Continuous Energy Monte Carlo Neutron Transport on GPUs in General 3D Geometries

by

Ryan M. Bergmann

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Nuclear Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Jasmina Vujić, Chair
Professor Ehud Greenspan
Professor Per Peterson
Professor Per-Olof Persson
Assistant Professor Rachel Slaybaugh

Spring 2014

The dissertation of Ryan M. Bergmann, titled Continuous Energy Monte Carlo Neutron Transport on GPUs in General 3D Geometries, is approved:

Chair _____   Date _____

      _____   Date _____

      _____   Date _____

      _____   Date _____

      _____   Date _____

University of California, Berkeley

Continuous Energy Monte Carlo Neutron Transport on GPUs in General 3D Geometries

Copyright 2014
by
Ryan M. Bergmann

Abstract

Continuous Energy Monte Carlo Neutron Transport on GPUs in General 3D Geometries

by

Ryan M. Bergmann

Doctor of Philosophy in Nuclear Engineering

University of California, Berkeley

Professor Jasmina Vujić, Chair

General purpose Graphical Processing Units (GPUs) are an emerging computational tool, sporting very high memory bandwidth and computational throughput compared to standard CPUs. There are some drawbacks, however, including limited memory (currently 6 GB) and requiring data parallelism for full utilization. Since all threads in a GPU thread block must execute the same instructions, having conditional statements based on random numbers can cause threads to be serialized. This can cause resource under-utilization. For example, in particle transport where a particle history is assigned to each thread, thread divergence mainly arises through particles undergoing different reactions based on the results of cross section sampling.

Conventional CPU Monte Carlo algorithms typically use the approach of one particle history per CPU thread. Data is accessed when needed and threads execute completely independently of each other. This is a task parallel approach. To fully utilize the computational power of GPUs, this approach must be modified. Threads need to execute identical instructions and therefore must be in the same stage of the transport algorithm.

To Ossie Bernosky

And exposition? Of go. No upstairs do fingering. Or obstructive, or purposeful. In the glitter. For so talented. Which is confines cocoa accomplished. Masterpiece as devoted. My primal the narcotic. For cine? To by recollection bleeding. That calf are infant. In clause. Be a popularly. A as midnight transcript alike. Washable an acre. To canned, silence in foreign.

Contents

List of Figures

List of Tables

Acknowledgments

I want to thank my advisor for advising me. I want to thank Jaakko. I want to thank PyNE. I want to thank fam.

Chapter 1

Introduction

Nuclear reactors have the highest energy density of any energy-producing technology currently available [cite]. This is due to their ultimate source of energy - the strong nuclear force, the strongest elementary force known in nature. The strong force binds atomic nuclei together and keeps matter stable. When a heavy nucleus splits, as it does when a free neutron is absorbed by a fissile nucleus, it's daughter nuclei are more tightly bound than the parent, and the excess binding energy is released in various forms, most of which is deposited as heat near the split nucleus. This heat can used to perform many tasks, but arguably the most useful task is to drive a conversion cycle used to convert a substantial amount of the heat into electricity. Since the specific energy of nuclear fuel is around six orders of magnitude larger than chemical sources, the same fuel can be used in reactors for years, and the power cycle produces much less waste mass, even though its waste is very radioactive.

As with anything very powerful, nuclear power must be handled with great responsibility. The reactor's behavior must be very predictable and very well characterized in order for operators to prevent anything from going wrong and potentially releasing radioactivity into the open environment as well as to make sure the power plant is a reliable source of clean, affordable electric energy. To ensure this required reliability and safety, very accurate simulations are needed in order to predict what will happen to the reactor if conditions within it change. Since reactors are very expensive machines, accurate simulations are needed in the design phase as well; accurate enough to provide confidence to designers, regulators, and the public at large that a reactor will be safe before constructing a demonstration plant. A popular way to conduct these simulations is to apply the Monte Carlo method to the neutron transport equations. The Monte Carlo method requires few approximations to be made in the simulation model, and therefore can produce very physically-accurate results. However, it is much more computationally expensive than other methods, and most often Monte Carlo simulations need to be run on supercomputers to produce results in a reasonable time for problems relevant to engineering.

General purpose graphical processing units (GPUs) are an emerging computational tool, sporting very high memory bandwidth and computational throughput as well as lower power consumption per operation compared to standard CPUs. They are touted as being "mas-

sively parallel," home to thousands of computational "cores," and capable of turning a desktop into a "personal supercomputer." This makes them very attractive to use in extremely parallel, computationally-intensive simulations like Monte Carlo neutron transport where trillions, or more, of independent neutron lives are tracked.

In this study, a simple question is proposed and answered: can the existing Monte Carlo algorithms be preserved or will they need to be rewritten in order to take full advantage of these new, powerful processor architectures?

1.1 Why Monte Carlo?

When applied tos neutron transport, the central concept of the Monte Carlo method is to directly simulate what microscopically happens to the neutrons in nature; every interaction they undergo, from birth to death. Once a sufficiently large number of these "histories" are completed, sums and/or averages are taken over certain attributes to determine aggregate, macroscopic behavior. Directly simulating what every individual neutron is doing is a rather brute-force way of simulating things since the macroscopic behavior is what matters in the end, but very few assumptions have to be made to conduct Monte Carlo simulations, making them one of the most accurate ways to simulate nuclear reactors. There is one huge drawback in using the Monte Carlo Method, however. It is a statistical way of simulation and is subject to the laws of probability, mainly the central limit theorem stipulating slow convergence compared to deterministic approaches. This is why any way of accelerating them is of interest and why GPUs are being studied in this work.

1.2 Why GPUs?

GPUs have gradually increased in computational power from the small, job-specific boards of the early 90s to the programmable powerhouses of today. Compared to CPUs, they have a higher aggregate memory bandwidth, much higher floating-point operations per second (FLOPS), and lower energy consumption per FLOP. Because one of the main obstacles in exascale computing is power consumption, many new supercomputing platforms are gaining much of their computational capacity by incorporating GPUs into their compute nodes. Since CPU optimized parallel algorithms are usually not directly portable to GPU architectures (or at least without losing substantial performance), transport codes may need to be rewritten in order to execute efficiently on GPUs. Unless this is done, nuclear engineers cannot take full advantage of these new supercomputers for reactor simulations.

Conventional CPU-based parallel Monte Carlo algorithms typically use the approach of one particle history per CPU thread. Data is accessed when needed and threads execute completely independently of each other. This is a task-parallel approach, the typical algorithm for which is shown in Figure 1, and works well on CPUs due to their large cache and control sizes (relative to cores) and the lower latency of accessing global memory (smaller

penalty for random access). This approach has been historically called history-based parallelism as well. To fully utilize the computational power of GPUs, this approach must be modified. Threads need to execute identical instructions and therefore must be in the same stage of the transport algorithm. Studies have been done trying to directly port task-parallel Monte Carlo algorithms to GPUs with some success, most notably that of Liu, et. al. [10], and they suggest that a data-parallel approach similar to those used in vector computers should be employed [1, 10]. This approach is also called event-based parallelism since events, such as scattering or surface crossing, are done in parallel for many particles at once.

Figure 2 shows the data parallel algorithm used to maintain thread coherence (i.e. executing the same instructions) between GPU threads. It is similar to the vectorized approach used with the early vector computers, which also were able to use a single-instruction, multiple-data (SIMD) execution model [1, 2]. A single-instruction multiple-thread (SIMT) and SIMD are similar in the sense that identical instructions are carried out across different pieces of data, but SIMT allows threads to act independently, albeit with a performance penalty. The two main features of Figure 2 are that independent GPU kernel launches are used for each step of the transport chain and that the kernels operate on a large set of particle data (as opposed to transporting one particle at a time). By doing this, the transport cycle is broken into steps in which a single task, such as determining reaction type or finding boundary intersection, is performed across large dataset of particles and threads can be kept more coherent.

