Christopher Meert and Abbas Jinia for putting up with the sound of me grinding coffee beans by hand in the office.

Thanks Mr. Jonathan Ling for introducing me to physics and getting me started down this path. It seems like only yesterday I was solving free body diagrams.

I would be remiss if I did not thank *the lucky couch* at Roxanne and Hayden Walker's house. Well over half this document was written within under the Pinskys' roof never

more than a few feet from *the lucky couch*.

I am fortunate to have a large extended family with Titos, Titas, Kuyas and Ates whose support and encouragement have taken me this far. Thanks Erika and Victor for all the money I will save over the next few years by crashing at your place instead of a hotel. Thanks Ariel for never once asking me how many pages I have left on this dissertation. And, of course, thanks Mom for being the best mom ever.

Preface

This dissertation is based on work supported by the Department of Energy's Office of Nuclear Energy under NEUP Project 18-15008: Development of Thermal Inelastic Scattering Covariance Data Capabilities with Demonstration of Light Water Evaluation. This work was also supported by the Consortium for Monitoring, Technology, and Verification (MTV).

Table of Contents

Dedication ii

Acknowledgements iii

Preface v

List of Figures ix

List of Tables xii

Abstract xiii

1 Introduction & Motivation 1

1.1 Features Unique to this Work 2

1.2 Outline of Dissertation 3

2 Background 6

2.1 Radiation Transport Fundamentals 6

2.1.1 Phase Space 7

2.1.2 Angular and Scalar Flux 9

2.1.3 Neutron Transport Equation 12

2.2 Monte Carlo Methods 14

2.2.1 Sampling Procedure 15

2.2.2 Sampling from a Probability Density Function 19

2.3 Free Gas Scattering . . . 23
2.3.1 Thermally-Averaged Cross Sections . . . 24
2.4 Thermal Neutron Scattering Law . . . 32
2.4.1 Incoherent Approximation . . . 38
2.5 Evaluations of Thermal Neutron Scattering Laws . . . 39
2.5.1 ENDF/B-VIII.0 Data . . . 39
2.5.2 Probability Density of State (PDOS) Data . . . 40
2.6 Uncertainty Quantification . . . 40
2.6.1 Uncertainty Propagation using a Taylor Series Approximation . . . 41
2.6.2 Differential Operator Sampling . . . 43
2.6.3 Automatic Differentiation . . . 45
2.6.4 Total Monte Carlo . . . 49
2.7 Covariance Formats . . . 50
2.7.1 ENDF-6 Format . . . 50
2.7.2 Generalized Nuclear Database Structure (GNDS) . . . 51
2.7.3 Covariance of Multidimensional Data . . . 54

3 Methodology **59**

3.1 Data Structures . . . 59
3.1.1 Direct Sampling . . . 59
3.1.2 Continuous Temperature Dependence . . . 62
3.1.3 Total Scattering Cross Section . . . 63
3.1.4 In-Memory Format . . . 63
3.1.5 Proper Orthogonal Decomposition . . . 67
3.1.6 Monoticity Considerations . . . 70
3.2 Cumulative Distribution Function Interpolation . . . 71
3.2.1 Piecewise Linear Interpolation . . . 71
3.2.2 Piecewise Quadratic Interpolation . . . 76

3.2.3 Piecewise Cubic Interpolation . . . 84
3.3 Sampling σ_s, β, and α . . . 88
3.3.1 Indirect Effects . . . 89

4 Test Problems **94**
4.1 Broomstick Problem . . . 94
4.2 One-Dimensional Slab . . . 97
4.3 Performance Metrics . . . 108

5 Conclusions & Future Work **110**
5.1 Conclusions . . . 110
5.2 Future Work . . . 113
5.2.1 Short Term Work . . . 113
5.2.2 Long Term Future Work . . . 114

Glossaries **116**
Acronyms . . . 116
Symbols . . . 119

Bibliography **125**

Appendix **130**
A.1 GNDS Evaluation of H in H_2O . . . 131
A.2 Logarithmic Derivatives . . . 135
A.3 Quadratic CDF Sampling . . . 136
A.3.1 Sensitivities . . . 138
A.3.2 Probability Density Function . . . 139
A.4 ENDF/B-VIII.0 Evaluation Gridpoints for H in H_2O . . . 145

List of Figures

2.1 Position and direction coordinates 8

2.2 Direction of flight coordinates 9

2.3 Outgoing energy distributions on H in H_2O modeled in MCNP® using ENDF/B-VIII.0 Thermal Neutron Scattering Law (TNSL) data 32

2.4 250 realizations of Probability Density of State (PDOS) courtesy Chapman [25] 40

2.5 Forward Automatic Differentiation (AD) 48

2.6 Reverse AD, reverse pass 49

2.7 Nodes in entity-relationship diagram. Left: Node without attributes. Right: Node with attributes. 51

2.8 Crow's foot notation. Left: Exactly one child required. Center: Zero or one child. Right: One or more children. 52

2.9 A `gridded2d` container containing simplified $S(\alpha, \beta)$ data as an Extensible Markup Language (XML) file and as an entity-relationship diagram. 52

2.10 Proposed TNSL covariance Generalized Nuclear Database Structure (GNDS) hierarchy 54

2.11 Finalized TNSL covariance GNDS hierarchy 55

2.12 AMPX pipeline for parsing GNDS formatted data 57

2.13 AMPX `Covariance` class inheritance 57

3.1 TNSL data at selected temperatures as found in ENDF/B-VIII.0 65

3.2 Scree plots of Proper Orthogonal Decomposition (POD) applied to each dataset. The left axes are singular values normalized to the dominant singular value. The right axes are the normalized Frobenius norms of the residual after performing a rank–$\tilde{r}$ approximation. The thin vertical lines indicate the order selected in this work. The data for $G(\beta \mid E, T)$ and $\hat{H}(\alpha \mid \beta, T)$ is stored after applying transformations discussed in Subsection 3.1.6. 69
3.3 Comparison of reconstructed Cumulative Distribution Function (CDF) after performing a rank–37 approximation of $G(\beta \mid E, T)$ for $E = 2.328 \times 10^{-4}$ eV and $T = 283.6$ K. 72
3.4 Comparison of reconstructed CDF after performing a rank–6 approximation of $\hat{H}(\alpha \mid \beta, T)$ for $\beta = 4.137 \times 10^{-10}$ and $T = 473.6$ K. 72
3.5 Matrix partitioning scheme for $G(\beta \mid E, T)$ dataset 73
3.6 A histogram Probability Density Function (PDF) characterized by $\lambda = 0.35$, and $F = 0.6$ 75
3.7 Quadratic and linear interpolation of α and its CDF 81
3.8 Points directly used in interpolation scheme. The $(\times)$ indicates the sampled point which must be interpolated from neighboring points $(\bullet)$. Empty circles $(\circ)$ indicate points which are not used in the interpolation scheme. The columns correspond to E/T gridpoints and rows correspond to CDF gridpoints. 90
3.9 Interpolation scheme used to evaluate $\beta_{\tilde{k}}$. Order of evaluation goes $\bullet \rightarrow \circledast \rightarrow \times$. 91

4.1 The broomstick problem 94
4.2 PDFs for $E = 0.002\,53$ eV 98
4.3 PDFs for $E = 0.0253$ eV 99
4.4 PDFs for $E = 0.056$ eV 100
4.5 PDFs for $E = 0.56$ eV 101

4.6 1-D Slab Problem 102
4.7 Sensitivities $\partial J_2^- / \partial \lambda$ 105
4.8 Absolute value of sensitivities and relative errors for leftgoing partial current J^-. Sensitivities are sorted by magnitude. 106
4.9 Absolute value of sensitivities and relative errors for rightgoing partial current J^+. Sensitivities are sorted by magnitude. 107
4.10 Differential Operator Sampling (DOS)/AD sensitivity code profiled 109

List of Tables

3.1 Grid sizes and ranges of TNSL data for H in H_2O as found in ENDF/B-VIII.0 66

3.2 Gridpoints selected in evaluation of $\sigma_s(E,T)$. 66

3.3 Gridpoints selected in evaluation of $G(\beta \mid E,T)$. 67

3.4 Gridpoints selected in evaluation of $\hat{H}(\alpha \mid \beta,T)$ 67

3.5 Elements in uncompressed matrix N, elements in compressed matrix $N_{\tilde{r}}$, compression truncation order $\tilde{r}$ and corresponding normalized Frobeinus norm of the residual. 70

3.6 Gridsizes from adaptive coarsening on $G(\beta \mid E,T)$ dataset 73

3.7 Gridsizes from adaptive coarsening on $\hat{H}(\alpha \mid \beta,T)$ dataset 74

3.8 Adaptive coarsening summary 74

4.1 Broomstick H in H_2O TNSL Datasets by MCNP 95

4.2 Error between POD method and MCNP bin counts in E' and μ 96

4.3 Central difference estimtes of differential change in estimator dJ_2^+ using $d\lambda = 0.05\lambda$ 103

4.4 Uncertainty quantification 103

4.5 Contributions to nuclear data uncertainty for J_1^- estimator 108

Abstract

This dissertation develops methods to enable uncertainty quantification on Thermal Neutron Scattering Law (TNSL) data. A specification for covariance data in the Generalized Nuclear Database Structure (GNDS) format is produced which supports the storage of covariances for multidimensional arrays. An evaluation for H in H_2O in the new format is created and submitted to the National Nuclear Data Center (NNDC).

Memory constraints due to intractably large covariance matrices are addressed by introducing Proper Orthogonal Decomposition (POD) to produce reasonably accurate low-rank approximations of TNSL data. This work demonstrates a reduction in the amount of memory required to store a covariance matrix from 44.73 GB to 7.16 GB by compressing σ_s, β, and α datasets while keeping the (normalized) Frobenius norms of the error below 1.69×10^{-5}, 1.14×10^{-3}, and 3.67×10^{-3}, respectively.

A Monte Carlo code is developed to use temperature dependent TNSL data by reconstructing σ_s, β, and α from POD coefficients at runtime. Linear and quadratic interpolation schemes in Cumulative Distribution Functions (CDFs) are investigated before Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) is selected to satisfy the monotonic CDF constraints as well as provide continuous Probability Density Functions (PDFs) suitable for Differential Operator Sampling (DOS).

Two test problems are demonstrated. A "broomstick" problem is used to compare the outgoing energy and angle distributions using POD against MCNP®, a production-level Monte Carlo code, when using the same ENDF/B-VIII.0 TNSL dataset. The ℓ^2 error remained below 8.46×10^{-3} for energy bins and below 4.19×10^{-3} for cosine bins across a range of energy/temperature pairs.

A 1D slab problem with continuous temperature dependence is performed using TNSL data based on independent identically distributed energy frequency spectra for H in H_2O

obtained from Chapman. Sensitivities of energy-dependent leakage rates are computed using a hybrid of DOS and Automatic Differentiation (AD). When compared to Total Monte Carlo (TMC), the reported means for each estimator were within 1%, but the reported uncertainties due to uncertainties in nuclear data were up to 124% greater in extreme cases.

Chapter 1

Introduction & Motivation

This work proposes a method for quantifying the uncertainty in integrated quantities due to uncertainties in continuous temperature Thermal Neutron Scattering Law (TNSL) data. Thermal neutron scattering plays an important role in systems where a moderator consists of chemically bound targets. As the nuclear power industry shifts towards advanced reactors, characterizing the behavior of such moderators for which there is little to no operating experience will be crucial for cost-effective and safe designs. Monte Carlo simulations of exotic moderators such as FLiBe and metal hydrides like ZrH and YH will be crucial for properly justifying operating margins during the design of nuclear reactors. For thermal reactor designs, properly modeling thermal neutron scattering is essential in order to enable high-fidelity Monte Carlo simulations.

To date, TNSL data is provided by evaluators at discrete temperatures. For systems where there exist fine temperature variations, this may be inadequate for accurate simulations. Prior work has proposed a method for on-the-fly evaluation of temperature-dependent TNSL, but Uncertainty Quantification (UQ) on such methods has not been performed. This work introduces a new method for performing UQ on temperature-dependent TNSL data which accounts for correlations across different temperatures. A demonstration using H in H_2O is performed for several test problems.

1.1 Features Unique to this Work

This dissertation develops formats, methods, and diagnostics for performing UQ on estimators when using temperature-dependent TNSL data. Difficulties which had previously precluded such analysis were addressed by introducing several new features which are novel to this work:

- Prior to this work, there was no format for storing TNSL covariance data. TNSL data used in this work depends on three variables: energy transfer, momentum transfer, and temperature. A TNSL covariance format is developed within the Generalized Nuclear Database Structure (GNDS) format. This work also extends covariances to data of arbitrarily many dimensions.

- For the necessary number of temperatures, typical TNSL covariance data sizes can be orders of magnitude larger than what can be feasibly stored on currently available hardware. This work introduces Proper Orthogonal Decomposition (POD) as a method of compressing TNSL data to a reasonable size in a form that can be used in a direct sampling scheme. A reduction in the TNSL dataset sizes corresponds to a significant reduction in the corresponding covariance data. This work demonstrates an 84% reduction in covariance dataset size while maintaining acceptable accuracy, enabling UQ on commodity hardware.

- Previous work on introducing continuous temperature dependence to TNSL data used functional expansions for the outgoing energy and momentum transfer of the scattered neutron. This work adds continuous temperature dependence to the total scattering cross section. This is necessary for accurate modeling of thermal neutron scattering because uncertainties in TNSL data must be propagated to uncertainties in collision probabilities and interaction rates.

- Automatic Differentiation (AD) is used in a Monte Carlo transport code to perform

Sensitivity Analysis (SA). Historically, derivatives for computing special estimators in Differential Operator Sampling (DOS) have been hard-coded into source code. AD allows the evaluation of derivatives which may be difficult or outright intractable to express using closed-form expressions. This enables UQ regardless of how complicated the TNSL sampling schemes may be.

- Monte Carlo radiation transport codes have historically used linear and quadratic interpolation for sampling between datapoints. This work uses Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) to reduce the number of datapoints required to obtain sufficient fidelity. The cubic interpolation scheme used in this work is chosen so that Cumulative Distribution Functions (CDFs) are guaranteed to be monotonic. Moreover, the sampling scheme for PCHIP uses a Newton iteration scheme which is guaranteed to converge.

- This work introduces a diagnostic for quantifying the sources of uncertainty within a single TNSL dataset when using a first order Taylor series expansion. This enables one to quantify which nuclear data parameters contribute the most to the overall nuclear data uncertainty when using temperature dependent TNSL data. This diagnostic is demonstrated for a TNSL dataset for H in H_2O in a 1D slab problem.

1.2 Outline of Dissertation

The remainder of this dissertation consists of four chapters:

- Chapter 2 performs a literature review and discusses background information and concepts that this dissertation builds upon. The neutron transport equation is derived and the Monte Carlo approach for solving it is outlined. Sampling schemes which are used in this work are derived. A discussion of free gas scattering is followed by a discussion on the TNSL which accounts for chemical binding effects at

sufficiently low incident neutron energies. Sources of TNSL data used in this project are discussed. UQ using a first order Taylor expansion is derived and several SA techniques for computing derivatives used for indirect effects are outlined. This is followed by an extensive discussion on work that was performed to add TNSL covariance data to the GNDS format as well as work done to implement the format in AMPX.

- Chapter 3 proposes several new methods for storing and sampling from TNSL data. Techniques for sampling outgoing energy and angle directly from TNSL data as well as data structures required to support the sampling schemes are discussed. Consideration of memory requirements of TNSL covariance data is used to motivate the introduction of POD as a data compression technique. A comparison of compression ratios achieved with POD is used to show that a significant reduction in memory footprint makes uncertainty quantification feasible.

  An extensive discussion on interpolation schemes used when sampling from TNSL datasets goes over linear, quadratic, and cubic polynomials for representing CDFs. The merits of each method are discussed along with attempts to address any pitfalls that were encountered. A discussion justifies why PCHIP was ultimately used as the interpolation scheme for storing energy and angular distribution in this work

- Chapter 4 demonstrates the new methods introduced using two test problems. A broomstick problem is used to show that POD with PCHIP is able reproduce outgoing energy and angle distributions by MCNP®. A one-dimensional slab problem with continuous temperature dependence is used to demonstrate UQ and SA using a first order Taylor series approximation using DOS and AD. Results are compared against Total Monte Carlo (TMC).

- Chapter 5 summarizes work performed in the course of this dissertation and reflects on the results obtained. Future avenues of research are divided into short term

projects which can feasibly be completed in a few months and long term projects which merit future dissertation work.

Chapter 2

Background

Before the new features of this dissertation can be discussed, it is useful to review the current state of the art. A brief overview of neutron transport theory is provided with an emphasis on modeling thermal scattering physics. Moreover, some work related in fulfillment of this dissertation is discussed, namely the development of formats for storing Thermal Neutron Scattering Law (TNSL) covariances.

2.1 Radiation Transport Fundamentals

Radiation transport is concerned with how radiation interacts with matter. This section discusses the transport equation, an integro-differential equation which models the distribution of neutrons in seven dimensional phase space.

Several key assumptions are made about the behavior of neutrons

- Neutrons travel in straight lines. The effect of gravity and other weak long-range forces are considered to be negligible over most length scales considered.
- Neutrons are not directly affected by the presence of nearby neutrons. The density of neutrons is low enough that neutron-neutron interactions are neglected.

- Statistical fluctuations are negligible. The density of neutrons is high enough that observed quantities have small variance.
- Neutron energies are high enough that wavelike behavior can be neglected. This dissertation specifically addresses this assumption in the context of scattering but for most materials this assumption is valid.

2.1.1 Phase Space

Neutrons are modeled as point particles occupying three dimensions in space, two dimensions in direction, one dimension in energy, and one dimension in time. Figure 2.1 shows position as a vector

$$\mathbf{x} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}} \tag{2.1}$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ are basis vectors for a three-dimensional Cartesian space. A differential volume about $\mathbf{x}$ is defined

$$dV := dx\, dy\, dz\,, \tag{2.2}$$

the product of a cube with dimensions dx, dy, and dz. The direction of flight is given by

$$\hat{\mathbf{\Omega}} = \Omega_x\hat{\mathbf{i}} + \Omega_y\hat{\mathbf{j}} + \Omega_z\hat{\mathbf{k}} \tag{2.3}$$

where we restrict $\hat{\mathbf{\Omega}}$ so that it lies on the unit sphere, giving a neutron 4π steradians of directions to travel. Figure 2.2 depicts it as it is conventionally expressed in polar coordinates

$$\Omega_x = \cos\theta \qquad := \mu \tag{2.4a}$$

$$\Omega_y = \sin\theta\cos\gamma = \left(1-\mu^2\right)\cos\gamma \tag{2.4b}$$

$$\Omega_z = \sin\theta\sin\gamma = \left(1-\mu^2\right)\sin\gamma \tag{2.4c}$$

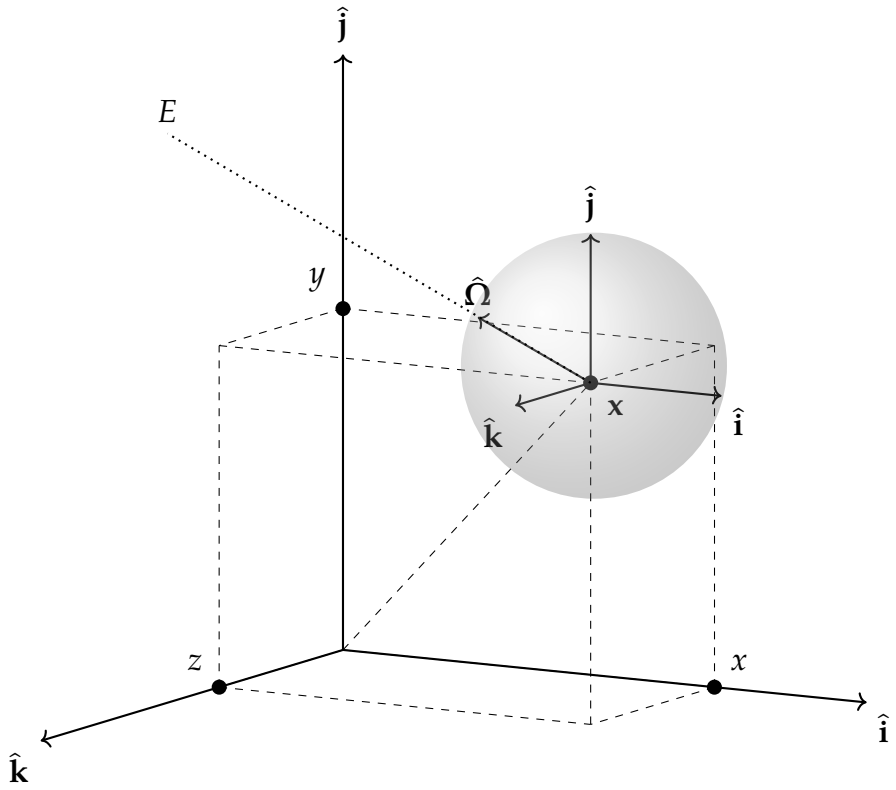


Figure 2.1: Position and direction coordinates

where θ is the cosine between $\hat{\mathbf{\Omega}}$ and polar axis $\hat{\mathbf{i}}$ and γ is an azimuthal rotation about $\hat{\mathbf{i}}$. A differential element of direction $d\Omega$ about $\hat{\mathbf{\Omega}}$ is equal to

$$d\Omega = d\mu\, d\gamma\,, \tag{2.5}$$

a differential element of the unit sphere. The energy E of a neutron is related to its velocity through the classical kinematic relation

$$E = \frac{1}{2} m v^2\,, \tag{2.6}$$

and is conventionally expressed in units of MeV.

For time-dependent problems, a time parameter t is necessary but this work focuses exclusively on time-independent problems so this will not be considered.

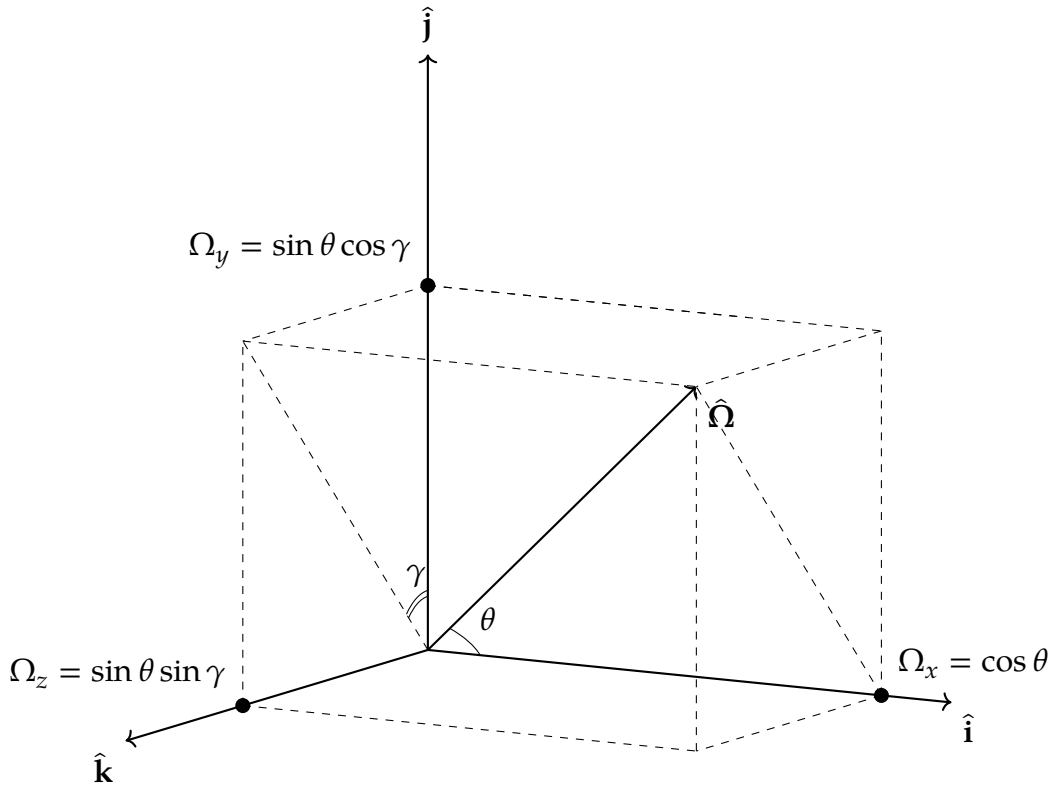


Figure 2.2: Direction of flight coordinates

2.1.2 Angular and Scalar Flux

We define the *neutron angular density* $n\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E\right)$ through the quantity

$$n\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E\right) dV\, d\Omega\, dE \tag{2.7}$$

as the expected number of neutrons to be found in dV about $\mathbf{x}$, $d\Omega$ about $\hat{\mathbf{\Omega}}$, and dE about E. We recall that we have assumed that the density of neutrons is high enough so that this quantity is not subject to statistical fluctuations. From the neutron angular density we derive another useful quantity, the *angular flux*

$$\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E\right) := v\left(E\right) n\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E\right) \tag{2.8}$$

where $v(E) = \sqrt{2E/m}$ is the (nonrelativistic) neutron speed. Two interpretations of the angular flux are useful: The volume interpretation and the surface interpretation.

The Volume Interpretation

We recall from eq. (2.7) that the angular flux gives the rate at which neutrons in dV about $\mathbf{x}$, $d\Omega$ about $\hat{\mathbf{\Omega}}$, dE about E produce path length. If a single neutron travels a differential segment of path length ds, the differential probability dp of interacting with a nuclide i and undergoing reaction j is

$$dp_{i,j} = N_i \sigma_{i,j}\, ds \tag{2.9}$$

where $N_i(\mathbf{x})$ is the *atomic number density* of the target nuclide i in units of atoms/cm^3 and $\sigma_{i,j}\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E\right)$ is the *microscopic cross section* for interacting with nuclide i and undergoing reaction j. The atomic number density is a bulk property of a material whereas the microscopic cross section is a property of an individual nuclide which must be measured experimentally and/or theoretically derived. The atomic number density and microscopic cross section are conventionally combined into the *macroscopic cross section*

$$\Sigma_{i,j} := N_i \sigma_{i,j}\,. \tag{2.10}$$

Dividing both sides of eq. (2.9) by a differential increment of time dt and using eq. (2.10), we get

$$r_{i,j}\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E\right) := \frac{dp_{i,j}}{dt} = \Sigma_{i,j}\frac{ds}{dt} = \Sigma_{i,j} v\,, \tag{2.11}$$

the expected rate with which a single neutron is expected to interact with nuclide i and undergo reaction j. Multiplying eq. (2.11) by the neutron angular density n we get

$$r_{i,j} n = \Sigma i, j v n = \Sigma i, j \psi\,, \tag{2.12}$$

the rate at which reaction j in nuclide i occurs due to *all* neutrons per unit volume, unit steradian of direction, and unit energy.

The rate given by eq. (2.12) is specific to neutrons with directions in $d\Omega$ about $\hat{\mathbf{\Omega}}$. We usually do not care about the direction of the incident neutrons which cause reactions so we integrate eq. (2.12) over all $\hat{\mathbf{\Omega}}$ and define

$$R_{i,j}(\mathbf{x}, E) := \int_{4\pi} \Sigma_{i,j} \psi \, d\Omega \tag{2.13}$$

as the rate at which reaction j in nuclide i occurs due to all neutrons per unit volume and unit energy. Assuming the macroscopic cross section has no directional dependence, we get

$$R_{i,j} = \Sigma_{i,j} \int_{4\pi} \psi \, d\Omega \tag{2.14}$$

so we can the *scalar neutron flux*

$$\phi := \int_{4\pi} \psi \, d\Omega \tag{2.15}$$

and express

$$R_{i,j}(\mathbf{x}, E) = \Sigma_{i,r}(\mathbf{x}, E) \, \phi(\mathbf{x}, E) \tag{2.16}$$

as the rate at which neutrons at $\mathbf{x}$ with energy E interact with nuclide i and undergo reaction j.

The Surface Interpretation

Given a differential surface area dA with unit normal vector $\hat{\mathbf{n}}$, the quantity $\left(\hat{\mathbf{\Omega}} \cdot \hat{\mathbf{n}}\right) \psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right)$ is the rate at which neutrons in $\hat{\mathbf{\Omega}}$ about $d\Omega$ and E about dE flow through dA at time t. For a surface ∂V which has outward-facing unit normal $\hat{\mathbf{n}}(\mathbf{x})$ for $\mathbf{x} \in \partial V$, the partial currents

$$J^{+}(t) := \int_0^{\infty} \int_{\partial V} \int_{\hat{\mathbf{\Omega}} \cdot \hat{\mathbf{n}} > 0} \left(\hat{\mathbf{\Omega}} \cdot \hat{\mathbf{n}}\right) \psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) d\Omega \, dA \, dE \tag{2.17a}$$

$$J^{-}(t) := \int_0^{\infty} \int_{\partial V} \int_{\hat{\mathbf{\Omega}} \cdot \hat{\mathbf{n}} < 0} \left(\hat{\mathbf{\Omega}} \cdot \hat{\mathbf{n}}\right) \psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) d\Omega \, dA \, dE \tag{2.17b}$$

give the rates at which neutrons cross surface ∂V in the outward and inward directions, respectively. The net current

$$J(t) = J^{+}(t) + J^{-}(t) \tag{2.18}$$

gives the *net* rate at which neutrons cross ∂V.

2.1.3 Neutron Transport Equation

In this subsection the time-dependent neutron transport equation is derived by considering mechanisms by which neutrons are gained and lost in a control volume of phase space dV about $\mathbf{x}$, $d\Omega$ about $\hat{\mathbf{\Omega}}$, E about dE between time t and $t + dt$.

Consider a volume V in space with boundary ∂V. Using the surface interpretation of the angular flux, we can express the net rate at which neutrons with energy E and direction-of-flight $\hat{\mathbf{\Omega}}$ leak out of V as

$$\int_{\partial V} \left(\hat{\mathbf{\Omega}} \cdot \hat{\mathbf{n}}\right) \psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) dA\,. \tag{2.19}$$

Invoking the divergence theorem, this is equivalent to

$$\int_{V} \left(\nabla \cdot \hat{\mathbf{\Omega}}\right) \psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) dV\,. \tag{2.20}$$

If we choose the volume V to be the differential element dV about $\mathbf{x}$, the integrand is constant under the integral and we have for the net leakage out of dV for neutrons with direction $\hat{\mathbf{\Omega}}$ about $d\Omega$ and energy E about dE:

$$\left(\nabla \cdot \hat{\mathbf{\Omega}}\right) \psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) dV\, d\Omega\, dE \tag{2.21}$$

Within dV, neutrons may collide and disappear from their current point in phase space. This can be due to processes like radiative capture, scattering, or fission. From the volume

interpretation of the angular flux, the total rate that path length is produced in dV by neutrons with energy E and direction $\hat{\mathbf{\Omega}}$ is

$$\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) dV \, d\Omega \, dE \, . \tag{2.22}$$

We must consider all nuclides i and all partial reactions j so we define

$$\Sigma_{\mathrm{t}}\left(\mathbf{x}, E\right) := \sum_{i} \sum_{j} \Sigma_{i,j} \tag{2.23}$$

to get the total rate at which neutrons are lost due to collisions as

$$\Sigma_{\mathrm{t}}\left(\mathbf{x}, E\right) \psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) dV \, d\Omega \, dE \, . \tag{2.24}$$

Neutrons can emerge into $\hat{\mathbf{\Omega}}$ about $d\Omega$ and E about dE from scattering. From the volume interpretation of the angular flux, the quantity

$$\Sigma_{\mathrm{s}}\left(E \rightarrow E', \hat{\mathbf{\Omega}} \cdot \hat{\mathbf{\Omega}}'\right) \psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) \tag{2.25}$$

gives the rate at which neutrons with energy E and direction $\hat{\mathbf{\Omega}}$ scatter and emerge with energy E' and direction $\hat{\mathbf{\Omega}}'$. For a differential volume dV, the rate that neutrons scatter into $\hat{\mathbf{\Omega}}$ about $d\Omega$ and E about dE is given by

$$dV \, d\Omega \, dE \int_{4\pi} \int_{0}^{\infty} \Sigma_{\mathrm{s}}\left(E' \rightarrow E, \hat{\mathbf{\Omega}}' \cdot \hat{\mathbf{\Omega}}\right) \psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}', E', t\right) dE' \, d\Omega' \, . \tag{2.26}$$

Neutrons can also emerge from fission, prompt fission, delayed fission, or fixed "internal" sources. Here the aggregate effect is collected into a term

$$S\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) dV \, d\Omega \, dE \, . \tag{2.27}$$

Once all the modes by which neutrons are lost and gained have been enumerated, the time derivative of the neutron angular density

$$\frac{\partial n}{\partial t}\, dV\, d\Omega\, dE = \frac{1}{v\,(E)}\frac{\partial}{\partial t}\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) dV\, d\Omega\, dE \tag{2.28}$$

can be set to the sum of the source terms less the loss terms. Combining eqs. (2.21), (2.24) and (2.26) to (2.28) and removing common factors, we have

$$\begin{aligned} &\frac{1}{v\,(E)}\frac{\partial}{\partial t}\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, Et\right) + \left(\nabla \cdot \hat{\mathbf{\Omega}}\right)\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) + \Sigma_{\mathrm{t}}\left(\mathbf{x}, E\right)\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) \\ &= \int_{4\pi}\int_{0}^{\infty} \Sigma_{\mathrm{s}}\left(E' \to E, \hat{\mathbf{\Omega}}' \cdot \hat{\mathbf{\Omega}}\right)\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}', E', t\right) dE'\, d\Omega' + S\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right), \end{aligned} \tag{2.29}$$

the *time-dependent neutron transport equation*. Along with boundary conditions

$$\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) = \psi^{\mathrm{b}}\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right), \quad \mathbf{x} \in \partial V, \quad \hat{\mathbf{\Omega}} \cdot \hat{\mathbf{n}} < 0, \quad 0 < E < \infty, \quad t > 0, \tag{2.30}$$

eqs. (2.29) and (2.30) constitute a seven-dimensional integro-differential equation which characterizes the distribution of neutrons in space and time. For steady state problems, the first term in eq. (2.29) is zero so it becomes

$$\begin{aligned} &\left(\nabla \cdot \hat{\mathbf{\Omega}}\right)\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) + \Sigma_{\mathrm{t}}\left(\mathbf{x}, E\right)\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right) \\ &= \int_{4\pi}\int_{0}^{\infty} \Sigma_{\mathrm{s}}\left(E' \to E, \hat{\mathbf{\Omega}}' \cdot \hat{\mathbf{\Omega}}\right)\psi\left(\mathbf{x}, \hat{\mathbf{\Omega}}', E', t\right) dE'\, d\Omega' + S\left(\mathbf{x}, \hat{\mathbf{\Omega}}, E, t\right), \end{aligned} \tag{2.31}$$

the *time independent neutron transport equation*.

2.2 Monte Carlo Methods

Without making suitable approximations, the neutron transport equation does not, in general, have a closed-form analytical solution. However, estimates can be obtained through Monte Carlo methods, a collection of techniques which date back to the 18$^{\mathrm{th}}$ century [1]

for solving problems using random number sequences to converge on quantities of interest. Modern Monte Carlo techniques for use on digital computers were pioneered by von Neumann, Ulam, and Metropolis [2] in Los Alamos in the 1940s to solve otherwise intractable problems.

Unlike deterministic methods, which rely on some appropriate discretization of the phase space to apply standard numerical methods, Monte Carlo methods perform random samples from some distribution in order to estimate quantities of interest. No discretization of phase space is required, practically limited only by floating point precision on computers. As a stochastic method, solutions from Monte Carlo methods are subject to statistical fluctuations and convergence issues may arise if an insufficient number of samples are performed. Various variance reductions techniques [3], [4] exist to reduce statistical uncertainties but they are not discussed here.

Monte Carlo methods allow one to estimate the expected value

$$\mathbb{E}[R] = \int_{\Omega} Rp(\omega)\, d\omega \tag{2.32}$$

of some random variable R (also known as a *response*) by drawing independent, identically distributed samples from a sample space Ω characterized by some Probability Density Function (PDF) $p(\omega)$. We recall from measure theory that random variables are neither random nor a variable. Rather the response R is simply a *scoring function*

$$R : \Omega \to \mathbb{R} \tag{2.33}$$

which maps from a sampled history to a real-valued score.

2.2.1 Sampling Procedure

We discuss the main steps of sampling a single history ω in a time-independent, fixed-source problem with no multiplication. For a system with no multiplication, the set of

possible histories Ω can be decomposed into $\Omega = X_0 \times ... \times X_{n'} \times ... \times X_N$ where $X_{n'}$ denotes the state of a particle after n' transitions. The number of transitions N is also a random number which corresponds to the number of transitions before a terminal state such as leakage or capture is reached. Sampling Ω yields a history $\omega = (x_0, ..., x_n)$ which consists of an initial state x_0 followed by n transitions until it reaches a terminal state. The PDF for sampling a given source state $p(x_0)$ is typically known or otherwise specified by the user. Subsequent particle states are obtained by the Markov property

$$p(x_{n'} \mid x_{n'-1}, ..., x_1, x_0) = p(x_{n'} \mid x_{n'-1}) \tag{2.34}$$

so the probability of sampling a history can be expressed

$$p(\omega) = p(x_0) \prod_{n'=1}^{n} p(x_{n'} \mid x_{n'-1}) \ . \tag{2.35}$$

Particle states can be characterized

$$x_{n'} = \begin{bmatrix} \mathbf{x}_{n'} \\ \hat{\mathbf{\Omega}}'_{n'} \\ E'_{n'} \\ r_{n'} \end{bmatrix} \tag{2.36}$$

where

$\mathbf{x}_{n'}$ = the particle's spatial position after the n'^{th} collision,

$\hat{\mathbf{\Omega}}'_{n'}$ = the particle's direction of flight after the n'^{th} collision,

$E'_{n'}$ = the particle's energy after the n'^{th} collision, and

$r_{n'}$ = the reaction that occured at the n'^{th} collision

so the transition probabilities given by eq. (2.34) can be further decomposed into

$$p\left(x_{n'} \mid x_{n'-1}\right) = \mathcal{T}_{n'} \mathcal{R}_{n'} \mathcal{E}_{n'} \mathcal{A}_{n'} \tag{2.37}$$

where the *transport kernels* are

$$\begin{aligned}
\mathcal{T}_{n'} &:= p\left(\mathbf{x}_{n'} \,\middle|\, \mathbf{x}_{n'-1}, \hat{\mathbf{\Omega}}'_{n'-1}, E'_{n'-1}, r_{n'-1}\right) \\
&= \text{ the probability of streaming uncollided from } \mathbf{x}_{n'-1} \text{ to } \mathbf{x}_{n'}, && \text{(2.38a)} \\
\mathcal{R}_{n'} &:= p\left(r_{n'} \,\middle|\, \mathbf{x}_{n'}, \hat{\mathbf{\Omega}}'_{n'-1}, E'_{n'-1}, r_{n'-1}\right) \\
&= \text{ the probability a collision at } \mathbf{x}_{n'} \text{ will result in reaction } r_{n'}, && \text{(2.38b)} \\
\mathcal{E}_{n'} &:= p\left(E'_{n'} \,\middle|\, \mathbf{x}_{n'}, \hat{\mathbf{\Omega}}'_{n'-1}, E'_{n'-1}, r_{n'}\right) \\
&= \text{ the probability reaction } r_{n'} \text{ will cause the particle to emerge with energy } E'_{n'}, \text{ and} && \text{(2.38c)} \\
\mathcal{A}_{n'} &:= p\left(\hat{\mathbf{\Omega}}'_{n'} \,\middle|\, \mathbf{x}_{n'}, \hat{\mathbf{\Omega}}'_{n'-1}, E'_{n'}, r_{n'}\right) \\
&= \text{ the probability a particle emerging with energy } E'_{n'} \text{ will do so with direction } \hat{\mathbf{\Omega}}'_{n'}. && \text{(2.38d)}
\end{aligned}$$

Although the transport kernels reprsenting marginalized conditional probabilities could have been marginalized in any order, the particular choice above corresponds to the order that each portion of phase space is actually sampled.

Transmission Kernel

Under surface tracking, the distance to the next collision is determined by sampling

$$l = -\frac{1}{\Sigma_{\mathrm{t}}} \log \xi \tag{2.39}$$

where Σ_{t} is the total cross section within the cell. This assumes that the total scattering cross section is uniform within the cell. When cross sections vary continuously within a

cell, delta tracking [5] must be used. A majorant cross section satisfying

$$\Sigma_{\mathrm{m}} > \Sigma_{\mathrm{t}}(\mathbf{x}) \; \forall \mathbf{x} \in V \tag{2.40}$$

within the cell volume V must be identified. The majorant must also be constant in V so that the distance to the next collision is determined by

$$l = -\frac{1}{\Sigma_{\mathrm{m}}} \log \xi \,. \tag{2.41}$$

Regardless of the tracking method, a particle can also leak out of the problem if it reaches a boundary. If this happens, the particle reaches a terminal state and the history ends.

Reaction Kernel

After sampling a collision distance using eq. (2.41), delta tracking samples whether or not the collision was real with probability

$$p_{\mathrm{real}} = \frac{\Sigma_{\mathrm{t}}}{\Sigma_{\mathrm{m}}} \,. \tag{2.42}$$

The true total cross section is obtained by summing over all nuclides i and partial reactions j to get

$$\Sigma_{\mathrm{t}} = \sum_i N_i \sum_j \Sigma_{i,j} \,. \tag{2.43}$$

If the collision was not true, or *virtual*, then it continues streaming as if it never collided. If the collision was true, a target nuclide must be sampled. The probability of colliding with nuclide i is

$$\frac{\Sigma_i}{\sum_{i'} \Sigma_{i'}} \,. \tag{2.44}$$

Next the reaction is sampled. The probability of undergoing reaction j in nuclide i is

$$\frac{\Sigma_{i,j}}{\sum_{j'} \Sigma_{i,j'}} \tag{2.45}$$

If the particle undergoes a reaction such as radiative capture, then its history ends.