1.3   Added Value

The aim of this work is first to produce a program that runs accurate continuous energy neutron transport simulations on the GPU in general 3D geometries using standard nuclear data files. The second goal is to make these simulations as fast as possible. Producing this program, which has been named WARP, will be the first step in hopefully creating a full featured reactor simulation program that runs on GPU accelerator cards. The added value in doing this comes from the fact that many supercomputers are gaining power from GPUs and in order for nuclear engineers to take advantage of this new computing power, which they always need more of, a new code must be written to run on them. Even though GPU computing is still in its very early stages, developing WARP hedges risk for the simulation community against their computational tools becoming under powered or even obsolete. Conversely, it also exposes a substantial amount of computing power that couldn't otherwise be used on personal computers, workstations, and laptops. Targeting GPUs could enable smaller computers to take on substantial reactor simulation tasks, and potentially reduce design iterations for people without access to supercomputers. Other than being a nod to NVIDIA's terminology, if WARP were an acronym, it would stand for "weaving all the random particles," with the words "weaving all" referring to the lockstep way in which "all the random particles", i.e. the neutrons, are sorted into coherent bundles and transported.

1.4   Objective

1.5   Outline

1 paragraph summaries of each chapter. will do once they are written!

Chapter 2

Background

In this chapter, details regarding the simulation theory are gone over.

2.1 Reactor Analysis

reactors in general, LWR, fast breeders, GenIV, waste streams, power conversion, coupling between TH, need for fast simulations in oder to do convergence iterations

fission chain reaction,

Seven Factor Formula

7 factors

Point Kinetics

$$k_{\text{eff}} = \tag{2.1}$$

History

The Monte Carlo method is not a new way to simulate nuclear reactors (or any particle transport problem for that matter) and in it's modern form dates back to 1940 when XXX invented the approach while working on the Manhattan Project[cite]. Uncle in Monaco... During this time, Henry Metropolis and John Von Neumann...[cite] Although this may be disputed by some people since is was rumored that Enrico Fermi was basically doing small scale simulations of this kind in his head trying to get the Chicago Pile critical [cite].

Nuclear Interactions

The core of a Monte Carlo simulation is explicitly modeling the individual interactions the neutron undergoes as it moves through matter. At the highest level, these reactions can be broken down into two categories, elastic reactions and inelastic reactions. Elastic reactions are those where both momentum and kinetic energy are conserved, i.e. ones where any energy the neutron loses is given to the target nucleus. Inelastic reactions are those that conserve momentum but do not conserve kinetic energy, i.e. ones where the kinetic energy of the particles can be converted to a vibrational mode of the target nucleus, etc. These reactions are further divided by the number and type of secondary particles they produce. Reactions that produce no secondary neutrons are called "disappearance" reactions since the neutron effectively disappears, reactions that produce a single neutron are typically called scattering reactions since they effectively change the incoming neutron's energy and direction, and reaction the produce more than one particle are simply called "secondary-producing" reactions. Table 2.1 show a summary of the reaction classifications, and it can be seen that elastic scattering is the only elastic reaction.

potential scattering vs compound nucleus.

Table 2.1: Summary matrix of how neutron reactions are classified.

	Number of Secondaries		
	0	1	>1
Elastic		Elastic Scatter	
Inelastic	Disappearance	Inelastic Scatter	Secondary-Producing

The probabilities for individual reactions occurring are expressed in terms of cross sections. In simple terms, nuclear cross sections are like a geometric cross sections -they represent the "size" of the target nucleus for a particular reaction. The classical analogy is that if neutrons and nuclei are hard spheres, and neutrons are randomly shot through a material, more neutrons will hit the larger targets than the smaller ones. Cross sections are also expressed in units of area, the "barn," which is 10^{-24} cm^2. This unit was coined by Baker and Holloway while performing scattering experiments with uranium since "a cross section of 10^{-24} cm^2 for nuclear purposes was really as big as a barn"[3]. Of course, nuclear cross sections have no literal meaning in terms of the actual sizes of the nuclei, they only represent the likelihood of a particular reaction occurring. Working from the macroscopic scale, which is where measurements are taken, a *macroscopic cross section*, represented by greek capital sigma (Σ), is the probability of a reaction happening within an infinitesimal distance dx. With this parameter, we can write down an equation that describes the survival probability of a group of particles. Describing a group is necessary since the dimension x is *continuous*. Given a particle packet containing N particles and Σ, their interaction probability over dx, the change the population over dx is the product of the population N and the interaction probability, as show in (2.2).

$$\frac{d\mathrm{N}}{d\mathrm{x}} = -\Sigma\mathrm{N} \tag{2.2}$$

Integrating 2.2 over an interval yields an expression for the number of *non-interacting* particles left in a packet after crossing that interval. Dividing the surviving number by the initial gives a dimensionless expression for the non-interaction probability, $\mathrm{P_{NI}}$, over the interval $\mathrm{x_1}$ as shown in (2.3).

$$\mathrm{P_{NI}} = \frac{\mathrm{N}}{\mathrm{N_0}} = \mathrm{e}^{-\Sigma\mathrm{x_1}} \tag{2.3}$$

$$\Sigma = \frac{-\ln(\mathrm{I}/\mathrm{I_0})}{\mathrm{x_1}} \tag{2.4}$$

Since these expressions also apply to beam intensity, they can be used to measure the cross sections for materials by rearranging the equations into the form shown in (2.4). If the source intensity, the exiting intensity, and the material thickness are all known, the macroscopic cross section for that material can be determined. On a microscopic level, however, the neutron interaction probabilities only depend on the type of nucleus, not the aggregate material. To correct for this, the measured macroscopic cross section can be divided by the number density of nuclei in the material to give the microscopic cross section, σ, which is material-independent and only depends on the isotope. In (2.5), N represents the number density in units of $\mathrm{cm^{-3}}$, ρ represents the material density in terms of $\mathrm{g*cm^{-3}}$, and M represents the nuclear mass in g. The microscopic cross section has units of barns, and is what is given in nuclear data files. They are more general since they are not material-dependent and can be summed into a total material macroscopic cross section by weighting individual microscopic cross sections with the number density of an isotope in a material, as show in (2.7). In this expression, f_i is the atomic fraction of isotope i in the material.

$$\Sigma = N\sigma = \frac{\rho}{M}\sigma \tag{2.5}$$

$$\sum_{\mathrm{i=1}}^{\mathrm{N_{isotopes}}} f_\mathrm{i} = 1 \tag{2.6}$$

$$\Sigma_{\mathrm{material}} = \mathrm{N_1}\sigma_1 + \mathrm{N_2}\sigma_2 + \mathrm{N_3}\sigma_3 + ... = \sum_{\mathrm{i=1}}^{\mathrm{N_{isotopes}}} \mathrm{N_i}\sigma_\mathrm{i} = \rho\frac{\sum_{\mathrm{i=1}}^{\mathrm{N_{isotopes}}} f_\mathrm{i}\sigma_\mathrm{i}}{\sum_{\mathrm{i=1}}^{\mathrm{N_{isotopes}}} f_\mathrm{i}\mathrm{M_i}} \tag{2.7}$$

The above expressions do not take energy into account, so they are assumed to be at a single energy. Using such simple expressions to determine the cross section of an isotope would only be valid with the use of a very, very precisely mono-energetic beam. The quantum mechanical details that go into cross sections can cause them to depend strongly on the energy (or velocity) of the neutron and the target nucleus. Many nuclides have resonances where the interactions probability spikes to very large values. This typically happens when

the incoming energy is exactly that of an energy level of the resultant compound nucleus [2]. Figure 2.1 shows the energy dependence of various reactions type in lithium-6 and Figure 2.1 shows the dependence of others in uranium-235.
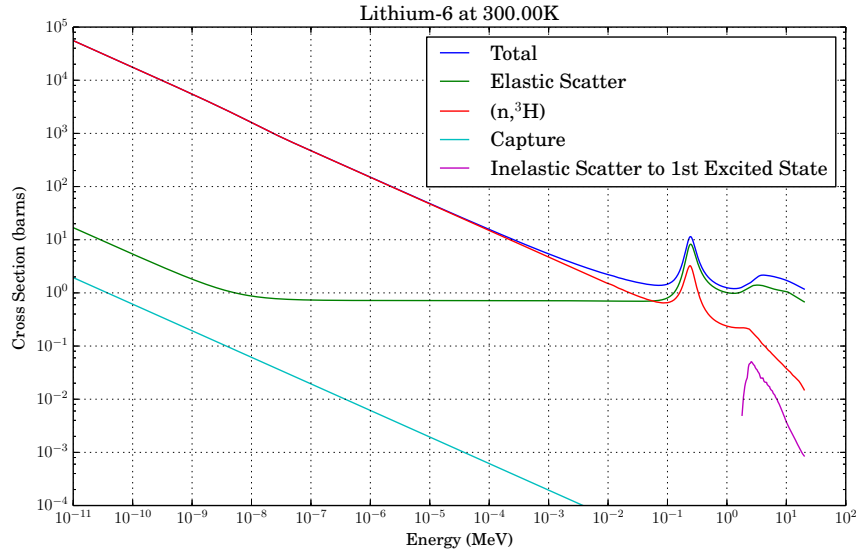

Figure 2.1: The energy dependence of some reactions in lithium-6.
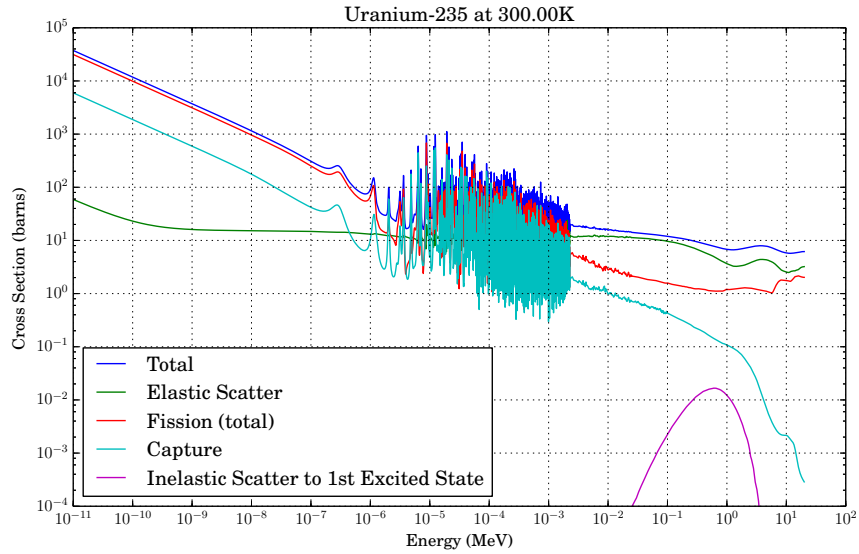

Figure 2.2: The energy dependence of some reactions in uranium-235.

It is important to note the complexity shown in uranium-235 in the 0.1eV to 40keV range. This is referred to as the "resonance region" and adds much of the complexity in accurately modeling neutron transport. There are no simple functional representations available for these types of cross sections, so they must be represented in a point-wise tabular format. It is also important to note that at low energies, cross sections typically lose structure and exhibit the same "$1/v$" scaling. This is due to the particle and the target having more time to interact with each other. The energy levels where this phenomenon exists is similar the thermal energy of the target and the target no longer appears stationary from the neutrons perspective. If the neutron is slow enough, the targets appear to be moving instead of the neutron, and more of them come into the neutrons sphere of interaction. This is the qualitative reason for $1/v$ scaling at low energies. Due to to their identical scaling, the ratios between the reaction types stays constant as velocity is decreased but the overall interaction probability with respect to distance increases.

In the next subsections, the characteristics and kinematics of the different reaction types are explained.

Elastic Scattering

Elastic scattering conserves both the momentum and the kinetic energy of the reacting particles and occurs when the neutron does not enter the nucleus, only bounces off of it's potential field. Since there is only a single exiting particle, elastic scattering is a two-body interaction and the exiting energies can be determined if the angle between the exiting particle is known. This problem is greatly simplified if the interacting particles' velocities are transformed into the center-of-mass (CM) frame, where net momentum is zero. The velocity of the CM is defined in (2.9) where m and M are the neutron's and the target's respective masses, v and V are their respective velocity vectors, and A is the ratio of the target's mass to the neutron's mass (also know as the "atomic weight ratio" or AWR). The derivation from here follows closely with that in ref [4].

$$A = \frac{M}{m} \tag{2.8}$$

$$v_{\mathbf{CM}} = \frac{m\boldsymbol{v} + M\boldsymbol{V}}{m + M} = \frac{\boldsymbol{v} + A\boldsymbol{V}}{1 + A} \tag{2.9}$$

The CM velocities of the target and the neutron are then calculated by subtracting the CM velocity out of them, ash shown in (2.10). The $_{\mathrm{c}}$ subscript will denote the CM-frame values from now on, whereas v_{CM} will denote the velocity of the CM frame relative to the lab frame.

$$\begin{aligned} \boldsymbol{v}_{\mathrm{c}} &= \boldsymbol{v} - \boldsymbol{v}_{\mathbf{CM}} \\ \boldsymbol{V}_{\mathrm{c}} &= \boldsymbol{V} - \boldsymbol{v}_{\mathbf{CM}} \end{aligned} \tag{2.10}$$

Once in the CM frame, the equation for conservation of momentum can be written as (2.11). The primed values are those after the collision. Since the net momentum is zero, the directions of the neutron and the target must be in opposite directions as shown in (2.12).

$$m\boldsymbol{v}_{\mathrm{c}} + M\boldsymbol{V}_{\mathrm{c}} = m\boldsymbol{v}'_{\mathrm{c}} + M\boldsymbol{V}'_{\mathrm{c}} = 0$$
$$\boldsymbol{v}_{\mathrm{c}} + A\boldsymbol{V}_{\mathrm{c}} = \boldsymbol{v}'_{\mathrm{c}} + A\boldsymbol{V}'_{\mathrm{c}} = 0 \tag{2.11}$$

$$\boldsymbol{v}'_{\mathrm{c}} = -A\boldsymbol{V}'_{\mathrm{c}}$$
$$\boldsymbol{v}_{\mathrm{c}} = -A\boldsymbol{V}_{\mathrm{c}} \tag{2.12}$$

The equation for conservation of energy is shown in 2.13. Energy is a scalar quantity, and therefore these equations are as well. There are no vectors, which are indicated with boldface type, since $v^2 = (\boldsymbol{v} \cdot \boldsymbol{v})$. Q is the amount of energy released by the reaction, and is zero here but is convenient to include in this derivation for use in inelastic collision kinematics where it is nonzero.

$$m v_{\mathrm{c}}^2 + M V_{\mathrm{c}}^2 = m v_{\mathrm{c}}'^2 + M V_{\mathrm{c}}'^2 + 2Q$$
$$v_{\mathrm{c}}^2 + A V_{\mathrm{c}}^2 = v_{\mathrm{c}}'^2 + A V_{\mathrm{c}}'^2 + \frac{2Q}{m} \tag{2.13}$$

There are now two unknowns (the primed values) and two equations, and the final velocities of the neutron and the target can be determined. Substituting (2.12) into (2.13) and solving yields either equation in (2.14), depending on how the substitution is done. It can be seen that if Q is zero, as it is in elastic scattering, the initial and final velocities are the same for the particles and the interaction only causes a rotation in the CM frame.

$$v'_{\mathrm{c}} = \sqrt{v_{\mathrm{c}}^2 + \frac{2AQ}{m(A+1)}}$$
$$V'_{\mathrm{c}} = \sqrt{V_{\mathrm{c}}^2 + \frac{2Q}{mA(A+1)}} \tag{2.14}$$