Energy and Angle Kernels

If the particle undergoes a reaction which changes its energy and direction, such as scattering, then an outgoing energy and direction must be sampled. The details of scattering are discussed next in Sections 2.3 and 2.4. Once sampled, the control flow loops back and samples the next distance to collision.

2.2.2 Sampling from a Probability Density Function

In a Monte Carlo simulation, we often wish to sample values of a random variable X whose PDF $f_X(x)$ is known. However, most pseudorandom number generator libraries only return samples of U, a uniform random variable which has PDF

$$f_U(u) = \begin{cases} 1 & 0 \leq u < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.46}$$

The goal is to convert samples of U into samples of X. Two methods, *direct inversion* and *rejection sampling*, are discussed here. Direct inversion obtains an analytic expression for the Cumulative Distribution Function (CDF) which is then inverted so that u, a sampled value of U, can be directly transformed into x, a sampled value of X. Rejection sampling bounds a PDF with a suitable function and repeatedly samples the bounding function until a sampled value is accepted. Both methods are discussed in detail by Spanier and Gelbard [6].

Direct Inversion

If a random variable X has a known CDF which is invertible, then direct inversion can be used to sample a value of x. The CDF $F_X(x)$ of a random variable X with PDF $f_X(x)$ is defined by

$$F_X(x) := \int_{-\infty}^{x} f_X(x')\,dx' \tag{2.47}$$

which is equivalent to the probability of sampling x' less than or equal to x:

$$p(x' \leq x) = F_X(x)\ . \tag{2.48}$$

If $F_X(x)$ has an inverse $F_X^{-1}(u)$, then we can define a new random variable $F_X^{-1}(U)$ and see that its CDF satisfies

$$p\left(F_X^{-1}(u) \leq x\right) = p\left(u \leq F_X(x)\right)\ . \tag{2.49}$$

It was legitimate to apply $F_X(\cdot)$ to both sides of the inequality because

- $F_X(x)$ had an inverse which meant it was strictly monotonic, so
- all u for which $F_X^{-1}(u) \leq x$ also satisfy $u \leq F_X(x)$, and
- all u for which $F_X^{-1}(u) > x$ also satisfy $u > F_X(x)$.

Finally, because U is a uniform random variable in $[0,1)$, we can express the righthand side of eq. (2.49) as

$$p\left(F_X^{-1}(u) \leq x\right) = F_X(x) \tag{2.50}$$

and conclude that $F_X^{-1}(U)$ and X are identically distributed. Thus, the procedure for sampling from X would be

1. sample U to obtain $\xi \in [0,1)$, then
2. evaluate $F_X^{-1}(\xi)$ to obtain x.

Rejection Sampling

In many cases, the CDF of X or its inverse F_X^{-1} cannot be found analytically. In such cases, rejection sampling can be a suitable alternative. Rejection sampling requires two things:

1. Any random variable Y with PDF $f_Y(x)$ for which the CDF $F_Y(x)$ is known and invertible, and

2. some constant c such that

$$\frac{1}{c}\frac{f_X(x)}{f_Y(x)} \leq 1 \tag{2.51}$$

   or, in other words, $cf_Y(x)$ bounds $f_X(x)$.

The sampling procedure then proceeds by sampling Y to obtain y and then accepting with probability

$$p(A \mid y) = \frac{1}{c}\frac{f_X(y)}{f_Y(y)} \tag{2.52}$$

where

$$A = \text{The event that the sampled value of } y \text{ is accepted as the sampled value of } x \tag{2.53}$$

If y is not accepted, the procedure repeatedly resamples y until a value is accepted. Rejection sampling gets its name because samples are either accepted or rejected. Specifically, the expected fraction of samples which are accepted is

$$p(A) = \int p(A \mid y)\, p(y)\, dy = \int \frac{1}{c}\frac{f_X(y)}{f_Y(y)} f_Y(y)\, dy = \frac{1}{c}\int f_X(y)\, dy = \frac{1}{c} \tag{2.54}$$

where we call $p(A)$ the *rejection efficiency*. Ideally, $f_Y(x)$ is as similar as possible to $f_X(x)$ so that c can approach 1 and the rejection efficiency can approach 1.

We have merely stated the procedure for rejection sampling. Demonstrating that an accepted value of y follows $f_X(y)$ is a straightforward application of Bayes' Theorem to

get

$$\begin{aligned} p(y \mid A) &= \frac{p(A \mid y)\, p(y)}{p(A)} \\ &= \frac{\frac{1}{c}\frac{f_X(y)}{f_Y(y)} f_Y(y)}{\frac{1}{c}} \\ &= f_X(y) \end{aligned} \tag{2.55}$$

where eqs. (2.52) and (2.54) have been used.

A generalization of the preceding discussion can be obtained by changing the constant c to instead be a function $g(x)$ such that eq. (2.51) becomes

$$\frac{1}{g(x)} \frac{f_X(x)}{f_Y(x)} = 1 \tag{2.56}$$

or, after rearranging,

$$f_X(x) = f_Y(x)\, g(x) \tag{2.57}$$

where $f_X(x)$ has been decomposed into $f_Y(y)$, a PDF which can be sampled by direct inversion, and $g(x)$, a function of x. If $g(x) \geq 0$, we can choose our acceptance probability to be

$$p(A \mid y) = \frac{g(y)}{\max_x g(x)} \tag{2.58}$$

which gives a rejection efficiency of

$$p(A) = \int p(A \mid y)\, p(y)\, dy = \int \frac{g(y)}{\max_x g(x)} f_Y(y)\, dy = \frac{1}{\max_x g(x)} \tag{2.59}$$

where we made use of eq. (2.57) in the last step. Again using Bayes' Theorem, we can use

eqs. (2.57) to (2.59) to write

$$\begin{aligned} p(y \mid A) &= \frac{p(A \mid y)\, p(y)}{p(A)} \\ &= \frac{\frac{g(y)}{\max_x g(x)} f_Y(y)}{\frac{1}{\max_x g(x)}} \\ &= g(y) f_Y(y) \\ &= f_X(y) \ , \end{aligned} \tag{2.60}$$

confirming that the sampling scheme draws samples of X.

Historically, rejection sampling has been preferred over direct inversion because computing random numbers was cheaper than computing transcendental functions such as exp, log, and tan. However, modern algorithms and dedicated Floating Point Unit (FPU) instructions on contemporary processors have the two methods comparable in performance. The rejection efficiency and runtime performance of each method must be considered when deciding which sampling scheme to implement.

2.3 Free Gas Scattering

When the speed of target nuclei are comparable to the speed of an incident neutron, the relative motion between the two must be accounted for in order to accurately model the scattering kinematics. Free gas scattering, initially outlined by Coveyou, Bate, and Osborn [7], models the scattering medium as unbound nuclei with a Maxwell-Boltzmann distribution in velocity. Free gas scattering assumes that the kinematics can be modeled as elastic scattering in the Center of Mass (COM) frame between two unbound particles. When this is not the case thermal neutron scattering laws, discussed later in Section 2.4, must be applied.

2.3.1 Thermally-Averaged Cross Sections

Recall from the volume interpretation of the angular flux eq. (2.11) that the expected rate with which a single neutron interacts with a single nuclide i and undergoes reaction j is

$$r_{i,j} = v_{\mathrm{n}}(E)\,\sigma_{i,j}(E)\ . \tag{2.61}$$

From hereon we drop the i and j subscripts as the following derivation applies to all nuclides and reactions, respectively. When the speed of the target nucleus is comparable to the speed of the incident neutron, the velocity and energies in eq. (2.61) must be replaced by the relative velocity

$$v_{\mathrm{r}} := \|\mathbf{v}_{\mathrm{n}} - \mathbf{v}_{\mathrm{T}}\| \tag{2.62}$$

between the incident neutron $\mathbf{v}_{\mathrm{n}}$ and the target nucleus $\mathbf{v}_{\mathrm{T}}$. Inserting eq. (2.62) into eq. (2.61), we get

$$r = \|\mathbf{v}_{\mathrm{n}} - \mathbf{v}_{\mathrm{T}}\|\,\sigma\left(\|\mathbf{v}_{\mathrm{n}} - \mathbf{v}_{\mathrm{T}}\|\right)\ . \tag{2.63}$$

A temperature-dependent cross section $\sigma(E,T)$ which preserves the reaction rate can be obtained by computing the average of eq. (2.63) with

$$\sigma(E,T)\,v_{\mathrm{n}}(E) = \int \|\mathbf{v}_{\mathrm{n}} - \mathbf{v}_{\mathrm{T}}\|\,\sigma\left(\|\mathbf{v}_{\mathrm{n}} - \mathbf{v}_{\mathrm{T}}\|\right) M(\mathbf{v}_{\mathrm{T}},T)\,d\mathbf{v}_{\mathrm{T}} \tag{2.64}$$

where we have weighted by the Maxwell-Boltzmann distribution

$$M(\mathbf{v}_{\mathrm{T}},T) = \left(\frac{Am_{\mathrm{n}}}{2\pi k_{\mathrm{B}}T}\right)^{\frac{3}{2}} \exp\left(-\frac{Am_{\mathrm{n}}v_{\mathrm{T}}^2}{2k_{\mathrm{B}}T}\right) \tag{2.65}$$

which gives the probability of finding a target nucleus with velocity $\mathbf{v}_{\mathrm{T}}$ when A is the atomic weight ratio of the target nucleus, m_{n} is the mass of a neutron, T is the temperature of the gas, and $v_{\mathrm{T}} := \|\mathbf{v}_{\mathrm{T}}\|$ is the speed of the target nuclei. The temperature-dependent

microscopic cross section is thus

$$\sigma\left(E,T\right)=\frac{1}{v_{\mathrm{n}}\left(E\right)}\int\left\|\mathbf{v}_{\mathrm{n}}-\mathbf{v}_{\mathrm{T}}\right\|\sigma\left(\left\|\mathbf{v}_{\mathrm{n}}-\mathbf{v}_{\mathrm{T}}\right\|\right)M\left(\mathbf{v}_{\mathrm{T}},T\right)d\mathbf{v}_{\mathrm{T}} \tag{2.66}$$

where the cross section inside the integral is taken to be at 0 K. The scattering cross section is typically assumed to be a constant

$$\sigma_{\mathrm{fr}}\coloneqq\sigma_{\mathrm{s}}\left(\left\|\mathbf{v}_{\mathrm{n}}-\mathbf{v}_{\mathrm{T}}\right\|\right) \tag{2.67}$$

when away from resonances so eq. (2.66) becomes

$$\sigma_{\mathrm{s}}\left(E,T\right)=\frac{\sigma_{\mathrm{fr}}}{v_{\mathrm{n}}\left(E\right)}\int\left\|\mathbf{v}_{\mathrm{n}}-\mathbf{v}_{\mathrm{T}}\right\|M\left(\mathbf{v}_{\mathrm{T}},T\right)d\mathbf{v}_{\mathrm{T}}\,. \tag{2.68}$$

In polar coordinates, this is

$$\sigma_{\mathrm{s}}\left(E,T\right)=\frac{\sigma_{\mathrm{fr}}}{v_{\mathrm{n}}\left(E\right)}\int_{0}^{2\pi}\int_{-1}^{+1}\int_{0}^{\infty}\left\|\mathbf{v}_{\mathrm{n}}-\mathbf{v}_{\mathrm{T}}\right\|M\left(\mathbf{v}_{\mathrm{T}},T\right){v_{\mathrm{T}}}^{2}\,dv_{\mathrm{T}}\,d\mu\,d\gamma \tag{2.69}$$

where μ is the polar coordinate of $\mathbf{v}_{\mathrm{T}}$ with respect to $\mathbf{v}_{\mathrm{n}}$ and γ is the azimuthal direction about $\mathbf{v}_{\mathrm{n}}$. Using the law of cosines, the relative speed can be expressed

$$\left\|\mathbf{v}_{\mathrm{n}}-\mathbf{v}_{\mathrm{T}}\right\|=\sqrt{{v_{\mathrm{n}}}^{2}+{v_{\mathrm{T}}}^{2}-2v_{\mathrm{n}}v_{\mathrm{T}}\mu}\,. \tag{2.70}$$

Combining eqs. (2.65), (2.69) and (2.70), we can express

$$\sigma_{\mathrm{s}}\left(E,T\right)=\frac{\sigma_{\mathrm{fr}}}{v_{\mathrm{n}}\left(E\right)}\int_{0}^{2\pi}\int_{-1}^{+1}\int_{0}^{\infty}\sqrt{v_{\mathrm{n}}^{2}+v_{\mathrm{T}}^{2}-2v_{\mathrm{n}}v_{\mathrm{T}}\mu}\left(\frac{Am_{\mathrm{n}}}{2\pi k_{\mathrm{B}}T}\right)^{\frac{3}{2}}\exp\left(-\frac{Am_{\mathrm{n}}v_{\mathrm{T}}^{2}}{2k_{\mathrm{B}}T}\right)v_{\mathrm{T}}^{2}\,dv_{\mathrm{T}}\,d\mu\,d\gamma \tag{2.71}$$

which can be integrated and then converted using a change of variables from velocity to energy to get

$$\sigma_{\mathrm{s}}(E,T) = \sigma_{\mathrm{fr}}\left[\left(1+\frac{k_{\mathrm{B}}T}{2AE}\right)\operatorname{erf}\left(\sqrt{\frac{AE}{k_{\mathrm{B}}T}}\right)+\sqrt{\frac{k_{\mathrm{B}}T}{\pi AE}}\exp\left(-\frac{AE}{k_{\mathrm{B}}T}\right)\right] \tag{2.72}$$

where the *error function* is defined by the integral

$$\operatorname{erf}(x) \coloneqq \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt\,. \tag{2.73}$$

The term in square brackets of eq. (2.72) represents an adjustment factor which must be applied to a zero-temperature cross section σ_{fr} during transport in order to correctly account for the effect of thermal motion on the scattering cross section. We consider two limiting behaviors by defining

$$a \coloneqq \frac{AE}{k_{\mathrm{B}}T} \tag{2.74}$$

to express eq. (2.72) as

$$\sigma_{\mathrm{s}}(E,T) = \sigma_{\mathrm{fr}}\left[\left(1+\frac{1}{2a}\right)\operatorname{erf}(\sqrt{a})+\sqrt{\frac{1}{\pi a}}\exp(-a)\right]. \tag{2.75}$$

At low energies, we have $a \to 0$ so we can approximate eq. (2.75) by a power series about $a = 0$ to get

$$\sigma_{\mathrm{s}}(E,T) = \sigma_{\mathrm{fr}}\left[\frac{1}{\sqrt{2\pi a}}+O\left(a^{\frac{1}{2}}\right)\right] \tag{2.76}$$

Dropping higher order terms and writing in terms of velocity, we get

$$\sigma_{\mathrm{s}}(E,T) = \frac{\sigma_{\mathrm{fr}}}{v_{\mathrm{n}}}\left[\sqrt{\frac{k_{\mathrm{B}}T}{\pi A m_{\mathrm{n}}}}\right], \quad E \ll \frac{k_{\mathrm{B}}T}{A} \tag{2.77}$$

so the scatter cross section behaves as $1/v_{\mathrm{n}}$ at low energies. At high energies we have $a \to \infty$ so the error function converges to one and the first term in eq. (2.75) becomes

one, the second term becomes zero, and the scattering cross section becomes temperature independent:

$$\sigma_s(E,T) = \sigma_{fr}, \quad E \gg \frac{k_B T}{A}. \tag{2.78}$$

In practice, codes such as MCNP® set a cutoff at $E = 500 k_B T/A$ at which the adjustment is not performed since the error introduced is less than 1% at energies above the cutoff and the error function can be relatively expensive to compute.

Sampling Energy and Direction

We have seen that the free gas scattering treatment introduces a temperature dependence to the scattering cross section. The free gas scattering treatment also affects the outgoing energy and direction of scattered neutrons. A sampling scheme for Monte Carlo simulations, based on Gelbard [8] and outlined by OpenMC [9] documentation, is briefly summarized here.

At low energies, the quantum mechanical treatment of scattering is accurate to the first term in the partial wave expansion. The resulting *S-wave scattering* is isotropic in the center of mass frame so the problem of modeling the scattering kinematics can be broken down into distinct steps:

1. Sample a temperature-dependent target velocity $\mathbf{v}_T$ in the lab frame
2. Convert neutron velocity to the COM frame where scattering is isotropic
3. Convert the outgoing neutron velocity in the COM frame $\mathbf{v}'_{n,CM}$ to an outgoing neutron velocity in the lab frame $\mathbf{v}'_n$

For the first step it is tempting to simply sample $\mathbf{v}_T$ from the Maxwell-Boltzmann distribution given by eq. (2.65) but this will not necessarily preserve the reaction rate given by the integrand of eq. (2.66). The correct distribution can be obtained by normalizing the integrand by

$$C := \int \|\mathbf{v}_n - \mathbf{v}_T\| \, \sigma_s(\|\mathbf{v}_n - \mathbf{v}_T\|) \, M(\mathbf{v}_T, T) \, d\mathbf{v}_T \tag{2.79}$$

to get

$$p\left(\mathbf{v}_{\mathrm{T}}\right)=\frac{1}{C}\left\|\mathbf{v}_{\mathrm{n}}-\mathbf{v}_{\mathrm{T}}\right\| \sigma_{\mathrm{s}}\left(\left\|\mathbf{v}_{\mathrm{n}}-\mathbf{v}_{\mathrm{T}}\right\|\right) M\left(\mathbf{v}_{\mathrm{T}}, T\right) , \tag{2.80}$$

a reaction rate preserving PDF. To sample from eq. (2.80), we need to express it in a functional form that is amenable to some sampling scheme. The $\left\|\mathbf{v}_{\mathrm{n}}-\mathbf{v}_{\mathrm{T}}\right\|$ term can be expressed using eq. (2.70). For the $\sigma\left(\left\|\mathbf{v}_{\mathrm{n}}-\mathbf{v}_{\mathrm{T}}\right\|\right)$ term we can again assume a constant cross section given by eq. (2.67). For the $M\left(\mathbf{v}_{\mathrm{T}}, T\right)$ term we can use eq. (2.65) and note that the Maxwell-Boltzmann distribution has no directional dependence so we can change to polar coordinates and integrate over polar and azimuthal coordinates to get

$$M\left(v_{\mathrm{T}}, T\right)=\left(\frac{2}{\pi}\right)^{\frac{1}{2}}\left(\frac{A m_{\mathrm{n}}}{k_{\mathrm{B}} T}\right)^{\frac{3}{2}} v_{\mathrm{T}}^{2} \exp\left(-\frac{A m_{\mathrm{n}} v_{\mathrm{T}}^{2}}{2 k_{\mathrm{B}} T}\right) , \tag{2.81}$$

or, in terms of the most probable velocity of a Maxwell-Boltzmann distribution

$$v_{\mathrm{p}} \coloneqq \sqrt{\frac{2 k_{\mathrm{B}} T}{A m_{\mathrm{n}}}} , \tag{2.82}$$

get

$$M\left(v_{\mathrm{T}}, T\right)=\frac{4}{\sqrt{\pi}} v_{\mathrm{p}}^{-3} v_{\mathrm{T}}^{2} \exp\left(-v_{\mathrm{p}}^{-2} v_{\mathrm{T}}^{2}\right) . \tag{2.83}$$

Combining eqs. (2.70) and (2.83), we can express eq. (2.80) as

$$p\left(v_{\mathrm{T}}\right)=\frac{\sigma_{\mathrm{fr}}}{C} \sqrt{v_{\mathrm{n}}^{2}+v_{\mathrm{T}}^{2}-2 v_{\mathrm{n}} v_{\mathrm{T}} \mu} \frac{4}{\sqrt{\pi}} v_{\mathrm{p}}^{-3} v_{\mathrm{T}}^{2} \exp\left(-v_{\mathrm{p}}^{-2} v_{\mathrm{T}}^{2}\right) . \tag{2.84}$$

Unfortunately, this PDF is not amenable to direct inversion sampling because its CDF does not have a closed form that can be inverted. However, we can apply the generalized rejection sampling scheme given by eqs. (2.57) to (2.60). We begin by decomposing eq. (2.84) like eq. (2.57) to get

$$p\left(v_{\mathrm{T}}\right)=\left[\frac{1}{C_{1}}\left(v_{\mathrm{n}}+v_{\mathrm{T}}\right) v_{\mathrm{p}}^{-3} v_{\mathrm{T}}^{2} \exp\left(-v_{\mathrm{p}}^{-2} v_{\mathrm{T}}^{2}\right)\right]\left[\frac{4 \sigma_{\mathrm{fr}}}{\sqrt{\pi} C_{2}} \frac{\sqrt{v_{\mathrm{n}}^{2}+v_{\mathrm{T}}^{2}-2 v_{\mathrm{n}} v_{\mathrm{T}} \mu}}{v_{\mathrm{n}}+v_{\mathrm{T}}}\right] \tag{2.85}$$

where we have multiplied and divided each factor in square brackets by $v_\mathrm{n} + v_\mathrm{T}$, the maximum value taken on by $\sqrt{v_\mathrm{n}^2 + v_\mathrm{T}^2 - 2v_\mathrm{n}v_\mathrm{T}\mu}$, so that the second factor becomes a bounded function with maximum value $\sigma_\mathrm{fr}/\sqrt{\pi}C_2$ and, like eq. (2.58), becomes the acceptance probability

$$p\left(A \mid v_\mathrm{T}\right) = \frac{\sqrt{v_\mathrm{n}^2 + v_\mathrm{T}^2 - 2v_\mathrm{n}v_\mathrm{T}\mu}}{v_\mathrm{n} + v_\mathrm{T}}. \tag{2.86}$$

We have also split the normalizing constant from eq. (2.84) into $C = C_1 C_2$ so that the first factor in eq. (2.85) is a proper PDF if we choose

$$C_1 \coloneqq \int \left(v_\mathrm{n} + v_\mathrm{T}\right) v_\mathrm{p}^{-3} v_\mathrm{T}^2 \exp\left(-v_\mathrm{p}^{-2} v_\mathrm{T}^2\right) dv_\mathrm{T} = \frac{v_\mathrm{p}}{4}\left(\sqrt{\pi} v_\mathrm{p}^{-1} v_\mathrm{n} + 2\right) \tag{2.87}$$

so the first term in eq. (2.85) becomes

$$q\left(v_\mathrm{T}\right) \coloneqq \left(\frac{4 v_\mathrm{n} v_\mathrm{p}^{-4} v_\mathrm{T}^2}{\left(\sqrt{\pi} v_\mathrm{p}^{-1} v_\mathrm{n} + 2\right)} + \frac{4 v_\mathrm{T} v_\mathrm{p}^{-4} v_\mathrm{T}^2}{\left(\sqrt{\pi} v_\mathrm{p}^{-1} v_\mathrm{n} + 2\right)}\right) \exp\left(-v_\mathrm{p}^{-2} v_\mathrm{T}^2\right). \tag{2.88}$$

In order to sample from $q\left(v_\mathrm{T}\right)$, it is useful to define dimensionless variables

$$x \coloneqq \frac{v_\mathrm{T}}{v_\mathrm{p}} \tag{2.89a}$$

$$y \coloneqq \frac{v_\mathrm{n}}{v_\mathrm{p}} \tag{2.89b}$$

then rearrange eq. (2.88) and apply change of variables $dx = v_\mathrm{p}^{-1}\, dv_\mathrm{T}$ to get

$$q\left(x\right) \coloneqq \left(\frac{\sqrt{\pi} y}{\sqrt{\pi} y + 2}\right) \frac{4}{\sqrt{\pi}} x^2 e^{-x^2} + \left(\frac{2}{\sqrt{\pi} y + 2}\right) 2x^3 e^{-x^2}. \tag{2.90}$$

A sampling scheme for eq. (2.90) can be obtained if we note that the terms in parenthesis

$$w_1 \coloneqq \frac{\sqrt{\pi} y}{\sqrt{\pi} y + 2} \tag{2.91a}$$

$$w_2 \coloneqq \frac{2}{\sqrt{\pi} y + 2} \tag{2.91b}$$

satisfy $w_1 + w_2 = 1$ and the corresponding factors

$$q_1(x) := \frac{4}{\sqrt{\pi}} x^2 e^{-x^2} \tag{2.92a}$$

$$q_2(x) := 2x^3 e^{-x^2} \tag{2.92b}$$

are both properly normalized PDFs with direct inversion sampling schemes detailed by Everett and Cashwell [10] under rules C61 and C49, respectively. Thus, the procedure for sampling a target speed v_T is as follows:

1. With probability w_1, sample x from eq. (2.92a), otherwise sample from eq. (2.92b)
2. Convert to a candidate target speed v_T using eq. (2.90)
3. Sample a candidate cosine μ between $\mathbf{v}_\mathrm{n}$ and $\mathbf{v}_\mathrm{T}$ uniformly in $[-1, +1)$
4. Accept v_T with probability given by eq. (2.86)

With a sampled target speed v_T obtained, we next find its direction. From the isotropic distribution of target velocities in the lab frame we can uniformly sample an azimuthal angle γ about $\mathbf{v}_\mathrm{n}$ in $[0, 2\pi]$. Recalling that μ is the polar coordinate of $\mathbf{v}_\mathrm{T}$ with respect to $\mathbf{v}_\mathrm{n}$ in the lab frame, we have

$$\frac{\mathbf{v}_\mathrm{n}}{v_\mathrm{n}} = \hat{\mathbf{\Omega}} = \begin{bmatrix} \Omega_x & \Omega_y & \Omega_z \end{bmatrix}^T \tag{2.93a}$$

$$\frac{\mathbf{v}_\mathrm{T}}{v_\mathrm{T}} = \hat{\mathbf{\Omega}}_\mathrm{T} = \begin{bmatrix} \Omega_{\mathrm{T},x} & \Omega_{\mathrm{T},y} & \Omega_{\mathrm{T},z} \end{bmatrix}^T . \tag{2.93b}$$

A lengthy coordinate transformation relates $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{\Omega}}_{\mathrm{T}}$ by

$$\Omega_{\mathrm{T},x} = \mu\Omega_x + \sqrt{\frac{1-\mu^2}{1-\Omega_z^2}}\left(\Omega_x\Omega_z\cos\gamma - \Omega_y\sin\gamma\right) \tag{2.94a}$$

$$\Omega_{\mathrm{T},x} = \mu\Omega_y + \sqrt{\frac{1-\mu^2}{1-\Omega_z^2}}\left(\Omega_y\Omega_z\cos\gamma + \Omega_x\sin\gamma\right) \tag{2.94b}$$

$$\Omega_{\mathrm{T},z} = \mu\Omega_z - \sqrt{1-\mu^2}\sqrt{1-\Omega_z^2}\cos\gamma\,. \tag{2.94c}$$

A singularity occurs when $\Omega_z = 1$ so care must be taken to ensure that the coordinate system is selected such that the neutron velocity does not lie entirely along $\hat{\mathbf{z}}$. The choice of coordinate system is arbitrary so eq. (2.94) can be performed in any other coordinate system.

With target velocity $\mathbf{v}_{\mathrm{T}}$ sampled, the COM velocity can be obtained through

$$\mathbf{v}_{\mathrm{CM}} = \frac{\mathbf{v}_{\mathrm{n}} + A\mathbf{v}_{\mathrm{T}}}{1+A} \tag{2.95}$$

and the neutron velocity in the COM frame by

$$\mathbf{v}_{\mathrm{n,CM}} = \mathbf{v}_{\mathrm{n}} - \mathbf{v}_{\mathrm{CM}}\,. \tag{2.96}$$

Scattering is isotropic in the COM frame so the outgoing neutron velocity can be sampled with and then converted to an outgoing velocity in the lab frame $\mathbf{v}'_{\mathrm{n}}$.

The preceding derivation assumed the constant cross section model given by eq. (2.68) but methods exist for sampling when eq. (2.67) does not hold. The Doppler Broadening Rejection Correction (DBRC) method [11] proceeds as usual with the constant cross section model but performs an additional rejection step at the end. DBRC is particularly relevant when resonances exist in the range of relative velocities being considered. This work does not make use of DBRC but it is mentioned here to note that such resonances could be handled if necessary.

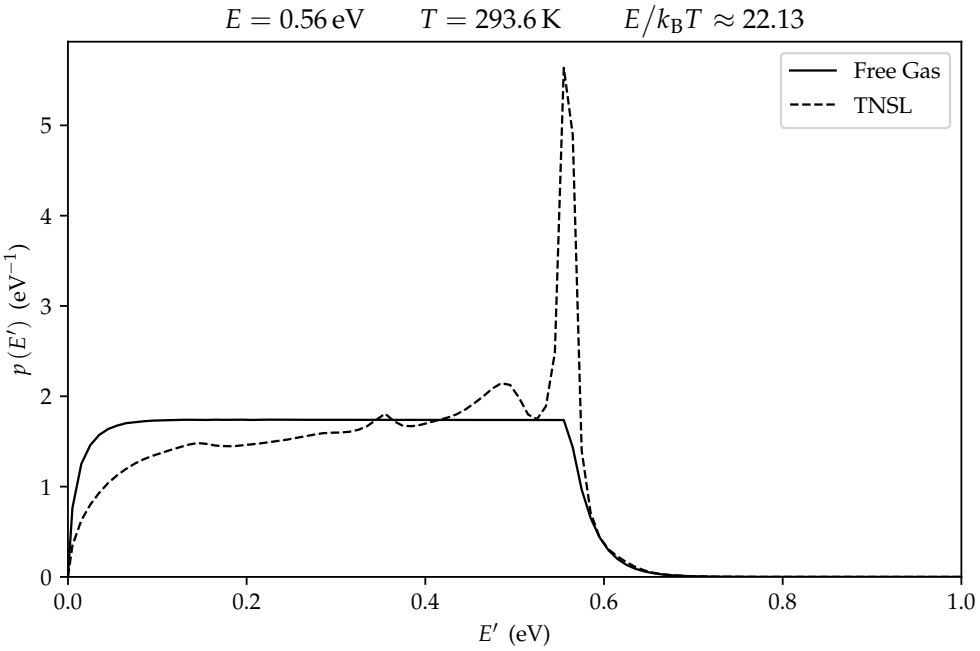


Figure 2.3: Outgoing energy distributions on H in H_2O modeled in MCNP using ENDF/B-VIII.0 TNSL data

2.4 Thermal Neutron Scattering Law

When the effects of interference between target nuclei or chemical binding cannot be neglected, the free gas model can yield poor results. Figure 2.3 shows a significant discrepancy in the outgoing energy spectra between free gas scattering and scattering with TNSL data.

At low energies, the de Broglie wavelength of a neutron is on the same order of magnitude as the interatomic spacing between atoms in the target molecule and quantum effects must be accounted for. The following derivation of the TNSL from scattering theory is based on a derivation by Squires [12] following the formalism of Van Hove [13].

Applying Fermi's golden rule to a neutron scattering off a system of nuclei, the Double Differential Cross Section (DDCS) is given by

$$\frac{d^2\sigma_\mathrm{s}}{d\Omega\, dE'} = \frac{k'}{k}\frac{1}{2\pi\hbar}\sum_{jj'}\overline{b_{j'}b_j}\int\langle j'j\rangle\exp\left(-i\omega t\right)dt \tag{2.97}$$

where

$$k \coloneqq |\boldsymbol{k}|\text{, the norm of the incident neutron wave vector}$$

$$k' \coloneqq |\boldsymbol{k}'|\text{, the norm of the outgoing neutron wave vector}$$

$$\hbar \coloneqq \text{ reduced Planck constant} \approx 1.054\,571\,817 \times 10^{-34}\,\mathrm{J\,s}$$

$$b \coloneqq \text{ the scattering length of the nuclei, here assumed to be real}$$

$$j \coloneqq \text{indexes each scatterer}$$

$$\omega \coloneqq (E - E')/\hbar$$

$$\langle j', j\rangle \coloneqq \langle \exp\left(-i\boldsymbol{\kappa}\cdot\boldsymbol{R}_{j'}(0)\right)\exp\left(i\boldsymbol{\kappa}\cdot\boldsymbol{R}_{j}(t)\right)\rangle$$

$$\boldsymbol{\kappa} \coloneqq \boldsymbol{k} - \boldsymbol{k}'$$

$$\boldsymbol{R}_j(t) \coloneqq \text{The position of nuclide } j \text{ at time } t$$

The $\langle j', j\rangle$ term correlates nuclei j' at $t' = 0$ with nuclei j at $t' = t$. It is obtained by defining

$$\langle A\rangle \coloneqq \sum_{\lambda} p_\lambda \langle\lambda|A|\lambda\rangle \tag{2.99}$$

which averages the expected value of an operator A at temperature T using p_λ, the probability that the scattering system is in the state λ, given the system obeys a Boltzmann distribution.

The $\overline{b_{j'}b_j}$ term is the expected value of the scattering lengths between nuclide $b_{j'}$ and b_j. Assuming scattering lengths between each nuclei are uncorrelated, we have

$$\overline{b_{j'}b_j} = \begin{cases} \left(\bar{b}\right)^2 & j' \neq j \\ \overline{b^2} & j' = j \end{cases} \tag{2.100}$$

so the sum in eq. (2.97) can be decomposed

$$\begin{aligned}\frac{d^2\sigma_s}{d\Omega\, dE'} = &\quad \frac{k'}{k}\frac{1}{2\pi\hbar}\left(\bar{b}\right)^2 \sum_{j'\neq j}\int \langle j', j\rangle \exp\left(-i\omega t\right) dt \\ &+ \frac{k'}{k}\frac{1}{2\pi\hbar}\overline{b^2}\sum_{j}\int \langle j', j\rangle \exp\left(-i\omega t\right) dt \\ = &\quad \frac{k'}{k}\frac{1}{2\pi\hbar}\left(\bar{b}\right)^2 \sum_{jj'}\int \langle j, j\rangle \exp\left(-i\omega t\right) dt \\ &+ \frac{k'}{k}\frac{1}{2\pi\hbar}\left(\overline{b^2} - \left(\bar{b}\right)^2\right)\sum_{j}\int \langle j', j\rangle \exp\left(-i\omega t\right) dt\,.\end{aligned} \tag{2.101}$$

The first term on the last line is the *coherent* scattering cross section

$$\left(\frac{d^2\sigma_s}{d\Omega\, dE'}\right)_{\text{coh}} := \frac{\sigma_{\text{coh}}}{4\pi}\frac{k'}{k}\frac{1}{2\pi\hbar}\sum_{jj'}\int \langle j', j\rangle \exp\left(-i\omega t\right) dt \tag{2.102a}$$

$$\sigma_{\text{coh}} := 4\pi\left(\bar{b}\right)^2 \tag{2.102b}$$

which accounts for interference effects between distinct nuclei and is strongly affected by the arrangement of atoms in the system. Coherent scattering is most evident in crystalline latices which exhibit neutron diffraction. The second term is the *incoherent* scattering cross section

$$\left(\frac{d^2\sigma_s}{d\Omega\, dE'}\right)_{\text{inc}} := \frac{\sigma_{\text{inc}}}{4\pi}\frac{k'}{k}\frac{1}{2\pi\hbar}\sum_{j}\int \langle j', j\rangle \exp\left(-i\omega t\right) dt \tag{2.103a}$$

$$\sigma_{\text{inc}} := 4\pi\left(\overline{b^2} - \left(\bar{b}\right)^2\right) \tag{2.103b}$$

which only accounts for correlations between the same nucleus at different times.

We can define *intermediate* and *self-intermediate* functions

$$I(\boldsymbol{\kappa}, t) := \sum_{jj'} \langle \exp\left(-i\boldsymbol{\kappa} \cdot \boldsymbol{R}_{j'}(0)\right) \exp\left(i\boldsymbol{\kappa} \cdot \boldsymbol{R}_j(t)\right) \rangle \tag{2.104a}$$

$$I_\mathrm{s}(\boldsymbol{\kappa}, t) := \sum_{j} \langle \exp\left(-i\boldsymbol{\kappa} \cdot \boldsymbol{R}_j(0)\right) \exp\left(i\boldsymbol{\kappa} \cdot \boldsymbol{R}_j(t)\right) \rangle \tag{2.104b}$$

and express the coherent and incoherent scattering cross sections as

$$\left(\frac{d^2\sigma_\mathrm{s}}{d\Omega\, dE'}\right)_\mathrm{coh} := \frac{\sigma_\mathrm{coh}}{4\pi}\frac{k'}{k}\frac{1}{2\pi\hbar}\int I(\boldsymbol{\kappa}, t) \exp(-i\omega t)\, dt \tag{2.105a}$$

$$\left(\frac{d^2\sigma_\mathrm{s}}{d\Omega\, dE'}\right)_\mathrm{inc} := \frac{\sigma_\mathrm{inc}}{4\pi}\frac{k'}{k}\frac{1}{2\pi\hbar}\int I_\mathrm{s}(\boldsymbol{\kappa}, t) \exp(-i\omega t)\, dt\,. \tag{2.105b}$$

We can define time Fourier transforms of eq. (2.104)

$$S(\boldsymbol{\kappa}, \omega) := \frac{1}{2\pi\hbar}\int I(\boldsymbol{\kappa}, t) \exp(-i\omega t)\, dt \tag{2.106a}$$

$$S_\mathrm{s}(\boldsymbol{\kappa}, \omega) := \frac{1}{2\pi\hbar}\int I_\mathrm{s}(\boldsymbol{\kappa}, t) \exp(-i\omega t)\, dt \tag{2.106b}$$

and express eq. (2.105)

$$\left(\frac{d^2\sigma_\mathrm{s}}{d\Omega\, dE'}\right)_\mathrm{coh} := \frac{\sigma_\mathrm{coh}}{4\pi}\frac{k'}{k} S(\boldsymbol{\kappa}, \omega) \tag{2.107a}$$

$$\left(\frac{d^2\sigma_\mathrm{s}}{d\Omega\, dE'}\right)_\mathrm{inc} := \frac{\sigma_\mathrm{inc}}{4\pi}\frac{k'}{k} S_\mathrm{s}(\boldsymbol{\kappa}, \omega)\,. \tag{2.107b}$$

We call $S(\boldsymbol{\kappa}, \omega)$ the *scattering function* and $S_\mathrm{s}(\boldsymbol{\kappa}, \omega)$ the *self-scattering function*. We can express the DDCS as

$$\frac{d^2\sigma_\mathrm{s}}{d\Omega\, dE'} = \frac{1}{4\pi}\frac{k'}{k}\left(\sigma_\mathrm{coh} S(\boldsymbol{\kappa}, \omega) + \sigma_\mathrm{inc} S_\mathrm{s}(\boldsymbol{\kappa}, \omega)\right) \tag{2.108}$$

or, defining $\sigma_b := \sigma_{coh} + \sigma_{inc}$,

$$\frac{d^2\sigma_s}{d\Omega\, dE'} = \frac{1}{4\pi}\frac{k'}{k}\sigma_b\left(\frac{\sigma_{coh}}{\sigma_b}S(\boldsymbol{\kappa},\omega) + \frac{\sigma_{inc}}{\sigma_b}S_s(\boldsymbol{\kappa},\omega)\right) \tag{2.109}$$

where we can identify

$$S(\boldsymbol{\kappa},\omega) := \frac{\sigma_{coh}}{\sigma_b}S(\boldsymbol{\kappa},\omega) + \frac{\sigma_{inc}}{\sigma_b}S_s(\boldsymbol{\kappa},\omega) \tag{2.110}$$

as the scattering law.

In nuclear engineering, it is more conventional to use the form

$$S(\alpha,\beta \mid T) := \frac{k_B T}{\hbar}e^{\frac{\beta}{2}}S(\boldsymbol{\kappa},\omega) \tag{2.111}$$

so the DDCS take the form

$$\sigma_s\left(E \to E', \hat{\boldsymbol{\Omega}}\cdot\hat{\boldsymbol{\Omega}}', T\right) = \frac{\sigma_b}{2k_B T}\sqrt{\frac{E'}{E}}e^{-\frac{\beta}{2}}S(\alpha,\beta \mid T) \tag{2.112}$$

where σ_b is the characteristic bound scattering cross section, k_B is the Boltzmann constant, T is the target nuclide temperature, E is the incident energy of the incoming neutron, E' is the outgoing energy of the scattered neutron, and $S(\alpha,\beta \mid T)$ is the TNSL. The TNSL is defined in terms of the dimensionless energy transfer

$$\beta := \frac{E' - E}{k_B T} \tag{2.113}$$

and the dimensionless momentum transfer

$$\alpha := \frac{E + E' - 2\mu\sqrt{EE'}}{Ak_B T} \tag{2.114}$$

where μ is the scattering cosine, and A is the atomic weight. Solving eq. (2.113) for E'

gives

$$E' = E + \beta k_B T \tag{2.115}$$

and solving eq. (2.114) for μ gives

$$\mu = \frac{E + E' - \alpha A k_B T}{2\sqrt{EE'}}. \tag{2.116}$$

The minimum and maximum possible values of β are

$$\beta_{min} = -\frac{E}{k_B T} \tag{2.117a}$$

$$\beta_{max} = \infty \tag{2.117b}$$

which correspond to $E' = 0$ and $E' = \infty$, respectively. A lower bound for β_{max} can be justified theoretically [14]. The formulation for TNSL treated the Probability Density of State (PDOS) in the harmonic approximation which allowed for excitations up to infinity. The realistic potential is unharmonic and limited by the binding energy E_b of the scattering nucleus in the molecule or lattice structure so an upper limit

$$\beta_{max} = \frac{E_b}{k_B T} \tag{2.118}$$

can be identified. Analysis by Ballinger [15] shows that upscattering above $\beta = 20$ is unlikely so it was used as a cutoff. The minimum and maximum possible values of α are

$$\alpha_{min} = \frac{\left(\sqrt{E} - \sqrt{E + \beta k_B T}\right)^2}{A k_B T} \tag{2.119a}$$

$$\alpha_{max} = \frac{\left(\sqrt{E} + \sqrt{E + \beta k_B T}\right)^2}{A k_B T} \tag{2.119b}$$

which correspond to $\mu = +1$ (forward scattering) and $\mu = -1$ (backscattering), respec-

tively.