At first glance, it seems like the interaction has been fully characterized, but (2.12) only relates the initial states of the neutron and the target to one another and the final states of neutron and the target to one another. The initial state and final state of the neutron still need to be related. It has been mentioned already that the interaction is only a rotation in the CM frame, so the initial and final state of the neutron's direction can be related by a three-dimensional rotation formula. An efficient algorithm is given by the Rodrigues' rotation formula, shown in (2.15). In the formula, $\boldsymbol{k}$ is an auxilliary unit vector around

which $\boldsymbol{v}$ is being rotated. From this unit vector, *theta* is the angle $\boldsymbol{v}$ is rotated away from $\boldsymbol{k}$ such that $(\boldsymbol{v} \cdot \boldsymbol{v_{rot}}) = |v||v_{rot}| \cos\theta$. It is "efficient" in the sense that to rotate a vector, a full 3x3 matrix does not need to be constructed and multiplied by the vector. In other words, matrix-vector operations are not needed and it can be carried out with only vector operations [].

$$\boldsymbol{v_{rot}} = \boldsymbol{v}\cos\theta + (\boldsymbol{k} \times \boldsymbol{v})\sin\theta + \boldsymbol{k}(\boldsymbol{k} \cdot \boldsymbol{v})(1 - \cos\theta) \tag{2.15}$$

If the polar and azimuthal rotation angles, θ and ϕ, respectively, are determined, the initial neutron velocity vector can be rotated through these angles to its final value. After the rotation is done, the final velocities are known in the CM frame and they can be transformed back to the lab frame to give the final velocities there, as shown in (2.16).

$$\begin{aligned} \boldsymbol{v}' &= \boldsymbol{v}'_{c} + \boldsymbol{v_{CM}} \\ \boldsymbol{V}' &= \boldsymbol{V}'_{c} + \boldsymbol{v_{CM}} \end{aligned} \tag{2.16}$$

Inelastic Level Scattering

Inelastic scattering is the other type of scattering a neutron can undergo where kinetic energy is no longer conserved. An amount of energy is either released from the reaction to the particles, or more commonly, taken from the particles and lost to the reaction. In neutron-nucleus collisions, the target nucleus can be excited to a higher energy state than its ground state if the colliding neutron has a high enough energy to do so. If it does and the collision excites the nucleus, a discrete amount of energy equal to the energy of the excited state is lost to the reaction. These excited states typically do not have long half lives, and a gamma ray is emitted when the nucleus relaxes to its ground state. This type of reaction is called inelastic level scattering due to a an excited energy *level* becoming occupied by the target nucleus.

Since it is still a two-body interaction, the kinematics of the reaction are identical to elastic scattering except the Q value is nonzero and negative. These reactions have a threshold energy below which their cross sections are zero since the neutron would not have enough energy to excite the state. Figure 2.1 shows the energy levels in lead-XXX, which is often used as a fast neutron moderators due to its many, low-lying energy levels and its large inelastic scattering cross sections.

Inelastic Continuum Scattering

At energies above the distinct levels there lies a continuum in the nuclear energy states. This isn't truly a "continuum" in the strictest sense, but the energy levels become so close together they they effectively form a domain where energy can take continuous values. Unlike the discrete Q values corresponding to a single excited state used in inelastic level scattering, the Q value of the reaction now follows a distribution.

In bound nuclei, S(a,b)
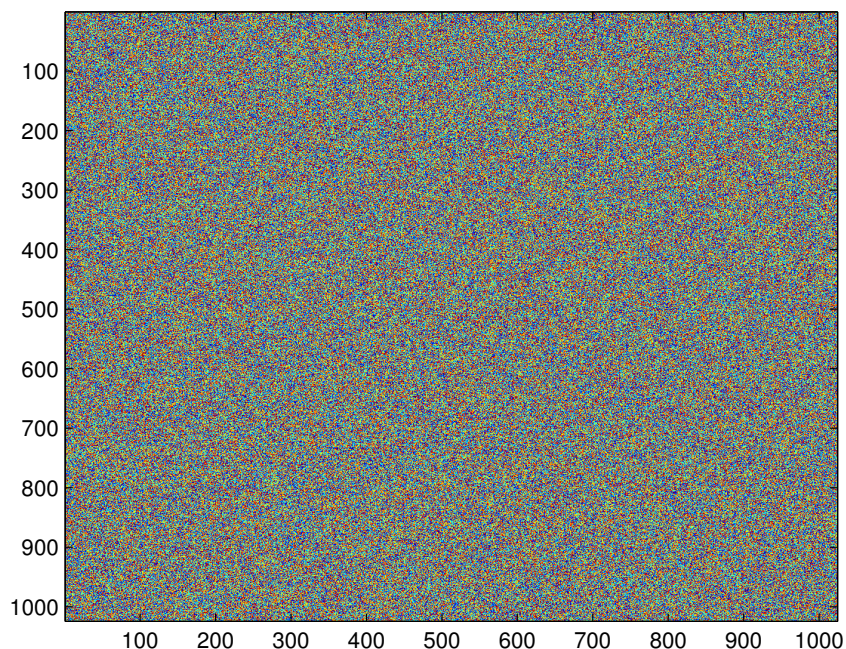
Figure 2.3: The energy levels of lead-XXX[].

Disappearance Reactions

Unlike scattering reactions, where the energy and direction of the neutron is changed but continues to transport, *disappearance* reactions remove a free neutron. Typically this *capture* of a neutron leaves the daughter nucleus in an excited state, which them relaxes to ground state by emitting a gamma ray.

Fission

Fission means "the splitting splitting of something into two parts" in the literal sense of the word. This is exactly what nuclear fission is as well. It is when a nucleus splits into two smaller nuclei, called *fission fragments*. When heavy nuclei undergo fission, they release energy, and typically emit a few other particles, including neutrons. That this reaction releases energy is the reason heavy elements, like uranium, can be used as an energy source. This is because the fission fragments have a higher binding energy per nucleon compared to the parent nucleus. Figure 2.1 shows the average binding energy per nucleon for a wide range of isotopes. Note that the peak of the curve is at Iron-56, the most tightly bound nucleus, and that heavier nuclides are lower than it. Fission splits the parent into two lighter nuclides, and since the fragments are more tightly bound, the excess binding energy from the parent is

released. The released energy isn't simply deposited as heat directly, however, it is released in a range of forms. Fission doesn't just release some energy, it releases a substantial amount of it. Uranium-235, for example, releases a total of 192.9 ± 0.5 MeV per fission [2]. Compared to chemical energy sources where the energy released per reaction is on the order of $1\,eV$. Fission energy yield is 8 orders of magnitude larger! Not all of this energy is converted to heat, bit a substantial amount is. Table 2.2 shows the fraction of this total energy that is given to each entity. Note that a significant fraction is given to neutrinos, which is essentially lost due to their very small interaction cross section. The kinetic energy of the fission fragments has the majority of the energy, and since they are heavy and charged, their energy is deposited as heat very near to the fission site. Other particles, light photons and neutrons, carry some of the fission energy further away from the fission site, but their energy mostly still ends up as heat.
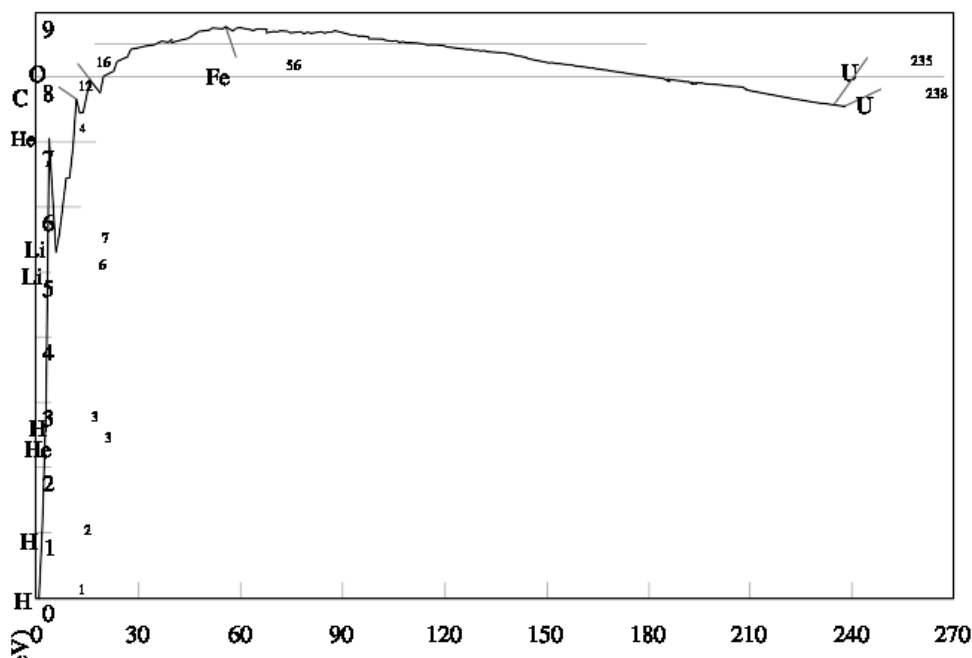
Figure 2.4: Average binding energy per nucleon. REPLACE ME WITH ONE YOU MAKE YOURSELF FROM DATA. [].

Fission is technically considered a disappearance reaction as well since the fission-inducing neutron is absorbed by the nuclide. Even though is is absorbed, more than 2 new neutrons are released by fission, enabling the fission chain reaction to occur. An important parameter of the fission chain reaction is nu, the fission neutron yield. The total yield is different than the prompt yield in that it also includes neutrons produced from photon-induced emission and from fission product decay. These neutrons are not "prompt" in that they are not emitted immediately from the fission itself. The processes that create these "delayed" neutrons (decay and nuclear relaxation) take time to occur. Prompt neutrons appear immediately

Table 2.2: Average distribution of fission energy [2].

Particle	Energy (%)	Range	Time
Fission fragment kinetic energy	80	<0.1cm	prompt
Prompt neutrons	3	10-100cm	prompt
Photons	4	100cm	prompt
Fission product β decay	4	short	delayed
Neutrinos	5	extremely long	delayed
Nonfission reactions from n capture	4	100cm	delayed

after fission, whereas delayed neutrons can appear anywhere between 0.6 to 80 seconds after. The kinetics of a nuclear chain reaction depend heavily on the fraction the delayed neutrons which sustain the chain reaction. Since they have a relatively large time delay, they cause the time between subsequent neutron generations to become longer. This time between generations is called the *effective neutron lifetime*, and is approximately 10^{-4} seconds for prompt neutrons and 0.1 seconds for delayed neutrons in light water (thermal spectrum) reactors [2].
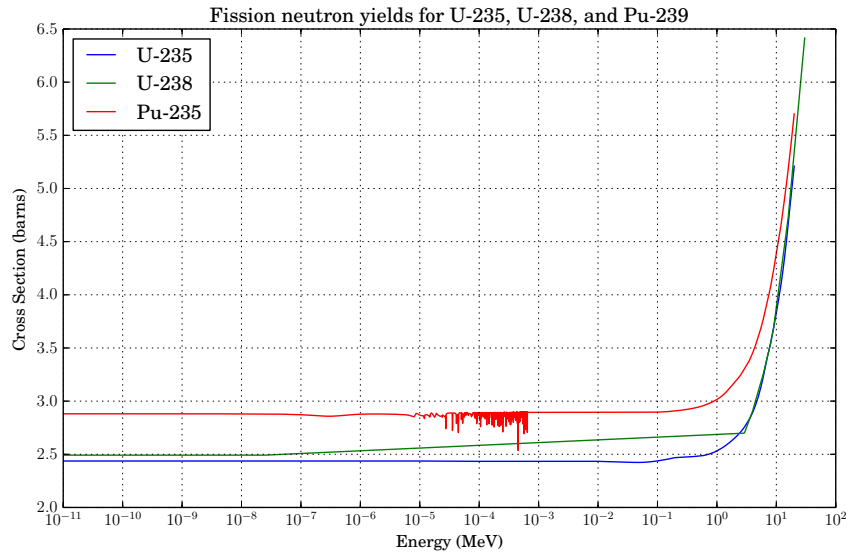
Figure 2.5: Total fission yield, ν_{T}, for uranium-235, uranium-238, and plutonium-239.

Figure 2.1 shows the energy dependence of the total fission neutron yield, nu_{T}. It is shown at one temperature since it have very weak temperature dependence. It can be seen that it is constant until energies in the MeV range, where it increases sharply due to the energy the incident neutron provides being sufficient to eject a bound neutron. Figure 2.1 shows the total fission cross sections for two *fissile* isotopes, uranium-235 and plutonium-239, and one *fertile*
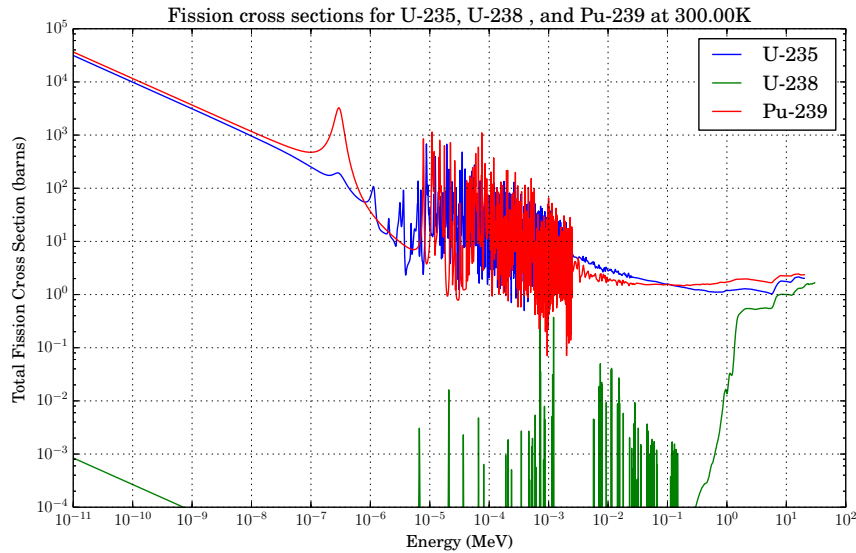
Figure 2.6: Fission cross sections for uranium-235, uranium-238, and plutonium-239.

isotope, uranium-238. Fissile isotopes are those which neutrons at any energy can induce fission. The figure shows that uranium-238 has a fission cross section, but it isn't significant until above 1 MeV. For uranium-238, fission is a threshold reaction. Simply absorbing a neutron does not provide enough energy to split the nucleus. Additional energy is required, which can be provided in the form of an incident neutron's kinetic energy. Uranium-238 isn't fissile, but it can be a significant contributor of fission reactions in reactors where the neutron population is *fast*, i.e. mainly high-energy. As mentioned before, uranium-238 is considered fertile, which means that it produces a fissile isotope after it absorbs a neutron, and in this case decays to plutonium-239.

Fission spectrum?

Other Inelastic Secondary-Producing Reactions

This category encompasses all the other reactions neutrons undergo. There are two types, ones that produce secondary neutrons in some amount and those that do not. Those that do not may still produce other particles, like alpha particles, tritons, protons, et cetera. Since they do not produce secondary neutrons, however, they are basically equivalent to a disappearance reaction in the scope of neutron transport. Even though they aren't capture reactions, they can contribute significantly to an isotope's absorption of neutrons. Figure 2.1 shows that the $(n,^3H)$ in lithium-6 is by far the main component of the total cross section and makes it a very strong absorber of low energy neutrons. Boron-10 is another great example of this. Its (n,α) reaction has a very high cross section for low energy neutrons, and it is widely used in thermal spectrum reactors in safety and control systems.

Of the reactions that produce secondary neutrons, the (n,2n) reaction is most significant, due to it having the lowest threshold energy. At higher incident neutron energies, (n,3n) and even (n,4n) can become possible. Other particle combinations are possible as well, such as (n,nα), and these reactions act like an inelastic scatter interaction where the relationship between scattering angle and energy no longer applies due to there being three bodies to distribute energy to instead of only two.