2.4.1 Incoherent Approximation

The temperature dependence of $S(\alpha, \beta \mid T)$ derives from $\langle j', j \rangle$ in eq. (2.97) which correlates nuclide j' at $t' = 0$ to nuclide j at $t' = t$. Recalling that the coherent term in eq. (2.102) was computed by summing over cases where $j' = j$ as well as $j' \neq j$, we can separate the coherent scattering function into a *distinct* and self part

$$S(\boldsymbol{\kappa}, \omega) = S_{\mathrm{d}}(\boldsymbol{\kappa}, \omega) + S_{\mathrm{s}}(\boldsymbol{\kappa}, \omega). \tag{2.120}$$

In the incoherent approximation, interference effects are neglected by setting $S_{\mathrm{d}}(\boldsymbol{\kappa}, \omega) = 0$ so eq. (2.108) becomes

$$\frac{d^2\sigma_{\mathrm{s}}}{d\Omega\, dE'} = \frac{1}{4\pi}\frac{k'}{k}\left(\sigma_{\mathrm{coh}} + \sigma_{\mathrm{inc}}\right) S_{\mathrm{s}}(\boldsymbol{\kappa}, \omega) \tag{2.121}$$

where $S_{\mathrm{s}}(\boldsymbol{\kappa}, \omega)$ only on depends on correlations between the same particle across time. The velocity autocorrelation function

$$\langle \mathbf{v}(0) \cdot \mathbf{v}(t) \rangle \tag{2.122}$$

can be obtained from Molecular Dynamics (MD) simulations by simulating an ensemble of scatterers interacting with each other through some potential model. The PDOS

$$\rho(\omega) = \int \cos(\omega t) \langle \mathbf{v}(0) \cdot \mathbf{v}(t) \rangle, \tag{2.123}$$

characterizing the density of quantum states in the scattering medium, is what codes such as NJOY accept as input.

2.5 Evaluations of Thermal Neutron Scattering Laws

Two sets of Thermal Neutron Scattering Law (TNSL) data were used in this work: ENDF/B-VIII.0 data in the File 7 format was used to compare new methods developed in this work against established codes. For uncertainty quantification, independent and identically distributed PDOS data was used to create new TNSL evaluations. This section discusses the theoretical models underlying each evaluation as well as any preprocesing that was required to prepare the TNSL data for this work.

2.5.1 ENDF/B-VIII.0 Data

The ENDF/B-VIII.0 evaluation [16] is the most recent release of nuclear data by the Cross Section Evaluation Working Group (CSEWG). ENDF/B-VIII.0 is released in coordination with other nuclear data formats such as the Joint Evaluated Fission and Fusion (JEFF) Nuclear Data Library [17] in Europe and the Japan Evaluated Nuclear Data Library (JENDL) [18] in Japan. CSEWG also issues the same nuclear data in the Generalized Nuclear Database Structure (GNDS) format.

ENDF/B-VIII.0 provides a thermal neutron scattering sublibrary containing 33 evaluations for 24 materials. The evaluation for hydrogen in light water (H in H_2O) [19] is based on the Centro Atómico Bariloche (CAB) [20] model for light water. Prior to the CAB model, all thermal scattering files for light and heavy water in major evaluated nuclear data files were produced with models from either the General Atomics (GA) model [21] or the Institut für Kernenergetik und Energiesysteme (IKE) model [22]. The GA model was first used by the GASKET [21] code and subsequently used by the LEAPR [23] module of NJOY to produce TNSL evaluations in ENDF/B-VI. The IKE model was used by Keinert and Mattes to add TNSL data to JEF-1 in 1984.

The CAB model introduces a frequency spectrum obtained from Groningen MAchine for Chemical Simulation (GROMACS), a MD software suite using a TIP4P/2005 water

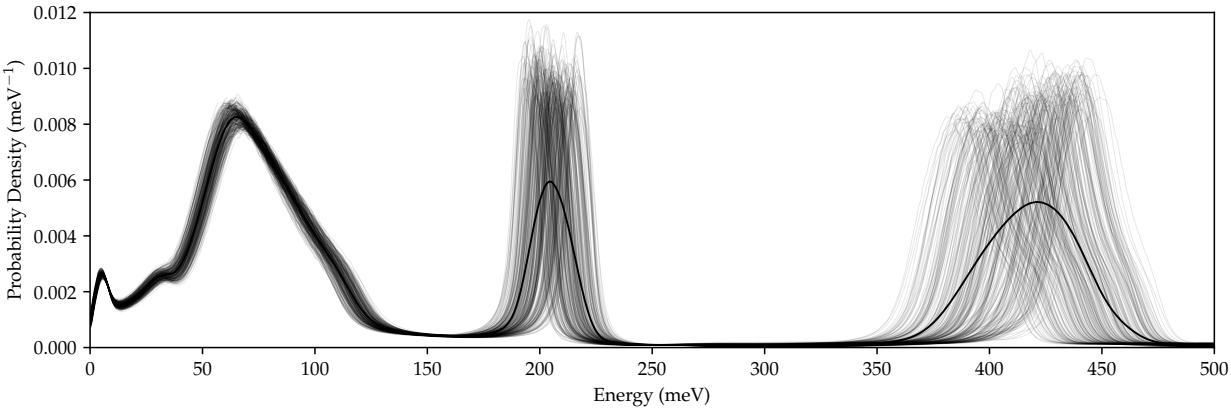


Figure 2.4: 250 realizations of PDOS courtesy Chapman [25]

model [24]. The frequency spectra are then used in the incoherent approximation and the Gaussian approximation to produce TNSL evaluations.

2.5.2 Probability Density of State (PDOS) Data

In order to perform uncertainty quantification, independent identically distributed realizations of TNSL data was required. Work by Chapman [25] developed a methodology for generating covariance data for thermal neutron scattering cross sections. The work used a Unified Monte Carlo (UMC) method to fit TIP4P/2005f model parameters of light water simulated using GROMACS and MCNP to double-differential scattering cross sections measured at Oak Ridge National Laboratory (ORNL) at the Spallation Neutron Source (SNS) [26].

2.6 Uncertainty Quantification

Historically, the approach to nuclear reactor design, safety evaluation, and licensing has been to make modeling assumptions so conservative so as to ensure that results are well within safety requirements [27]. The traditional approach can be overly conservative, compensating for a lack of knowledge about the modeling methodology or uncertainty in nuclear data by introducing needlessly large safety margins.

In recent times, a set of approaches which can broadly be called Best Estimate Plus Uncertainty (BEPU) [28] emphasizes accurate computer simulations using nuclear data with well-characterized uncertainties. Uncertainties in the nuclear data are propagated through computer simulations and reported in integral estimates. Despite the name, the approach remains conservative in that sources of uncertainty are better known and can thus be better addressed directly.

Sensitivity Analysis (SA) and Uncertainty Quantification (UQ) are broad fields [29] of study but this work relies on two methods. A "sandwich rule" approach propagates uncertainties in nuclear data covariances using sensitivities through a Taylor series approximation. A more computationally expensive Total Monte Carlo (TMC) method performs multiple Monte Carlo simulations on independent, identically distributed nuclear data to more directly estimate the effect of nuclear data uncertainties on estimators.

2.6.1 Uncertainty Propagation using a Taylor Series Approximation

We now derive a formula used to propagate uncertainties in nuclear data to estimated quantities using a multivariate Taylor series approximation. We begin by modeling nuclear data as two random variables X and Y but we will extend this to an arbitrary number of random variables later. Consider a response $R = g(X, Y)$ where we assume that the scoring function g is differentiable so that a Taylor series expansion in X and Y can be performed about the mean $(\mathbb{E}[X], \mathbb{E}[Y])$ to approximate

$$r = g(x, y) \approx g(\mathbb{E}[X], \mathbb{E}[Y]) + (x - \mathbb{E}[X]) \frac{\partial g}{\partial x} + (y - \mathbb{E}[Y]) \frac{\partial g}{\partial y} \tag{2.124}$$

where the partial derivatives are evaluated at $(\mathbb{E}[X], \mathbb{E}[Y])$. The expected value of R is

$$\mathbb{E}[R] \approx \mathbb{E}[g(\mathbb{E}[X], \mathbb{E}[Y])] + (\mathbb{E}[X] - \mathbb{E}[X]) \frac{\partial g}{\partial x} + (\mathbb{E}[Y] - \mathbb{E}[Y]) \frac{\partial g}{\partial y} \tag{2.125}$$

but the first term is a constant and the second and third terms vanish so we have

$$\mathbb{E}[R] \approx g(\mathbb{E}[X], \mathbb{E}[Y]) . \tag{2.126}$$

From the definition of variance we have

$$\mathbb{V}[R] = \mathbb{E}\left[(R - \mathbb{E}[R])^2\right] . \tag{2.127}$$

Substituting eq. (2.124) for R and eq. (2.126) for $\mathbb{E}[R]$, we have

$$\begin{aligned}
\mathbb{V}[R] \approx \mathbb{E}\Bigg[\Bigg(& g(\mathbb{E}[X], \mathbb{E}[Y]) + (X - \mathbb{E}[X]) \frac{\partial g}{\partial x} + (Y - \mathbb{E}[Y]) \frac{\partial g}{\partial y} \\
& - g(\mathbb{E}[X], \mathbb{E}[Y]) \Bigg)^2 \Bigg] \\
= \mathbb{E}\left[\left((X - \mathbb{E}[X]) \frac{\partial g}{\partial x} + (Y - \mathbb{E}[Y]) \frac{\partial g}{\partial y}\right)^2\right] & \\
= \mathbb{E}\Bigg[(X - \mathbb{E}[X])^2 \left(\frac{\partial g}{\partial x}\right)^2 + (Y - \mathbb{E}[Y])^2 \left(\frac{\partial g}{\partial y}\right)^2 & \\
2(X - \mathbb{E}[X])(Y - \mathbb{E}[Y]) \frac{\partial g}{\partial x} \frac{\partial g}{\partial y}\Bigg] . &
\end{aligned} \tag{2.128}$$

From the linearity of expectation we get

$$\begin{aligned}
\mathbb{V}[R] = & \quad \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] \left(\frac{\partial g}{\partial x}\right)^2 + \mathbb{E}\left[(Y - \mathbb{E}[Y])^2\right] \left(\frac{\partial g}{\partial y}\right)^2 \\
& + 2\mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] \frac{\partial g}{\partial x} \frac{\partial g}{\partial y} \\
= & \ \mathbb{V}[X] \left(\frac{\partial g}{\partial x}\right)^2 + \mathbb{V}[Y] \left(\frac{\partial g}{\partial y}\right)^2 + 2\,\mathrm{Cov}(X, Y) \frac{\partial g}{\partial x} \frac{\partial g}{\partial y} .
\end{aligned} \tag{2.129}$$

The result can be generalized to n random variables $X_1, \ldots, X_n$ for $R = g(X_1, \ldots, X_n)$ with

$$\mathbb{V}[R] = \sum_{i=1}^{n} \mathbb{V}[X_i] \left(\frac{\partial g}{\partial x_i}\right)^2 + 2 \sum_{i=1}^{n} \sum_{j>i}^{n} \mathrm{Cov}\left(X_i, X_j\right) . \tag{2.130}$$

A more compact expression can be obtained by defining a covariance matrix

$$\mathbf{C} \coloneqq \begin{bmatrix} \mathrm{Cov}\left(X_1, X_1\right) & \cdots & \mathrm{Cov}\left(X_1, X_n\right) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}\left(X_n, X_1\right) & \cdots & \mathrm{Cov}\left(X_n, X_n\right) \end{bmatrix} \tag{2.131}$$

and sensitivity vector

$$\mathbf{s} \coloneqq \begin{bmatrix} \frac{\partial g}{\partial x_1} \\ \vdots \\ \frac{\partial g}{\partial x_n} \end{bmatrix} \tag{2.132}$$

so that we can express eq. (2.130) as

$$\mathbb{V}\left[R\right] \approx \mathbf{s}^T \mathbf{C} \mathbf{s} \tag{2.133}$$

which is sometimes called the *sandwich rule* because it places a covariance matrix between two sensitivity vectors. A key assumption made in deriving this expression is that the function g can be well approximated by a linear function within a few standard deviations about the mean. If this is not the case, higher order expansions may be performed, but those require higher order moments which are not always easy to estimate within statistical uncertainty.

2.6.2 Differential Operator Sampling

Sensitivities are problem dependent and must be calculated using one of several methods. Differential Operator Sampling (DOS), first proposed by Mikhailov [30] in 1967, allows one to compute the sensitivity of integral quantities without changing the random walk of each history. Rief [31] formalized the approach in the context of computing gradients and higher order derivatives of quantities obtained from solutions to the neutron transport equation.

We recall that Monte Carlo methods produce estimates of a response

$$\mathbb{E}[R] = \int_{\Omega} R(\omega)\, p(\omega)\, d\omega \tag{2.134}$$

by drawing independent, identically distributed samples of ω. Sensitivities can be computed by taking the derivative of eq. (2.134)

$$\frac{d\mathbb{E}[R]}{d\lambda} = \frac{d}{d\lambda} \int_{\Omega} Rp(\omega)\, d\omega \tag{2.135}$$

and, assuming $d\Omega/d\lambda = 0$, move the derivative operator inside the integral to obtain, after applying the product rule,

$$\frac{d\mathbb{E}[R]}{d\lambda} = \int_{\Omega} \left(\frac{dR}{d\lambda} p(\omega) + R \frac{dp(\omega)}{d\lambda} \right) d\omega\,. \tag{2.136}$$

Factoring out $Rp(\omega)$ gives

$$\frac{d\mathbb{E}[R]}{d\lambda} = \int_{\Omega} \left(\frac{1}{R}\frac{dR}{d\lambda} + \frac{1}{p(\omega)} \frac{dp(\omega)}{d\lambda} \right) Rp(\omega)\, d\omega \tag{2.137}$$

which looks identical to eq. (2.134) if one makes the substitution

$$R \rightarrow \left(\frac{1}{R}\frac{dR}{d\lambda} + \frac{1}{p(\omega)} \frac{dp(\omega)}{d\lambda} \right) R \tag{2.138}$$

which suggests that the sensitivity can be estimated using the unperturbed PDF but applying the modified response given by eq. (2.138). The first term,

$$\frac{1}{R}\frac{dR}{d\lambda}\,, \tag{2.139}$$

is called the *direct effect* and the second term,

$$\frac{1}{p(\omega)}\frac{dp(\omega)}{d\lambda}, \tag{2.140}$$

is called the *indirect effect*. The direct effect accounts for changes in the response while the indirect effect accounts for changes in the random walk due to a perturbation. The direct effect is computed at the time scores are tallied but the indirect effect generally must be stored for the entire duration of a history.

The indirect effect given by eq. (2.140) can be expanded using eq. (2.35) and the logarithmic product rule eq. (4) to get

$$\frac{1}{p(\omega)}\frac{dp(\omega)}{d\lambda} = \frac{\frac{d}{d\lambda}p(x_0)}{p_{X_0}} + \sum_{c'=1}^{c}\frac{\frac{d}{d\lambda}p(x_{c'} \mid x_{c'-1})}{p(x_{c'} \mid x_{c'-1})} \tag{2.141}$$

2.6.3 Automatic Differentiation

Calculation of direct and indirect effects require the evaluation of various derivatives. Options for computing derivatives on a computer generally fall into four categories [32]:

1. *Analytic Methods* in which one manually calculates a closed-form expression for the derivative and implements the function. This requires a new expression to be calculated and implemented whenever the differentiated function changes. Moreover, it can become laborious and prone to error for complicated expressions.

2. *Numeric Differentiation* in which finite differences are used to approximate derivatives. A function is evaluated at $f(x)$ and a nearby point $f(x+h)$ to approximate the limit

   $$\frac{df}{dx} = \lim_{h\to 0}\frac{f(x+h) - f(x)}{h}. \tag{2.142}$$

   Choosing h too large results in approximation errors but choosing h too small will result in rounding errors from catastrophic cancellation [33], a phenomenon where

subtracting two numbers of similar magnitude results in large relative errors in the difference. For computing gradients, numeric differentiation can become prohibitively expensive because a separate calculation must be done for each partial derivative.

3. *Symbolic Differentiation* wherein a computer algebra system automatically applies the standard rules of differentiation to arrive at a symbolic expression for the derivative. This method is free from approximation and rounding errors that occur in numeric differentiation but suffers from "expression swell" [34] where derivative expressions become exponentially complex if simplifications (such as factorization or the application of well-known algebraic identities) are not performed. Finally, symbolic differentiation only operates on closed-form expressions so constructs such as loops and conditionals cannot be captured.

4. *Automatic Differentiation* which is discussed here.

Automatic Differentiation (AD) was introduced [35] as a procedure for the evaluation of total and/or partial derivatives of arbitrary algebraic functions on a computer. A key tool in AD is the use of *dual numbers* [36], a number system similar to complex numbers. A dual number is defined

$$d = a + b\epsilon \tag{2.143}$$

where a and b are real. Whereas i in the complex number system satisfies $i^2 = -1$, the ϵ in dual numbers satisfies $\epsilon^2 = 0$.

Any function with a Taylor series in $x \in \mathbb{R}$

$$f(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_n x^n \tag{2.144}$$

has a dual-valued counterpart

$$\begin{aligned} f(d) &= f(a + b\epsilon) \\ &= c_0 + c_1 (a + b\epsilon) + c_2 (a + b\epsilon)^2 + \ldots + c_n (a + b\epsilon)^n \\ &= c_0 + c_1 (a) + c_2 (a)^2 + \ldots + c_n (a)^n \\ &\quad + c_1 b\epsilon + 2c_2 ab\epsilon + \ldots + nc_n a^{n-1} b\epsilon \\ &= f(a) + f'(a) b\epsilon \end{aligned} \tag{2.145}$$

which, by carrying out the evaluation of $f(d)$, simultaneously evaluates $f(a)$ and its derivative $f'(a)$.

There are two distinct types of AD: forward and reverse. Both modes operate by recursively solving terms in the chain rule

$$\frac{\partial f(x(y(z)))}{\partial z} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial y} \frac{\partial y}{\partial z}. \tag{2.146}$$

In forward mode, terms are evaluated from right to left

$$\frac{\partial f(x(y(z)))}{\partial z} = \frac{\partial f}{\partial x} \left(\frac{\partial x}{\partial y} \left(\frac{\partial y}{\partial z} \right) \right), \tag{2.147}$$

storing intermediate results as dual numbers by augmenting standard datatypes such as `double` in C++. This process is expressed as a computational graph in fig. 2.5. The value stored at each node is expressed as a tuple $\{a, b\}$ where the first entry is the value being computed and the second entry is the dual number containing the corresponding derivative. Leaf nodes represent independent variables. The root node corresponds to the function being differentiated. Differentiation of $f(x, y) = \sin x + xy$ with respect to x begins by seeding the dual number of x to 1 and the dual number of all other independent variables to 0. As computation progresses towards the root node, derivatives are propagated

$$\left\{\sin x + xy, \frac{\partial(\sin x+xy)}{\partial x} = \frac{\partial(\sin x+xy)}{\partial(\sin x)}\frac{\partial(\sin x)}{\partial x} + \frac{\partial(\sin x+xy)}{\partial(xy)}\frac{\partial(xy)}{\partial x} = \cos x + y\right\}$$

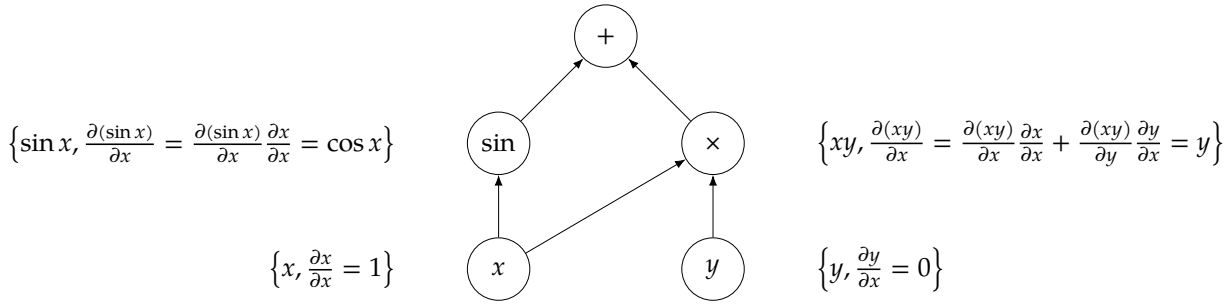


Figure 2.5: Forward AD

through the dual variables w according to

$$w_i = \sum_j \frac{\partial w_i}{\partial w_j} w_j \tag{2.148}$$

where j indexes all of node i's predecessors.

Forward AD is simpler to implement because the evaluation of dual numbers follows the normal order of computation. Implementations such as autodiff [37] are able to implement forward mode AD using template metaprogramming in order to optimize arithmetic operations at compile time. One significant drawback of forward AD is that the entire calculation must be re-run for each independent variable one wishes to differentiate with respect to.

The other mode of AD is reverse mode which solves eq. (2.146) in the reverse order

$$\frac{\partial f\left(x\left(y\left(z\right)\right)\right)}{\partial z} = \left(\left(\frac{\partial f}{\partial x}\right)\frac{\partial x}{\partial y}\right)\frac{\partial y}{\partial z} \tag{2.149}$$

using two passes: the first pass occurs similarly to forward mode. The values of each node are evaluated but the dual variables are not. It is followed by a reverse pass, shown in fig. 2.6, where the dual variables are propagated from the root to the leaves according to eq. (2.148), but the j now indexes all of node i's successors.

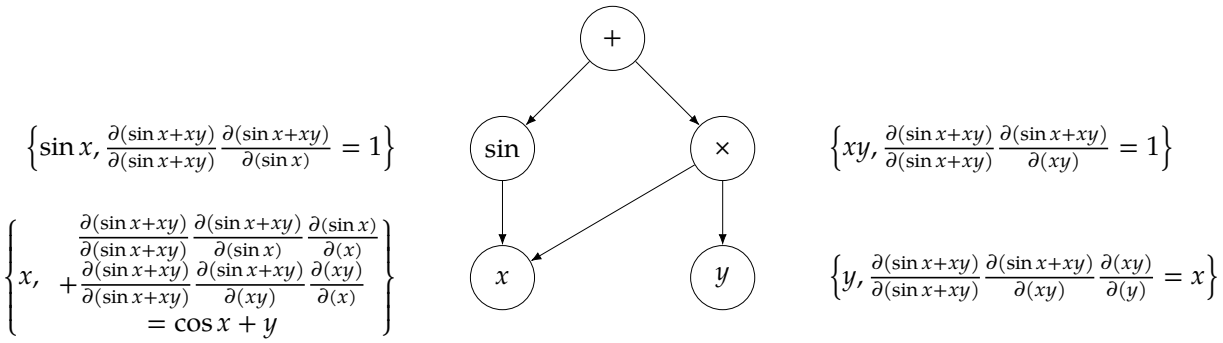


Figure 2.6: Reverse AD, reverse pass

Reverse mode AD is more difficult to implement since the reverse pass requires information about values evaluated during the forward pass. However, it has the significant advantage that all derivatives can be computed in a single forward/reverse pass. This makes it particularly suitable for cases where sensitivities with respect to many parameters must be computed.

2.6.4 Total Monte Carlo

Total Monte Carlo (TMC) is a "brute-force" approach to propagating nuclear data uncertainties. Rather than compute covariance matrices and estimate sensitivities in a single Monte Carlo simulation, TMC performs multiple runs with each run using independent and identically distributed realizations of nuclear data. Standard sample statistics can then be applied on the resulting estimators to yield estimates of the variance due to nuclear data uncertainty.

Initially used to connect uncertainties in microscopic nuclear physics models to macroscopic reactor design models [38], TMC has become a viable approach to UQ thanks to the ever-increasing availability of computing power.

This work will use TMC to compare against a new UQ methodology. A sufficiently large number of histories must be performed to ensure that statistical uncertainties are

small compared to systematic uncertainties due to differences in nuclear data.

2.7 Covariance Formats

The sensitivities given in eq. (2.132) are generally unique to a specific problem and must be computed by end users. The covariance matrix eq. (2.131) is unique to a set of nuclear data and must be provided by evaluators. This section discusses existing formats for exchanging nuclear data, their limitations, and work performed to address issues with storing multidimensional covariance data.

2.7.1 ENDF-6 Format

The ENDF-6 format is the de facto standard for distributing nuclear data. The format is used by major nuclear data libraries like Evaluated Nuclear Data File (ENDF) [16] in the United States, JEFF [38] in Europe, and JENDL [18] in Japan. The first release of ENDF nuclear data, ENDF/B-I in 1966, was the result of a project funded by the Atomic Energy Commission (AEC). It has since received regular updates up to the most recent release ENDF/B-VIII.0 by CSEWG in 2018. Nuclear data is continuously being improved with new experimental data and improved theoretical models so nuclear data libraries will continue to be updated in the foreseeable future.

ENDF-6 is the most recent version of the format used for storing nuclear data. It improves upon earlier formats by allowing for higher incident energies, more complete descriptions of secondary particle distributions, and charged particle production data. The ENDF-6 format also introduces nuclear data sub-libraries for decay data, fission product yield data, photo-atomic data and thermal neutron scattering data. In addition to central, or mean, values, ENDF-6 also provides covariance formats for some nuclear data such as fission multiplicities, resonance parameters, and neutron cross sections, to name a few.

Despite these improvements, the ENDF format remains bound to original design deci-

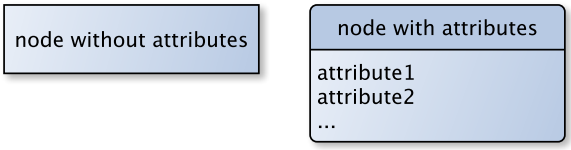


Figure 2.7: Nodes in entity-relationship diagram. Left: Node without attributes. Right: Node with attributes.

sions which were guided largely in part by a need to be economical with the space afforded on physical punchcards. At present, ENDF-6 supports formats for storing covariance data of some nuclear data such as fission multiplicities, resonance parameters, and neutron cross sections but does not support them for others. In particular, there is no format for storing TNSL covariance data in the ENDF-6 format nor are there plans for introducing it in the near future as the nature of the ENDF format and the codes that parse it make it quite difficult to extend.

2.7.2 Generalized Nuclear Database Structure (GNDS)

In 2012 the Working Party of Evaluation Cooperation (WPEC) began efforts to modernize the ENDF-6 format. A set of detailed requirements for a next generation nuclear data format, GNDS [39], outlined design goals and features for a format intended to eventually supersede ENDF-6. A key notion in GNDS is separation of formats and data structure. Whereas ENDF-6 was quite particular about the format for storage efficiency on a physical medium, GNDS does not require a particular format for data serialization. Instead, GNDS specifies a hierarchical structure which can be expressed in an appropriate meta-language such as HDF5, ROOT, JSON, or Python classes.

Entity-relationship diagrams provide a compact notation for representing nodes as boxes as shown in fig. 2.7. Node names are indicated in headings and any attributes as subheadings, if any. Crow's foot notation, depicted in fig. 2.8, specifies the relationships between nodes by indicating the expected number of children nodes for a given parent

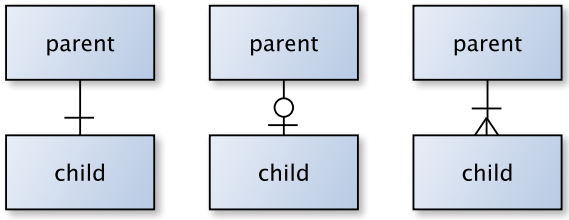
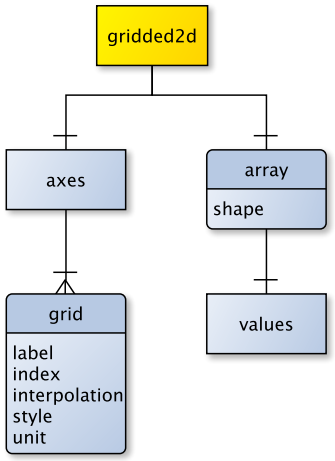


Figure 2.8: Crow's foot notation. Left: Exactly one child required. Center: Zero or one child. Right: One or more children.

```
<gridded2d>
  <axes>
    <grid label="beta" index="2"
     ↪ interpolation="lin-log" style="points">
      <values>0.000000 0.300000 ... </values>
    </grid>
    <grid label="alpha" index="1"
     ↪ interpolation="lin-log" style="points">
      <values>0.000500 0.200000 ... </values>
    </grid>
    <axis label="S" index="0" />
  </axes>
  <array shape="1,26,19">
    <values>4.672386e+00 3.098872e-01 ... </values>
  </array>
</gridded3d>
```

(a) XML file



(b) Entity-relationship diagram

Figure 2.9: A `gridded2d` container containing simplified $S(\alpha, \beta)$ data as an XML file and as an entity-relationship diagram.

node. Figure 2.9 shows an example of a `gridded2d` node used to store simplified $S(\alpha, \beta)$ data as both an abstract hierarchical entity-relationship diagram and as a concrete Extensible Markup Language (XML) file.

Covariance data in GNDS are treated quite generally. The requirements document [39] makes this goal explicit:

> The legacy ENDF format stores nuclear data with a very complex set of schemes: the ENDF manual takes over 80 pages to describe seven distinct types of data. Arguably, there should be one "simple" data structure, after all a covariance matrix is, at its heart, just a matrix.

At the beginning of this project, GNDS supported covariances for any 1D dataset $\begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix}$ by storing the corresponding covariance matrix

$$\mathbf{C} \coloneqq \begin{bmatrix} \mathrm{Cov}\left(X_1, X_1\right) & \cdots & \mathrm{Cov}\left(X_1, X_n\right) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}\left(X_n, X_1\right) & \cdots & \mathrm{Cov}\left(X_n, X_n\right) \end{bmatrix} \tag{2.150}$$

as a 2D array using the `gridded2d` container. However, support for multidimensional dataset was not supported. Covariance for a $m \times n$ 2D dataset can be stored by flattening

$$\begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,n-1} & X_{1,n} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,n-1} & X_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ X_{m-1,1} & X_{m-1,2} & \cdots & X_{m-1,n-1} & X_{m-1,n} \\ X_{m,1} & X_{m,2} & \cdots & X_{m,n-1} & X_{m,n} \end{bmatrix} \rightarrow \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{m,n-1} & X_{m,n} \end{bmatrix} \tag{2.151}$$

where we have chosen *row-major* order so the first (row) index in the flattened array changes the slowest. The resulting 1D array has a corresponding covariance matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \cdots & \mathbf{C}_{1,m} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{m,1} & \cdots & \mathbf{C}_{m,m} \end{bmatrix} \tag{2.152}$$

where each element is now a submatrix

$$\mathbf{C}_{m',m''} \coloneqq \begin{bmatrix} \mathrm{Cov}\left(X_{m',1}, X_{m'',1}\right) & \cdots & \mathrm{Cov}\left(X_{m',1}, X_{m'',n}\right) \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}\left(X_{m',n}, X_{m'',1}\right) & \cdots & \mathrm{Cov}\left(X_{m',n}, X_{m'',n}\right) \end{bmatrix}. \tag{2.153}$$

We note that for submatrix $\mathbf{C}_{m',m''}$ the row indices (m', m'') remain constant while the

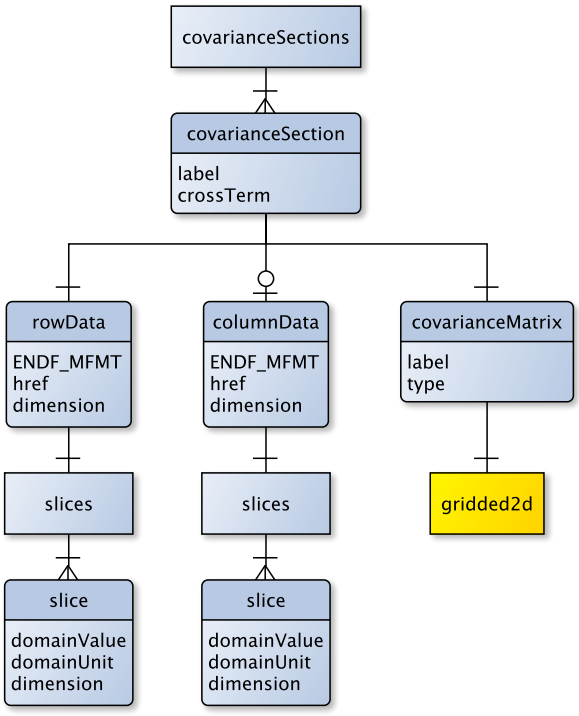


Figure 2.10: Proposed TNSL covariance GNDS hierarchy

column indices n vary.

2.7.3 Covariance of Multidimensional Data

Work was performed in developing [40] and implementing [41] an appropriate format for storing TNSL covariances using existing low-level data containers [40]. The structure of the multidimensional covariance data given by eq. (2.152) led to proposing a GNDS hierarchy given by fig. 2.10. At the top level, a `covarianceSections` node, already used to specify covariances between different observables [42], represents the collection of submatrices given by eq. (2.152). Each submatrix is represented by a `covarianceSection` given by eq. (2.153).

The `rowData` node specifies the type (`ENDF_MFMT`) of data the `covarianceMatrix` refers to along with a hyperlink (`href`) to the data for which it is specifying covariances. The `dimension` attribute corresponds to the `index` attribute inside a `grid` node. For example, if

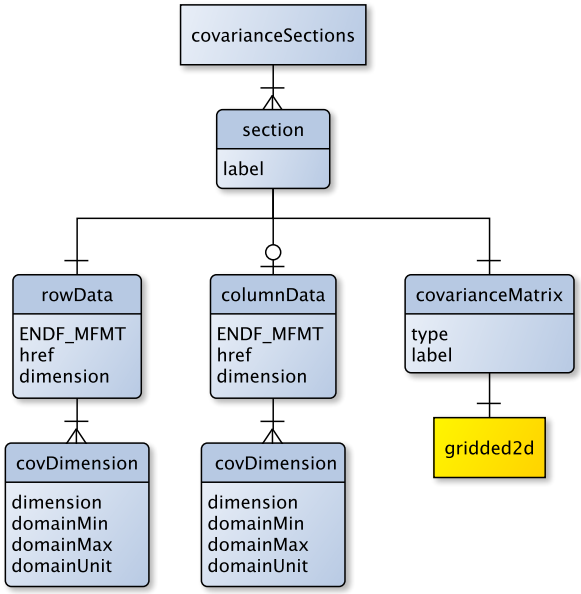


Figure 2.11: Finalized TNSL covariance GNDS hierarchy

the `href` attribute links to the `gridded2d` from fig. 2.9 and `dimension` $= 1$, then the rows of the `covarianceMatrix` are interpreted to correspond to α values because the `grid` stores α values with `index=1`.

This work proposed `slice` nodes to indicate which variables remain constant within the `covarianceMatrix`. These can be thought of as the subscripts for a submatrix in eq. (2.152). Using the example from above, setting `dimension` $= 2$ indicates that the β value remains constant throught the covariance matrix. The constant value is indicated by the `domainValue` attribute. Note that one or more `slice` objects are permitted Under the `slices` node so this scheme generalizes to data of arbitrarily many dimensions.

Finally, the `covarianceMatrix` node contains a `gridded2d` containing the actual covariances. From fig. 2.9, we see that the `gridded2d` stores elements inside an `array` node. Some basic forms of data compression like diagonal arrays, block arrays, and symmetric arrays are supported by the `array` node.

After discussion with collaborators in the international data community, the specification shown in fig. 2.11 was settled upon. Notable changes include the removal of the

`slices` node and the replacement of `slice` in favor of `covDimension` which supports a range between `domainMin` and `domainMax` rather than a single point `domainValue`. If covariances at a point are required, an evaluator can still set `domainMin` = `domainMax`. The new format, which can be used to specify covariances for multidimensional data, is expected to be included in the GNDS 2.0 specification.

Evaluations for H in H_2O

An evaluation of covariance data in the new GNDS format was created using 1000 random realizations of ENDF-6 formatted File 7 [43] TNSL data and submitted to National Nuclear Data Center (NNDC). Excerpts of the evaluation for the mean and covariance data of $S(\alpha, \beta \mid T)$ are provided in Section A.1. The new evaluations were used to test new GNDS features in the For Updating Data and Generating Evaluations (FUDGE) code.

Data Processing Codes

AMPX is a nuclear data processing code used to generate continuous-energy and multigroup cross sections used by SCALE, a modeling and simulation suite for nuclear safety analysis maintained by Reactor and Nuclear Systems Division (RNSD) at ORNL. AMPX processes ENDF-6 formatted TNSL data but also supports GNDS formatted data. The pipeline for parsing GNDS data is shown in fig. 2.12. A XML document is parsed into GNDS containers which are then used to construct GNDS data structures. The GNDS data structures are then loaded into memory in as C++ classes for use interal to AMPX.

This work implemented covariances for multidimensional data at the GNDS data structures level. Other data types, such as `ResonanceCovariance`, already implemented covariance classes as a vector of vectors. This work adapted existing data structures by creating a base class `Covariance` from which other covariance-type classes can inherit. An intermediate `SymmetricCovariance` class was added to reflect the fact that resonance covariances and $S(\alpha, \beta)$ covariances are self-covariances. Cross covariances would be handled by sep-

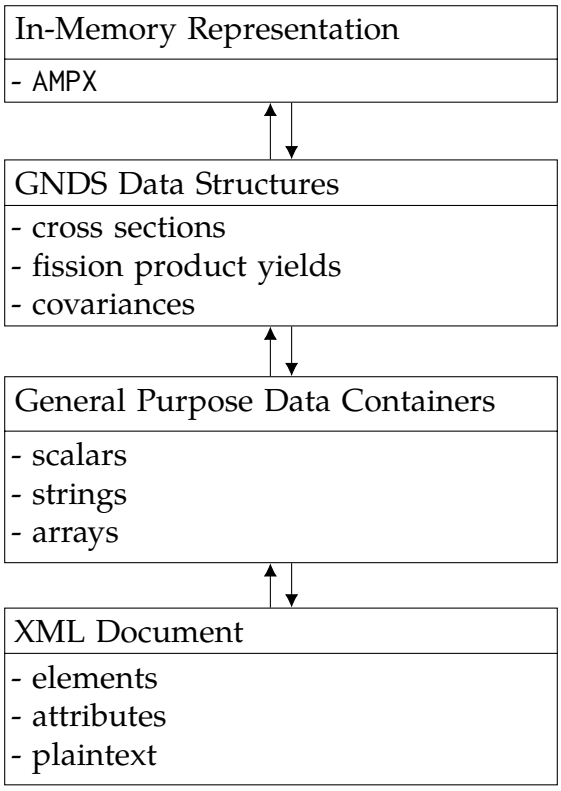


Figure 2.12: AMPX pipeline for parsing GNDS formatted data

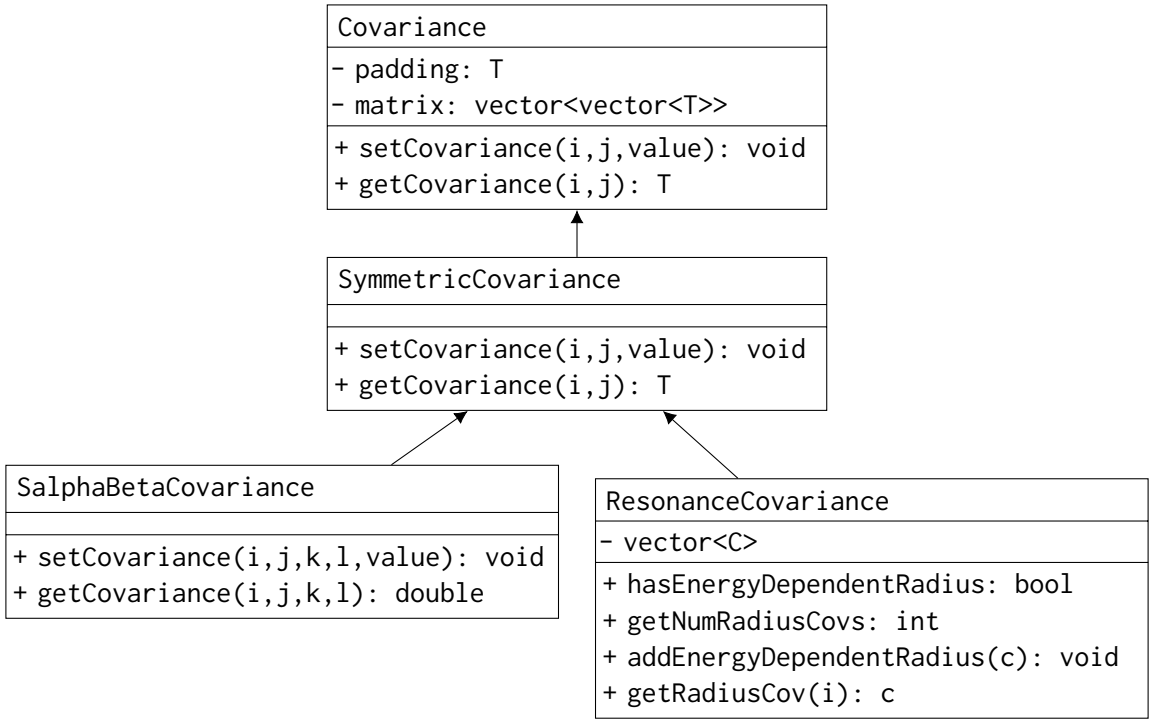


Figure 2.13: AMPX `Covariance` class inheritance

arate class for asymmetric covariances. An outline of the new class hierarchy is shown in fig. 2.13. Existing unit tests were kept intact and new ones were introduced to test the new $S(\alpha, \beta)$ covariance class. The new classes and accompanying unit tests were pushed to a feature branch in the SCALE repository.