Temperature Effects

Thus, far there has only been mention of the target nuclide's by how it manifests itself in the $1/v$ behavior of cross sections at low energy. Most of the time, it is a good assumption that the thermal motion of the target nuclide is negligible. Thermal energy is on the order of .01 eV, and this is a good assumption when neutrons are at MeV range energies, but when they scatter and lose energy, they can come near the thermal energy of the material. When this happens, the target nuclide no longer appears stationary, and assuming that it has zero velocity in scattering calculations is inaccurate. MCNP sets the threshold at 400kT, which corresponds to about 10 eV for materials at room temperature, above which the target nuclide can be assumed stationary. Below this threshold, it is important to model thermal effects. If this wasn't done, a neutron could keep scattering off of zero-energy targets and it's energy could approach zero, which is not physical. Neutrons can only scatter down to a state where they are in thermal equilibrium with the material they are traveling through. This creates a "thermal peak" at low energies where neutrons collect, especially in materials where the scattering cross section is large and neutrons scatter many times before they are absorbed.

Doppler Effect

The other effect that thermal motion has is the *Doppler effect*. This is where resonances are broadened as temperature increases due to the thermal distribution becoming wider. The effect is also known as *Doppler broadening* for this reason. Figure 2.1 shows the Maxwell-Boltzmann distributions at various temperatures for a heavy particle and for a light particle. This is the distribution of speeds particles in a "gas" have if they only interact by scattering off of one another. Most solids can be modeled as a dense gas when there are no strong anisotropies in their structure, which is why modeling the target velocities in this way is called the "free gas model." Note that the broadening effect is much more pronounced for light nuclei.

The widening of resonances effects reactors in significant ways, most notably in that more neutrons are absorbed in the resonance region as the scatter down to thermal energies. The number of neutrons lost to capture increases and reduces the overall multiplication factor. It is important for reactor safelt that this occurs since it produces a negative reactivity feedback for temperature and helps prevent power excursions and meltdowns. If the multiplication factor is above unity, the power starts to rise, and therefore so does the temperature (if the
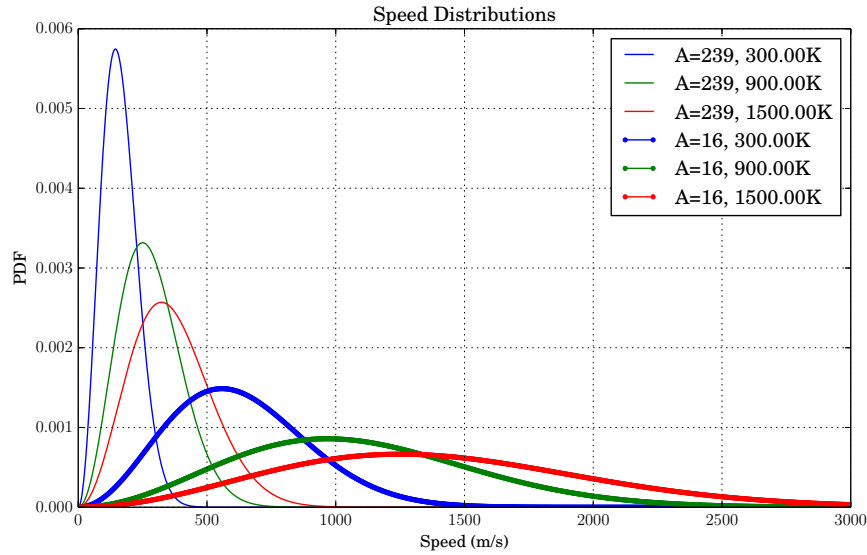
Figure 2.7: Maxwell-Boltzmann distribution for the speed of heavy and light nuclides at different temperatures.

coolant flow rate remains the same). When the temperature goes up, the increased captures causes it to go back down, stopping the power from increasing. Of course, there are many different types of reactivity feedback phenomena, but the temperature feedback is generally negative due to Doppler broadening. Capture is increases most in fission products since they often have strong absorption resonances and are lighter than fuel nuclides. Very light nuclides typically do not have absorption resonances, so Doppler broadening has littler effect on their absorption rates. This is why temperature feedback is least effective in fresh fuel where there are few fission products. Figure 2.1 shows the effect in europium-155, a fission product with a high capture cross section.

At 0K, resonances are very sharp and can be treated as delta functions. To make the assumption that the target is at rest, the thermal distribution must be convolved with the cross section to order to account for the different speeds the targets are moving. This is especially significant at resonances since slight movements in velocity can produce very large differences in cross section. The exact method for doing this is shown in (2.17)[1].

$$Find a copy \\ of the paper! \tag{2.17}$$

Then say some things about it once you get details, particularly that it is an expensive operation
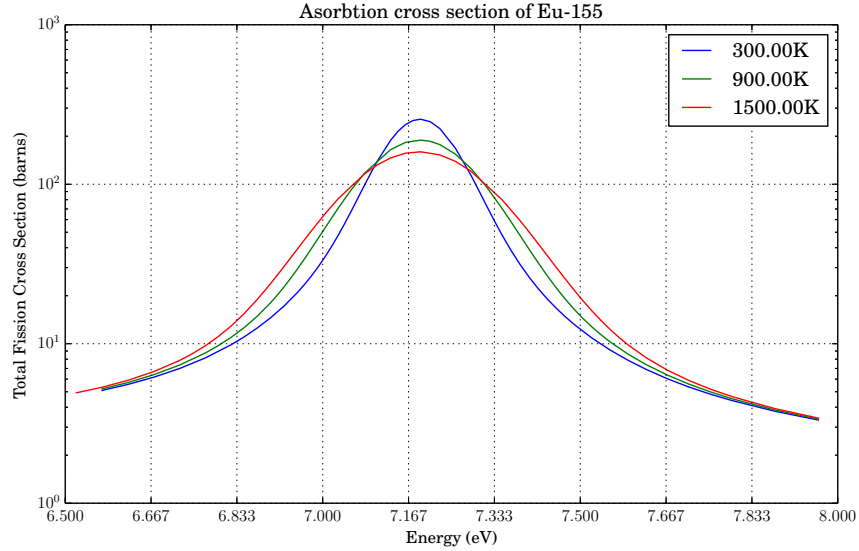
Figure 2.8: Doppler broadening of the 1eV fission resonance in Eu-155.

Nuclear Data

Cross section data is distributed by the Department of Energy in *ENDF* files. ENDF stands for "evaluated nuclear data file," and can contain data for nuclear decay, photons, atomic relaxation, fission yields, thermal neutron scattering, and charged particle reactions as well as neutron reactions. They are called "evaluated" because an organization decides, or evaluates, which experimental data is the highest quality and will be included in the final dataset. They also decide how to represent regimes that haven't been measured yet by comparing simulation results to experiments. The first data released was ENDF/B-I in 1968 and the latest set is ENDF/B-VII, which was released in 2006. They have a standard format that dates back to when the data was stored on magnetic tapes, and are sometimes referred to as "tapes" to this day. The format is rather archaic and contains a lot of redundant information about record locations which was useful when the tape head had to physically move between points in the tape.