Chapter 3

Methodology

A new scheme is proposed for storing and sampling from Thermal Neutron Scattering Law (TNSL) data. The two key components of the scheme are Proper Orthogonal Decomposition (POD) for obtaining an optimally compressed version of the original dataset suitable for direct sampling and Piecewise Cubic Hermite Interpolating Polynomials (PCHIPs) for interpolating values at any given Cumulative Distribution Function (CDF) value.

3.1 Data Structures

3.1.1 Direct Sampling

During a thermal neutron scattering event, an outgoing energy E' and direction $\hat{\mathbf{\Omega}}'$ is sampled using temperature-dependent TNSL data in the form of $S(\alpha, \beta \mid T)$. The direct sampling method outlined by Ballinger [15] accomplishes this by first performing the change of variables $(\mu, E') \rightarrow (\alpha, \beta)$ on eq. (2.112) with Jacobian determinant

$$\det \begin{bmatrix} \frac{\partial \alpha}{\partial \mu} & \frac{\partial \alpha}{\partial E'} \\ \frac{\partial \beta}{\partial \mu} & \frac{\partial \beta}{\partial E'} \end{bmatrix} = \det \begin{bmatrix} -\frac{2\sqrt{EE'}}{Ak_{\mathrm{B}}T} & \frac{1-\mu\sqrt{\frac{E}{E'}}}{Ak_{\mathrm{B}}T} \\ 0 & \frac{1}{k_{\mathrm{B}}T} \end{bmatrix} = -\frac{2\sqrt{EE'}}{A\left(k_{\mathrm{B}}T\right)^2} \tag{3.1}$$

to obtain the scattering Double Differential Cross Section (DDCS) from eq. (2.112) in terms of α and β to be

$$\sigma_{\mathrm{s}}(\alpha, \beta \mid E, T) = \frac{\sigma_{\mathrm{b}} A k_{\mathrm{B}} T}{4E} e^{-\frac{\beta}{2}} S(\alpha, \beta \mid T). \tag{3.2}$$

The total scattering cross section can be obtained by integrating over all possible values of α and β to get the total cross section for scattering

$$\sigma_{\mathrm{s}}(E, T) = \int_{\beta_{\min}}^{\beta_{\max}} \int_{\alpha_{\min}}^{\alpha_{\max}} \sigma_{\mathrm{s}}(\alpha', \beta' \mid E, T) \, d\alpha' \, d\beta'. \tag{3.3}$$

Normalizing eq. (3.2) by eq. (3.3), we can define

$$f(\alpha, \beta \mid E, T) := \frac{e^{-\frac{\beta}{2}} S(\alpha, \beta \mid T)}{\int_{\beta_{\min}}^{\beta_{\max}} \int_{\alpha_{\min}}^{\alpha_{\max}} e^{-\frac{\beta'}{2}} S(\alpha', \beta' \mid T) \, d\alpha' \, d\beta'} \tag{3.4}$$

which satisfies all the properties of a Probability Density Function (PDF). Decomposing into marginal and conditional PDFs, we have

$$f(\alpha, \beta \mid E, T) = g(\beta \mid E, T) h(\alpha \mid \beta, E, T) \tag{3.5}$$

where we first sample a β from $g(\beta \mid E, T)$ and then sample a α from $h(\alpha \mid \beta, E, T)$. The first PDF is obtained by marginalizing α from eq. (3.4) and the second PDF is obtained by solving for $h(\alpha \mid \beta, E, T)$ in eq. (3.5) then inserting eqs. (3.4) and (3.6a):

$$g(\beta \mid E, T) = \frac{e^{-\frac{\beta}{2}} \int_{\alpha_{\min}}^{\alpha_{\max}} S(\alpha', \beta \mid T) \, d\alpha'}{\int_{\beta_{\min}}^{\beta_{\max}} \int_{\alpha_{\min}}^{\alpha_{\max}} S(\alpha', \beta' \mid T) \, d\alpha' \, d\beta'} \tag{3.6a}$$

$$h(\alpha \mid \beta, E, T) = \frac{S(\alpha, \beta \mid T)}{\int_{\alpha_{\min}}^{\alpha_{\max}} S(\alpha', \beta \mid T) \, d\alpha'} \tag{3.6b}$$

Rather than storing PDFs, we store CDFs since Monte Carlo methods uniformly sample

$\xi \in [0, 1)$ which are used to look up various CDF grids [4]. For eq. (3.6), the corresponding CDFs are

$$G\left(\beta \mid E, T\right) = \frac{\int_{\beta_{\min}}^{\beta} e^{-\frac{\beta'}{2}} \int_{\alpha_{\min}}^{\alpha_{\max}} S\left(\alpha', \beta' \mid T\right) d\alpha' \, d\beta'}{\int_{\beta_{\min}}^{\beta_{\max}} \int_{\alpha_{\min}}^{\alpha_{\max}} S\left(\alpha', \beta' \mid T\right) d\alpha' \, d\beta'} \tag{3.7a}$$

$$H\left(\alpha \mid \beta, E, T\right) = \frac{\int_{\alpha_{\min}}^{\alpha} S\left(\alpha', \beta \mid T\right) d\alpha'}{\int_{\alpha_{\min}}^{\alpha_{\max}} S\left(\alpha', \beta \mid T\right) d\alpha'} \,. \tag{3.7b}$$

$G\left(\beta \mid E, T\right)$ can be stored in a 3-D array with some choice of CDF, E and T grids. Likewise, $H\left(\alpha \mid \beta, E, T\right)$ can be stored in a 4-D array with some choice of CDF, β, E, and T grids. The E dimension in $H\left(\alpha \mid \beta, E, T\right)$ can be eliminated by noting that the dependence on E in eq. (3.7b) only occurs at the limits of integration $\alpha_{\min}$ and $\alpha_{\max}$. This permits one to define

$$\hat{H}\left(\alpha \mid \beta, T\right) := \frac{\int_{0}^{\alpha} S\left(\alpha', \beta \mid T\right) d\alpha'}{\int_{0}^{\alpha_{\infty}} S\left(\alpha', \beta \mid T\right) d\alpha'} \tag{3.8}$$

where α_{∞} is some cutoff value, typically

$$\alpha_{\infty} := \max_{\beta, E, T} \alpha_{\max}\left(\beta, E, T\right) \,, \tag{3.9}$$

and then, given any E, compute the corresponding $\alpha_{\min}$ and $\alpha_{\max}$ then evaluate eq. (3.7b) on-the-fly though

$$H\left(\alpha \mid \beta, E, T\right) = \frac{\hat{H}\left(\alpha \mid \beta, T\right) - \hat{H}_{\min}}{\hat{H}_{\max} - \hat{H}_{\min}} \tag{3.10}$$

where

$$\hat{H}_{\min} := \hat{H}\left(\alpha_{\min} \mid \beta, T\right) \tag{3.11a}$$

$$\hat{H}_{\max} := \hat{H}\left(\alpha_{\max} \mid \beta, T\right) \tag{3.11b}$$

are the CDF values that would return $\alpha_{\min}$ and $\alpha_{\max}$, respectively. Although $\hat{H}$ is indepen-

dent of energy, the values for $\hat{H}_{\min}$ and $\hat{H}_{\max}$ have an implicit energy dependence through $\alpha_{\min}$ and $\alpha_{\max}$, respectively. Searching the CDF grid for $\hat{H}_{\min}$ and $\hat{H}_{\max}$ incurs some time complexity but, in return, the E dimension in the nuclear data can be eliminated. Whereas sampling $\hat{H}(\alpha \mid \beta, T)$ results in an α between 0 and α_∞, rescaling the CDF using eq. (3.10) results in an α between $\alpha_{\min}$ and $\alpha_{\max}$.

3.1.2 Continuous Temperature Dependence

The direct sampling method in Subsection 3.1.1 stores a 3-D array in $G(\beta \mid E, T)$ and another 3-D array in $\hat{H}(\alpha \mid \beta, T)$. Previous work by Pavlou and Ji [44] removed the need to store T dependence by introducing functional expansions

$$\beta(T) = \sum_n c_n^{(\beta)} f_n^{(\beta)}(T) \tag{3.12a}$$

$$\alpha(T) = \sum_n c_n^{(\alpha)} f_n^{(\alpha)}(T) \tag{3.12b}$$

so that, rather than sampling a single value of β or α, the sampling scheme returns a set of coefficients $\{c_0, c_1, ...\}$ from which a continuous temperature dependent value can be obtained using eq. (3.12). The resulting CDF datasets, $G(\beta(T) \mid E)$ and $\hat{H}(\alpha(T) \mid E)$, are still 3-D arrays but provide some opportunity for data compression. This methodology was demonstrated for hydrogen in light water and carbon in graphite. However, some limitations exist with this method:

1. The basis functions $\{f_0(T), f_1(T), ...\}$ must be empirically selected. Moreover, an appropriate basis for β may not necessarily be appropriate for α. A set of basis functions must be tailored for each new target nuclide using root-mean-square error to measure the quality of the fit.

2. Plotting β (or α) versus $G(\beta \mid E, T)$ (or $H(\alpha \mid \beta, E, T)$) should yield a monotonic increasing graph. Otherwise, a negative PDF is implied which has no physical mean-

ing. Equation (3.12) provides no guarantees of monoticity. Alternatively, one could store PDFs instead of CDFs but the problem changes to finding a basis for nonnegative functions.

3.1.3 Total Scattering Cross Section

When constructing distributions $G(\beta \mid E, T)$ and $\hat{H}(\alpha \mid \beta, T)$, information about the total scattering cross section is discarded when the normalization is performed in eq. (3.4). The total scattering cross section is given by eq. (3.3) and is clearly dependent on $S(\alpha, \beta \mid T)$. Previous work focused only on functional expansions in β and α but this work will also consider the effect of TNSL data on the total scattering cross section.

3.1.4 In-Memory Format

Energy and temperature dependent TNSL data for representing $G(\beta \mid E, T)$ can be represented as an $m \times n$ matrix

$$A = \begin{array}{c} \begin{matrix} E_1 & E_1 & \cdots & E_k & E_k \\ T_1 & T_2 & \cdots & T_{l-1} & T_l \end{matrix} \\ \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \cdots & \beta_{1,n-1} & \beta_{1,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \beta_{m,1} & \beta_{m,2} & \cdots & \beta_{m,n-1} & \beta_{m,n} \end{bmatrix} \end{array} \begin{matrix} G_1 \\ \vdots \\ G_m \end{matrix} \tag{3.13}$$

where row m' corresponds to a particular CDF value $G_{m'} \in \{G_1, \ldots, G_m\}$ and column n' corresponds to a particular pair of incident energy $E_{k'} \in \{E_1, \ldots, E_k\}$ and target temperature $T_{l'} \in \{T_1, \ldots, T_l\}$. A given k' and l' corresponds to flattened index $n' = (k' - 1)\, l + l'$. There are k incident energies and l target temperatures so there are a total of $k \times l = n$

columns. For TNSL data for representing $\hat{H}(\alpha \mid \beta, T)$, we simply make the substitutions

$$E \to |\beta| \tag{3.14a}$$

$$T \to T \tag{3.14b}$$

$$G \to \hat{H} \tag{3.14c}$$

$$\beta_{m,n} \to \alpha_{m,n} \tag{3.14d}$$

where eq. (3.14a) indicates that the incident energy has been replaced by the absolute value of the dimensionless energy transfer that was sampled, eq. (3.14b) emphasizes that the temperature grids for the β and α datasets need not be the same, eq. (3.14c) indicates that the CDF grids may also differ, and eq. (3.14d) indicates that the sampled value is now α instead of β. For representing $\sigma_s(E, T)$, we make substitutions

$$E, T \to T \tag{3.15a}$$

$$G \to E \tag{3.15b}$$

$$\beta_{m,n} \to \sigma_{sm,n} \tag{3.15c}$$

where eq. (3.15a) indicates that each column now only corresponds to temperature, eq. (3.15b) that the rows now correspond to energy, and eq. (3.15c) that the stored values correspond to scattering cross section.

To motivate the need for POD, we consider typical data storage requirements. TNSL data for H in H_2O, obtained from ENDF/B-VIII.0, is summarized in Table 3.1 and plotted at select temperatures in Figure 3.1. The full list of gridpoints used by the ENDF/B-VIII.0 evaluation is given in Section A.4. It should be noted that the ENDF/B-VIII.0 dataset was evaluated with the ENDF-6 flag `LAT=1` so that $T = 293.6\,\mathrm{K}$ is used to evaluate β and α regardless of the actual temperature. This is adjusted to the true temperature T by multiplying β or α by $T/293.6\,\mathrm{K}$.

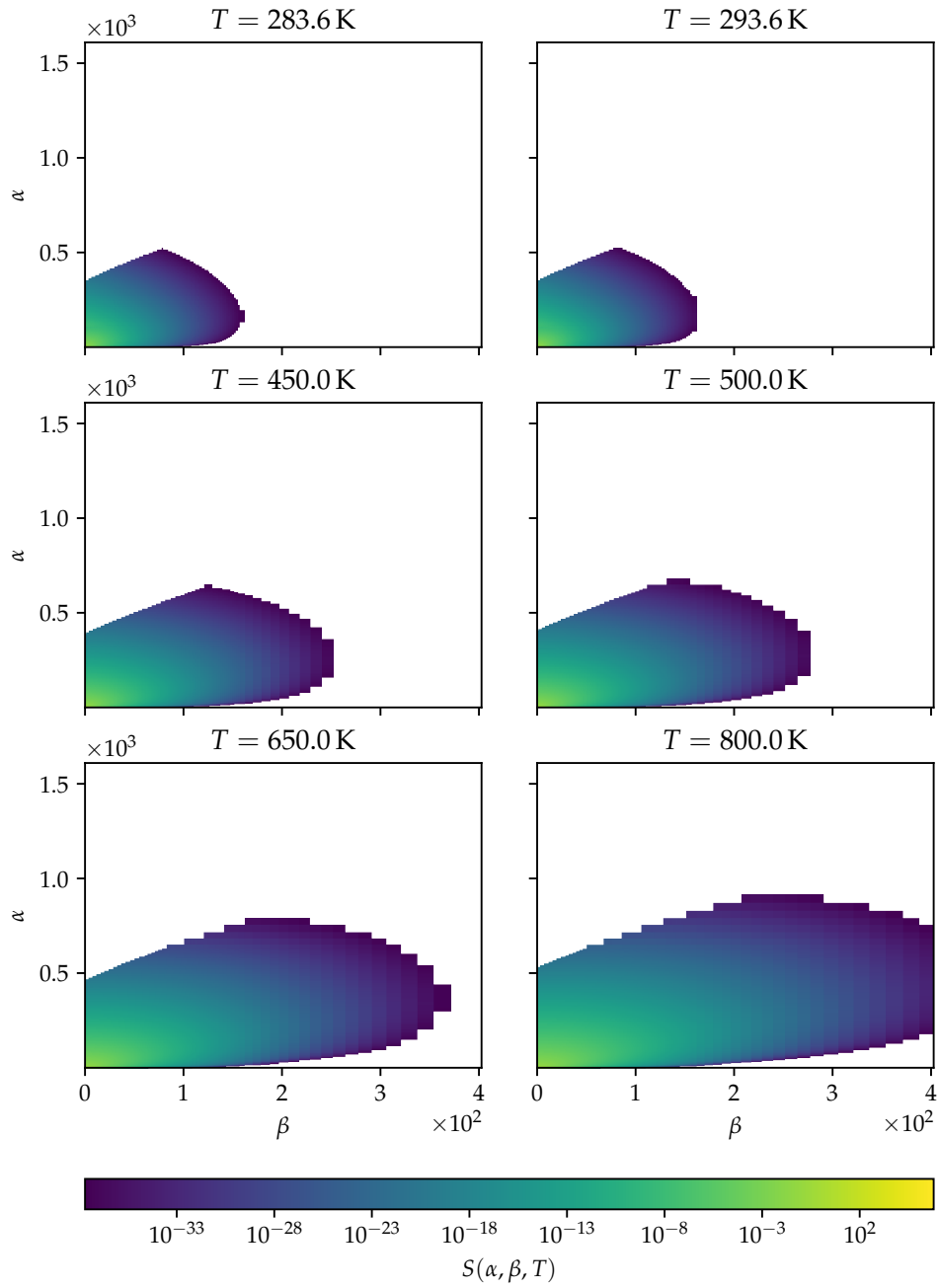


Figure 3.1: TNSL data at selected temperatures as found in ENDF/B-VIII.0

Table 3.1: Grid sizes and ranges of TNSL data for H in H_2O as found in ENDF/B-VIII.0

	β	α	T
N	317	222	18
min	0	1.002×10^{-5}	283.6 K
max	395.26	1.581×10^{3}	800.0 K

Table 3.2: Gridpoints selected in evaluation of $\sigma_s(E, T)$.

	E (eV)	T (K)
N	250	18
min	1.000×10^{-5}	283.6
max	3.949×10^{0}	800.0
spacing	logarithmic	3

Grids selected for evaluating $\sigma_s(E, T)$, $G(\beta \mid E, T)$, and $\hat{H}(\alpha \mid \beta, T)$ are detailed in Table 3.2, Table 3.3, and Table 3.4, respectively. Given the selected grids, a total of $(250 \times 18) + (100 \times 50 \times 18) + (25 \times 25 \times 18) = 105750$ numbers must be stored. If each number is stored as an 8-byte double-precision floating-point value, about 0.846 MB of memory is required. However, if uncertainty quantification is to be performed, a covariance matrix with $105750 \times (105750 - 1)/2 \approx 5.6 \times 10^{9}$ entries, occupying about 44.73 GB of memory, would be needed. While still manageable on contemporary hardware, this amount of storage only applies to one target. A particular simulation may require more than one TNSL dataset. Moreover, choosing a slightly finer grid or adding an additional parameter (such as porosity) can quickly increase the data requirements. Consider the effect of increasing the number of temperatures by a factor of 10 or, similarly, adding evaluations at 10 different porosities. The resulting covariance matrix requirements would increase by a factor of 100 so the previous example would require about 4.5 TB – an impractical amount on commodity hardware in the near future.

Table 3.3: Gridpoints selected in evaluation of $G\,(\beta \mid E, T)$.

	G	E (eV)	T (K)
N	100	50	18
min	1.001×10^{-3}	1.00×10^{-5}	283.6
max	9.990×10^{-1}	3.95×10^{0}	800.0
spacing	linear	logarithmic	3

Table 3.4: Gridpoints selected in evaluation of $\hat{H}\,(\alpha \mid \beta, T)$

	$\hat{H}$	$\lvert\beta\rvert$	T (K)
N	25	25	18
min	1.998×10^{-3}	4.137×10^{-10}	283.6
max	9.980×10^{-1}	1.615×10^{2}	800.0
spacing	linear	adaptive	3

3.1.5 Proper Orthogonal Decomposition

One method of reducing the memory burden is POD, a method that was initially employed in computational fluid dynamics [45] but has since found use for reduced-order modeling in radiation transport [46], [47]. POD decomposes eq. (3.13) by noting that a matrix $A \in \mathbb{R}^{m \times n}$ has a decomposition

$$A = U \Sigma V^T \tag{3.16}$$

where $U \in \mathbb{R}^{m \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$, $V^T \in \mathbb{R}^{r \times n}$, and $r = \text{rank}(A)$. The left and right matrices satisfy

$$U^T U = 1 \qquad \text{and} \qquad V^T V = 1 \tag{3.17}$$

and so are called *orthogonal* (*unitary* in the complex case). The middle matrix Σ is diagonal and conventionally has non-negative entries sorted in descending order: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$. The Singular Value Decomposition (SVD) has the property that choosing some $\tilde{r} < r$ and modifying Σ so that $\sigma_r = 0$ for $r > \tilde{r}$ results in

$$\tilde{A} = U \tilde{\Sigma} V^T\,, \tag{3.18}$$

an optimal rank–$\tilde{r}$ approximation of A with respect to minimizing the Frobenius norm $\|\tilde{A} - A\|_{\mathrm{F}}$. In particular,

$$\|\tilde{A} - A\|_{\mathrm{F}} = \|U\left(\tilde{\Sigma} - \Sigma\right)V^T\|_{\mathrm{F}} = \|\tilde{\Sigma} - \Sigma\|_{\mathrm{F}} = \sum_{r'>\tilde{r}} \sigma_{r'}^2\,, \tag{3.19}$$

where the fact that the Frobenius norm is invariant under unitary operations is used.

When using eq. (3.18), one can also truncate the U, Σ and V matrices, defining $\tilde{U} \in \mathbb{R}^{m\times\tilde{r}}$, $\tilde{\Sigma} \in \mathbb{R}^{\tilde{r}\times\tilde{r}}$, and $\tilde{V}^T \in \mathbb{R}^{\tilde{r}\times n}$ to obtain an equivalent $\tilde{A}$ and potentially reduce the number of elements that must be stored from $m\times n$ to only $(m+1+n)\times\tilde{r}$. After performing SVD, the matrix elements of eq. (3.13) can be reconstructed through

$$\beta_{ij} = \sum_r U_{ir}\Sigma_{rr}V_{jr} \tag{3.20}$$

as needed so that the full matrix never has to be allocated memory.

The selection of a low rank $\tilde{r} < r$ can be informed by examining the singular values and corresponding Frobenius norms from a low-order reconstruction through eq. (3.20) as detailed in fig. 3.2. One way to choose $\tilde{r}$ is by specifying a maximum allowable error $\epsilon_{\max}$ and choosing the smallest $\tilde{r}$ which satisfies

$$\epsilon_{\tilde{r}} = \frac{\|\tilde{A} - A\|_{\mathrm{F}}}{\|A\|_{\mathrm{F}}} = \frac{\sum_{r'>\tilde{r}}\sigma_{r'}^2}{\sum_{r'}\sigma_{r'}^2} \leq \epsilon_{\max}\,. \tag{3.21}$$

This work applies additional criteria to selecting $\tilde{r}$, to be discussed in Subsection 3.1.6.

Using the approximated datasets with ranks given in Table 3.5, the uncompressed covariance matrix discussed in Subsection 3.1.4 is reduced in size from 44.73 GB to 7.16 GB of memory, a reduction in size by about 84% and a reasonable size for contemporary commodity hardware.

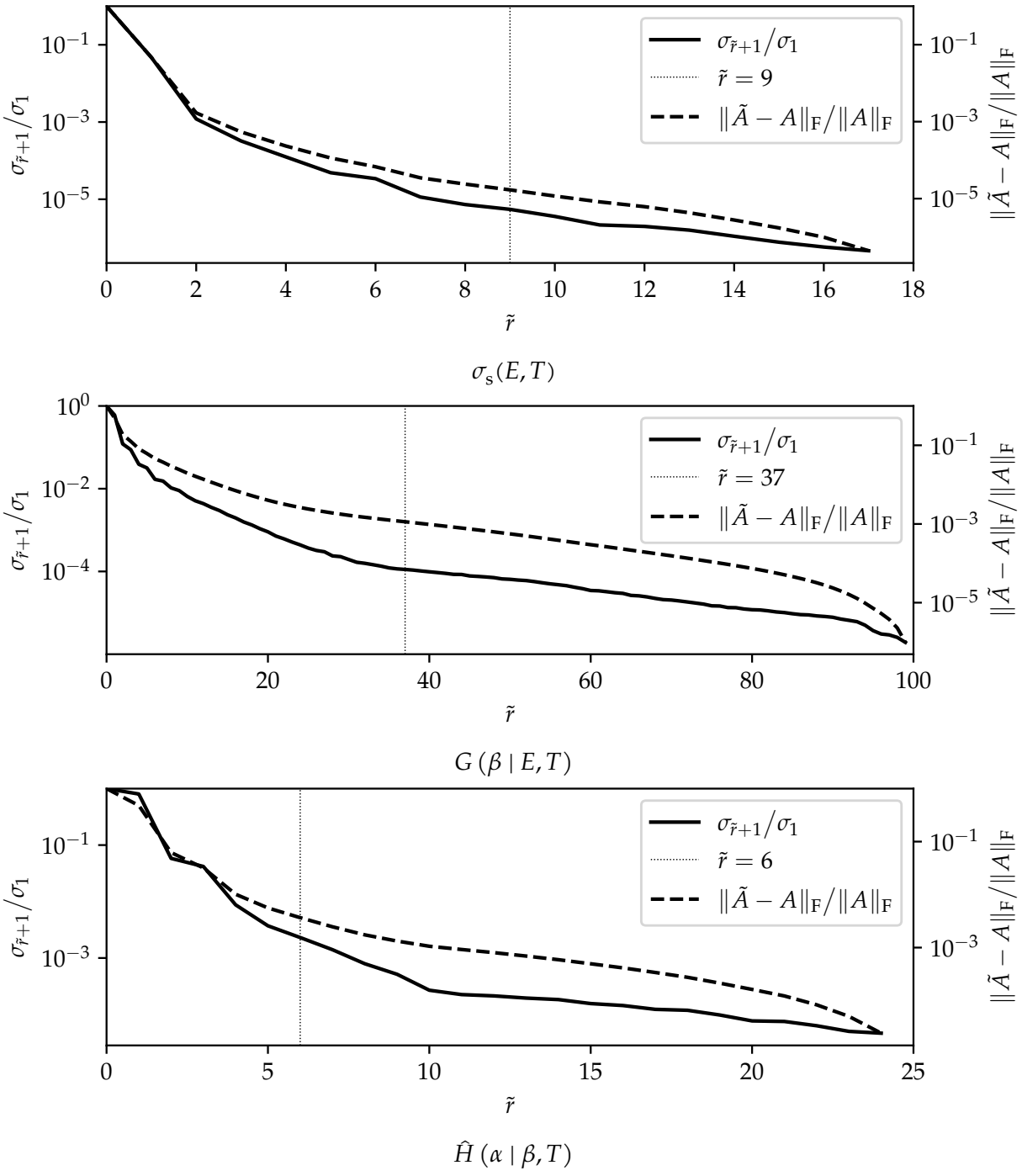


Figure 3.2: Scree plots of POD applied to each dataset. The left axes are singular values normalized to the dominant singular value. The right axes are the normalized Frobenius norms of the residual after performing a rank–$\tilde{r}$ approximation. The thin vertical lines indicate the order selected in this work. The data for $G(\beta \mid E, T)$ and $\hat{H}(\alpha \mid \beta, T)$ is stored after applying transformations discussed in Subsection 3.1.6.

Table 3.5: Elements in uncompressed matrix N, elements in compressed matrix $N_{\tilde{r}}$, compression truncation order $\tilde{r}$ and corresponding normalized Frobeinus norm of the residual.

Dataset	N	$N_{\tilde{r}}$	$\frac{N_{\tilde{r}}}{N}$	$\tilde{r}$	$\frac{\|\tilde{A}-A\|_{\mathrm{F}}}{\|A\|_{\mathrm{F}}}$
$\sigma_{\mathrm{s}}(E,T)$	4500	2421	0.5380	9	1.689×10^{-5}
$G(\beta \mid E,T)$	90000	37037	0.4115	37	1.138×10^{-3}
$\hat{H}(\alpha \mid \beta,T)$	11250	2856	0.2539	6	3.668×10^{-3}

3.1.6 Monoticity Considerations

SVD forms an optimal rank–r approximation to eq. (3.13) with respect to the Frobenius norm but it may also result in undesirable artifacts. From the definition of the CDF we require that the plot of the CDF $G(\beta \mid E,T)$ versus β or $\hat{H}(\alpha \mid \beta,T)$ versus α is always monotonically nondecreasing. A decreasing CDF corresponds to a negative PDF which has no physical interpretation. Higher-order modes have a tendency to produce nonmonotonic CDF curves, especially when the values do not change much between subsequent CDF values. Figures 3.3a and 3.4a show instances where subsequent values are all similar in magnitude so even the slightest perturbation can introduce nonmonotonic intervals. Because POD seeks to minimize the difference between the original and reconstructed matrices, transformations which increase the spacing between similar values can mitigate nonmonotonic regions. One can perform the transformations

$$\beta \to \log(\beta - \beta_{\min}) \tag{3.22a}$$

$$\alpha \to \log(\alpha), \tag{3.22b}$$

perform POD to store coefficients, and then undo the transformations at runtime. The subtraction in eq. (3.22a) is to ensure that all values in the argument of the logarithm are nonnegative. Values of α are always positive so eq. (3.22b) does not do this. One consequence of the transformations is that larger magnitude values may become less accurate

but the results in Table 3.5 show that the Frobenius norms of the residuals were below 1% for each dataset. A transformation for the $\sigma_s(E,T)$ dataset is not necessary since there are no requirements of monoticity on what is essentially a PDF.

The criteria for selecting truncation orders shown in Table 3.5 occured in two steps:

1. $\tilde{r}$ was initialized to the smallest value satisfying a maximum error tolerance, then
2. $\tilde{r}$ was increased until all entries were monotonic nondecreasing.

3.2 Cumulative Distribution Function Interpolation

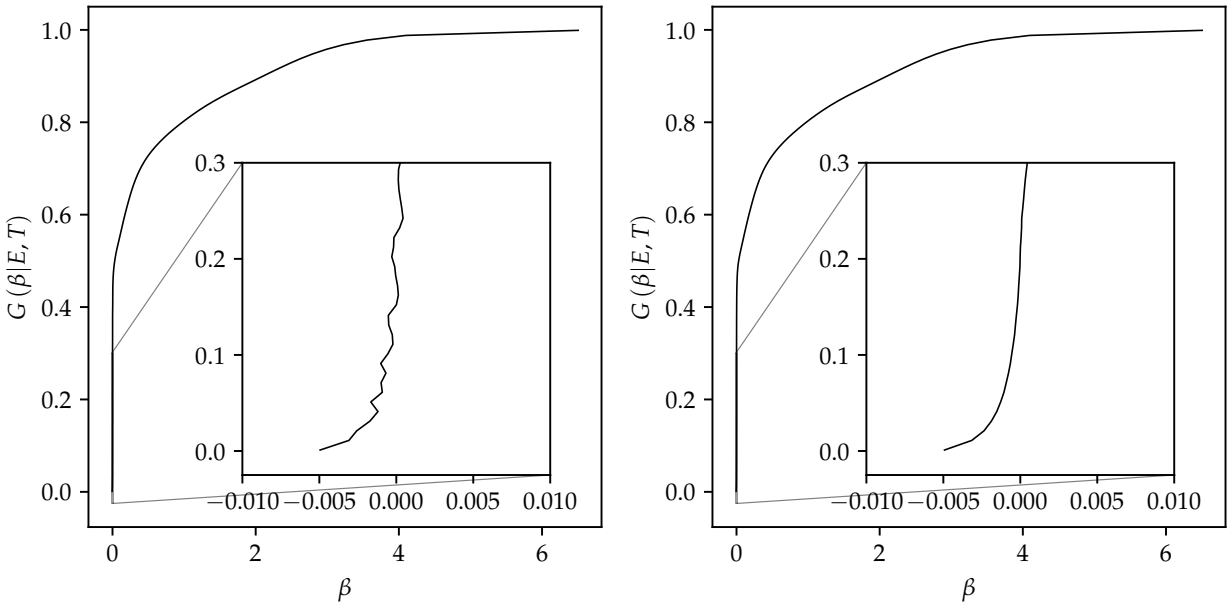
At runtime, a uniform random variable ξ is sampled and the corresponding β or α must be returned. Intermediate values must be interpolated. This is most easily done using piecewise linear interpolation in CDF but such interpolation is not suitable for sensitivity analysis using Differential Operator Sampling (DOS). Piecewise quadratic interpolation is also investigated but is deemed unsuitable because it cannot satisfy the requirement that PDFs are always nonnegative.

3.2.1 Piecewise Linear Interpolation

Piecewise linear interpolation is a natural starting point because it is simple to implement and fast to compute at runtime. Moreover, if datapoints are prepared so that the CDF is monotonic, then the interpolated curve is also monotonic. Linear interpolation was thoroughly investigated for use in CDF interpolation. Some issues were encountered and workarounds were tried, but it was ultimately abandoned in favor of higher order polynomials.

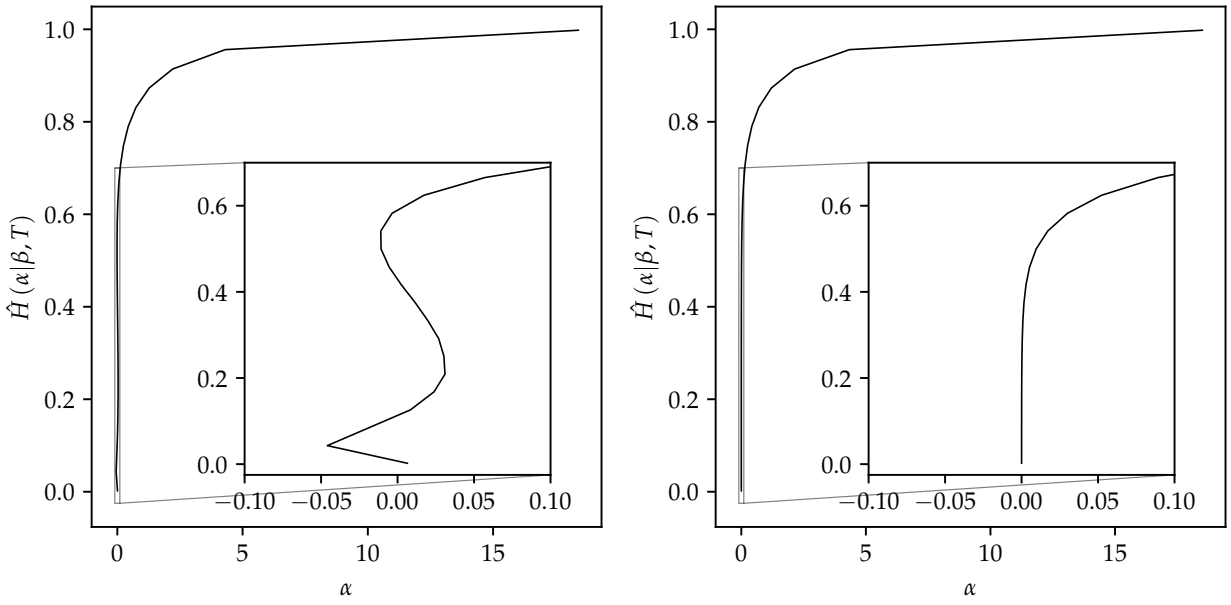
Partitioning Scheme

One drawback of linear interpolation is that the placement of CDF gridpoints can potentially skip very sharp, localized features in the CDF. One workaround to address this was



(a) Untransformed (b) Transformed

Figure 3.3: Comparison of reconstructed CDF after performing a rank–37 approximation of $G(\beta \mid E, T)$ for $E = 2.328 \times 10^{-4}\,\text{eV}$ and $T = 283.6\,\text{K}$.



(a) Untransformed (b) Transformed

Figure 3.4: Comparison of reconstructed CDF after performing a rank–6 approximation of $\hat{H}(\alpha \mid \beta, T)$ for $\beta = 4.137 \times 10^{-10}$ and $T = 473.6\,\text{K}$.

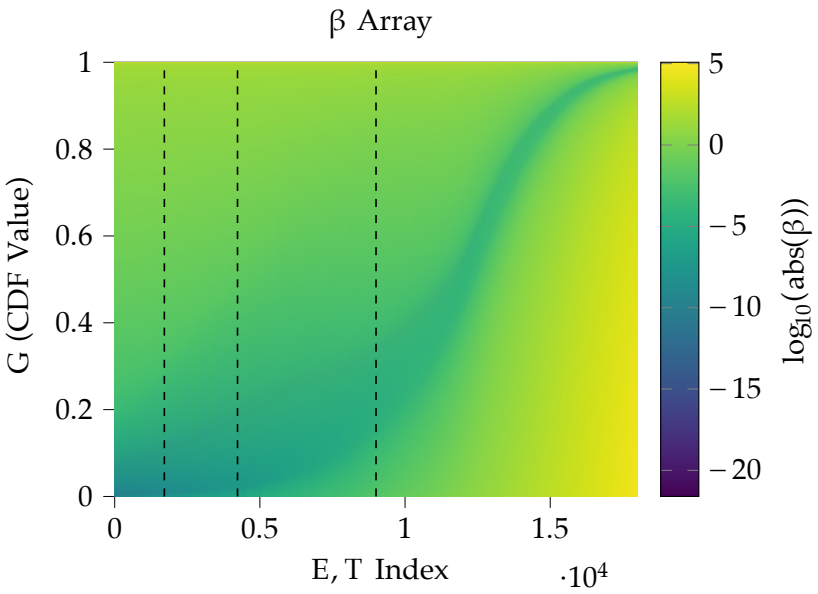


Figure 3.5: Matrix partitioning scheme for $G(\beta \mid E, T)$ dataset

Table 3.6: Gridsizes from adaptive coarsening on $G(\beta \mid E, T)$ dataset

E Boundary	CDF Gridsize		E Gridsize		T Gridsize	
	Fine	Coarse	Fine	Coarse	Fine	Coarse
1.00×10^{-5}	1000	21	95	3	18	6
3.41×10^{-5}	1000	23	140	4	18	7
2.07×10^{-4}	1000	25	265	8	18	7
6.32×10^{-3}	1000	36	500	96	18	18

a partitioning scheme [48] which split the full matrix into distinct partitions based on similar features. Figure 3.5 illustrates a matrix partitioning scheme dividing the $G(\beta \mid E, T)$ dataset into four partitions. Each partition is free to specify its own CDF grid as well as its own energy and temperature grid in the case of $G(\beta \mid E, T)$ and its own β and temperature grid in the case of $\hat{H}(\alpha \mid \beta, T)$. The coarsened gridpoints listed in Tables 3.6 and 3.7 were obtained using an adaptive coarsening scheme which exhaustively removed one CDF, energy (or β), or temperature gridpoint at a time until a relative error tolerance was exceeded.

The results in Table 3.8 show that a significant reduction in the datasets was observed while keeping relative error below 1×10^{-3}. However, large absolute errors given by ℓ_∞ in Table 3.8 shows that some local features are being missed despite their small contribution

Table 3.7: Gridsizes from adaptive coarsening on $\hat{H}(\alpha \mid \beta, T)$ dataset

β Boundary	CDF Gridsize		β Gridsize		T Gridsize	
	Fine	Coarse	Fine	Coarse	Fine	Coarse
1.00×10^{-5}	1000	12	50	4	18	5
3.41×10^{-5}	1000	20	450	7	18	7
2.07×10^{-4}	1000	26	450	33	18	14
6.32×10^{-3}	1000	28	50	12	18	11

Table 3.8: Adaptive coarsening summary

Dataset	Original (MB)	Coarse (MB)	Relative ℓ_2 Error	Absolute $\ell\infty$ Error
$G(\beta \mid E, T)$	1.44×10^{2}	2.12×10^{-1}	9.95×10^{-4}	4.52×10^{-1}
$\hat{H}(\alpha \mid \beta, T)$	1.44×10^{2}	8.77×10^{-3}	9.92×10^{-4}	9.87×10^{-1}

to the ℓ_2 error. These sharp features were generally found to correspond to the peaks such as the $\beta = 0$ elastic scattering peak in the case of the $G(\beta \mid E, T)$ dataset.

The selection of grid boundaries was performed manually. Although an algorithm could in principle be developed to check every possible partitioning scheme, a naïve greedy implementation would be exponential in complexity. One of the motivations for developing the POD methodology was to obviate the need for empirical fits as was done in functional expansion tallies so this process would ideally be automated using some suitable algorithm.

Differential Operator Sampling Incompatibility

The final and most serious drawback which made linear interpolation unsuitable for this work was its incompatibility with DOS. We present a case where a perturbation $d\lambda$ results in an indirect effect that contains a Dirac delta function at some special value λ: it has finite contribution to the integral in eq. (2.138) but almost never gets sampled because the probability of sampling exactly λ is zero.

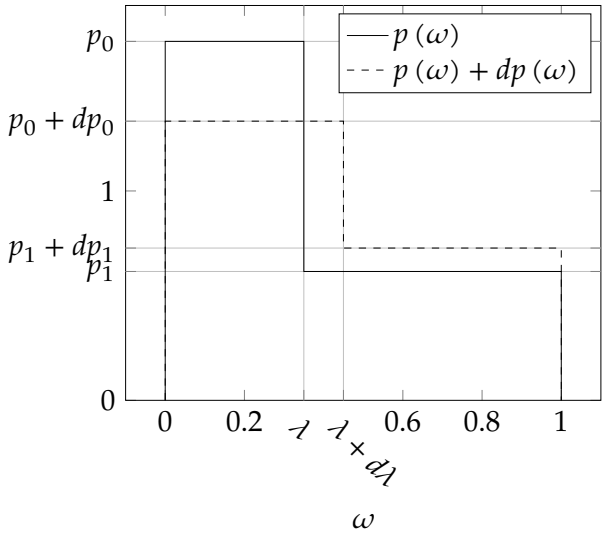


Figure 3.6: A histogram PDF characterized by $\lambda = 0.35$, and $F = 0.6$

Consider the piecewise constant (or histogram) PDF supported on $[0, 1)$ given by

$$p(\omega) = p_0 + H(\omega - \lambda)(p_1 - p_0) \tag{3.23}$$

where

- $p_0 := \frac{F}{\lambda}$ is the PDF when $0 \leq \omega < \lambda$,
- $p_1 := \frac{1-F}{1-\lambda}$ is the PDF when $\lambda \leq \omega < 1$,
- and

$$H(\omega) := \begin{cases} 0 & x < \omega \\ & \\ 1 & x \geq \omega \end{cases}, \tag{3.24}$$

  is the Heaviside step function.