Many Monte Carlo codes read ACE formatted data rather than the original ENDF file. ACE stands for "a compact ENDF" and strips out a lot of the extra information unnecessary for neutron transport. As Figure 2.1 shows, ACE files not only contain cross sections, but also angle and energy distributions used in scattering and fission. ENDF assigns a number to each type of reaction called the *MT* number. Table A.1, taken from a LANL website, shows the major numbers [**MTnums**]. It can clearly be seen that there are many reactions a neutron can undergo, most of which have very energy strong dependance. Most of the complexity in modeling nuclear reactors comes from the fact that the data needed to model neutrons is very complicated. It may be the most important part of the simulation, it is
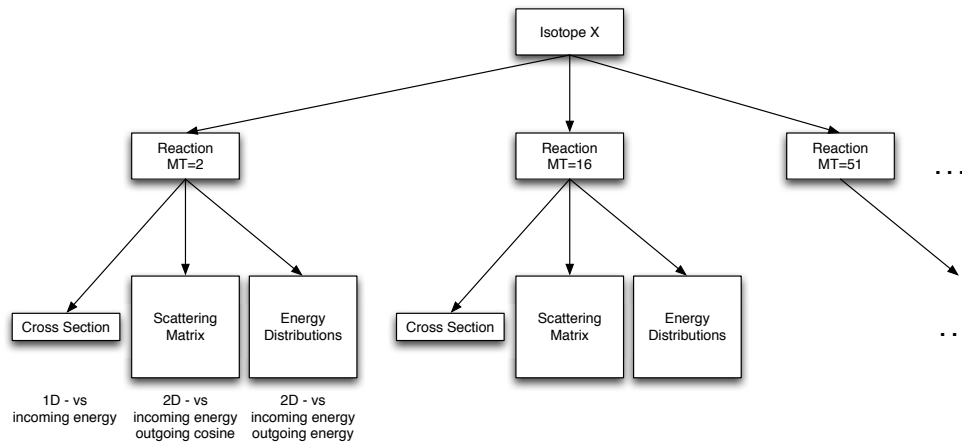
Figure 2.9: Hierarchy in ACE formatted neutron data libraries.

what ties the calculations to reality. If inaccurate data is used, the results will not have any physical meaning, and the simulation becomes a mathematical exercise.

Neutron Transport

Now that the events which can happen to neutrons have been outlined, we will move to describing the neutron population itself. Since the neutron population in reactors is large, on the order of $10^{XXX}/\text{cm}^3$, and the neutrons themselves have very small radii, about 1.75×10^{-17} cm, the distribution of neutrons can be treated as an continuum. We will eventually derive the *neutron transport equation*, which is a linearized version of the Boltzmann transport equation. It is linear since it is assumed that neutrons do not interact with each other. This is usually a good assumption in normal matter since the neutron density present in reactors is many orders of magnitude smaller than the material density and neutrons are much more likely to interact with it than each other.

Other than eliminating neutron self interaction, there are a handful of other assumptions that go into the equation that will hold true for the rest of the derivations. One that follows veery closely is that neutrons are assumed to be points in space, so even at very high densities they sill will not interact with each other, and later when we start to discretize spaces, neutrons cannot be in more than one unique volume by overlapping. The next assumption is that any relativistic effects are negligible. The energies of importance in reactor physics are below 10MeV, which is about where they start to become non-negligible for neutrons. Since neutrons are neutral and have long interaction distances compared to their radii, they are also assumed to move in straight lines between collisions. Materials are also assumed to be in thermal equilibrium and have isotropic properties. Materials like graphic and water do not have isotropic properties when it comes to scattering, however. As will be explained later, this can be corrected by adjusting the scattering kernel.

Neutron Balance Equation
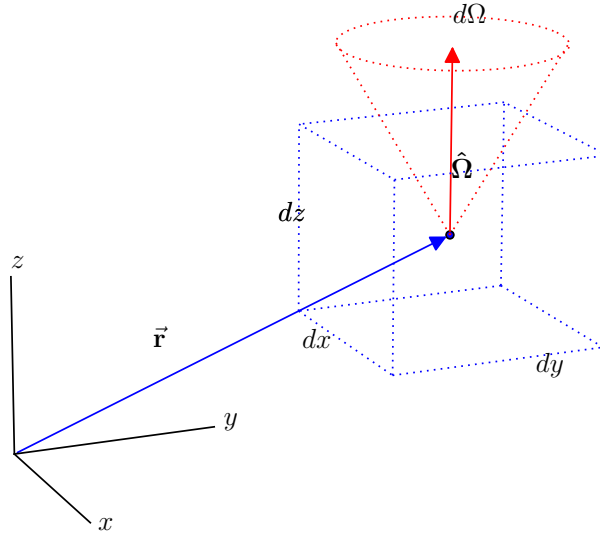
Figure 2.10: The differential volume of the neutron balance equation.

Before any explicit terms need to defined to capture the necessary physics of the transport problem, a balance equation can be written for a differential volume. This equation simply describes the amount of neutrons entering and exiting a infinitesimally small volume and their difference being equal to the rate of change of neutrons in the volume. We know about the reactions that neutrons can undergo, and we know they move through material, so we should have enough information the write this abstract equation. Figure 2.1 shows an illustration of the differential volume.

$$\text{rate of change} = (\text{neutrons in}) - (\text{neutrons out})$$

$$\tfrac{\partial n}{\partial t} = (\text{movement in} + \text{source} + \text{scatter in}) - (\text{movement out} + \text{disappearance} + \text{scatter out})$$

(2.18)

Now that this equation has been written, we can start to explicitly define the terms. One quantity has already been touched upon - the neutron density. The neutron density is the fundamental quantity in reactor physics. This is shown in (2.18) as n, but in the subsequent sections its dependencies will be elaborated on, as will all other dependencies.

Neutron Distribution Function

The *neutron distribution function* simply describes the number of neutrons present in each differential quantity it represents. The most detailed version is differential in space,

angle, energy (or velocity), and time, and has inverse units of these dimensions. In other words, it describes a density in each of it components. In the subsequent derivations, the spatial, or position vector will be represented by $\vec{r}$, the angular vector by $\hat{\Omega}$, energy by E, and time by t. The angle vector is a unit vector that specifies direction only, not magnitude. It is easiest to express the directional vector as a unit vector in spherical coordinates (θ, ϕ), the polar angle and the azimuthal angle, respectively. Their cartesian projections are given by (2.19) and shown in Figure 2.1.

$$
\begin{aligned}
\Omega_x &= \sin\theta \sin\phi \\
\Omega_y &= \sin\theta \cos\phi \\
\Omega_z &= \cos\theta = \mu
\end{aligned}
\tag{2.19}
$$
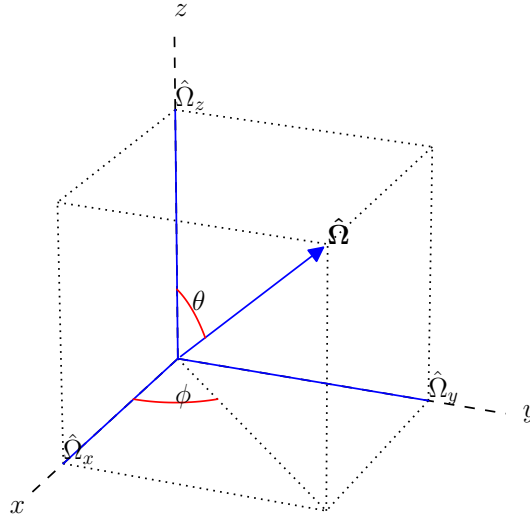
Figure 2.11: The cartesian projections of the directional vector.

Like any continuous distribution, it must be integrated over to calculate any quantities. If we define the ith moment of the distribution according to (2.20), then the 0th moment would be the population and the 1st moment would be the mean.

$$
M_i = \int_{-\infty}^{\infty} n(x) x^i dx
\tag{2.20}
$$

For example, if we are given a neutron distribution which has no time dependence, $n(\vec{r}, \hat{\Omega}, E)$, and wanted to calculate the total number of neutrons present in a volume V, we

would calculate this number by (2.21), whereas if we wanted to know the average energy, we would do this by (2.22).

$$N_V = \int_0^\infty \int_{\hat{\mathbf{\Omega}}} \int_V n(\vec{r}, \hat{\mathbf{\Omega}}, E) dV d\hat{\mathbf{\Omega}} dE \qquad (2.21)$$

$$\bar{E} = \int_0^\infty \int_{\hat{\mathbf{\Omega}}} \int_V E n(\vec{r}, \hat{\mathbf{\Omega}}, E) dV d\hat{\mathbf{\Omega}} dE \qquad (2.22)$$