A specific instance is plotted in fig. 3.6. Such a PDF arises when values of ω are given for a fixed CDF grid and intermediate values of ω are determined by linear interpolation in CDF. A key property of eq. (3.23) is that the CDF at λ, $F(\lambda)$, is preserved.

The indirect effect, given by eq. (2.140), requires the derivative of $p(\omega)$ with respect to

λ. It is given by

$$\frac{dp(\omega)}{d\lambda} = \frac{dp_0}{d\lambda} + H(\omega - \lambda)\frac{d}{d\lambda}(p_1 - p_0) + \delta(\omega - \lambda)(p_1 - p_0) . \tag{3.25}$$

For simplicity, we choose a constant response $R(\omega) = 1$ so the direct effect, given by eq. (2.139), is zero. Thus the modified response becomes

$$R(\omega) \to \frac{1}{p_0}\frac{dp_0}{d\lambda} + H(\omega - \lambda)\frac{\frac{d}{d\lambda}(p_1 - p_0)}{p_1 - p_0} + \delta(\omega - \lambda) . \tag{3.26}$$

In order to estimate $d\mathbb{E}[X]/d\lambda$, one can only obtain through Monte Carlo a finite number of samples $\omega_1, \omega_2, \ldots, \omega_N$. The probability of sampling exactly $\omega = \lambda$ is zero. However, the contribution of a response at $\omega = \lambda$ to the integral in eq. (2.137) is finite because

$$\int_{\lambda}^{\lambda+d\lambda} \delta(\omega - \lambda)\, d\omega = 1 . \tag{3.27}$$

This issue can be avoided by choosing an interpolation scheme that allows for continuous PDF so that discontinuities in PDF (and consequently Dirac deltas when computing $dp(\omega)/d\lambda$) do not appear.

3.2.2 Piecewise Quadratic Interpolation

The next-simplest interpolation scheme is piecewise quadratic interpolation in CDF. Quadratic interpolation has precedents in other transport codes. Laws 4, 44, and 61 in the ENDF-6 format, for example, use quadratic interpolation to represent data such as outgoing energy distributions for inelastic scattering. Unlike most implementations which typically result in discontinuous PDFs, this work uses the extra degree of freedom to match adjacent PDFs. This addresses the issue encountered with DOS when piecewise linear interpolation was used. However, it will be shown that this scheme comes with its own set of pitfalls.

We begin by deriving the construction of piecewise quadratic interpolants which re-

sult in piecewise continuous (C^0 smooth) PDFs. Given a set of value and CDF pairs $((x_0, F_0), \ldots, (x_M, F_M))$, we seek to represent CDFs as a piecewise function

$$F(x) = \begin{cases} F^{(1)}(x), & x \in [x_0, x_1) \\ F^{(2)}(x), & x \in [x_1, x_2) \\ \vdots & \\ F^{(M)}(x), & x \in [x_{M-1}, x_M) \end{cases} \tag{3.28}$$

which is differentiable everywhere. Defining the derivative

$$f(x) \coloneqq \frac{dF}{dx}, \tag{3.29}$$

each quadratic curve $F^{(m)}(x)$ must satisfy

$$F_{m-1} = F^{(m)}(x_{m-1}) \tag{3.30a}$$

$$F_m = F^{(m)}(x_m) \tag{3.30b}$$

$$f_{m-1} = f^{(m)}(x_{m-1}). \tag{3.30c}$$

Equations (3.30a) and (3.30b) ensure that adjacent curves are (C^0) smooth by requiring that adjacent curves $F^{(m)}(x)$ and $F^{(m+1)}(x)$ take on the same value F_m at x_m. Equation (3.30c) can be used to ensure that adjacent curves are (C^1) differentiable by setting f_{m-1} to be equal to $f^{(m-1)}(x_{m-1})$, the derivative of the previous curve at x_{m-1}.

A quadratic polynomial in CDF is the lowest order polynomial that can always satisfy eq. (3.30). The corresponding PDF is linear so the difference between CDFs can be obtained exactly through the trapezoid rule

$$F_m - F_{m-1} = \frac{1}{2}(f_m + f_{m-1})(x_m - x_{m-1}) \tag{3.31}$$

which can be rearranged to obtain the recurrence relation

$$f_m = 2\frac{F_m - F_{m-1}}{x_m - x_{m-1}} - f_{m-1}\,. \tag{3.32}$$

Thus, once f_0 is specified, all f_m can be determined. Defining

$$c_m := \begin{cases} f_0, & m = 0 \\ 2\frac{F_m - F_{m-1}}{x_m - x_{m-1}}, & m = 1, 2, \ldots, M \end{cases}, \tag{3.33}$$

we have

$$f_m = (-1)^m \sum_{m'=0}^{m} (-1)^{m'} c_{m'}\,. \tag{3.34}$$

Before seeking coefficients for each quadratic polynomial, we introduce a variable transformation in terms of a local interpolation factor

$$r^{(m)}(x) := \frac{x - x_{m-1}}{x_m - x_{m-1}} \tag{3.35a}$$

$$\tilde{F}^{(m)}\left(r^{(m)}\right) = F^{(m)}(x) \tag{3.35b}$$

$$\tilde{f}^{(m)}\left(r^{(m)}\right) = f^{(m)}(x)\,(x_m - x_{m-1}) \tag{3.35c}$$

to simplify the arithmetic. This also has the benefit of transforming to more "natural" length scales which mitigates against arithmetic underflow or overflow from some particular choice of units. Under variable transformation eq. (3.35), the constraints in eq. (3.30) become

$$F_{m-1} = \tilde{F}^{(m)}(0) \tag{3.36a}$$

$$F_m = \tilde{F}^{(m)}(1) \tag{3.36b}$$

$$f_{m-1}\,(x_m - x_{m-1}) = \tilde{f}^{(m)}(0)\,. \tag{3.36c}$$

We now seek expressions to identify $\tilde{a}^{(m)}$, $\tilde{b}^{(m)}$, and $\tilde{c}^{(m)}$ in

$$\tilde{F}^{(m)}\left(r^{(m)}\right) = \tilde{a}^{(m)}\left(r^{(m)}\right)^2 + \tilde{b}^{(m)} r^{(m)} + \tilde{c}^{(m)} \tag{3.37a}$$

$$\tilde{f}^{(m)}\left(r^{(m)}\right) = 2\tilde{a}^{(m)} r^{(m)} + \tilde{b}^{(m)} \tag{3.37b}$$

by evaluating eq. (3.37) at the appropriate $r^{(m)}$ specified in eq. (3.36) to get

$$F_{m-1} = \tilde{c}^{(m)} \tag{3.38a}$$

$$F_m = \tilde{a}^{(m)} + \tilde{b}^{(m)} + \tilde{c}^{(m)} \tag{3.38b}$$

$$f_{m-1}\left(x_m - x_{m-1}\right) = \tilde{b}^{(m)} \tag{3.38c}$$

which is easily solved to obtain

$$\tilde{a}^{(m)} = \frac{1}{2}\left(f_m - f_{m-1}\right)\left(x_m - x_{m-1}\right) \tag{3.39a}$$

$$\tilde{b}^{(m)} = f_{m-1}\left(x_m - x_{m-1}\right) \tag{3.39b}$$

$$\tilde{c}^{(m)} = F_{m-1} \tag{3.39c}$$

where eq. (3.31) was used in eq. (3.39a). Thus, eq. (3.37a) becomes

$$\begin{aligned}\tilde{F}^{(m)}\left(r^{(m)}\right) = \frac{1}{2}\left(f_m - f_{m-1}\right)\left(x_m - x_{m-1}\right)\left(r^{(m)}\right)^2 \\ + f_{m-1}\left(x_m - x_{m-1}\right) r^{(m)} \\ + F_{m-1}\,.\end{aligned} \tag{3.40}$$

Transforming back to x using eq. (3.35a), eq. (3.40) becomes

$$\begin{aligned} F^{(m)}(x) = &\frac{1}{2}\left(f_m - f_{m-1}\right)\left(x_m - x_{m-1}\right)\left(\frac{x - x_{m-1}}{x_m - x_{m-1}}\right)^2 \\ &+ f_{m-1}\left(x_m - x_{m-1}\right)\left(\frac{x - x_{m-1}}{x_m - x_{m-1}}\right) \\ &+ F_{m-1}\,. \end{aligned} \tag{3.41}$$

Collecting powers of x, we get

$$\begin{aligned} F^{(m)}(x) = &\left(\frac{1}{2}\frac{f_m - f_{m-1}}{x_m - x_{m-1}}\right)x^2 \\ &+ \left(-\frac{f_m - f_{m-1}}{x_m - x_{m-1}}x_{m-1} + f_{m-1}\right)x \\ &+ \left(\frac{1}{2}\frac{f_m - f_{m-1}}{x_m - x_{m-1}}x_{m-1}^2 - f_{m-1}x_{m-1} + F_{m-1}\right). \end{aligned} \tag{3.42}$$

which is quadratic in x. From eq. (3.29), we see that differentiating eq. (3.42) gives

$$\begin{aligned} f^{(m)}(x) &= 2\left(\frac{1}{2}\frac{f_m - f_{m-1}}{x_m - x_{m-1}}\right)x + \left(-\frac{f_m - f_{m-1}}{x_m - x_{m-1}}x_{m-1} + f_{m-1}\right) \\ &= f_{m-1} + \left(\frac{f_m - f_{m-1}}{x_m - x_{m-1}}\right)\left(x - x_{m-1}\right)\,, \end{aligned} \tag{3.43}$$

a PDF which is linear in x.

The choice of f_0 has a significant impact on the quality of fit as it is the initial value of the recurrence relation in eq. (3.32). We show that there exists a value of the first PDF which minimizes oscillations. For the m^{th} curve, the effect can be quantified by computing the area between $F^{(m)}(x)$, a quadratic interpolant given by eq. (3.42), and

$$G^{(m)}(x) = \left(\frac{F_m - F_{m-1}}{x_m - x_{m-1}}\right)\left(x - x_{m-1}\right) + F_{m-1}\,, \tag{3.44}$$

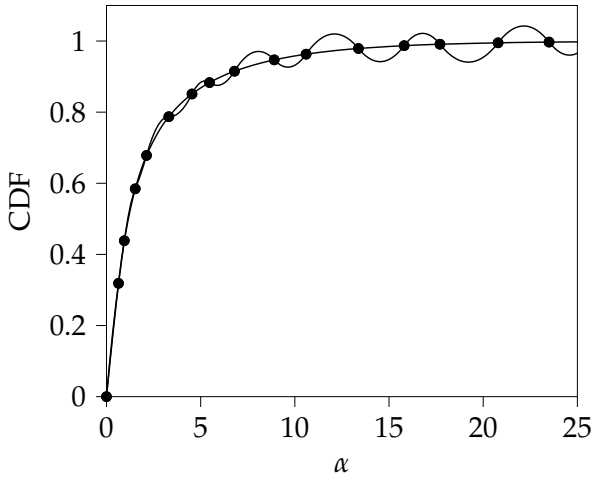


Figure 3.7: Quadratic and linear interpolation of α and its CDF

the linear interpolant, by evaluating

$$A^{(m)} := \int_{x_{m-1}}^{x_m} F^{(m)}(x) - G^{(m)}(x)\, dx\,. \tag{3.45}$$

A piecewise linear interpolation and a piecewise quadratic interpolation with a poor choice of f_0 is shown in fig. 3.7 to illustrate the area between the two interpolation methods. Again using local interpolation factors eq. (3.35) and the trapezoid rule eq. (3.31), eq. (3.44) becomes

$$\tilde{G}^{(m)}\left(r^{(m)}\right) = \frac{1}{2}\left(f_m + f_{m-1}\right)\left(x_m - x_{m-1}\right) r^{(m)} + F_{m-1} \tag{3.46}$$

and eq. (3.45) becomes

$$A^{(m)} = \left(x_m - x_{m-1}\right) \int_0^1 \tilde{F}^{(m)}(r) - \tilde{G}^{(m)}(r)\, dr\,. \tag{3.47}$$

Inserting eq. (3.40) and eq. (3.46) into eq. (3.47) and simplifying gives

$$A^{(m)} = \left(x_m - x_{m-1}\right)\left\{\frac{1}{2}\left(f_m - f_{m-1}\right)\left(x_m - x_{m-1}\right)\right\} \int_0^1 \left(r^2 - r\right) dr\,. \tag{3.48}$$

Note that the term in $\{\cdot\}$ happens to be $\tilde{a}^{(m)}$ from eq. (3.39a), the quadratic coefficient of

$\tilde{F}^{(m)}\left(r^{(m)}\right)$. This is an eminently sensible result since setting $\tilde{a}^{(m)}$ to zero results in linear interpolation so $\tilde{F}$ coincides with $\tilde{G}$, and the area between the two curves, $A^{(m)}$, becomes zero. The integral in eq. (3.48) is simply $-1/6$ so we have

$$A^{(m)} = -\frac{1}{12}\left(f_m - f_{m-1}\right)\left(x_m - x_{m-1}\right)^2 . \tag{3.49}$$

We can now define

$$L\left(f_0\right) := \sum_{m'=1}^{M}\left(A^{(m')}\right)^2 \tag{3.50}$$

that quantitatively measures how much the quadratic fit oscillates about the linear fit. Squaring each term ensures that areas are nonnegative. Inserting eq. (3.49) into eq. (3.50) and differentiating, we have

$$\frac{dL}{df_0} = 2\left(-\frac{1}{12}\right)^2 \sum_{m'=1}^{M}\left(f_{m'} - f_{m'-1}\right)\left(x_{m'} - x_{m'-1}\right)^4 \frac{d}{df_0}\left(f_{m'} - f_{m'-1}\right) . \tag{3.51}$$

The optimal value of f_0 is a stationary point $dL/df_0 = 0$ so we have

$$0 = \sum_{m'=1}^{M}\left(f_{m'} - f_{m'-1}\right)\left(x_{m'} - x_{m'-1}\right)^4 \frac{d}{df_0}\left(f_{m'} - f_{m'-1}\right) . \tag{3.52}$$

The derivative inside the sum can be evaluated by differentiating eq. (3.34) to get

$$\frac{df_{m'}}{df_0} = (-1)^{m'} \sum_{m''=0}^{m'} (-1)^{m''} \frac{dc_{m''}}{df_0} \tag{3.53}$$

then noting from eq. (3.33) that the only nonzero term in the sum is at $m'' = 0$ when $dc_0/df_0 = 1$ which gives

$$\frac{df_{m'}}{df_0} = (-1)^{m'} \tag{3.54}$$

so the last term in eq. (3.52) is just $2\,(-1)^{m'}$. The leading 2 can be divided out so, eq. (3.52)

becomes

$$0 = \sum_{m'=1}^{M} (-1)^{m'} \left(f_{m'} - f_{m'-1}\right) \left(x_{m'} - x_{m'-1}\right)^4 . \tag{3.55}$$

We now turn our attention to $f_{m'} - f_{m'-1}$. Using eq. (3.34), we get

$$\begin{aligned}
f_{m'} - f_{m'-1} &= \left((-1)^{m'} \sum_{m''=0}^{m'} (-1)^{m''} c_{m''} \right) - \left((-1)^{m'-1} \sum_{m''=0}^{m'-1} (-1)^{m''} c_{m''} \right) \\
&= \left((-1)^{m'} \sum_{m''=0}^{m'} (-1)^{m''} c_{m''} \right) + \left((-1)^{m'} \sum_{m''=0}^{m'-1} (-1)^{m''} c_{m''} \right) \\
&= \left((-1)^{m'} \sum_{m''=0}^{m'} (-1)^{m''} c_{m''} \right) + \left((-1)^{m'} \sum_{m''=0}^{m'} (-1)^{m''} c_{m''} \right) - c_{m'} \\
&= 2 \left((-1)^{m'} \sum_{m''=0}^{m'} (-1)^{m''} c_{m''} \right) - c_{m'} \\
&= 2 (-1)^{m'} f_0 + 2 \left((-1)^{m'} \sum_{m''=1}^{m'} (-1)^{m''} c_{m''} \right) - c_{m'}
\end{aligned} \tag{3.56}$$

so eq. (3.55) becomes

$$\begin{aligned}
0 &= \sum_{m'=1}^{M} (-1)^{m'} \left(2 (-1)^{m'} f_0 + 2 \left((-1)^{m'} \sum_{m''=1}^{m'} (-1)^{m''} c_{m''} \right) - c_{m'} \right) \left(x_{m'} - x_{m'-1}\right)^4 \\
&= \sum_{m'=1}^{M} \left(2 f_0 + 2 \left(\sum_{m''=1}^{m'} (-1)^{m''} c_{m''} \right) - (-1)^{m'} c_{m'} \right) \left(x_{m'} - x_{m'-1}\right)^4
\end{aligned} \tag{3.57}$$

which can be solved for f_0 to get

$$f_0 = - \frac{\sum_{m'=1}^{M} \left(2 \left(\sum_{m''=1}^{m'} (-1)^{m''} c_{m''} \right) - (-1)^{m'} c_{m'} \right) \left(x_{m'} - x_{m'-1}\right)^4}{2 \sum_{m'=1}^{M} \left(x_{m'} - x_{m'-1}\right)^4} \tag{3.58}$$

Equation (3.58) is optimal in the sense that it minimizes eq. (3.50), the area between the piecewise quadratic and the piecewise linear interpolations. From eq. (3.30), it also guarantees that the corresponding PDF is C^0 smooth so it does not suffer from the pitfalls discussed in Subsection 3.2.1. However, this approach does not have enough degrees of

freedom to also satisfy the requirement that the CDFs be monotonic. Monotonic curves can be obtained, but requires manual selection of grids.

Finally, analytic expressions for the sampling scheme, discussed in Subsection A.3.1, are quite complicated to evaluate because every CDF gridpoint is related to every other CDF gridpoint through the recurrence relation eq. (3.32). For these reasons, one more degree of freedom was added to arrive at piecewise cubic interpolation.

3.2.3 Piecewise Cubic Interpolation

With simpler options exhausted, we turn to cubic interpolation. Cubic polynomials are commonly used in Bernstein-Bézier curves [49] and splines [50] because in addition to specifying endpoints, one has enough degrees of freedom to specify the derivative at both endpoints. Moreover, there exist methods for guaranteeing that interpolated splines are monotonic.

As was done in the quadratic case, we seek a piecewise function

$$F(x) = \begin{cases} F^{(1)}(x), & x \in [x_0, x_1) \\ F^{(2)}(x), & x \in [x_1, x_2) \\ & \vdots \\ F^{(M)}(x), & x \in [x_{M-1}, x_M) \end{cases} \tag{3.59}$$

which is differentiable everywhere. Unlike the quadratic case, we have enough degrees of freedom this time to specify the derivative at each point where the each cubic curve meets.

Thus the constraints become

$$F_{m-1} = F^{(m)}\left(x_{m-1}\right) \tag{3.60a}$$

$$F_m = F^{(m)}\left(x_m\right) \tag{3.60b}$$

$$f_{m-1} = f^{(m)}\left(x_{m-1}\right) \tag{3.60c}$$

$$f_m = f^{(m)}\left(x_m\right) . \tag{3.60d}$$

Whereas the quadratic constraints eq. (3.30) uniquely determined the PDF at x_m, the additional degree of freedom eq. (3.60d) grants the cubic polynomial the freedom to choose its derivative on the right endpoint.

In terms of a local interpolation factor

$$r^{(m)}(x) := \frac{x - x_{m-1}}{x_m - x_{m-1}} \tag{3.61}$$

a cubic Hermite polynomial satisfying eq. (3.60) is given by

$$\begin{aligned} F^{(m)}\left(r^{(m)}\right) = \quad & F_{m-1} h_{00}\left(r^{(m)}\right) \\ & + f_{m-1}\left(x_m - x_{m-1}\right) h_{10}\left(r^{(m)}\right) \\ & + F_m h_{01}\left(r^{(m)}\right) \\ & + f_m\left(x_m - x_{m-1}\right) h_{11}\left(r^{(m)}\right) . \end{aligned} \tag{3.62}$$

where

$$h_{00}(t) = 2t^3 - 3t^2 + 1 \tag{3.63a}$$

$$h_{10}(t) = t^3 - 2t^2 + t \tag{3.63b}$$

$$h_{01}(t) = -2t^3 + 3t^2 \tag{3.63c}$$

$$h_{11}(t) = t^3 - t^2 \tag{3.63d}$$

form the basis functions for a cubic Hermite spline. Collecting terms with powers of $r^{(m)}$ we can alternatively express

$$\begin{aligned} F^{(m)}\left(r^{(m)}\right) = \quad & \left(r^{(m)}\right)^3 \left(2F_{m-1} + f_{m-1}\left(x_m - x_{m-1}\right) - 2F_m + f_m\left(x_m - x_{m-1}\right)\right) \\ & + \left(r^{(m)}\right)^2 \left(-3F_{m-1} - 2f_{m-1}\left(x_m - x_{m-1}\right) + 3F_m - f_m\left(x_m - x_{m-1}\right)\right) \\ & + r^{(m)}\left(f_{m-1}\left(x_m - x_{m-1}\right)\right) \\ & + F_{m-1} \end{aligned} \tag{3.64}$$

which is used in practice.

Unlike the case of quadratic interpolation, there is no longer a recursion relation between subsequent PDFs. Rather, each f_m can be freely specified. Work by Fritsch and Butland [51] provide an algorithm for constructing monotone piecewise cubic interpolants. At a point where two interpolants meet, there are two secant lines

$$s_m := \frac{F_m - F_{m-1}}{x_m - x_{m-1}} \tag{3.65a}$$

$$s_{m+1} := \frac{F_{m+1} - F_m}{x_{m+1} - x_m} \tag{3.65b}$$

which correspond to the average derivatives of interval m and $m+1$, respectively. Initial work by Butland [52] identified the harmonic mean

$$f\left(s_m, s_{m+1}\right) = \begin{cases} \frac{2s_m s_{m+1}}{s_m + s_{m+1}} & s_m s_{m+1} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.66}$$

as a suitable choice for f_m. Qualitatively smoother plots [51] can be obtained by using

$$f(s_m, s_{m+1}) = \begin{cases} 0 & s_m s_{m+1} \leq 0 \\ \text{sgn}(s_m) \frac{m s_m s_{m+1}}{s_m + (m-1)s_{m+1}} & |s_m| \geq |s_{m+1}| \\ \frac{2 s_m s_{m+1}}{s_m + s_{m+1}} & \text{otherwise} \end{cases} \tag{3.67}$$

where $m = 2$ recovers Butland's formula and $m = 3$ was found to produce "smoother" plots. The above aproaches are desirable in that they are "local": Perturbations in f_m only affect points between x_{m-1} and x_{m+1}

One key disadvantage of using cubic polynomials is that inverse sampling no longer has a simple formula for finding roots. While there exists a closed form solution for cubic polynomial roots, it involves expensive cubic root operations for which direct implementations may lead to significant precision loss. Work by Yuskel [53] found that root finding for cubics can be done in a performant and numerically stable manner provided some conditions are satisfied. Recalling that care was taken to ensure that the data was monotonic nondecreasing in Subsection 3.1.6, we know that each uniform random value ξ must map to a unique interval m. Moreover, we know from the intermediate value theorem that a single real root must exist between $m-1$ and m. Newton iterations take the form

$$x^{(i)} = x^{(i-1)} - \frac{F\left(x^{(i-1)}\right)}{\frac{dF}{dx}\left(x^{(i-1)}\right)} \tag{3.68}$$

where the derivatives are obtained by differentiating eq. (3.64) to get a function which is quadratic in x. Convergence issues for cubics can be avoided by forcing the next guess to remain in the interval $[x_{m-1}, x_m]$ by performing

$$x^{(i)} \rightarrow \max\left(x_{\min}, \min\left(x_{\max}, x^{(i)}\right)\right) \tag{3.69}$$

after each iteration. There is one special case [53]

$$\frac{F\left(x^{(i)}\right)}{\frac{dF}{dx}\left(x^{(i)}\right)} = -\frac{F\left(x^{(i-1)}\right)}{\frac{dF}{dx}\left(x^{(i-1)}\right)}, \tag{3.70}$$

which can cause an infinite loop. However, the CDF data is prepared so that $F \geq 0$ and $dF/dx \geq 0$ are always true so this case is never encountered.

3.3 Sampling σ_s, β, and α

This section discusses how σ_s, β and α are reconstructed then sampled using $\sigma_s(E,T)$, $G(\beta \mid E,T)$, and $\hat{H}(\alpha \mid \beta,T)$ datasets at runtime, respectively. The process is nearly identical between σ_s, β, and α so this section will only discuss sampling β in detail but eq. (3.14) can be applied to obtain the process for α. The key difference when sampling α is that the CDF must be rescaled, a process which is given by eq. (3.10) and discussed in Subsection 3.1.1

We begin with an incident neutron with energy E scattering off a moderator at temperature T. A uniform random variable ξ is sampled and we must return the corresponding value of β from $G(\beta \mid E,T)$. A matrix representing $G(\beta \mid E,T)$ has an E grid $\{E_1,\ldots,E_k\}$, temperature grid $\{T_1,\ldots,T_l\}$, and CDF grid $\{\hat{H}_1,\ldots,\hat{H}_m\}$. In each grid, elements are sorted in strictly increasing order so one can identify indices k', l', and m' satisfying

$$E_{k'-1} \leq E < E_{k'} \tag{3.71a}$$

$$T_{l'-1} \leq T < T_{l'} \tag{3.71b}$$

$$G_{m'-1} \leq \xi < G_{m'} \tag{3.71c}$$

from which interpolation can be performed in G then T. No interpolation is performed in E since the procedure for sampling β follows the MCNP® implementation of ENDF-6 Law

4. Instead, an E interpolation factor

$$r_E = \frac{E - E_{k'-1}}{E_{k'} - E_{k'-1}} \in [0, 1) \tag{3.72}$$

is used to sample $\tilde{k}$ with Probability Mass Function (PMF)

$$p\left(\tilde{k}\right) = \begin{cases} r_E & \tilde{k} = k' \\ 1 - r_E & \tilde{k} = k' - 1 \end{cases} . \tag{3.73}$$

Defining the flattened index

$$n' := \left(\tilde{k} - 1\right) l + l' , \tag{3.74}$$

the eight points to be used in the interpolation scheme are depicted in fig. 3.8. Equation (3.20) is used to reconstruct each β from POD coefficients at the eight points. Linear interpolation in temperature is then performed to get four points at the requested temperature T. Finally, cubic interpolation is performed on the four points at T using PCHIP and Newton iteration (discussed in Subsection 3.2.3) to interpolate in CDF. A cartoon of this process is shown in fig. 3.9 Notably, this entire process does not require decompressing the entire POD-compressed dataset into memory. Only the coefficients which are required to reconstruct the requested β value are used.

3.3.1 Indirect Effects

The data structures introduced in this work store evaluated nuclear data for which sensitivities must be calculated. The indirect effect of perturbations in the $\sigma_s(E, T)$, $G(\beta \mid E, T)$, and $H(\alpha \mid \beta, E, T)$ datasets are propagated using DOS and Automatic Differentiation (AD) to estimated quantities. This section examines the indirect effects on each transport kernel which was introduced in eq. (2.37)

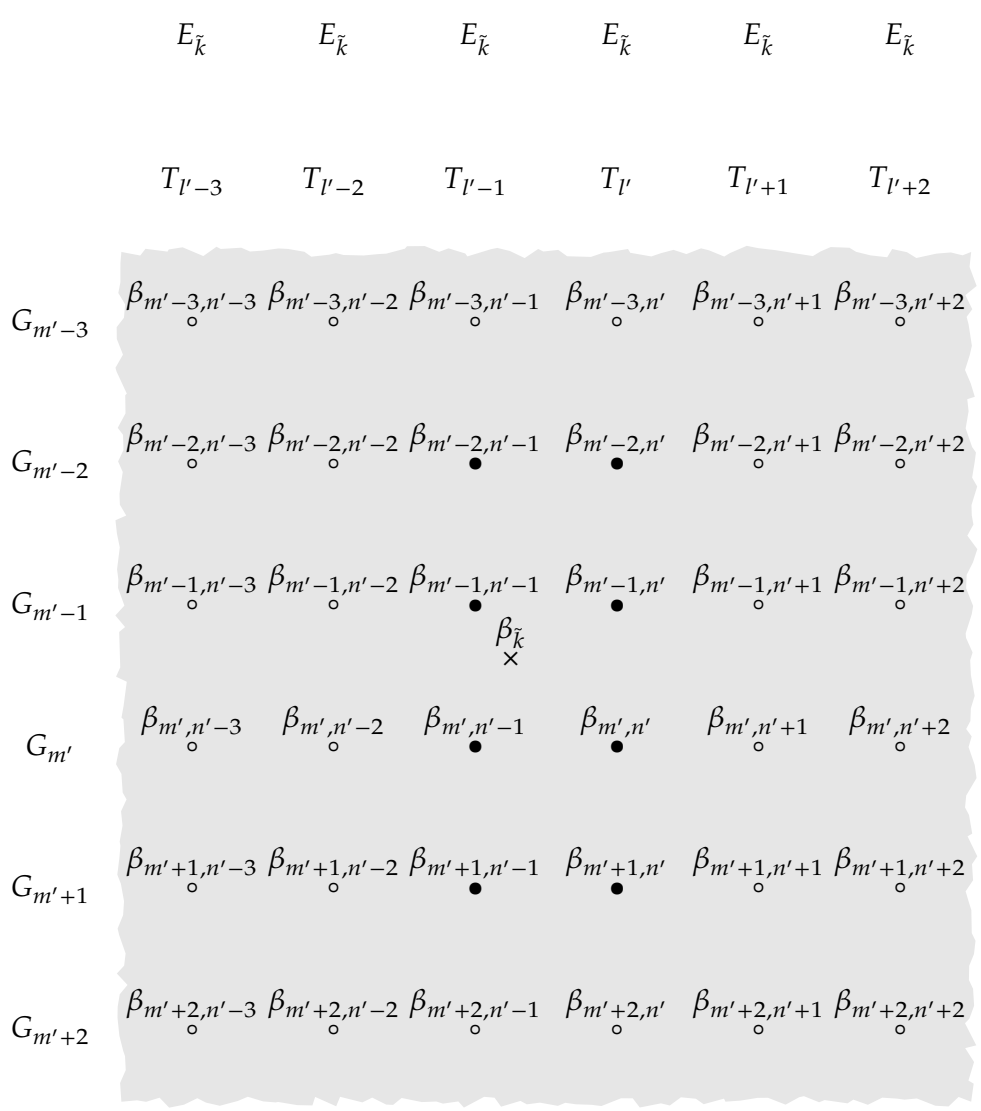


Figure 3.8: Points directly used in interpolation scheme. The ($\times$) indicates the sampled point which must be interpolated from neighboring points ($\bullet$). Empty circles ($\circ$) indicate points which are not used in the interpolation scheme. The columns correspond to E/T gridpoints and rows correspond to CDF gridpoints.

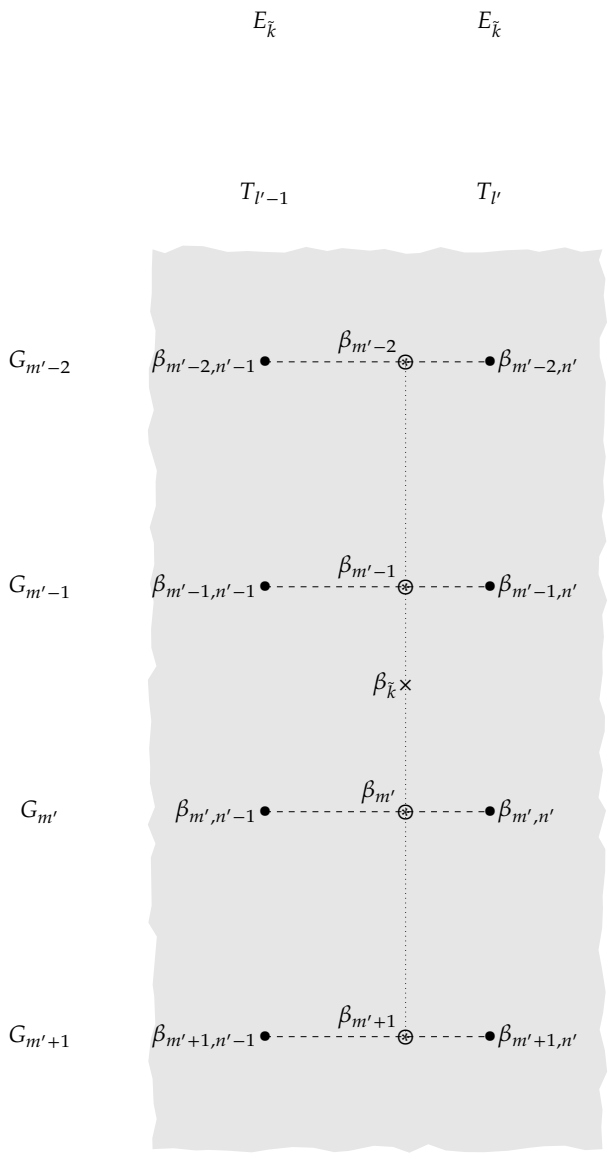


Figure 3.9: Interpolation scheme used to evaluate $\beta_{\tilde{k}}$. Order of evaluation goes $\bullet \to \circledast \to \times$.

Transmission Kernel

The continuous temperature dependence in this work necessitates the use of delta tracking to sample the distance to the next collision. Perturbations to TNSL data will affect the scattering cross section but will not affect the majorant cross section so the indirect effect on the transmission kernel is

$$\frac{\frac{d}{d\lambda}\mathcal{T}}{\mathcal{T}} = 0\,. \tag{3.75}$$

It is important to note that the majorant was not simply set to $\max_{E,T}\sigma_{\mathrm{s}}\left(E,T\right)$ as that would make the majorant a function of $\sigma_{\mathrm{s}}(E,T)$ and therefore subject to perturbations.

Reaction Kernel

Perturbations to TNSL data will affect the probability of sampling each type of reaction. Specifically, perturbations of data used to evaluate the scattering cross section $\sigma_{\mathrm{s}}(E,T)$ will affect the probability of sampling a true versus virtual collision.

If nuclide i uses perturbed TNSL data to evaluate $\sigma_{\mathrm{s}}(E,T)$, then the probability of sampling a true collision changes by

$$\frac{dp_{\mathrm{real}}}{d\sigma_{\mathrm{s},i}} = \frac{N_i}{\Sigma_{\mathrm{m}}} \tag{3.76}$$

so the indirect effect in the reaction kernel becomes

$$\frac{\frac{d}{d\lambda}\mathcal{R}}{\mathcal{R}} = \frac{N_i}{\Sigma_{\mathrm{t}}}\,. \tag{3.77}$$

Energy Kernel

When perturbations in $G\left(\beta \mid E,T\right)$ data are considered, the probability of sampling an given outgoing energy E' is affected. The probability of sampling E' is related to the probability of sampling β by

$$p\left(E'\right) = p\left(\beta\right)\frac{1}{k_{\mathrm{B}}T}\,. \tag{3.78}$$

$1/k_{\mathrm{B}}T$ is a constant with respect to β so, by eq. (2), the indirect effect in the energy kernel becomes

$$\frac{\frac{d}{d\lambda}\mathcal{E}}{\mathcal{E}} = \frac{\frac{dp(\beta)}{d\lambda}}{p\left(\beta\right)}. \tag{3.79}$$

Calculation of $dp\left(\beta\right)/d\lambda$ is performed for every POD coefficient using reverse mode AD.

Angle Kernel

When perturbations in $\hat{H}\left(\alpha \mid \beta, T\right)$ data are considered, the probability of sampling a scattering cosine μ is affected. The probability of sampling μ is related to the probability of sampling α by

$$p\left(\mu\right) = p\left(\alpha\right)\frac{2\sqrt{EE'}}{Ak_{\mathrm{B}}T}. \tag{3.80}$$

Just as was the case with the energy kernel, the Jacobian is a constant with respect to α so, by eq. (2), the indirect effect in the angle kernel becomes

$$\frac{\frac{d}{d\lambda}\mathcal{A}}{\mathcal{A}} = \frac{\frac{dp(\alpha)}{d\lambda}}{p\left(\alpha\right)} \tag{3.81}$$

where the righthand side is also computed using reverse mode AD.

Chapter 4

Test Problems

Two problems are examined to benchmark the new methodology proposed: A "broomstick" problem is used to compare the outgoing energy and angle distributions of Thermal Neutron Scattering Law (TNSL) data stored as Proper Orthogonal Decomposition (POD) coefficients versus the MCNP® implementation. A 1-D slab is used to examine a problem with continuous temperature dependence and perform uncertainty quantification on partial currents for two energy groups.

4.1 Broomstick Problem

The broomstick problem consists of a long, thin cylinder with a point source directed along the axis of the cylinder, illustrated in fig. 4.1. The volume of the cylinder is modeled as H in H_2O. This work only requires a single POD dataset to model thermal scattering at all

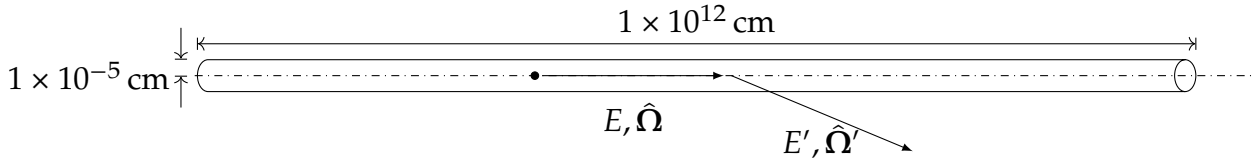


Figure 4.1: The broomstick problem

Table 4.1: Broomstick H in H_2O TNSL Datasets by MCNP

T	MCNP A
293.6 K	`h-h2o.40`
450.0 K	`h-h2o.48`
500.0 K	`h-h2o.50`
650.0 K	`h-h2o.56`

temperatures whereas MCNP requires appropriate ACE files, listed in Table 4.1, be used for each particular temperature. Both datasets are based on the ENDF/B-VIII.0 evaluation for H in H_2O, discussed in detail in Subsection 2.5.1. Full-order POD expansion were used in this problem so that truncation errors from using a low-rank approximation do not enter into the comparisons with MCNP.

The broomstick problem is designed so that particles which scatter immediately leave the volume. Surface tallies can then estimate the energy and direction for once-scattered particles. This broomstick problem provides a way of estimating the Probability Density Functions (PDFs) for β and α when using TNSL data.

In both cases, 200 equally-spaced energy bins were used, 200 equally-spaced cosine bins were used, and 1×10^8 histories were run. Temperatures were selected to cover a wide range of temperatures (293.6 K to 650.0 K) as well as evaluations which are nearby in temperature (450.0 K and 500.0 K). The results, summarized in Table 4.2, give the ℓ^2 error

$$\ell^2 \text{ Error} := \sqrt{\sum_b |\tilde{c}_b - c_b|^2} \tag{4.1}$$

where $\tilde{c}_b$ is the number of counts in bin b using this work's method and c_b is the number of counts in bin b given by MCNP. The $\max_b |\Delta c_b|$ error is the largest magnitude in difference in counts across all bins.

Qualitatively, there are some artifacts in the POD implementation. At low incident energies, shown in fig. 4.2, plots for $p(E)$ seem to oscillate. At higher energies, this phenomenon does not appear to happen. Plots of $p(\mu)$ seem to disagree with MCNP, espe-

Table 4.2: Error between POD method and MCNP bin counts in E' and μ

			E'		μ	
Figure	E (eV)	T (K)	ℓ^2 Error	$\max_b \lvert\Delta c_b\rvert$	ℓ^2 Error	$\max_b \lvert\Delta c_b\rvert$
4.2	0.00253	293.6	7.43×10^{-3}	3.28×10^{-3}	2.85×10^{-3}	7.89×10^{-4}
		450.0	4.03×10^{-3}	1.39×10^{-3}	3.39×10^{-3}	1.38×10^{-3}
		500.0	4.12×10^{-3}	1.53×10^{-3}	3.44×10^{-3}	1.42×10^{-3}
		650.0	4.71×10^{-3}	2.06×10^{-3}	3.98×10^{-3}	2.81×10^{-3}
4.3	0.0253	293.6	8.46×10^{-3}	7.00×10^{-3}	2.31×10^{-3}	1.13×10^{-3}
		450.0	1.67×10^{-3}	1.16×10^{-3}	2.90×10^{-3}	1.32×10^{-3}
		500.0	1.37×10^{-3}	8.32×10^{-4}	3.16×10^{-3}	1.49×10^{-3}
		650.0	1.13×10^{-3}	7.36×10^{-4}	3.81×10^{-3}	2.16×10^{-3}
4.4	0.056	293.6	7.15×10^{-3}	5.51×10^{-3}	2.38×10^{-3}	1.10×10^{-3}
		450.0	1.79×10^{-3}	1.01×10^{-3}	3.24×10^{-3}	1.34×10^{-3}
		500.0	1.61×10^{-3}	8.19×10^{-4}	3.52×10^{-3}	1.48×10^{-3}
		650.0	1.31×10^{-3}	4.42×10^{-4}	4.18×10^{-3}	2.12×10^{-3}
4.5	0.56	293.6	2.50×10^{-3}	1.43×10^{-3}	2.83×10^{-3}	2.07×10^{-3}
		450.0	1.86×10^{-3}	1.42×10^{-3}	2.89×10^{-3}	1.94×10^{-3}
		500.0	1.82×10^{-3}	1.41×10^{-3}	2.91×10^{-3}	1.84×10^{-3}
		650.0	1.74×10^{-3}	1.40×10^{-3}	3.09×10^{-3}	1.66×10^{-3}

cially near $\mu = -1$ and $\mu = +1$. Interestingly, the are also oscillations in MCNP near backward and forward scattering. Figures 4.2 to 4.5 do contain 3σ uncertainty bands, but they are about the same thickness as the lines themselves. The discrepancy is likely not due to statistical uncertainty. It is unclear if the discrepancies are due to implementation differences or differences in the TNSL evaluations. MCNP constructs the full Double Differential Cross Section (DDCS) from TNSL data at runtime, which is one crucial implementation difference.

4.2 One-Dimensional Slab

A 1-D slab problem is constructed to demonstrate uncertainty quantification using POD coefficients. A 2 cm thick volume is modeled as purely scattering H in H_2O. The thickness of the slab corresponds to approximately 2.67 mean free paths. Incident on the left face is an isotropic flux

$$p(\mu) = \begin{cases} 2\mu & \text{if } 0 < \mu \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

where μ is with respect to a vector pointing into the left face from outside. The incident energy was fixed at $E = 0.1\,\text{eV}$. The temperature within the slab was set to linearly vary from $T_{\text{left}} = 350\,\text{K}$ at the left face to $T_{\text{right}} = 650\,\text{K}$ at the right face. Scattering above 4 eV was treated with free gas scattering and scattering below 4 eV was treated with TNSL data.

Four estimators are examined:

1. J_2^-, the partial current leaking out of the left face below E_{cutoff},
2. J_1^-, the partial current leaking out of the left face above E_{cutoff},
3. J_2^+, the partial current leaking out of the right face below E_{cutoff}, and
4. J_1^+, the partial current leaking out of the right face above E_{cutoff}

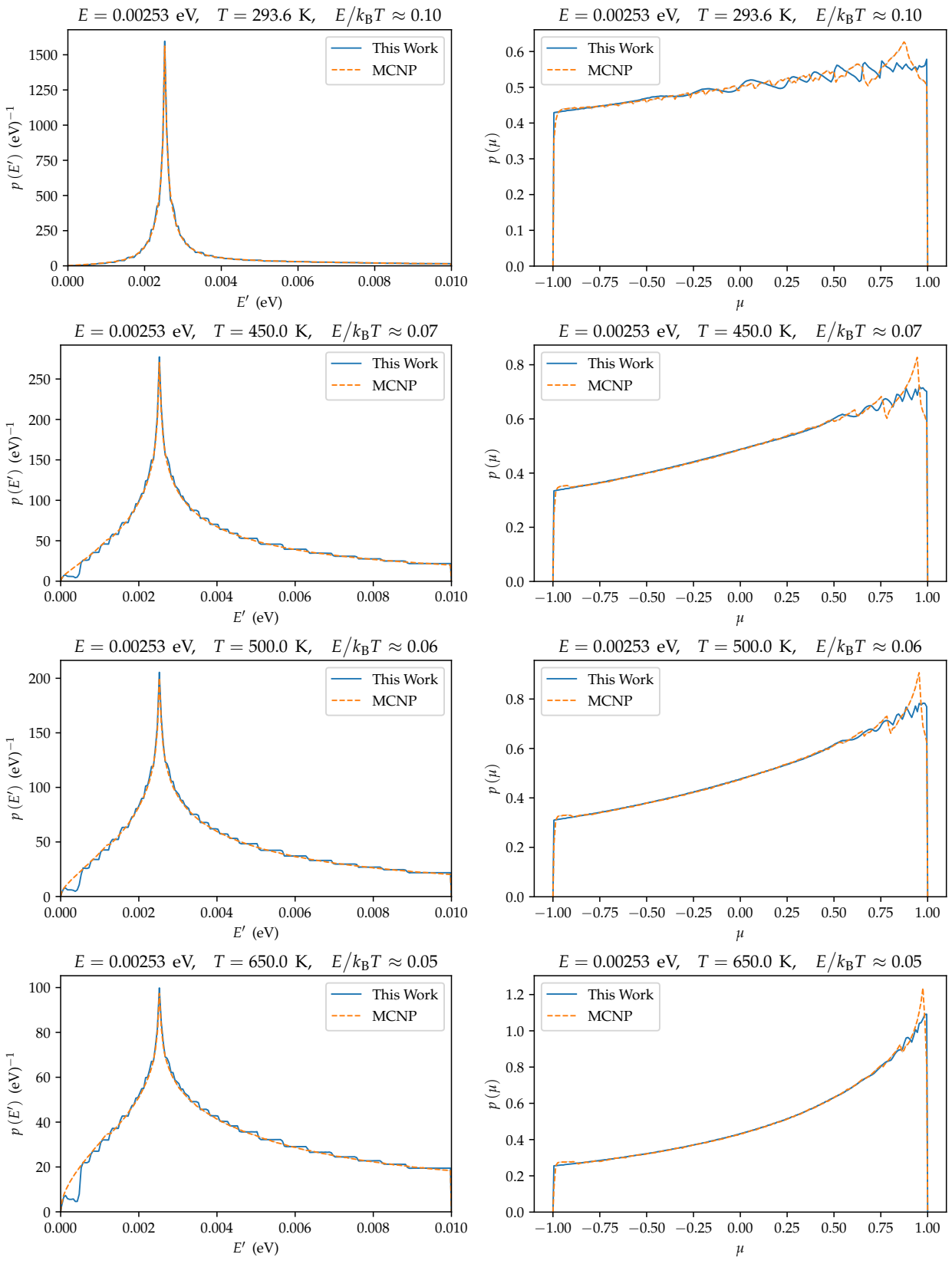


Figure 4.2: PDFs for $E = 0.002\,53$ eV

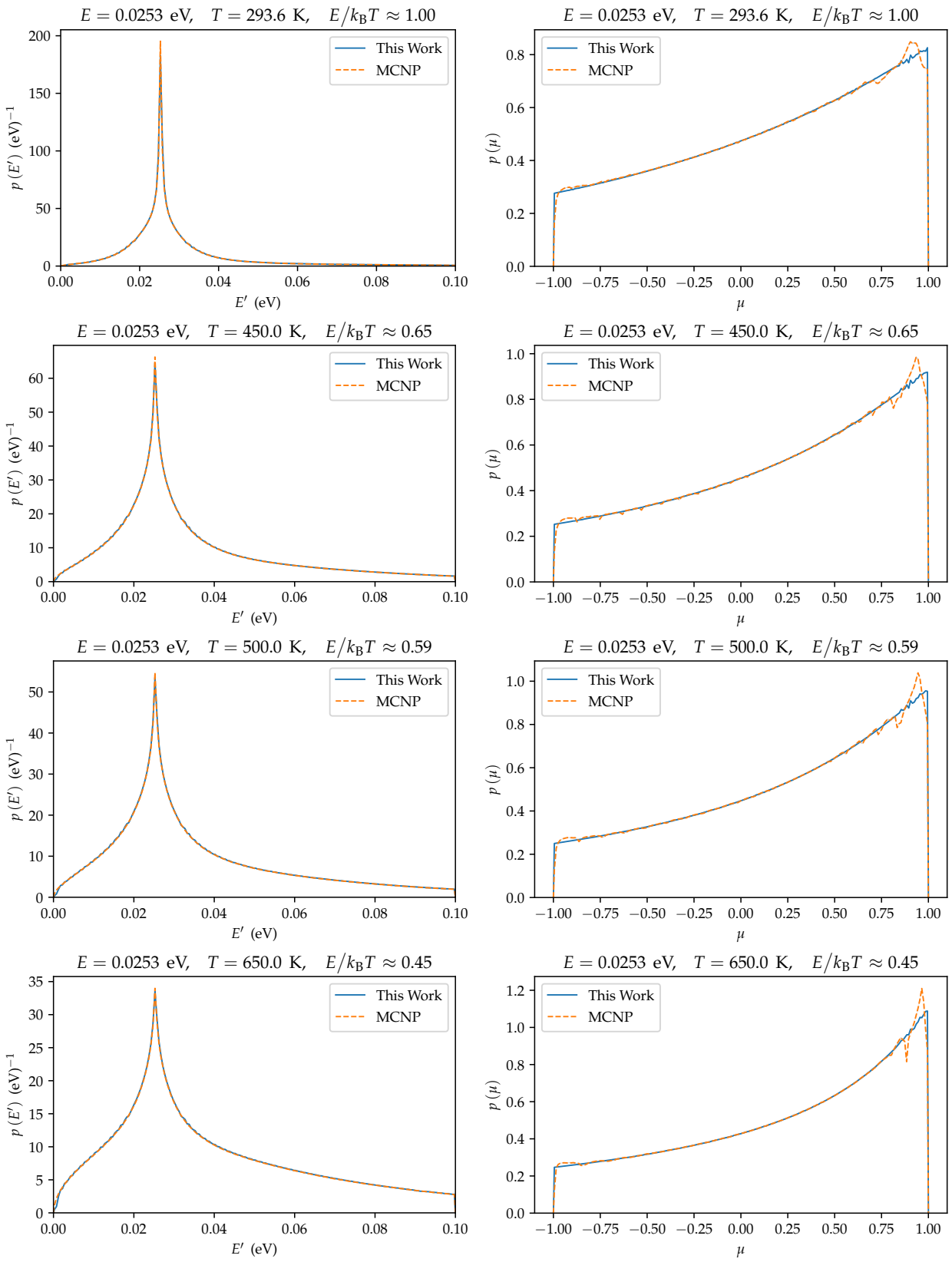


Figure 4.3: PDFs for $E = 0.0253\,\text{eV}$

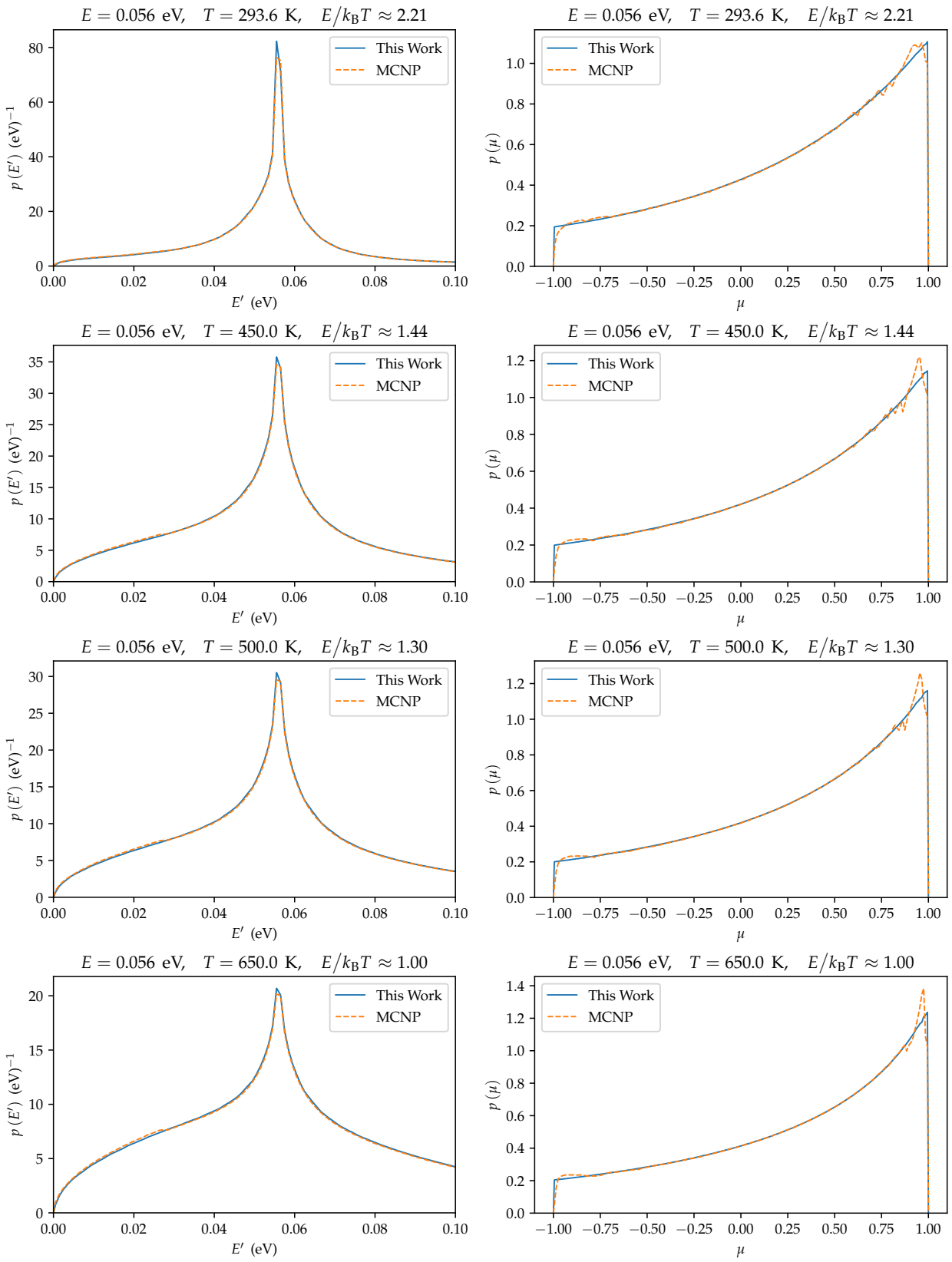


Figure 4.4: PDFs for $E = 0.056\,\text{eV}$

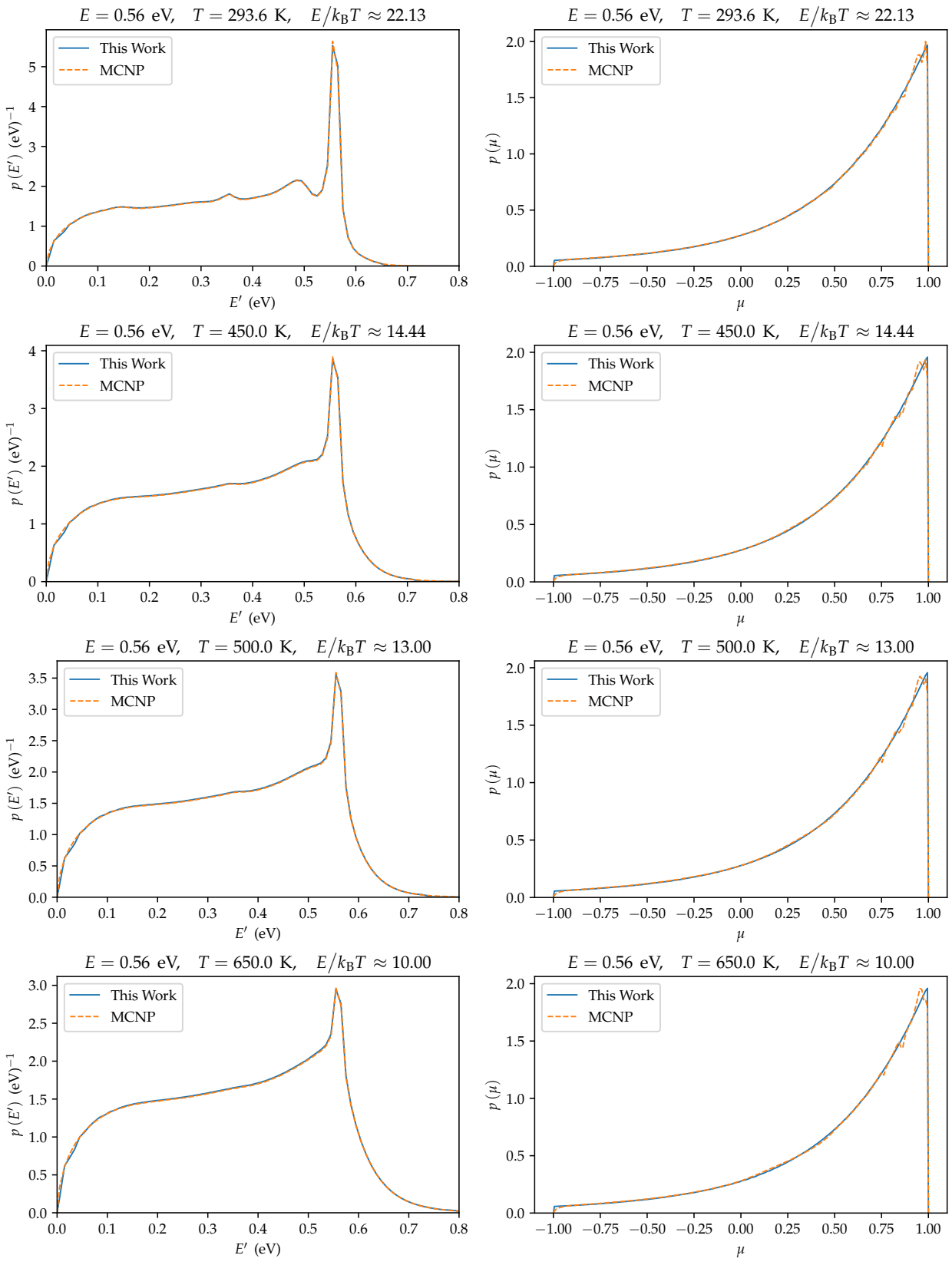


Figure 4.5: PDFs for $E = 0.56\,\text{eV}$

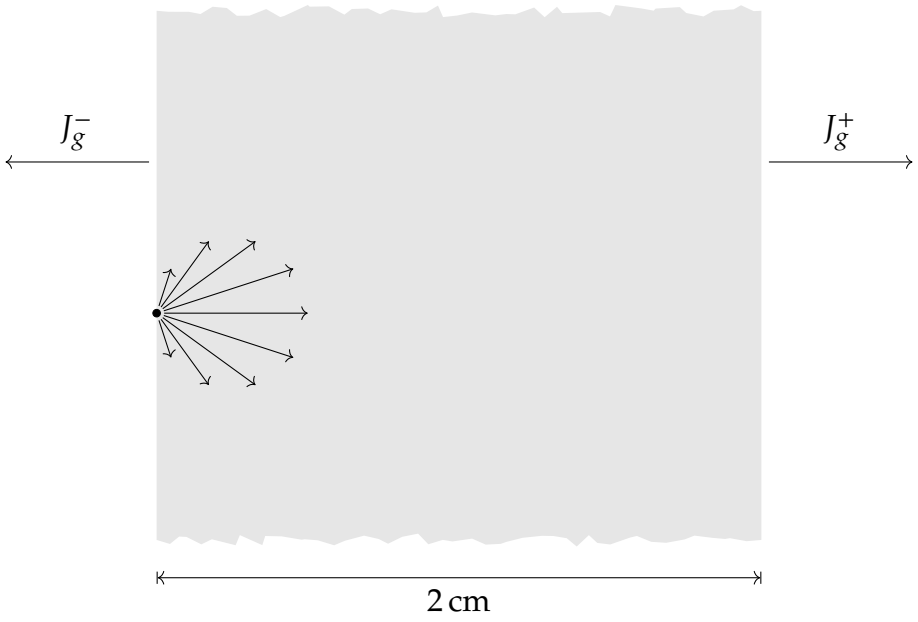


Figure 4.6: 1-D Slab Problem

where $E_{\text{cutoff}} = 0.0486\,\text{K}$ was selected because it is the mean energy of neutrons following a Maxwell-Boltzmann distribution at $T = 500\,\text{K}$, the average temperature in the slab.

TNSL data by Chapman, discussed in Subsection 2.5.2, was used with gridpoints detailed in Tables 3.2 to 3.4 and the corresponding expansion orders in Table 3.4.

Sensitivities of each estimator with respect to all 42 314 POD coefficients was performed using a combination of Differential Operator Sampling (DOS) and Automatic Differentiation (AD). The sensitivities for a single estimator are verified using central differencing in Table 4.3. Running 5×10^7 histories took approximately 11.44 h of wall time on 8 cores. For comparison, a Total Monte Carlo (TMC) run was performed, running each of the POD files using 5×10^7 histories. The overall runtime was approximately 10.97 h. Sample statistics were used to compute the mean values of each of the estimators.

The mean values for each of the estimators are listed in Table 4.4. Estimates between this work and TMC were similar in magnitude but disagreed even when statistical uncertainty is taken into account. We do not expect the mean of estimators using different nuclear data to be equal to the estimator using the mean of different nuclear data. In other words, we have $g\left(\mathbb{E}\left[X\right]\right) \neq \mathbb{E}\left[g\left(x\right)\right]$.

Table 4.3: Central difference estimtes of differential change in estimator dJ_2^+ using $d\lambda = 0.05\lambda$

		This Work		Central Difference	
Dataset	Matrix	Mean	Std. Dev.	Mean	Std. Dev
$\sigma_s(E,T)$	U	3.50×10^{-4}	2.88×10^{-6}	3.48×10^{-4}	3.55×10^{-5}
	Σ	-1.65×10^{-4}	2.89×10^{-6}	-1.68×10^{-4}	3.55×10^{-5}
	V	3.34×10^{-4}	3.96×10^{-6}	3.26×10^{-4}	3.55×10^{-5}
$G(\beta \mid E,T)$	U	3.32×10^{-3}	3.40×10^{-3}	-4.55×10^{-6}	3.55×10^{-5}
	Σ	-1.17×10^{-2}	7.35×10^{-3}	-4.37×10^{-3}	3.55×10^{-5}
	V	-6.14×10^{-3}	6.60×10^{-3}	-3.20×10^{-7}	3.55×10^{-5}
$\hat{H}(\alpha \mid \beta,T)$	U	-1.68×10^{-4}	1.50×10^{-5}	-1.31×10^{-4}	3.55×10^{-5}
	Σ	3.34×10^{-5}	1.75×10^{-7}	3.05×10^{-5}	3.55×10^{-5}
	V	1.84×10^{-5}	3.53×10^{-7}	1.83×10^{-5}	3.55×10^{-5}

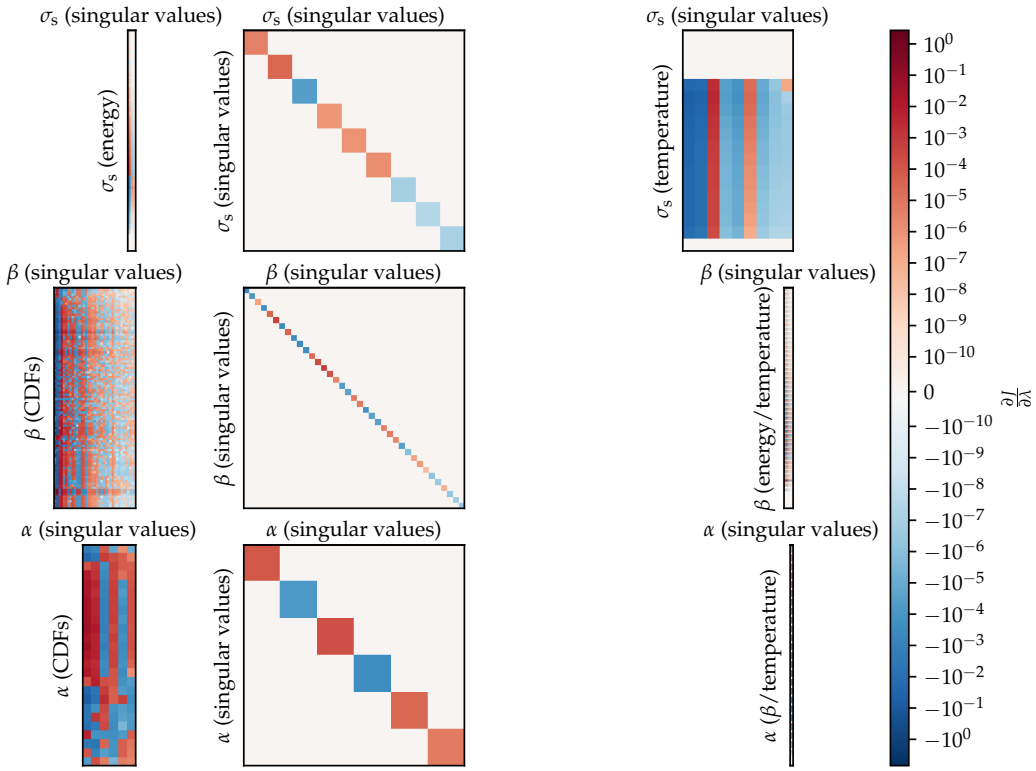
Table 4.4: Uncertainty quantification

	Mean		Statistical Uncertainty		ND Uncertainty	
	This Work	TMC	This Work	TMC	This Work	TMC
J_2^-	2.36×10^{-1}	2.34×10^{-1}	6.01×10^{-5}	5.98×10^{-5}	4.50×10^{-3}	3.79×10^{-3}
J_1^-	4.30×10^{-1}	4.32×10^{-1}	7.00×10^{-5}	7.00×10^{-5}	1.49×10^{-2}	6.65×10^{-3}
J_2^+	6.77×10^{-2}	6.72×10^{-2}	3.55×10^{-5}	3.54×10^{-5}	2.86×10^{-3}	2.22×10^{-3}
J_1^+	2.66×10^{-1}	2.67×10^{-1}	6.25×10^{-5}	6.26×10^{-5}	8.55×10^{-3}	7.96×10^{-3}

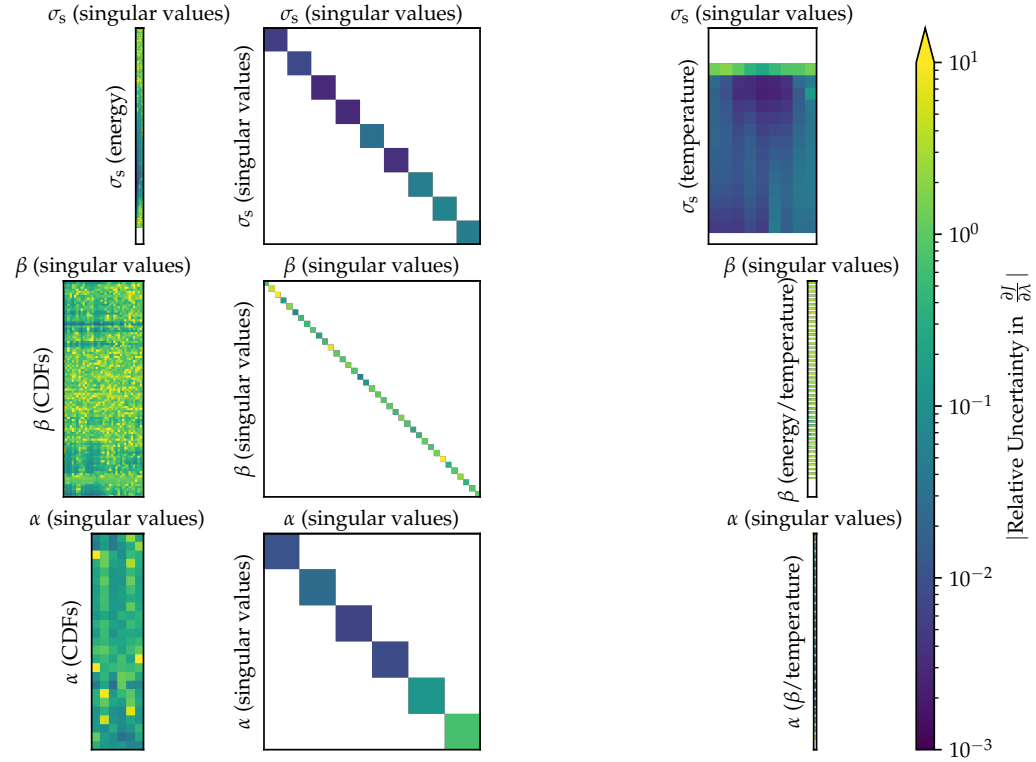
The reported statistical uncertainties in Table 4.4 give the uncertainty due to sampling a finite number of samples from the set of all possible histories. The statistical uncertainty reported using this work's method is the variance of the mean from one run. The statistical uncertainty reported using TMC is the average of the uncertainties reported across all 250 runs. To properly estimate the statistical uncertainty in the TMC runs, one would need covariance information across estimators from different runs.

The Nuclear Data (ND) uncertainties give uncertainties propagated from the TNSL dataset used. The ND uncertainty was detectable above the statistical uncertainty in each case. The DOS/AD approach tended to give slightly greater uncertainty in each case. Recall that one key assumption in the first order Taylor expansion of uncertainty is that the uncertainty is linear in each of the parameters. This is not necessarily the case when the function being expanded changes rapidly about the mean or if the uncertainty for each parameter is sufficiently large. Figure 4.7a shows sensitivity estimates for J_2^- with respect to each parameter perturbed. Figure 4.7b shows the absolute value of the relative uncertainty in each sensitivity estimate. Some parameters which are never used have infinite uncertainty. For example, temperatures outside of the range $\left[T_{\text{left}}, T_{\text{right}}\right]$ are never used and thus do not affect any of the estimators. Figures 4.8 and 4.9 show the sensitivities sorted by magnitude. Figure 4.8 give the sensitivities for the leftoing partial current J^- and fig. 4.9 for the rightgoing partial current J^+. In both cases, POD coefficients which did not contribute much to estimator sensitivities generally had larger relative uncertainties. A handful of coefficients do fall below a relative error of 0.1 which is generally considered to be a good threshold for reliable results.

One benefit of using first order Taylor expansions to compute uncertainties is that the contribution of each parameter to the overall uncertainty can be computed. Submatrices of the full covariance matrix can be summed over, allowing one to decompose contributions as shown in Table 4.5 for J_1^-. Most of the uncertainty in the leakage rate was found to be due to self-covariances in $G\left(\beta \mid E, T\right)$, the dataset for sampling β. Intuitively, this makes

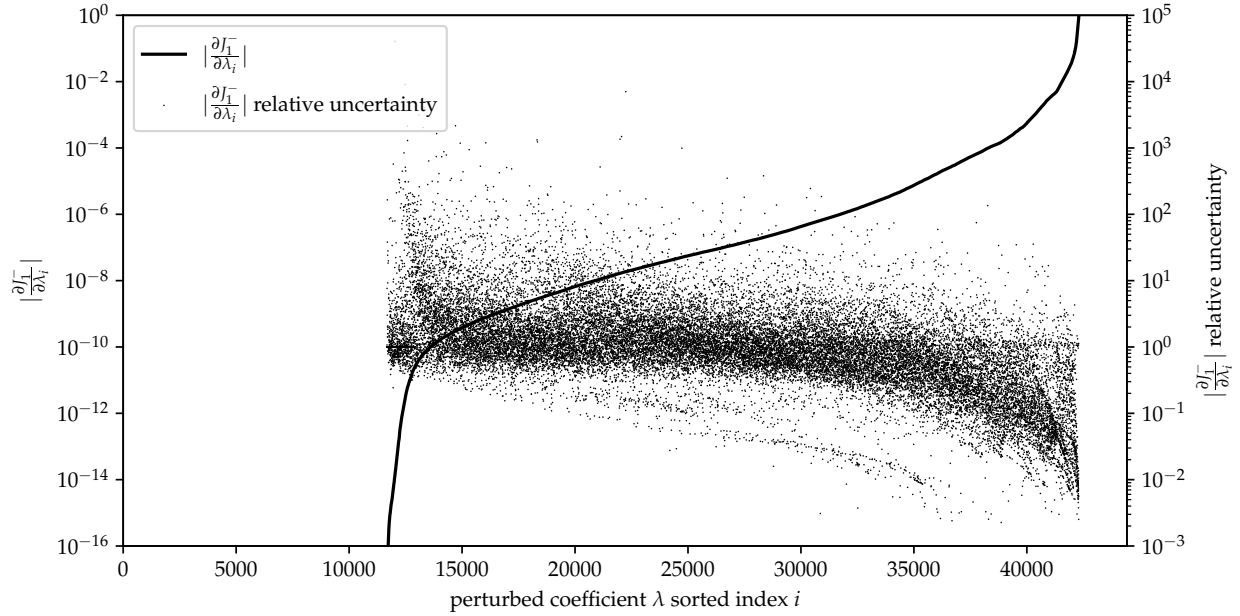


(a) Sensitivity estimates

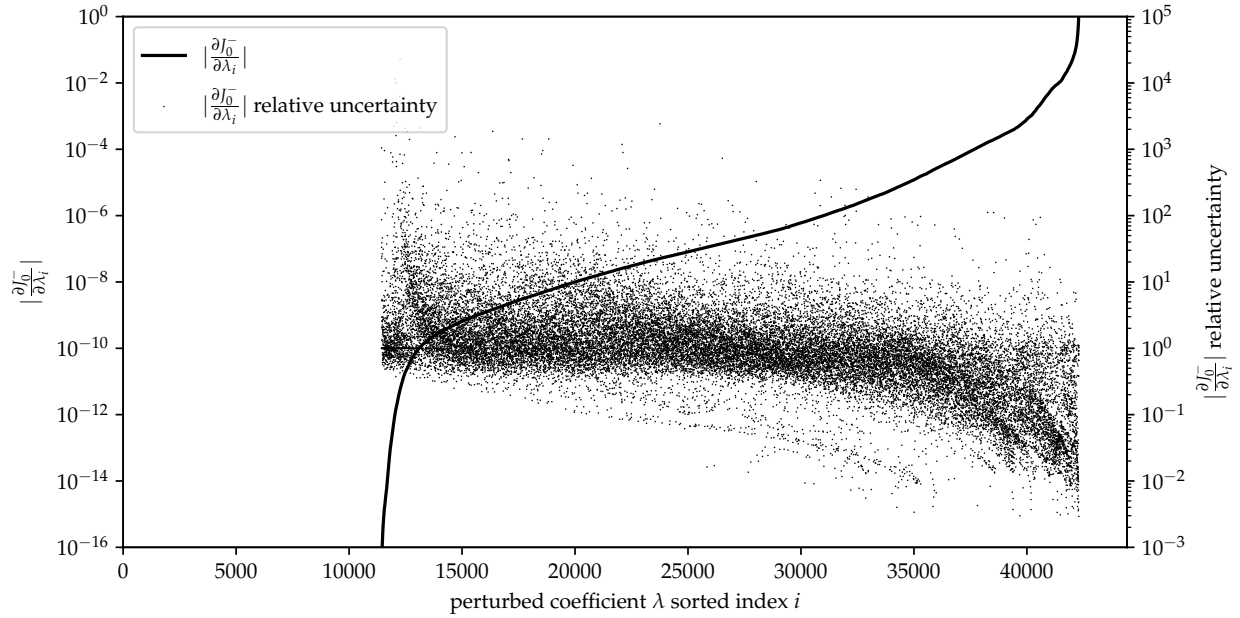


(b) Absolute value of relative uncertainty in sensitivity estimates

Figure 4.7: Sensitivities $\partial J_2^- / \partial \lambda$

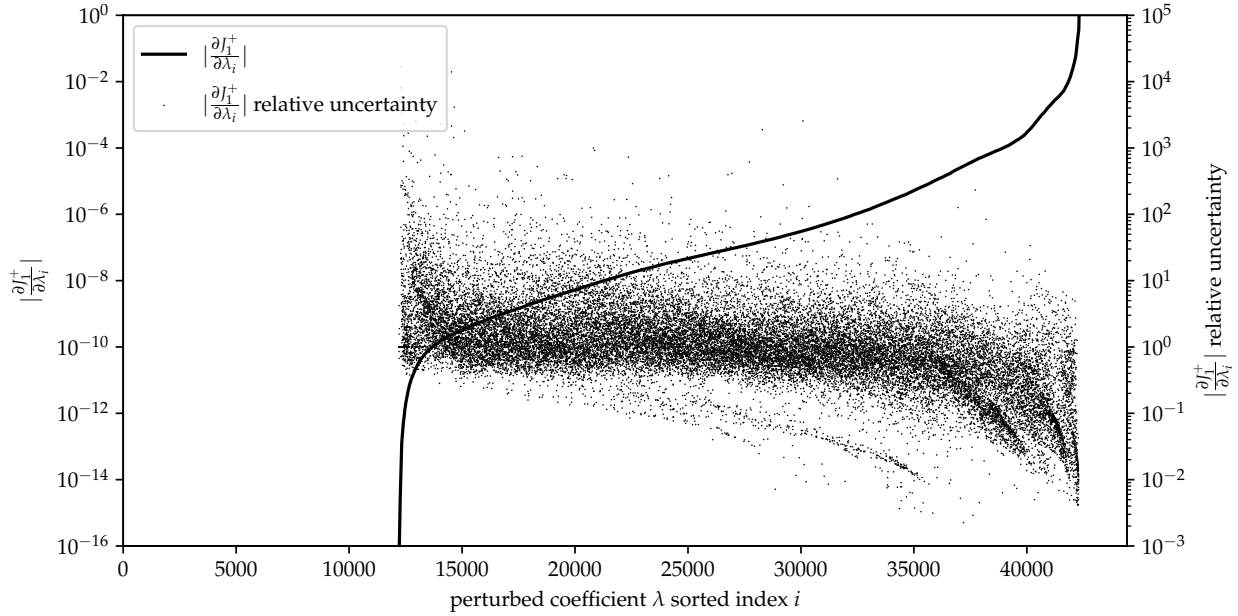


(a) $E' < E_{\text{cutoff}}$

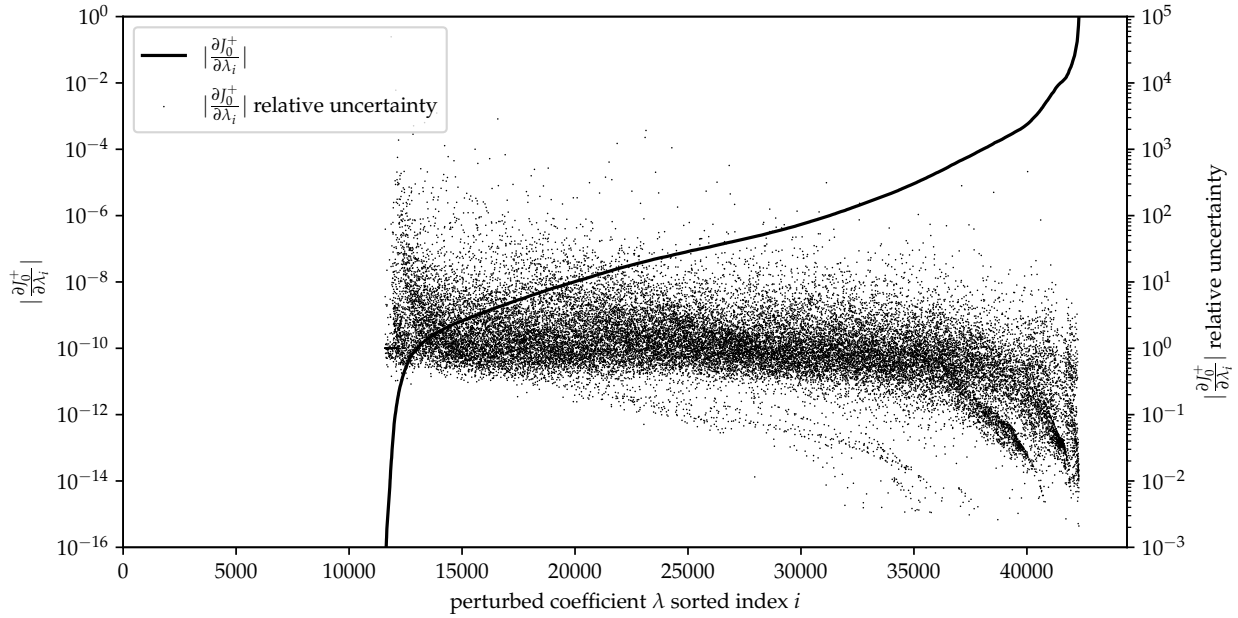


(b) $E' > E_{\text{cutoff}}$

Figure 4.8: Absolute value of sensitivities and relative errors for leftgoing partial current J^-. Sensitivities are sorted by magnitude.



(a) $E' < E_{\text{cutoff}}$



(b) $E' > E_{\text{cutoff}}$

Figure 4.9: Absolute value of sensitivities and relative errors for rightgoing partial current J^+. Sensitivities are sorted by magnitude.

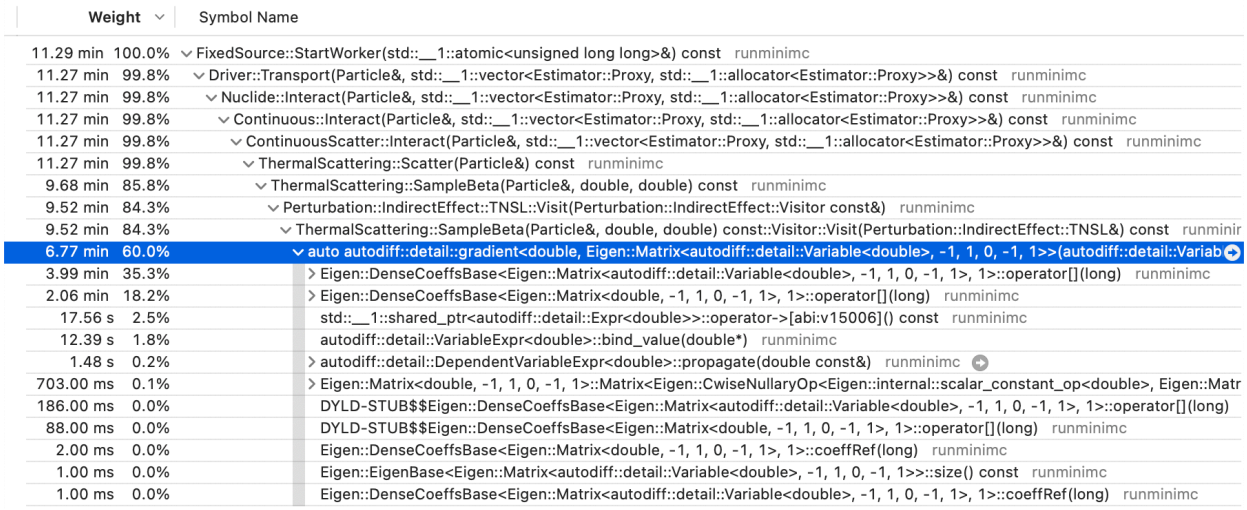
Table 4.5: Contributions to nuclear data uncertainty for J_1^- estimator

	$\sigma_s(E,T)$	$G(\beta \mid E,T)$	$\hat{H}(\alpha \mid \beta,T)$
$\sigma_s(E,T)$	5.59×10^{-6}	6.01×10^{-6}	1.50×10^{-5}
$G(\beta \mid E,T)$	-	1.76×10^{-4}	5.31×10^{-6}
$\hat{H}(\alpha \mid \beta,T)$	-	-	1.33×10^{-5}

sense as the energy spectrum has the strongest effect on the thermalization process versus the total cross section or scattering angle.

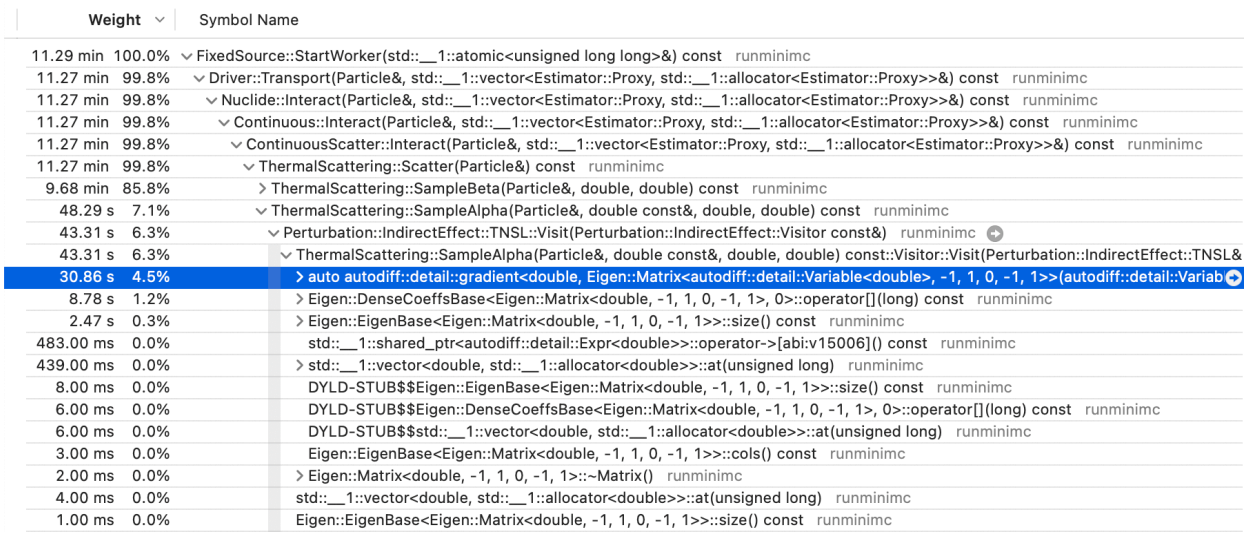
4.3 Performance Metrics

Wall times for both DOS/AD and TMC were similar. A snippet of the DOS/AD method on the 1D slab problem was profiled using `Instruments.app`, part of the `Xcode` Integrated Development Environment (IDE) developed by Apple Inc. Figure 4.10a shows the time spent by the code when sampling β and Figure 4.10b shows the same for sampling α. About 85.8% of the code's time was spent sampling β while about 7.1% of time was spent sampling α. This is roughly in line with the size of each data set given in Table 3.5. Within each sampling routine, computation of the indirect effects using AD took up a significant portion of each routine. 98.3% of the time in sampling β was spent computing the indirect effects with respect to each TNSL parameter. 89.6% of the time in sampling α was spent in doing the same. It was noted that disabling DOS/AD and skipping the calculation of indirect effects did not speed up the code significantly. Digging deeper into the code profiler revealed that the computation of the indirect effects is not itself expensive, but *allocating space for the AD variables on the heap* contributed to the majority of the time spent in computing indirect effects. Replacing AD variables with regular `double` resulted in significant speedup.



Weight		Symbol Name
11.29 min	100.0%	FixedSource::StartWorker(std::__1::atomic<unsigned long long>&) const runminimc
11.27 min	99.8%	Driver::Transport(Particle&, std::__1::vector<Estimator::Proxy, std::__1::allocator<Estimator::Proxy>>&) const runminimc
11.27 min	99.8%	Nuclide::Interact(Particle&, std::__1::vector<Estimator::Proxy, std::__1::allocator<Estimator::Proxy>>&) const runminimc
11.27 min	99.8%	Continuous::Interact(Particle&, std::__1::vector<Estimator::Proxy, std::__1::allocator<Estimator::Proxy>>&) const runminimc
11.27 min	99.8%	ContinuousScatter::Interact(Particle&, std::__1::vector<Estimator::Proxy, std::__1::allocator<Estimator::Proxy>>&) const runminimc
11.27 min	99.8%	ThermalScattering::Scatter(Particle&) const runminimc
9.68 min	85.8%	ThermalScattering::SampleBeta(Particle&, double, double) const runminimc
9.52 min	84.3%	Perturbation::IndirectEffect::TNSL::Visit(Perturbation::IndirectEffect::Visitor const&) runminimc
9.52 min	84.3%	ThermalScattering::SampleBeta(Particle&, double, double) const::Visitor::Visit(Perturbation::IndirectEffect::TNSL&) const runminir
6.77 min	60.0%	auto autodiff::detail::gradient<double, Eigen::Matrix<autodiff::detail::Variable<double>, -1, 1, 0, -1, 1>>(autodiff::detail::Variab
3.99 min	35.3%	Eigen::DenseCoeffsBase<Eigen::Matrix<autodiff::detail::Variable<double>, -1, 1, 0, -1, 1>, 1>::operator[](long) runminimc
2.06 min	18.2%	Eigen::DenseCoeffsBase<Eigen::Matrix<double, -1, 1, 0, -1, 1>, 1>::operator[](long) runminimc
17.56 s	2.5%	std::__1::shared_ptr<autodiff::detail::Expr<double>>::operator->[abi:v15006]() const runminimc
12.39 s	1.8%	autodiff::detail::VariableExpr<double>::bind_value(double*) runminimc
1.48 s	0.2%	autodiff::detail::DependentVariableExpr<double>::propagate(double const&) runminimc
703.00 ms	0.1%	Eigen::Matrix<double, -1, 1, 0, -1, 1>::Matrix<Eigen::CwiseNullaryOp<Eigen::internal::scalar_constant_op<double>, Eigen::Matr
186.00 ms	0.0%	DYLD-STUB$$Eigen::DenseCoeffsBase<Eigen::Matrix<autodiff::detail::Variable<double>, -1, 1, 0, -1, 1>, 1>::operator[](long)
88.00 ms	0.0%	DYLD-STUB$$Eigen::DenseCoeffsBase<Eigen::Matrix<double, -1, 1, 0, -1, 1>, 1>::operator[](long) runminimc
2.00 ms	0.0%	Eigen::DenseCoeffsBase<Eigen::Matrix<double, -1, 1, 0, -1, 1>, 1>::coeffRef(long) runminimc
1.00 ms	0.0%	Eigen::EigenBase<Eigen::Matrix<autodiff::detail::Variable<double>, -1, 1, 0, -1, 1>>::size() const runminimc
1.00 ms	0.0%	Eigen::DenseCoeffsBase<Eigen::Matrix<autodiff::detail::Variable<double>, -1, 1, 0, -1, 1>, 1>::coeffRef(long) runminimc

(a) Focus on `SampleBeta` subtree



Weight		Symbol Name
11.29 min	100.0%	FixedSource::StartWorker(std::__1::atomic<unsigned long long>&) const runminimc
11.27 min	99.8%	Driver::Transport(Particle&, std::__1::vector<Estimator::Proxy, std::__1::allocator<Estimator::Proxy>>&) const runminimc
11.27 min	99.8%	Nuclide::Interact(Particle&, std::__1::vector<Estimator::Proxy, std::__1::allocator<Estimator::Proxy>>&) const runminimc
11.27 min	99.8%	Continuous::Interact(Particle&, std::__1::vector<Estimator::Proxy, std::__1::allocator<Estimator::Proxy>>&) const runminimc
11.27 min	99.8%	ContinuousScatter::Interact(Particle&, std::__1::vector<Estimator::Proxy, std::__1::allocator<Estimator::Proxy>>&) const runminimc
11.27 min	99.8%	ThermalScattering::Scatter(Particle&) const runminimc
9.68 min	85.8%	ThermalScattering::SampleBeta(Particle&, double, double) const runminimc
48.29 s	7.1%	ThermalScattering::SampleAlpha(Particle&, double const&, double, double) const runminimc
43.31 s	6.3%	Perturbation::IndirectEffect::TNSL::Visit(Perturbation::IndirectEffect::Visitor const&) runminimc
43.31 s	6.3%	ThermalScattering::SampleAlpha(Particle&, double const&, double, double) const::Visitor::Visit(Perturbation::IndirectEffect::TNSL&
30.86 s	4.5%	auto autodiff::detail::gradient<double, Eigen::Matrix<autodiff::detail::Variable<double>, -1, 1, 0, -1, 1>>(autodiff::detail::Variab
8.78 s	1.2%	Eigen::DenseCoeffsBase<Eigen::Matrix<double, -1, 1, 0, -1, 1>, 0>::operator[](long) const runminimc
2.47 s	0.3%	Eigen::EigenBase<Eigen::Matrix<double, -1, 1, 0, -1, 1>>::size() const runminimc
483.00 ms	0.0%	std::__1::shared_ptr<autodiff::detail::Expr<double>>::operator->[abi:v15006]() const runminimc
439.00 ms	0.0%	std::__1::vector<double, std::__1::allocator<double>>::at(unsigned long) runminimc
8.00 ms	0.0%	DYLD-STUB$$Eigen::EigenBase<Eigen::Matrix<double, -1, 1, 0, -1, 1>>::size() const runminimc
6.00 ms	0.0%	DYLD-STUB$$Eigen::DenseCoeffsBase<Eigen::Matrix<double, -1, 1, 0, -1, 1>, 0>::operator[](long) const runminimc
6.00 ms	0.0%	DYLD-STUB$$std::__1::vector<double, std::__1::allocator<double>>::at(unsigned long) runminimc
3.00 ms	0.0%	Eigen::EigenBase<Eigen::Matrix<double, -1, 1, 0, -1, 1>>::cols() const runminimc
2.00 ms	0.0%	Eigen::Matrix<double, -1, 1, 0, -1, 1>::~Matrix() runminimc
4.00 ms	0.0%	std::__1::vector<double, std::__1::allocator<double>>::at(unsigned long) runminimc
1.00 ms	0.0%	Eigen::EigenBase<Eigen::Matrix<double, -1, 1, 0, -1, 1>>::size() const runminimc

(b) Focus on `SampleAlpha` subtree

Figure 4.10: DOS/AD sensitivity code profiled

Chapter 5

Conclusions & Future Work

The primary objective of this work was to demonstrate that uncertainty quantification on Thermal Neutron Scattering Law (TNSL) data was feasible. Until this work, uncertainty quantification due to chemical effects had not been performed because of several issues which this dissertation addresses. These issues were addressed in turn to arrive at a demonstration of uncertainty quantification for H in H_2O. Several avenues for future work were identified which naturally follow from the findings of this work.

5.1 Conclusions

This dissertation introduces, for the first time, a covariance format for TNSL data. The lack of support for TNSL covariance data in existing formats led to changes in the Generalized Nuclear Database Structure (GNDS) covariance format, enabling support for covariances of multidimensional data in general. Collaborating with the international nuclear data community, a new format was proposed for inclusion into the GNDS 2.0 specification. The new format was implemented in the AMPX data processing code. An evaluation of TNSL data for H in H_2O was produced and sent to the National Nuclear Data Center (NNDC). The new evaluation was used by For Updating Data and Generating Evaluations (FUDGE)

developers to exercise previously untested features of the GNDS parsing component of FUDGE.

Memory requirements had previously made the storage of TNSL covariance data unfeasible as multidimensional data tends to require a significant amount of memory. To address this, Proper Orthogonal Decomposition (POD) was introduced to compress elements of the TNSL dataset by using Singular Value Decomposition (SVD) to construct a low rank approximation of the original dataset. This work developed a scheme for evaluating temperature-dependent values of σ_s, β, and α at runtime with minimal penalty on runtime performance. The scheme improves upon previous work using functional expansion by Pavlou and Ji [44] by removing the need to perform empirical fits to TNSL data.

Several methods for sensitivity analysis were examined in order to perform uncertainty quantification on temperature dependent TNSL data. Initial attempts using linear interpolation with an partitioning scheme along with an adaptive coarsening scheme initially showed promising results. However, it proved difficult to capture sharp localized features in TNSL data. More importantly, linear interpolation was found to be incompatible with Differential Operator Sampling (DOS) for computing sensitivities due to the appearance of Dirac deltas when computing derivatives.

Quadratic interpolation, already used in existing Monte Carlo codes, was attempted next. The extra degree of freedom allowed the construction of piecewise continuous Probability Density Functions (PDFs) which was amenable to DOS. However, the requirement that PDFs match at the boundaries made perturbations in any one datapoint propagate across the entire PDF curve. This made expressions for DOS indirect effects intractable so Automatic Differentiation (AD) was introduced for the first time in a Monte Carlo radiation transport code. Nevertheless, monotonicity of the resulting Cumulative Distribution Function (CDF) curves could never be guaranteed in for all possible datasets so this approach was abandoned.

Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) was ultimately used be-

cause it addressed the requirements that the PDFs be continuous and the corresponding CDFs be monotonic. Moreover, PCHIP restored local control – perturbations in any one datapoint only affected neighboring datapoints so indirect effects were significantly simpler to compute. This work implemented a scheme for sampling from cubic polynomials using Newton's method with guaranteed convergence. DOS and AD were used to compute sensitivities due to perturbations in CDF datapoints for β and α.

The methodology for storing temperature dependent TNSL data was examined in a broomstick problem. Outgoing energy and scattering cosines were found to agree with distributions from MCNP® for a wide range of incident neutron energies and target temperatures. In both methods, the same ENDF/B-VIII.0 TNSL dataset for H in H_2O was used. Some artifacts were present in both this work and MCNP but were likely due to implementation differences in their methods for sampling energy and scattering cosine.

A one-dimensional slab problem was performed to demonstrate uncertainty quantification on the effect of chemical binding effects on H in H_2O. Leakage rates were estimated and compared against Total Monte Carlo (TMC) using the same code in both cases. A slight discrepancy in mean values can be attributed to expectation values using nuclear data means not necessarily being equal to the mean of expectation values using different nuclear data. Nuclear data uncertainties were conservatively overestimated by the DOS/AD method when compared to TMC. Sources of this discrepancy include large relative uncertainties for TNSL parameters which are infrequently used and the assumption that uncertainties are linear in the nuclear data parameters which are perturbed.

This work accounted for uncertainties in the total cross section, the outgoing energy distribution, and the outgoing scattering cosines due to uncertainties in physical models for chemical binding effects. The first order Taylor expansion for uncertainty provides a method for quantifying the contributions of each dataset to the overall uncertainty. This provides a useful diagnostic for identifying which datasets contribute to nuclear data error when designing nuclear reactors.

5.2 Future Work

During the course of this work, several avenues for future research were identified. We separate them into short term work, projects which could reasonably be performed by an individual over the course of a few months, and long term work, projects which might merit future dissertation work.

5.2.1 Short Term Work

The covariance format for TNSL data used existing GNDS containers. The `gridded2` container, in particular, is used to store covariance matrix elements. The current specification supports basic types of compression such as storage of diagonal matrices, block matrices, and symmetric matrices. The methodology introduced here suggests that a new compression method based on SVD be introduced. A new GNDS container could implement this by storing three separate matrices (U, Σ, and V) under a `svd` node.

This work primarily examined POD compression on nuclear data parameters to reduce the size of the sensitivity matrices. Previous work [41] had examined eigenvalue decomposition $\mathbf{C} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ on the covariance matrix to obtain low rank approximations of the full covariance matrix. Both approaches, POD on TNSL data and eigenvalue decomposition on the covariance matrix, could be applied at the same time to achieve even further compression. One bottleneck in uncertainty quantification was memory storage for the full covariance matrix when performing first order Taylor series expansion for uncertainty. Performing uncertainty quantification on the eigenvalue decomposition factored form would remove the need to load the full covariance matrix into memory.

AD was initially introduced in this work to address intractably complicated expressions for indirect effects when using piecewise quadratic interpolation for CDFs. Although PCHIP was ultimately used, AD was kept for computing derivatives. It was found that a significant amount of runtime was spent allocating memory for reverse mode AD compu-

tation graphs. Because perturbations in one datapoint only affect neighboring datapoints in PCHIP, analytic expressions can be used for computing indirect effects and potentially realize a significant speedup.

5.2.2 Long Term Future Work

Covariance matrices for multidimensional data here were obtained by flattening multidimensional arrays. Compression methods such as SVD are agnostic to the multivariate structure of data which are being compressed. Recent work on Kronecker-structured covariance models for multiway data [54] generalizes covariance matrices to higher dimensions by modeling it as a Kronecker product of lower dimension matrices. This dissertation could be extended to perform eigenvalue decomposition on Kronecker product covariance matrices to account for the multidimensional structure of $S(\alpha, \beta \mid T)$.

This work focused exclusively on chemical binding effects for H in H_2O. It was selected because it was one material for which covariance data was obtainable. Multiple realizations of Probability Density of State (PDOS) data for H in H_2O, provided by Chapman, allowed the construction of independent, identically distributed samples of $S(\alpha, \beta \mid T)$ data. Future work would examine if methodology proposed here will apply to other moderators such as graphite. Because increasing expansion orders with POD *always* improves the quality of the fit, there is good reason to believe that this method will generalize to other moderators. It remains to be seen if SVD will result in sufficiently good compression ratios for other materials.

The stated motivation for modeling temperature dependence of TNSL data was to enable the design of advanced reactors with exotic moderators. For problems where the effect of fine changes in temperature is not well known, the methodology proposed here can be used to evaluate TNSL data on relatively fine temperature grids. The ENDF/B-VIII.0 TNSL dataset used here was evaluated at 18 distinct temperatures. Future work could produce evaluations of TNSL data on a temperature grid finer than standard evaluated

nuclear data files to see if the additional resolution in temperature is justified.

Acronyms

Acronym	Description	Page List
ACE	A Compact ENDF Format Specification [55]. Nuclear data format primarily used by MCNP.	95
AD	Automatic Differentiation	ix, xi, xiv, 2–4, 46–49, 89, 93, 102, 104, 108, 109, 111–113
AEC	Atomic Energy Commission	50
BEPU	Best Estimate Plus Uncertainty	41
CAB	Centro Atómico Bariloche	39
CDF	Cumulative Distribution Function	x, xiii, 3, 4, 19–21, 28, 59–64, 70–77, 81, 84, 88–90, 111–113, 120, 122, 136, 138, 139
COM	Center of Mass	23, 27, 31, 123
CSEWG	Cross Section Evaluation Working Group	39, 50
DBRC	Doppler Broadening Rejection Correction	31
DDCS	Double Differential Cross Section	32, 35, 36, 60, 97, 120
DOS	Differential Operator Sampling	xi, xiii, xiv, 3, 4, 43, 71, 74, 76, 89, 102, 104, 108, 109, 111, 112
ENDF	Evaluated Nuclear Data File	50–52, 117

Acronym	Description	Page List
ENDF-6	The format specification for evaluated nuclear data files detailed in the ENDF-6 Formats Manual [56]. Related to, but distinct from, any particular release of nuclear data such as ENDF/B-VIII.0.	50, 51, 56, 64, 76, 88
ENDF/B-VIII.0	The 8th Major Release of the Evaluated Nuclear Data File (ENDF) [16]	ix, xii, xiii, 32, 39, 50, 64–66, 95, 112, 114, 117
FPU	Floating Point Unit	23
FUDGE	For Updating Data and Generating Evaluations	56, 110, 111
GA	General Atomics	39
GNDS	The Generalized Nuclear Database Structure [39].	ix, xiii, 2, 4, 39, 51–57, 110, 111, 113
GROMACS	Groningen MAchine for Chemical Simulation	39, 40
HDF5	Hierarchal Data Format	51
IDE	Integrated Development Environment	108
IKE	Institut für Kernenergetik und Energiesysteme	39
JEFF	Joint Evaluated Fission and Fusion	39, 50
JENDL	Japan Evaluated Nuclear Data Library	39, 50
MCNP	Monte Carlo N-Particle®, a general-purpose particle transport code developed by Los Alamos National Laboratory [57].	ix, xii, xiii, 4, 27, 32, 40, 88, 94–97, 112, 116
MD	Molecular Dynamics	38, 39
ND	Nuclear Data	104
NNDC	National Nuclear Data Center	xiii, 56, 110

Acronym	Description	Page List
OpenMC	A Monte Carlo neutron and photon transport simulation code developed by the Computational Reactor Physics Group at the Massachusetts Institute of Technology [9].	27
ORNL	Oak Ridge National Laboratory	40, 56
PCHIP	Piecewise Cubic Hermite Interpolating Polynomial	xiii, 3, 4, 59, 89, 111–114
PDF	Probability Density Function	x, xiii, 15, 16, 19–22, 28–30, 44, 60, 62, 63, 70, 71, 75–77, 80, 83, 85, 86, 95, 98–101, 111, 112, 123, 124, 139
PDOS	Probability Density of State	ix, 37–40, 114
PMF	Probability Mass Function	89
POD	Proper Orthogonal Decomposition	x, xii, xiii, 2, 4, 59, 64, 67, 69, 70, 74, 89, 93–97, 102, 104, 111, 113, 114
RNSD	Reactor and Nuclear Systems Division	56
SA	Sensitivity Analysis	3, 4, 41
SNS	Spallation Neutron Source	40
SVD	Singular Value Decomposition	67, 68, 70, 111, 113, 114
TIP4P	Transferable Intermolecular Potential, 4 Points	39, 40

Acronym	Description	Page List
TMC	Total Monte Carlo	xiv, 4, 41, 49, 102–104, 108, 112
TNSL	Thermal Neutron Scattering Law	ix, xii, xiii, 1–4, 6, 32, 36, 37, 39, 40, 51, 54–56, 59, 63–66, 92, 94, 95, 97, 102, 104, 108, 110–114, 120
UMC	Unified Monte Carlo	40
UQ	Uncertainty Quantification	1–4, 41, 49
WPEC	Working Party of Evaluation Cooperation	51
XML	Extensible Markup Language	ix, 52, 56

Symbols

Symbol	Description	Page List
A	Atomic weight ratio. Ratio of target nuclide mass to neutron mass m_n. For hydrogen-1, $A \approx 0.999167339$.	24–28, 31, 36, 37, 59, 60, 93
D_ℓ	Logarithmic derivative operator. Typically uses λ as the independent parameter to be perturbed. Discussed in Section A.2.	135, 138, 139
E'	Outgoing energy of scattered neutron	xii, 13, 16, 17, 32–38, 59, 92, 93, 96, 106, 107

Symbol	Description	Page List
E	Incident energy of neutron	x, xii, 8–14, 24–27, 33, 36, 37, 59–64, 66, 67, 69–74, 88–90, 92, 93, 95–104, 106–108, 120–124
F	Generic variable representing a CDF. In this work it typically refers to $G(\beta \mid E, T)$ or $\hat{H}(\alpha \mid \beta, T)$.	20, 21, 77–82, 84–86, 88, 136–140
$G(\beta \mid E, T)$	Marginal CDF in β. Defined by eq. (3.7a).	x, xii, 61–63, 66, 67, 69, 70, 72–74, 88, 89, 92, 103, 104, 108, 120
G	Shorthand for $G(\beta \mid E, T)$.	62–64, 88
$H(\alpha \mid \beta, E, T)$	Conditional CDF in α given β. Defined by eq. (3.7b).	61, 62, 89, 122
$I(\boldsymbol{\kappa}, t)$	Intermediate function. Defined by eq. (2.104a)	35
$I_{\mathrm{s}}(\boldsymbol{\kappa}, t)$	Self-intermediate function. Defined by eq. (2.104b)	35
J	Partial current. Defined by eq. (2.17).	11
M	A Maxwell-Boltzmann distribution. Defined by eq. (2.65).	24, 25, 27, 28
R	Response or scoring function. Defined by eq. (2.33)	15, 41–44
$S(\alpha, \beta \mid T)$	Functional form of the TNSL. Related to scattering Double Differential Cross Section (DDCS) through eq. (2.112)	36, 38, 56, 59–61, 63, 114, 123
$S(\boldsymbol{\kappa}, \omega)$	Scattering function. Defined by eq. (2.106a).	35, 36, 38
$S_{\mathrm{d}}(\boldsymbol{\kappa}, \omega)$	Distinct scattering function.	38
$S_{\mathrm{s}}(\boldsymbol{\kappa}, \omega)$	Self-scattering function. Defined by eq. (2.107a)	35, 36, 38

Symbol	Description	Page List
T	Temperature of target nuclide	x, xii, 24–28, 36–38, 56, 59–64, 66, 67, 69–74, 88–90, 92, 93, 95, 96, 103, 104, 108, 114, 120–124, 147
Σ	Macroscopic cross section. Defined by eq. (2.10).	10, 11, 13, 14, 17–19
α_∞	Maximum possible value of α across all β, E, and T. Defined by eq. (3.9)	62
$\alpha_{\max}$	The maximum possible value for α. Defined by eq. (2.119b)	37, 60–62, 122
$\alpha_{\min}$	The minimum possible value for α. Defined by eq. (2.119a)	37, 60–62, 122
α	Dimensionless momentum transfer. Defined by eq. (2.114)	ix, x, xii, xiii, 36–38, 52, 55, 56, 58–64, 66, 67, 69–74, 81, 88, 89, 93, 95, 103, 108, 111, 112, 114, 120–124, 147
$\beta_{\max}$	The maximum possible value for β. Defined by eq. (2.117b)	37, 60, 61
$\beta_{\min}$	The minimum possible value for β. Defined by eq. (2.117a)	37, 60, 61

Symbol	Description	Page List
β	Dimensionless energy transfer. Defined by eq. (2.113)	ix, x, xii, xiii, 36–38, 52, 55, 56, 58–64, 66–74, 88, 89, 91–93, 95, 103, 104, 108, 111, 112, 114, 120–124, 146
$\hat{H}(\alpha \mid \beta, T)$	Energy-independent CDF used to evaluate $H(\alpha \mid \beta, E, T)$ on-the-fly. Defined by eq. (3.8).	x, xii, 61–64, 66, 67, 69, 70, 72–74, 88, 93, 103, 108, 120, 122
$\hat{H}_{\max}$	Value of $\hat{H}$ when evaluated at $\alpha_{\max}$. Defined by eq. (3.11b).	61, 62
$\hat{H}_{\min}$	Value of $\hat{H}$ when evaluated at $\alpha_{\min}$. Defined by eq. (3.11a).	61, 62
$\hat{H}$	Shorthand for $\hat{H}(\alpha \mid \beta, T)$.	61, 62, 64, 67, 88, 122
$\hat{\mathbf{\Omega}}'$	Outgoing direction of scattered neutron	13, 16, 17, 36, 59, 123
$\hat{\mathbf{\Omega}}_{\mathrm{T}}$	Direction of flight of target nuclei in the lab frame	30, 31
$\hat{\mathbf{\Omega}}$	Incident direction of neutron	7–14, 30, 31, 36, 123
$\hbar$	Reduced Planck constant. Equal to $h/2\phi$ where $h := 6.626\,070\,15 \times 10^{-34}\,\mathrm{J\,s}$	33, 36
λ	Generic variable representing a parameter to be perturbed.	x–xii, 44, 45, 74–76, 92, 93, 103, 105, 119, 135, 139, 144
$\mathbb{E}$	The expected value of a random variable X is denoted $\mathbb{E}[X]$	15, 41, 42, 44, 76, 102
$\mathbb{V}$	The variance value of a random variable X is denoted $\mathbb{V}[X]$	42, 43
$\mathcal{A}$	Angle kernel. Defined by eq. (2.38d).	17, 93

Symbol	Description	Page List
$\mathscr{E}$	Energy kernel. Defined by eq. (2.38c).	17, 93
$\mathscr{R}$	Reaction kernel. Defined by eq. (2.38b).	17, 92
$\mathscr{T}$	Transmission kernel. Defined by eq. (2.38a).	17, 92
μ	Scattering cosine. Defined by $\hat{\mathbf{\Omega}} \cdot \hat{\mathbf{\Omega}}'$	xii, 36, 37, 59, 93, 95–97
ϕ	Scalar flux. Defined by eq. (2.15).	11
ψ	Angular flux. Defined by eq. (2.8).	9–14
σ_{b}	Characteristic bound scattering cross section	36, 60
σ_{coh}	Coherent scattering cross section. Defined by eq. (2.102)	34–36, 38
σ_{inc}	Incoherent scattering cross section. Defined by eq. (2.103)	34–36, 38
σ_{s}	Scattering cross section. Defined as a function of $S(\alpha, \beta \mid T)$ by eq. (3.3).	xii, xiii, 25–28, 32, 34–36, 38, 60, 64, 66, 69–71, 88, 89, 92, 103, 108, 111
σ	Microscopic cross section	10, 24–29
$\boldsymbol{\kappa}$	Scattering vector $\boldsymbol{k} - \boldsymbol{k}'$	33, 35, 36, 38, 120
$\boldsymbol{R}$	Position vector of a nucleus	33, 35
$\boldsymbol{k}$	Neutron wave vector	33, 123
$\mathbf{v}_{\mathrm{CM}}$	Center of Mass (COM) velocity. Defined by eq. (2.95)	31
$\mathbf{v}_{\mathrm{T}}$	Velocity of target nuclei in the lab frame	24, 25, 27, 28, 30, 31, 124
$\mathbf{v}_{\mathrm{n,CM}}$	Velocity of incident neutron in the center of mass frame	27, 31
$\mathbf{v}_{\mathrm{n}}$	Velocity of incident neutron in the lab frame	24, 25, 27, 28, 30, 31, 124
$\mathbf{x}$	Spatial position of neutron	7, 9–14, 16–18
ξ	A continuous uniform random variable in $[0, 1)$	17, 18, 20, 61, 71, 87, 88, 136–139
$f(\alpha, \beta \mid E, T)$	Bivariate PDF in α and β. Defined by eq. (3.4).	60, 124

Symbol	Description	Page List
f	Generic variable representing a PDF. Defined by eq. (3.29). Not to be confused with $f(\alpha, \beta \mid E, T)$.	19–23, 77–83, 85–87, 136–139, 144
$g(\beta \mid E, T)$	Marginal PDF in β. Defined by eq. (3.6a).	60
$h(\alpha \mid \beta, E, T)$	Conditional PDF in α given β. Defined by eq. (3.6b).	60
k_B	Boltzmann constant. Defined to be exactly $1.380\,649 \times 10^{-23}\,\mathrm{J\,K^{-1}}$. In this dissertation, the more convenient quantity $8.617\,333\,262 \times 10^{-5}\,\mathrm{eV\,K^{-1}}$ is used.	24–28, 36, 37, 59, 60, 92, 93
m_n	Mass of a neutron. Approximately equal to $1.674\,927\,498\,04 \times 10^{-27}\,\mathrm{kg}$.	24–26, 28, 119
n	Neutron angular density. Discussed in eq. (2.7).	9, 10, 14
v_T	Speed $\|\mathbf{v}_T\|$ of target nuclide in the lab frame	24, 25, 28–30
v_n	Speed $\|\mathbf{v}_n\|$ of incident neutron in the lab frame	24–26, 28–30
v_p	The most probable velocity or mode of a Maxwell-Boltzmann distribution. Defined by eq. (2.82).	28, 29
v_r	Relative velocity between incident neutron and target nucleus. Defined by eq. (2.62).	24
x	Generic variable representing realization of a random variable. In this work it typically refers to β or α.	77–88, 136–144

Bibliography

[1] G. Buffon, *Histoire de L'Académie Royale des Sciences*, pp. 43–45, 1733.

[2] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335–341, Sep. 1949.

[3] H. Kahn and A. W. Marshall, "Methods of reducing sample size in monte carlo computations," *Journal of the Operations Research Society of America*, vol. 1, no. 5, pp. 263–278, 1953. (visited on 08/25/2023).

[4] I. Lux and L. Koblinger, *Monte Carlo Particle Transport Methods*. CRC Press, 1991.

[5] E. R. Woodcock, T. Murphy, P. J. Hemmings, and T. C. Longworth, "Techniques used in the gem code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry," Argonne National Laboratory, Tech. Rep. ANL-7050, 1965.

[6] J. Spanier and E. M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley, 1969.

[7] R. R. Coveyou, R. R. Bate, and R. K. Osborn, "Effect of moderator temperature upon neutron flux in infinite, capturing medium," *Journal of Nuclear Energy*, vol. 2, no. 3, pp. 153–167, 1956.

[8] E. M. Gelbard, "Epithermal scattering in VIM," Argonne National Laboratory, Tech. Rep. FRA-TM-123, 1979.

[9] P. K. Romano, N. E. Horelik, B. R. Herman, A. G. Nelson, B. Forget, and K. Smith, "Openmc: A state-of-the-art Monte Carlo code for research and development," *Annals of Nuclear Energy*, vol. 82, pp. 90–97, 2015, ISSN: 0306-4549. DOI: 10.1016/j.anucene.2014.07.048.

[10] C. J. Everett and E. D. Cashwell, "Third Monte Carlo sampler. revision and extension of samplers I and II," Los Alamos National Laboratory, Tech. Rep. LA-9721-MS, 1983.

[11] B. Becker, R. Dagan, and G. Lohnert, "Proof and implementation of the stochastic formula for ideal gas, energy eependent scattering kernel," *Annals of Nuclear Energy*, vol. 36, no. 4, pp. 470–474, 2009. DOI: 10.1016/j.anucene.2008.12.001.

[12] G. L. Squires, *Introduction to the theory of thermal neutron scattering*. Cambridge University Press, 1978.

[13] L. Van Hove, "Correlations in space and time and born approximation scattering in systems of interacting particles," *Phys. Rev.*, vol. 95, pp. 249–262, 1 Jul. 1954. DOI: 10.1103/PhysRev.95.249.

[14] M. Mattes and J. Keinert, "Thermal neutron scattering data for the moderator materials H_2O, D_2O, and ZrH_x in ENDF-6 format and as ACE library for MCNP(X) codes," Institute for Nuclear Technology and Energy Systems (IKE), Tech. Rep. INDC(NDS)-0470, 2005.

[15] C. T. Ballinger, "The direct $S(\alpha, \beta)$ method for thermal neutron scattering," in *Mathematics and Computations, Reactor Physics, and Environmental Analyses: Proceedings of the International Conference, Portland, Oregon, April 30 – May 4, 1995*, vol. 1, American Nuclear Society, 1995, pp. 134–143.

[16] D. Brown, M. Chadwick, R. Capote, *et al.*, "ENDF/B-VIII.0: The 8^{th} major release of the nuclear reaction data library with CIELO-project cross sections, new standards and thermal scattering data," *Nuclear Data Sheets*, vol. 148, pp. 1–142, Feb. 2018. DOI: 10.1016/j.nds.2018.02.001.

[17] A. Koning, R. Forrest, M. Kellett, R. Mills, H. Henriksson, and Y. Rugama, "The JEFF-3.1 nuclear data library," Nuclear Energy Agency, Tech. Rep. 6190, 2006.

[18] O. Iwamoto, N. Iwamoto, S. Kunieda, *et al.*, "Japanese evaluated nuclear data library version 5: JENDL-5," *Journal of Nuclear Science and Technology*, vol. 60, no. 1, pp. 1–60, 2023. DOI: 10.1080/00223131.2022.2141903.

[19] Marquez Damian, Jose Ignacio, Granada, Jose Rolando, Cantargi, Florencia, and Roubtsov, Danila, "New evaluation of thermal neutron scattering libraries for light and heavy water," *EPJ Web Conf.*, vol. 146, p. 13 001, 2017.

[20] J. Márquez Damián, J. Granada, and D. Malaspina, "CAB models for water: A new evaluation of the thermal neutron scattering laws for light and heavy water in ENDF-6 format," *Annals of Nuclear Energy*, vol. 65, pp. 280–289, 2014.

[21] J. U. Koppel, J. R. Triplett, and Y. D. Naliboff, "GASKET a unified code for thermal neutron scattering," General Atomic, Tech. Rep. GA-7417, 1966.

[22] J. Keinert and M. Mattes, "JEF-1 scattering law data," IKE Stuttgart, Tech. Rep. IKE 6-147, 1984.

[23] R. E. MacFarlane, "New thermal neutron scattering files for ENDF/B-VI release 2," Los Alamos National Laboratory, Tech. Rep. LA-12639-MS, 1994.

[24] M. A. González and J. L. F. Abascal, "A flexible model for water based on TIP4P/2005," *The Journal of Chemical Physics*, vol. 135, no. 22, p. 224 516, Dec. 2011. DOI: 10.1063/1.3663219.

[25] C. W. Chapman, G. Arbanas, A. I. Kolesnikov, *et al.*, "Methodology for generating covariance data of thermal neutron scattering cross sections," *Nuclear Science and Engineering*, vol. 195, no. 1, pp. 13–32, 2021.

[26] G. E. Granroth, A. I. Kolesnikov, T. E. Sherline, *et al.*, "SEQUOIA: A newly operating chopper spectrometer at the SNS," *Journal of Physics: Conference Series*, vol. 251, no. 1, p. 012 058, Nov. 2010. DOI: 10.1088/1742-6596/251/1/012058.

[27] L. L. Briggs, "Uncertainty quantification approaches for advanced reactor analyses," Argonne National Laboratory, Tech. Rep. ANL-GenIV-110, 2008.

[28] F. D'Auria, C. Camargo, and O. Mazzantini, "The Best Estimate Plus Uncertainty BEPU approach in licensing of current nuclear reactors," *Nuclear Engineering and Design*, vol. 248, pp. 317–328, 2012. DOI: 10.1016/j.nucengdes.2012.04.002.

[29] D. G. Cacuci, *Sensitivity and uncertianty analysis*. Chapman & Hall, 2003.

[30] G. Mikhailov, "Monte-Carlo calculation of derivatives of functionals from the solution of the transfer equation according to the parameters of the system," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 274–281, 1967. DOI: 10.1016/0041-5553(67)90162-0.

[31] H. Rief, "Stochastic perturbation analysis applied to neutral particle transport," in *Advances in Nuclear Science and Technology*. 1996, pp. 69–140. DOI: 10.1007/0-306-47810-2_2.

[32] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 5595–5637, Jan. 2017, ISSN: 1532-4435.

[33] D. Goldberg, "What every computer scientist should know about floating-point arithmetic," *ACM Comput. Surv.*, vol. 23, no. 1, pp. 5–48, Mar. 1991. DOI: 10.1145/103162.103163.

[34] G. F. Corliss, "Applications of differentiation arithmetic," in *Reliability in Computing: The Role of Interval Methods in Scientific Computing*. Academic Press Professional, Inc., 1988, pp. 127–148.

[35] R. E. Wengert, "A simple automatic derivative evaluation program," *Commun. ACM*, vol. 7, no. 8, pp. 463–464, Aug. 1964. DOI: 10.1145/355586.364791.

[36] Clifford, "Preliminary sketch of biquaternions," *Proceedings of The London Mathematical Society*, pp. 381–395, 1871.

[37] A. M. M. Leal, *autodiff, a modern, fast and expressive C++ library for automatic differentiation*, https://autodiff.github.io, 2018. [Online]. Available: https://autodiff.github.io.

[38] A. Koning and D. Rochman, "Towards sustainable nuclear energy: Putting nuclear physics to work," *Annals of Nuclear Energy*, vol. 35, no. 11, pp. 2024–2030, 2008, ISSN: 0306-4549. DOI: 10.1016/j.anucene.2008.06.004.

[39] D. Brown, "Detailed requirements for a next generation nuclear data structure," Brookhaven National Laboratory, Tech. Rep. BNL-112394-2016-IR, Jul. 2016.

[40] A. G. Tumulak, H. Park, V. Sobes, W. S. Yang, and B. C. Kiedrowski, "Progress on the development of thermal scattering covariance formats and processing tools," in *Transactions of the American Nuclear Society*, vol. 121, Nov. 2019, pp. 1377–1380.

[41] A. G. Tumulak, D. Wiarda, A. Holcomb, and B. C. Kiedrowski, "Thermal scattering covariance data processing and compression," in *Transactions of the American Nuclear Society*, vol. 123, Nov. 2020, pp. 136–139.

[42] D. A. Brown, B. R. Beck, J. L. Conlin, W. Haeck, C. M. Mattoon, and D. Wiarda, "Wpec subgroup-38 final report part ii: Specifications for a new database structure," Lawrence Livermore National Laboratory, Tech. Rep. LLNL-TR-774621, May 2019.

[43] G. Noguere, *1000 H in H2O files based on JEFF-311*, https://www.oecd-nea.org/download/wpec/sg42/, 2018.

[44] A. T. Pavlou and W. Ji, "On-the-fly sampling of temperature-dependent thermal neutron scattering data for Monte Carlo simulations," *Annals of Nuclear Energy*, vol. 71, pp. 411–426, 2014. DOI: 10.1016/j.anucene.2014.04.028.

[45] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual Review of Fluid Mechanics*, vol. 25, no. 1, pp. 539–575, 1993.

[46] A. G. Buchan, A. A. Calloo, M. G. Goffin, *et al.*, "A POD reduced order model for resolving angular direction in neutron/photon transport problems," *Journal of Computational Physics*, vol. 296, pp. 138–157, 2015.

[47] P. German and J. C. Ragusa, "Reduced-order modeling of parameterized multi-group diffusion k-eigenvalue problems," *Annals of Nuclear Energy*, vol. 134, no. C, pp. 144–157, 2019.

[48] A. G. Tumulak and B. C. Kiedrowski, "A continuous-temperature thermal scattering treatment using proper orthogonal decomposition," in *Transactions of the American Nuclear Society*, vol. 126, Jun. 2022, pp. 1377–1380.

[49] P. Barry, *Encyclopaedia of Mathematics*, M. Hazewinkel, Ed. Kluwer Academic Publishers, 1997, vol. 1, p. 119.

[50] E. F. Eisele, *Encyclopaedia of Mathematics*, M. Hazewinkel, Ed. Kluwer Academic Publishers, 1997, vol. 1, pp. 119–120.

[51] F. N. Fritsch and J. Butland, "A method for constructing local monotone piecewise cubic interpolants," *SIAM Journal on Scientific and Statistical Computing*, vol. 5, no. 2, pp. 300–304, 1984.

[52] J. Butland, "A method of interpolating reasonable-shaped curves through any data," *Proc. Computer Graphics*, vol. 80, pp. 409–422, 1980.

[53] C. Yuksel, "A fast & robust solution for cubic & higher-order polynomials," in *ACM SIGGRAPH 2022 Talks*, ser. SIGGRAPH '22, Vancouver, BC, Canada: Association for Computing Machinery, 2022.

[54] Y. Wang, Z. Sun, D. Song, and A. Hero, "Kronecker-structured covariance models for multiway data," *Statistics Surveys*, vol. 16, pp. 238–270, 2022. DOI: 10.1214/22-SS139.

[55] J. L. Conlin and P. Romano, "A compact ENDF (ace) format specification," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-19-29016, Sep. 2019. DOI: 10.2172/1561065.

[56] A. Trkov and D. A. Brown, "ENDF-6 formats manual: Data formats and procedures for the evaluated nuclear data files," Jan. 2018. DOI: 10.2172/1425114. [Online]. Available: https://www.osti.gov/biblio/1425114.

[57] J. A. Kulesza, T. R. Adams, J. C. Armstrong, *et al.*, "MCNP® Code Version 6.3.0 Theory & User Manual," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-22-30006, Rev. 1, Sep. 2022. DOI: 10.2172/1889957. [Online]. Available: https://www.osti.gov/biblio/1889957.

[58] B. K. Horn, "Numerically stable method for solving quadratic equations," 2005. [Online]. Available: https://people.csail.mit.edu/bkph/articles/Quadratics.pdf.

Appendix

A.1 GNDS Evaluation of H in H_2O

```
<?xml version="1.0"?>
<reactionSuite evaluation="demo tsl evaluation" format="2.0" projectile="n"
 ↪ projectileFrame="lab">
  <styles>
    <evaluated date="2021-12-19" label="eval" library="demo library">
      <projectileEnergyDomain max="3.999930" min="0" unit="eV" />
      <temperature value="293.6" unit="K" />
      <documentation>
        <authors>
          <author name="Aaron G. Tumulak" />
        </authors>
      </documentation>
    </evaluated>
  </styles>
  <reactions>
    <reaction ENDF_MT="4" label="tsl">
      <doubleDifferentialCrossSection label="eval">
        <thermalNeutronScatteringLaw_incoherentInelastic label="eval" pid="n" frame="lab"
         ↪ primaryScatterer="?">
          <scatteringAtoms>
            <scatteringAtom pid="H" primaryScatterer="true" numberPerMolecule="2">
              <mass label="H1 atomic weight ratio" value="9.991700-1" />
              <e_max unit="eV" value="3.999930+0" />
              <boundAtomCrossSection unit="b" value="81.98005757" />
              <selfScatteringKernel symmetric="true">
                <gridded3d>
                  <axes>
                    <grid label="temperature" index="3" interpolation="lin-log"
                     ↪ style="points" unit="K">
                      <values>293.6</values>
                    </grid>
                    <grid label="beta" index="2" interpolation="lin-log" style="points">
                      <values>0.000000 0.300000 0.800000 ... </values>
                    </grid>
                    <grid label="alpha" index="1" interpolation="lin-log" style="points">
                      <values>0.000500 0.200000 0.500000 ... </values>
                    </grid>
                    <axis label="S" index="0" />
                  </axes>
                  <array shape="1,26,19">
                    <values>4.672386e+00 3.098872e-01 1.923857e-01 ... </values>
                  </array>
                </gridded3d>
              </selfScatteringKernel>
            </scatteringAtom>
          </scatteringAtoms>
        </thermalNeutronScatteringLaw_incoherentInelastic>
      </doubleDifferentialCrossSection>
    </reaction>
  </reactions>
</reactionSuite>
```

Listing 1: Evaluation for water. Contents of <value> elements are truncated.

```xml
<?xml version="1.0"?>
<covarianceSuite evaluation="demo tsl evaluation" projectile="n"
 ↪  interaction="thermalNeutronScatteringLaw" version="2.0">
  <externalFiles>
    <externalFile label="mean_tsl" path="./reactionSuite.xml" />
  </externalFiles>
  <styles>
    <evaluated date="2021-12-19" label="eval" library="demo library">
      <projectileEnergyDomain max="3.999930" min="0" unit="eV" />
      <temperature value="293.6" unit="K" />
      <documentation>
        <authors>
          <author name="Aaron G. Tumulak" />
        </authors>
      </documentation>
    </evaluated>
  </styles>
  <covarianceSections>
    <covarianceSection label="beta=0.000000 beta_p=0.000000" crossTerm="false">
      <!--covarianceSection subtree-->
    </covarianceSection>
    <covarianceSection label="beta=0.300000 beta_p=0.000000" crossTerm="true">
      <!--covarianceSection subtree-->
    </covarianceSection>
    <covarianceSection label="beta=0.300000 beta_p=0.300000" crossTerm="false">
      <!--covarianceSection subtree-->
    </covarianceSection>
    <covarianceSection label="beta=0.800000 beta_p=0.000000" crossTerm="true">
      <!--covarianceSection subtree-->
    </covarianceSection>
    <covarianceSection label="beta=0.800000 beta_p=0.300000" crossTerm="true">
      <!--covarianceSection subtree-->
    </covarianceSection>
    <covarianceSection label="beta=0.800000 beta_p=0.800000" crossTerm="false">
      <!--covarianceSection subtree-->
    </covarianceSection>
    <!--continued for all covariance submatrices-->
  </covarianceSections>
</covarianceSuite>
```

Listing 2: Collection of `covarianceSections`

```xml
<covarianceSection label="beta=0.300000 beta_p=0.000000" crossTerm="true">
  <rowData ENDF_MFMT="7,4" href="path/to/gridded3d" dimension="1">
    <slices>
      <slice domainValue="293.6" domainUnit="K" dimension="3" />
      <slice domainValue="0.300000" dimension="2" />
    </slices>
  </rowData>
  <columnData ENDF_MFMT="7,4" href="path/to/gridded3d" dimension="1">
    <slices>
      <slice domainValue="293.6" domainUnit="K" dimension="3" />
      <slice domainValue="0.000000" dimension="2" />
    </slices>
  </columnData>
  <covarianceMatrix label="eval" type="absolute">
    <gridded2d>
      <axes>
        <grid label="alpha" index="2" style="points">
          <link href="path/to/gridded3d/axes/grid[@label='alpha']/values" />
        </grid>
        <grid label="alpha_p" index="1" style="points">
          <link href="path/to/gridded3d/axes/grid[@label='alpha']/values" />
        </grid>
        <axis label="covariances" index="0" />
      </axes>
      <array shape="19, 19">
        <values>4.181457e-06 -6.569185e-08 -3.922927e-08 ... </values>
      </array>
    </gridded2d>
  </covarianceMatrix>
</covarianceSection>
```

Listing 3: Contents of a single `covarianceSection`

A.2 Logarithmic Derivatives

This dissertation makes extensive use of the *logarithmic derivative* operator D_ℓ which, given a differentiable function $f(\lambda)$, takes the form

$$D_\ell(f(\lambda)) := \frac{1}{f}\frac{df}{d\lambda} \tag{1}$$

where the term in the inner parenthesis may be used to explicitly denote what the derivatives are with respect to. The logarithmic derivative of many functions can often be greatly simplified by the use of certain identities. Some useful ones are listed here.

- Constant Multiplication ($dc/d\lambda = 0$)

$$D_\ell(cf) = D_\ell(f) \tag{2}$$

- Sum Rule

$$D_\ell\left(\sum_i f_i\right) = \frac{\sum_i f_i D_\ell(f_i)}{\sum_i f_i} \tag{3}$$

- Product Rule

$$D_\ell\left(\prod_i f_i\right) = \sum_i D_\ell(f_i) \tag{4}$$

- Quotient Rule

$$D_\ell\left(\frac{f}{g}\right) = D_\ell(f) - D_\ell(g) \tag{5}$$

- Power Rule ($dn/d\lambda = 0$)

$$D_\ell(\lambda^n) = \frac{n}{\lambda} \tag{6}$$

- Chain Rule ($f(\lambda) = f(g(\lambda))$)

$$D_\ell(f(\lambda)) = gD_\ell(f(g))\, D_\ell(g(\lambda)) \tag{7}$$

A.3 Quadratic CDF Sampling

Given a sampled Cumulative Distribution Function (CDF) value ξ, we wish to identify the corresponding x. Once m satisfying $F_{m-1} \leq \xi < F_m$ is found, we can insert ξ into the lefthand side of eq. (3.40) and rearrange to obtain

$$0 = \frac{1}{2}\left(f_m - f_{m-1}\right)\left(r^{(m)}\right)^2 + f_{m-1}r^{(m)} - \frac{\xi - F_{m-1}}{x_m - x_{m-1}} \tag{8}$$

which is a quadratic equation in $r^{(m)}$. We wish to prove that the discriminant satisfies

$$f_{m-1}^2 + 2\left(f_m - f_{m-1}\right)\frac{\xi - F_{m-1}}{x_m - x_{m-1}} \geq 0\,. \tag{9}$$

Recall that we have selected m so that $F_{m-1} \leq \xi$ and can prepare data so that it is monotonic nondecreasing ($x_m - x_{m-1} \geq 0$) allowing us to establish that

$$\frac{\xi - F_{m-1}}{x_m - x_{m-1}} \geq 0\,. \tag{10}$$

We handle two cases separately:

- $f_m - f_{m-1} \geq 0$: Equation (9) is trivially true.
- $f_m - f_{m-1} < 0$: Equation (9) is true if and only if

$$f_{m-1}^2 \geq -2\left(f_m - f_{m-1}\right)\frac{\xi - F_{m-1}}{x_m - x_{m-1}}\,. \tag{11}$$

  Since we required that $\xi < F_m$, we can say that if

$$f_{m-1}^2 \geq -2\left(f_m - f_{m-1}\right)\frac{F_m - F_{m-1}}{x_m - x_{m-1}} \tag{12}$$

holds, then eq. (11) also holds. Using eq. (3.32), eq. (12) can be expressed

$$f_{m-1}^2 \geq -\left(f_m - f_{m-1}\right)\left(f_m + f_{m-1}\right) = -f_m^2 + f_{m-1}^2 \tag{13}$$

which is true so eq. (9) holds.

If eq. (9) holds then there is at least one real root. Recalling that in eq. (8) the constant term is $-\frac{\xi - F_{m-1}}{x_m - x_{m-1}} \leq 0$ and the linear term is $f_{m-1} \geq 0$, we wish to find $r^{(m)} \in [0, 1)$. For a nonnegative linear term, there are two numerically stable [58] solutions:

$$r_1^{(m)} = \frac{-f_{m-1} - \sqrt{f_{m-1}^2 + 2\left(f_m - f_{m-1}\right)\frac{\xi - F_{m-1}}{x_m - x_{m-1}}}}{f_m - f_{m-1}} \tag{14a}$$

$$r_2^{(m)} = \frac{-2\frac{\xi - F_{m-1}}{x_m - x_{m-1}}}{-f_{m-1} - \sqrt{f_{m-1}^2 + 2\left(f_m - f_{m-1}\right)\frac{\xi - F_{m-1}}{x_m - x_{m-1}}}}. \tag{14b}$$

There are two cases for the quadratic term:

- $f_m - f_{m-1} \geq 0$: Equation (14a) is $r_1^{(m)} \leq 0$ so it is nonviable. Equation (14b) is $r_2^{(m)} \geq 0$ so it is the only possible choice.

- $f_m - f_{m-1} < 0$: Equations (14a) and (14b) are both nonngative. However, defining

$$\tilde{r}_1^{(m)} := 1 - r_1^{(m)} = \frac{f_m + \sqrt{f_{m-1}^2 + 2\left(f_m - f_{m-1}\right)\frac{\xi - F_{m-1}}{x_m - x_{m-1}}}}{f_m - f_{m-1}} \tag{15}$$

  we note that $\tilde{r}_1^{(m)} \leq 0$ which implies that $r_1^{(m)} \geq 1$ so eq. (14a) is again nonviable and eq. (14b) is the only possible choice.

Thus we deduce that the correct solution is

$$r^{(m)} = \frac{2\frac{\xi - F_{m-1}}{x_m - x_{m-1}}}{f_{m-1} + \sqrt{f_{m-1}^2 + 2\left(f_m - f_{m-1}\right)\frac{\xi - F_{m-1}}{x_m - x_{m-1}}}}. \tag{16}$$

Inserting eq. (3.35a) into the lefthand side of eq. (16) and solving for x, we get

$$x = x_{m-1} + \frac{2\left(\xi - F_{m-1}\right)}{f_{m-1} + \sqrt{f_{m-1}^2 + 2\left(\xi - F_{m-1}\right)\frac{f_m - f_{m-1}}{x_m - x_{m-1}}}} \tag{17}$$

as the value of x that is sampled with CDF ξ.

A.3.1 Sensitivities

For sensitivity analysis, we are interested in computing the logarithmic derivative of eq. (17). Adopting notation from eq. (1) the logarithmic derivative of eq. (17) is

$$D_\ell\left(x\right) = D_\ell\left(x_{m-1} + \frac{2\left(\xi - F_{m-1}\right)}{f_{m-1} + \sqrt{f_{m-1}^2 + 2\left(\xi - F_{m-1}\right)\frac{f_m - f_{m-1}}{x_m - x_{m-1}}}}\right) \tag{18}$$

for which eq. (3) can be applied. The logarithmic derivative of the first term is $D_\ell\left(x_{m-1}\right)$ and the second term is

$$\begin{aligned} & D_\ell\left(\frac{2\left(\xi - F_{m-1}\right)}{f_{m-1} + \sqrt{f_{m-1}^2 + 2\left(\xi - F_{m-1}\right)\frac{f_m - f_{m-1}}{x_m - x_{m-1}}}}\right) \\ = & D_\ell\left(\frac{1}{f_{m-1} + \sqrt{f_{m-1}^2 + 2\left(\xi - F_{m-1}\right)\frac{f_m - f_{m-1}}{x_m - x_{m-1}}}}\right) \\ = & -D_\ell\left(f_{m-1} + \sqrt{f_{m-1}^2 + 2\left(\xi - F_{m-1}\right)\frac{f_m - f_{m-1}}{x_m - x_{m-1}}}\right) \end{aligned} \tag{19}$$

where eq. (2) then eq. (5) were applied. The logarithmic derivative of the first term is $D_\ell\left(f_{m-1}\right)$ and the second term is

$$\begin{aligned} & D_\ell\left(\sqrt{f_{m-1}^2 + 2\left(f_m - f_{m-1}\right)\frac{\xi - F_{m-1}}{x_m - x_{m-1}}}\right) \\ = & \frac{1}{2} D_\ell\left(f_{m-1}^2 + 2\left(f_m - f_{m-1}\right)\frac{\xi - F_{m-1}}{x_m - x_{m-1}}\right) \end{aligned} \tag{20}$$

where eqs. (6) and (7) were used. The logarithmic derivative of the first term in eq. (20) is $2D_\ell(f_{m-1})$ and the second term is

$$D_\ell\left(2\left(f_m - f_{m-1}\right)\frac{\xi - F_{m-1}}{x_m - x_{m-1}}\right) = D_\ell\left(f_m - f_{m-1}\right) - D_\ell\left(x_m - x_{m-1}\right) . \tag{21}$$

Thus, by making use of eqs. (18) to (21) and the sum rule eq. (3), we can express eq. (18) entirely in terms of $D_\ell\left(x_m\right)$, $D_\ell\left(x_{m-1}\right)$, $D_\ell\left(f_m\right)$, and $D_\ell\left(f_{m-1}\right)$.

A.3.2 Probability Density Function

We now discuss how to evaluate the logarithmic derivative of the Probability Density Function (PDF) f on any CDF gridpoint m. Beginning with eq. (3.34), we have

$$D_\ell\left(f_m\right) = D_\ell\left(\left(-1\right)^m \sum_{m'=0}^{m} \left(-1\right)^{m'} c_{m'}\right) . \tag{22}$$

Since properties of logarithmic derivatives are not very helpful for sums and f_m and each $c_{m'}$ are computed during the sampling process anyways, we will just compute the derivative of eq. (3.34) directly with

$$\frac{df_m}{d\lambda} = \left(-1\right)^m \sum_{m'=0}^{m} \left(-1\right)^{m'} \frac{dc_{m'}}{d\lambda} \tag{23}$$

where the derivative of $c_{m'}$ can be obtained by differentiating eq. (3.33) to obtain

$$\frac{dc_m}{d\lambda} := \begin{cases} \frac{df_0}{d\lambda}, & m = 0 \\ -2\frac{F_m - F_{m-1}}{\left(x_m - x_{m-1}\right)^2} = -\frac{c_m}{x_m - x_{m-1}}, & m = 1, 2, \ldots, M \end{cases} \tag{24}$$

where the terms for $m > 0$ are computed during the sampling of x.

The case for $m = 0$ is handled next. We separate computing the derivative of eq. (3.58) into distinct steps. First, we compute the derivative of the numerator by splitting the sum-

mand into the product of two terms, differentiating each term separately, then combining the derivatives using the product rule. Then, we compute the derivative of the denominator. Finally, we combine the differentiated numerator and denominator through the quotient rule.

The derivative of the first term of the summand of the numerator of eq. (3.58) is given by

$$
\begin{aligned}
&\frac{d}{dx_m}\left(2\left(\sum_{m''=1}^{m'}(-1)^{m''}c_{m''}\right)-(-1)^{m'}c_{m'}\right)\\
&=2\left(\sum_{m''=1}^{m'}(-1)^{m''}\frac{dc_{m''}}{dx_m}\right)-(-1)^{m'}\frac{dc_{m'}}{dx_m}\\
&=2\left(\sum_{m''=1}^{m'}(-1)^{m''}\frac{d}{dx_m}\left(2\frac{F_{m''}-F_{m''-1}}{x_{m''}-x_{m''-1}}\right)\right)-(-1)^{m'}\frac{d}{dx_m}\left(2\frac{F_{m'}-F_{m'-1}}{x_{m'}-x_{m'-1}}\right)\\
&=2\left(\sum_{m''=1}^{m'}(-1)^{m''}2\left(F_{m''}-F_{m''-1}\right)\frac{d}{dx_m}\left(x_{m''}-x_{m''-1}\right)^{-1}\right)\\
&\quad-(-1)^{m'}2\left(F_{m'}-F_{m'-1}\right)\frac{d}{dx_m}\left(x_{m'}-x_{m'-1}\right)^{-1}\\
&=2\left(\sum_{m''=1}^{m'}(-1)^{m''}2\left(-1\right)\frac{F_{m''}-F_{m''-1}}{\left(x_{m''}-x_{m''-1}\right)^2}\frac{d}{dx_m}\left(x_{m''}-x_{m''-1}\right)\right)\\
&\quad-(-1)^{m'}2\left(-1\right)\frac{F_{m'}-F_{m'-1}}{\left(x_{m'}-x_{m'-1}\right)^2}\frac{d}{dx_m}\left(x_{m'}-x_{m'-1}\right)\\
&=-2\left(\sum_{m''=1}^{m'}(-1)^{m''}2\frac{F_{m''}-F_{m''-1}}{\left(x_{m''}-x_{m''-1}\right)^2}\left(\delta_{m,m''}-\delta_{m,m''-1}\right)\right)\\
&\quad+(-1)^{m'}3\frac{F_{m'}-F_{m'-1}}{\left(x_{m'}-x_{m'-1}\right)^2}\left(\delta_{m,m'}-\delta_{m,m'-1}\right)\\
&=-2\left(\sum_{m''=1}^{m'}\frac{(-1)^{m''}c_{m''}}{x_{m''}-x_{m''-1}}\left(\delta_{m,m''}-\delta_{m,m''-1}\right)\right)+\frac{(-1)^{m'}c_{m'}}{x_{m'}-x_{m'-1}}\left(\delta_{m,m'}-\delta_{m,m'-1}\right)
\end{aligned}
\tag{25}
$$

where we have used eq. (3.33) in the third and last line. The derivative of the second term

of the summand of the numerator of eq. (3.58) is given by

$$\begin{aligned}
&\frac{d}{dx_m}\left(\left(x_{m'}-x_{m'-1}\right)^4\right)\\
&=4\left(x_{m'}-x_{m'-1}\right)^3\frac{d}{dx_m}\left(x_{m'}-x_{m'-1}\right)\\
&=4\left(x_{m'}-x_{m'-1}\right)^3\left(\delta_{m,m'}-\delta_{m,m'-1}\right).
\end{aligned} \tag{26}$$

Using the linearity of the derivative operator and the product rule, the derivative of the numerator of eq. (3.58) is

$$\begin{aligned}
&\frac{d}{dx_m}\left(\sum_{m'=1}^{M}\left(2\left(\sum_{m''=1}^{m'}\left(-1\right)^{m''}c_{m''}\right)-\left(-1\right)^{m'}c_{m'}\right)\left(x_{m'}-x_{m'-1}\right)^4\right)\\
&=\sum_{m'=1}^{M}\Bigg(\frac{d}{dx_m}\left(2\left(\sum_{m''=1}^{m'}\left(-1\right)^{m''}c_{m''}\right)-\left(-1\right)^{m'}c_{m'}\right)\left(x_{m'}-x_{m'-1}\right)^4\\
&\qquad+\left(2\left(\sum_{m''=1}^{m'}\left(-1\right)^{m''}c_{m''}\right)-\left(-1\right)^{m'}c_{m'}\right)\frac{d}{dx_m}\left(x_{m'}-x_{m'-1}\right)^4\Bigg).
\end{aligned} \tag{27}$$

Inserting eqs. (25) and (26), we have

$$\sum_{m'=1}^{M}\Bigg(\quad \left(-2\left(\sum_{m''=1}^{m'}\frac{(-1)^{m''}c_{m''}}{x_{m''}-x_{m''-1}}\left(\delta_{m,m''}-\delta_{m,m''-1}\right)\right)+\frac{(-1)^{m'}c_{m'}}{x_{m'}-x_{m'-1}}\left(\delta_{m,m'}-\delta_{m,m'-1}\right)\right)$$
$$\times\left(x_{m'}-x_{m'-1}\right)^{4}$$
$$+\left(2\left(\sum_{m''=1}^{m'}(-1)^{m''}c_{m''}\right)-(-1)^{m'}c_{m'}\right)\left(4\left(x_{m'}-x_{m'-1}\right)^{3}\left(\delta_{m,m'}-\delta_{m,m'-1}\right)\right)\Bigg) \tag{28a}$$

$$=\sum_{m'=1}^{M}\Bigg(-2\left(\sum_{m''=1}^{m'}\frac{(-1)^{m''}c_{m''}}{x_{m''}-x_{m''-1}}\left(\delta_{m,m''}-\delta_{m,m''-1}\right)\right)\left(x_{m'}-x_{m'-1}\right)^{4}$$
$$+(-1)^{m'}c_{m'}\left(x_{m'}-x_{m'-1}\right)^{3}\left(\delta_{m,m'}-\delta_{m,m'-1}\right)$$
$$+2\left(\sum_{m''=1}^{m'}(-1)^{m''}c_{m''}\right)4\left(x_{m'}-x_{m'-1}\right)^{3}\left(\delta_{m,m'}-\delta_{m,m'-1}\right)$$
$$-4(-1)^{m'}c_{m'}\left(x_{m'}-x_{m'-1}\right)^{3}\left(\delta_{m,m'}-\delta_{m,m'-1}\right)\Bigg) \tag{28b}$$

$$=\sum_{m'=1}^{M}\Bigg(-2\left(\sum_{m''=1}^{m'}\frac{(-1)^{m''}c_{m''}}{x_{m''}-x_{m''-1}}\left(\delta_{m,m''}-\delta_{m,m''-1}\right)\right)\left(x_{m'}-x_{m'-1}\right)^{4}$$
$$+2\left(\sum_{m''=1}^{m'}(-1)^{m''}c_{m''}\right)4\left(x_{m'}-x_{m'-1}\right)^{3}\left(\delta_{m,m'}-\delta_{m,m'-1}\right)$$
$$-3(-1)^{m'}c_{m'}\left(x_{m'}-x_{m'-1}\right)^{3}\left(\delta_{m,m'}-\delta_{m,m'-1}\right)\Bigg). \tag{28c}$$

We first look at the first term in the summand of eq. (28c), using the Kronecker delta's

sifting property to obtain

$$
\begin{aligned}
&\sum_{m'=1}^{M} -2\left(\sum_{m''=1}^{m'} \frac{(-1)^{m''} c_{m''}}{x_{m''} - x_{m''-1}} \left(\delta_{m,m''} - \delta_{m,m''-1}\right)\right) \left(x_{m'} - x_{m'-1}\right)^4 \\
&= \sum_{m'=1}^{M} -2\left(\frac{(-1)^{m} c_{m} \mathbf{1}_{1\leq m\leq m'}}{x_m - x_{m-1}} - \frac{(-1)^{m+1} c_{m+1} \mathbf{1}_{0\leq m\leq m'-1}}{x_{m+1} - x_m}\right) \left(x_{m'} - x_{m'-1}\right)^4 \qquad (29a) \\
&= -2\frac{(-1)^{m} c_m}{x_m - x_{m-1}} \left(\sum_{m'=1}^{M} \mathbf{1}_{1\leq m\leq m'} \left(x_{m'} - x_{m'-1}\right)^4\right) \\
&\quad +2\frac{(-1)^{m+1} c_{m+1}}{x_{m+1} - x_m} \left(\sum_{m'=1}^{M} \mathbf{1}_{0\leq m\leq m'-1} \left(x_{m'} - x_{m'-1}\right)^4\right) \qquad (29b) \\
&= -2\frac{(-1)^{m} c_m}{x_m - x_{m-1}} \left(\sum_{m'=m}^{M} \left(x_{m'} - x_{m'-1}\right)^4\right) + 2\frac{(-1)^{m+1} c_{m+1}}{x_{m+1} - x_m} \left(\sum_{m'=m+1}^{M} \left(x_{m'} - x_{m'-1}\right)^4\right). \qquad (29c)
\end{aligned}
$$

Likewise, the second term in the summand of eq. (28c) can be expressed

$$
\begin{aligned}
&\sum_{m'=1}^{M} 2\left(\sum_{m''=1}^{m'} (-1)^{m''} c_{m''}\right) 4\left(x_{m'} - x_{m'-1}\right)^3 \left(\delta_{m,m'} - \delta_{m,m'-1}\right) \\
&= 2\left(\sum_{m'=1}^{m} (-1)^{m'} c_{m'}\right) 4\left(x_m - x_{m-1}\right)^3 - 2\left(\sum_{m'=1}^{m+1} (-1)^{m'} c_{m'}\right) 4\left(x_{m+1} - x_m\right)^3 .
\end{aligned} \qquad (30)
$$

The third and last term in the summand of eq. (28c) is just

$$
\begin{aligned}
&\sum_{m'=1}^{M} -3\left(-1\right)^{m'} c_{m'} \left(x_{m'} - x_{m'-1}\right)^3 \left(\delta_{m,m'} - \delta_{m,m'-1}\right) \\
&= -3\left(-1\right)^{m} c_m \left(x_m - x_{m-1}\right)^3 + 3\left(-1\right)^{m+1} c_{m+1} \left(x_{m+1} - x_m\right)^3 .
\end{aligned} \qquad (31)
$$

Inserting eqs. (29c), (30) and (31) into eq. (28c), we have

$$\begin{aligned}
&-2\frac{(-1)^m c_m}{x_m - x_{m-1}}\left(\sum_{m'=m}^{M}\left(x_{m'} - x_{m'-1}\right)^4\right) + 2\frac{(-1)^{m+1} c_{m+1}}{x_{m+1} - x_m}\left(\sum_{m'=m+1}^{M}\left(x_{m'} - x_{m'-1}\right)^4\right) \\
&+2\left(\sum_{m'=1}^{m}(-1)^{m'} c_{m'}\right) 4\left(x_m - x_{m-1}\right)^3 - 2\left(\sum_{m'=1}^{m+1}(-1)^{m'} c_{m'}\right) 4\left(x_{m+1} - x_m\right)^3 \\
&-3(-1)^m c_m \left(x_m - x_{m-1}\right)^3 + 3(-1)^{m+1} c_{m+1}\left(x_{m+1} - x_m\right)^3
\end{aligned} \tag{32}$$

as the derivative of the numerator of eq. (3.58).

Next we compute the derivative of the denominator of eq. (3.58)

$$\begin{aligned}
&\frac{d}{dx_m}\left(2\sum_{m'=1}^{M}\left(x_{m'} - x_{m'-1}\right)^4\right) \\
&= 2\sum_{m'=1}^{M}\frac{d}{dx_m}\left(\left(x_{m'} - x_{m'-1}\right)^4\right) \\
&= 2\sum_{m'=1}^{M} 4\left(x_{m'} - x_{m'-1}\right)^3\left(\delta_{m,m'} - \delta_{m,m'-1}\right) \\
&= 2\left(4\left(x_m - x_{m-1}\right)^3 - 4\left(x_{m+1} - x_m\right)^3\right)
\end{aligned} \tag{33}$$

where eq. (26) was used in the third step. The derivative of the numerator is given by eq. (32) and the denominator by eq. (33) so the quotient rule can be used to calculate $df_0/d\lambda$ as it appears in eq. (24).

A.4 ENDF/B-VIII.0 Evaluation Gridpoints for H in H_2O

Listing 1: β gridpoints

```
0.0000e+00, 5.0170e-06, 6.6893e-06, 8.9190e-06, 1.1892e-05, 1.5856e-05, 2.1141e-05,
2.8189e-05, 3.7585e-05, 5.0113e-05, 6.6817e-05, 8.9090e-05, 1.1879e-04, 1.5838e-04,
2.1118e-04, 2.8157e-04, 3.7542e-04, 5.0056e-04, 6.6742e-04, 8.8989e-04, 1.1865e-03,
1.5820e-03, 2.1094e-03, 2.8125e-03, 3.7500e-03, 5.0000e-03, 1.0000e-02, 1.5000e-02,
2.0000e-02, 2.5000e-02, 3.0000e-02, 4.0000e-02, 5.0000e-02, 6.0000e-02, 7.0000e-02,
8.0000e-02, 1.0000e-01, 1.2500e-01, 1.5000e-01, 1.7500e-01, 2.0000e-01, 2.2500e-01,
2.5000e-01, 3.0000e-01, 3.5000e-01, 4.0000e-01, 4.5000e-01, 5.0000e-01, 5.5000e-01,
6.0000e-01, 6.5000e-01, 7.0000e-01, 7.5000e-01, 8.0000e-01, 8.5000e-01, 9.0000e-01,
9.5000e-01, 1.0000e+00, 1.0500e+00, 1.1000e+00, 1.1500e+00, 1.2000e+00, 1.2500e+00,
1.3000e+00, 1.3500e+00, 1.4000e+00, 1.4500e+00, 1.5000e+00, 1.5500e+00, 1.6000e+00,
1.6500e+00, 1.7000e+00, 1.7500e+00, 1.8000e+00, 1.8500e+00, 1.9000e+00, 1.9500e+00,
2.0000e+00, 2.0500e+00, 2.1000e+00, 2.1500e+00, 2.2000e+00, 2.2500e+00, 2.3000e+00,
2.3500e+00, 2.4000e+00, 2.4500e+00, 2.5000e+00, 2.5500e+00, 2.6000e+00, 2.6500e+00,
2.7127e+00, 2.7700e+00, 2.8300e+00, 2.9000e+00, 2.9600e+00, 3.0300e+00, 3.1100e+00,
3.1800e+00, 3.2600e+00, 3.3400e+00, 3.4300e+00, 3.5200e+00, 3.6100e+00, 3.7100e+00,
3.8100e+00, 3.9200e+00, 4.0300e+00, 4.1400e+00, 4.2600e+00, 4.3900e+00, 4.5200e+00,
4.6500e+00, 4.8000e+00, 4.9400e+00, 5.1000e+00, 5.2600e+00, 5.4255e+00, 5.6000e+00,
5.7000e+00, 5.9700e+00, 6.1700e+00, 6.3700e+00, 6.5900e+00, 6.8100e+00, 7.0400e+00,
7.2900e+00, 7.5400e+00, 7.8100e+00, 7.9000e+00, 8.0000e+00, 8.1028e+00, 8.2000e+00,
8.2800e+00, 8.3700e+00, 8.6700e+00, 8.9800e+00, 9.3000e+00, 9.6400e+00, 1.0000e+01,
1.0400e+01, 1.0815e+01, 1.1160e+01, 1.1570e+01, 1.2000e+01, 1.2460e+01, 1.2980e+01,
1.3528e+01, 1.3940e+01, 1.4480e+01, 1.5030e+01, 1.5620e+01, 1.6206e+01, 1.6403e+01,
1.6800e+01, 1.7000e+01, 1.7500e+01, 1.8200e+01, 1.8920e+01, 1.9400e+01, 1.9950e+01,
2.0700e+01, 2.1630e+01, 2.2100e+01, 2.2660e+01, 2.3500e+01, 2.4308e+01, 2.4506e+01,
2.4800e+01, 2.6200e+01, 2.7020e+01, 2.7500e+01, 2.8050e+01, 2.8900e+01, 2.9730e+01,
3.0200e+01, 3.0760e+01, 3.1500e+01, 3.2410e+01, 3.2609e+01, 3.2806e+01, 3.2900e+01,
3.4000e+01, 3.5300e+01, 3.6150e+01, 3.6600e+01, 3.7180e+01, 3.7900e+01, 3.8800e+01,
3.9890e+01, 4.0200e+01, 4.0513e+01, 4.0909e+01, 4.1000e+01, 4.1540e+01, 4.2000e+01,
4.3200e+01, 4.4200e+01, 4.5280e+01, 4.6000e+01, 4.7000e+01, 4.7990e+01, 4.8300e+01,
4.8615e+01, 4.9209e+01, 4.9600e+01, 5.0670e+01, 5.1200e+01, 5.2500e+01, 5.3380e+01,
5.3900e+01, 5.4410e+01, 5.5200e+01, 5.6000e+01, 5.6720e+01, 5.7120e+01, 5.8400e+01,
5.9800e+01, 6.1200e+01, 6.2510e+01, 6.3800e+01, 6.5230e+01, 6.6500e+01, 6.7900e+01,
6.8400e+01, 6.8930e+01, 6.9800e+01, 7.0610e+01, 7.1100e+01, 7.1640e+01, 7.2200e+01,
7.2920e+01, 7.3334e+01, 7.4000e+01, 7.4800e+01, 7.5600e+01, 7.6400e+01, 7.7200e+01,
7.8000e+01, 7.8900e+01, 7.9800e+01, 8.0700e+01, 8.1600e+01, 8.2500e+01, 8.3400e+01,
8.4300e+01, 8.5200e+01, 8.6100e+01, 8.7000e+01, 8.8000e+01, 8.9000e+01, 9.0000e+01,
9.1000e+01, 9.2000e+01, 9.3000e+01, 9.4000e+01, 9.5000e+01, 9.6000e+01, 9.7000e+01,
9.8000e+01, 9.9000e+01, 1.0000e+02, 1.0120e+02, 1.0240e+02, 1.0360e+02, 1.0480e+02,
1.0600e+02, 1.0720e+02, 1.0840e+02, 1.0960e+02, 1.1080e+02, 1.1200e+02, 1.1350e+02,
1.1500e+02, 1.1650e+02, 1.1800e+02, 1.1950e+02, 1.2100e+02, 1.2250e+02, 1.2400e+02,
1.2550e+02, 1.2700e+02, 1.2850e+02, 1.3000e+02, 1.3200e+02, 1.3400e+02, 1.3600e+02,
1.3800e+02, 1.4000e+02, 1.4200e+02, 1.4400e+02, 1.4600e+02, 1.4800e+02, 1.5000e+02,
1.5200e+02, 1.5400e+02, 1.5600e+02, 1.5810e+02, 1.6600e+02, 1.7431e+02, 1.8302e+02,
1.9217e+02, 2.0178e+02, 2.1187e+02, 2.2246e+02, 2.3359e+02, 2.4526e+02, 2.5753e+02,
2.7040e+02, 2.8392e+02, 2.9812e+02, 3.1303e+02, 3.2868e+02, 3.4511e+02, 3.6237e+02,
3.8049e+02, 3.9526e+02
```

Listing 2: α gridpoints

```
1.0023e-05, 1.3363e-05, 1.7818e-05, 2.3757e-05, 3.1676e-05, 4.2235e-05, 5.6314e-05,
7.5085e-05, 1.0011e-04, 1.3348e-04, 1.7798e-04, 2.3730e-04, 3.1641e-04, 4.2188e-04,
5.6250e-04, 7.5000e-04, 1.0000e-03, 1.5000e-03, 2.5000e-03, 3.5000e-03, 5.0000e-03,
7.0000e-03, 1.0000e-02, 1.5000e-02, 2.5000e-02, 3.5000e-02, 5.0000e-02, 7.0000e-02,
1.0000e-01, 1.2500e-01, 1.5000e-01, 2.0000e-01, 2.5000e-01, 3.0000e-01, 3.2500e-01,
3.5000e-01, 3.7500e-01, 4.0000e-01, 4.2500e-01, 4.5000e-01, 4.7500e-01, 5.0000e-01,
5.2500e-01, 5.5000e-01, 5.8000e-01, 6.1000e-01, 6.5000e-01, 6.9000e-01, 7.3000e-01,
7.8000e-01, 8.3000e-01, 8.8000e-01, 9.4000e-01, 1.0000e+00, 1.0800e+00, 1.1600e+00,
1.2400e+00, 1.3300e+00, 1.4300e+00, 1.5400e+00, 1.6600e+00, 1.7900e+00, 1.9400e+00,
2.0900e+00, 2.2600e+00, 2.4800e+00, 2.7127e+00, 2.8900e+00, 3.1100e+00, 3.3800e+00,
3.6700e+00, 3.9800e+00, 4.3200e+00, 4.6500e+00, 5.0000e+00, 5.4255e+00, 6.0000e+00,
6.5600e+00, 7.1300e+00, 7.6000e+00, 8.1026e+00, 8.8000e+00, 9.5000e+00, 1.0200e+01,
1.0815e+01, 1.1700e+01, 1.2600e+01, 1.3528e+01, 1.4400e+01, 1.5300e+01, 1.6205e+01,
1.7233e+01, 1.8200e+01, 1.8920e+01, 2.0300e+01, 2.1630e+01, 2.2900e+01, 2.4308e+01,
2.5600e+01, 2.7020e+01, 2.8400e+01, 2.9730e+01, 3.1000e+01, 3.2410e+01, 3.3440e+01,
3.4466e+01, 3.6150e+01, 3.7180e+01, 3.8800e+01, 4.0513e+01, 4.1540e+01, 4.2570e+01,
4.4200e+01, 4.6000e+01, 4.7000e+01, 4.8615e+01, 4.9600e+01, 5.1200e+01, 5.2500e+01,
5.4410e+01, 5.5200e+01, 5.6720e+01, 5.8400e+01, 5.9800e+01, 6.1200e+01, 6.2510e+01,
6.3800e+01, 6.5230e+01, 6.6500e+01, 6.7900e+01, 6.8930e+01, 7.0610e+01, 7.1640e+01,
7.2920e+01, 7.5900e+01, 8.0000e+01, 8.4000e+01, 8.9000e+01, 9.4000e+01, 1.0000e+02,
1.0500e+02, 1.1300e+02, 1.2063e+02, 1.2600e+02, 1.3200e+02, 1.4000e+02, 1.4700e+02,
1.5400e+02, 1.6200e+02, 1.7000e+02, 1.7700e+02, 1.8400e+02, 1.9100e+02, 1.9900e+02,
2.0800e+02, 2.1800e+02, 2.2700e+02, 2.3700e+02, 2.4600e+02, 2.5500e+02, 2.6500e+02,
2.7572e+02, 2.8400e+02, 2.9358e+02, 3.0200e+02, 3.1100e+02, 3.2000e+02, 3.2900e+02,
3.3800e+02, 3.4700e+02, 3.5600e+02, 3.6500e+02, 3.7400e+02, 3.8300e+02, 3.9200e+02,
4.0100e+02, 4.1000e+02, 4.1900e+02, 4.2800e+02, 4.3700e+02, 4.4600e+02, 4.5500e+02,
4.6400e+02, 4.7300e+02, 4.8200e+02, 4.9100e+02, 5.0000e+02, 5.0900e+02, 5.1800e+02,
5.2700e+02, 5.3600e+02, 5.4500e+02, 5.5400e+02, 5.6300e+02, 5.7200e+02, 5.8100e+02,
5.9000e+02, 5.9700e+02, 6.0400e+02, 6.1100e+02, 6.1800e+02, 6.2500e+02, 6.3290e+02,
6.6454e+02, 6.9777e+02, 7.3266e+02, 7.6929e+02, 8.0776e+02, 8.4815e+02, 8.9055e+02,
9.3508e+02, 9.8184e+02, 1.0309e+03, 1.0825e+03, 1.1366e+03, 1.1934e+03, 1.2531e+03,
1.3158e+03, 1.3815e+03, 1.4506e+03, 1.5231e+03, 1.5810e+03
```

Listing 3: T gridpoints (K)

```
283.6, 293.6, 300.0, 323.6, 350.0, 373.6, 400.0, 423.6, 450.0, 473.6, 500.0, 523.6, 550.0,
573.6, 600.0, 623.6, 650.0, 800.0
```