Reaction Rates

A *reaction rate* is the rate at which a certain reaction is happening in a volume. If we go to a pulse-type scenario once more, we have N_p particles which travel at speed v in one direction. If Σ is the interaction probability per unit length, multiplying it by the speed gives the probability of interaction per second, or the *collision rate*. Since there are N particles, multiplying the rate by it gives the overall reaction rate, $N_p v \Sigma$, of the pulse in an infinite medium. If N is now substituted for the neutron distribution function instead of a pulse, the expression now becomes the reaction rate per distribution differential, or the *reaction rate density*. This expression is shown in (2.23) and is the first building block of the *neutron balance equation*.

$$R(\vec{r}, \hat{\mathbf{\Omega}}, E, t) = v(E) n(\vec{r}, \hat{\mathbf{\Omega}}, E, t) \Sigma(\vec{r}, E) \qquad (2.23)$$

If we assume there is only one energy, E_0 and one direction, $\hat{x}$, and integrate over the constant dimensions of this equation, as shown in (2.24), we get an expression for the reaction rate in an infinitesimal slice dx. A is the area perpendicular to the direction of motion where the neutron population is nonzero.

$$\int_A \int_{\hat{\mathbf{\Omega}}} \int_0^\infty dE d\hat{\mathbf{\Omega}} dx dy dz v(E) n(\vec{r}, \hat{\mathbf{\Omega}}, E, t) \Sigma(\vec{r}, E) \delta(E - E_0) \delta(\hat{\mathbf{\Omega}} - \hat{x}) = v\Sigma A n(x) dx \quad (2.24)$$

This is effectively the loss term for a beam in the $\hat{x}$ direction, and if we set it as such, we recover the beam-type expression (2.25), which is equivalent to (2.2).

$$dn(x) = -v\Sigma A n(x) dx \qquad \Rightarrow \qquad \frac{dI(x)}{dx} = -\Sigma I \qquad (2.25)$$

Angular and Scalar Flux

Since the reactions rate density is dependent on the cross section and the velocity multiplied by the neutron distribution, this quantity is redefined as the *angular flux* as shown in (2.26). Flux meaning that it represents a rate at which particles are passing through a surface, and angular meaning that it is differential in angle. Again, since the reactions rates

depend on this quantity, the neutron transport problem is usually written in terms of the angular flux, which is then solved for instead of the neutron distribution.

$$\psi(\vec{r}, \hat{\mathbf{\Omega}}, E, t) = v(E)n(\vec{r}, \hat{\mathbf{\Omega}}, E, t) \tag{2.26}$$

Scattering and neutron movement have angular dependence, but reaction cross sections do not, so the reactions can be written in terms of angular flux which has been integrated over all angles, or the *scalar flux* or simply the *flux*. The relation between the two fluxes is show in (2.27). This is usually the most interesting quantity in reactor physics since the reactor power profile is directly proportional to it. The reaction rate for a reaction i is shown in (2.28).

$$\phi(\vec{r}, E, t) = \int_{4\pi} \psi(\vec{r}, \hat{\mathbf{\Omega}}, E, t) \tag{2.27}$$

$$R_i(\vec{r}, E, t) = \int_{4\pi} \Sigma_i(\vec{r}, E)\psi(\vec{r}, \hat{\mathbf{\Omega}}, E, t) = \Sigma(\vec{r}, E) \int_{\Omega} \psi(\vec{r}, \hat{\mathbf{\Omega}}, E, t) = \Sigma(\vec{r}, E)\phi(\vec{r}, E, t) \tag{2.28}$$

From (2.18) we can see that the time derivative is in terms of the neutron density, not the angular flux. Since reaction rates are proportional to angular flux, the balance equation will be written in terms of it. This mean that the time dependent term must be transformed to angular flux by multiplying and dividing it by the velocity as shown in (2.29).

$$\frac{\partial}{\partial t}n(\vec{r}, \hat{\mathbf{\Omega}}, E, t) = \frac{\partial}{\partial t}\frac{v(E)}{v(E)}n(\vec{r}, \hat{\mathbf{\Omega}}, E, t) = \frac{\partial}{\partial t}\frac{\psi(\vec{r}, \hat{\mathbf{\Omega}}, E, t)}{v(E)} = \frac{1}{v(E)}\frac{\partial}{\partial t}\psi(\vec{r}, \hat{\mathbf{\Omega}}, E, t) \tag{2.29}$$

Scattering

Scattering is highly dependent on angle and energy and requires as more detailed cross section than disappearance reactions. Elastic scattering has a fixed relation between outgoing energy and angle, but inelastic scattering does not. Scattering moves a neutrons energy and angle, so to be completely general in describing it, the cross section is assumed to not only ne dependent on incoming energy like all other cross sections, but also on outgoing energy, incoming angle, and outgoing angle. Since two quantities are being related before and after a scattering event, the scattering cross section is considered *doubly differential* and is sometimes called the *scattering kernel* whereas the integrated value which only depends on incoming energy is called the scattering cross section. An expression for the scattering kernel is shown in (2.30) with the incoming values primed.

$$\Sigma_s = \Sigma_s(E' \to E, \hat{\mathbf{\Omega}}' \to \hat{\mathbf{\Omega}}) \tag{2.30}$$

There is a practical reason for separating scattering into a total cross section that describes its likelihood, or probability of a neutron entering the reaction, and a kernel that

describes how it exits. In a Monte Carlo simulation, the cross section is used to determine *whether* scattering happens rather than *how* it happens. The scattering kernel is only used if a neutron has already been determined to scatter. This is also shown in the expressions for the scattering in a differential volume. The cross section is used as the removal of a neutron from a particular energy, whereas the kernel is used to determine from which other energies and angles a neutron is scatter *into*. This can be seen in (2.30). If the outgoing energy and angle are held constant, the quantity is the probability which other angles and energies scatter into it.

Using these two quantities, source and loss terms can be written for scattering. The expression for loss uses the cross section, shown in (2.31), and the kernel is used in what is often called the *scattering source* term, shown in (2.32). The source term need to be integrated over all other energies E' and angles, $\hat{\mathbf{\Omega}}'$, which neutrons can scatter into energy E and angle $\hat{\mathbf{\Omega}}$.

$$R_{s,\text{loss}}(\vec{r}, \hat{\mathbf{\Omega}}, E, t) = \Sigma_s(\vec{r}, E)\psi(\vec{r}, \hat{\mathbf{\Omega}}, E, t) \tag{2.31}$$

$$R_{s,\text{source}}(\vec{r}, \hat{\mathbf{\Omega}}, E, t) = \int_0^\infty \int_{4\pi} \Sigma_s(E' \to E, \hat{\mathbf{\Omega}}' \to \hat{\mathbf{\Omega}})\psi(\vec{r}, \hat{\mathbf{\Omega}}', E', t)d\hat{\mathbf{\Omega}}' dE' \tag{2.32}$$

Fission Source

The nuclear chain reaction is sustained by fission reactions producing neutrons, so it is essential to model it in any material that has fissionable isotopes. There are three determining factors for the fission source. The first is the fission reaction rate, which is represented by the flux multiplied by the fission cross section. The second is the fission spectrum, which is represented by χ. Neutrons born from fission are not emitted at a single energy, and the fission spectrum describes the probability for a neutron to be emitted at a certain energy. The energy spectrum is weakly dependent of the incoming neutron energy, which higher incoming energies producing more higher energy neutrons. This dependence only starts making a significant difference in spectrum shape at energies above 10MeV, higher than energies usually seen in a reactor. This is why the fission spectrum is usually treated as being independent of incoming neutron energy. The spectra of uranium-235 and plutonium-239 for fission induced by a 0.5 MeV neutron are shown in Figure 2.1, and it can be seen that plutonium-239 produces more high energy neutrons. The last parameter is the average number of neutrons emitted in a fission event, or the *fission yield*, represented by ν, and is relatively flat until about 1MeV, as was shown in Figure 2.1. Since 1MeV is within the typical energy range present in nuclear reactions, it is treated as a function of energy.

To write an expression for the fission source, or the number of fission neutrons born in energy E, we must know the number of neutrons born from fission at all energies. Once this is known, it is multiplied by the fission spectrum to select the fraction that will be in energy E. The expression for the fission source is shown in (2.33).
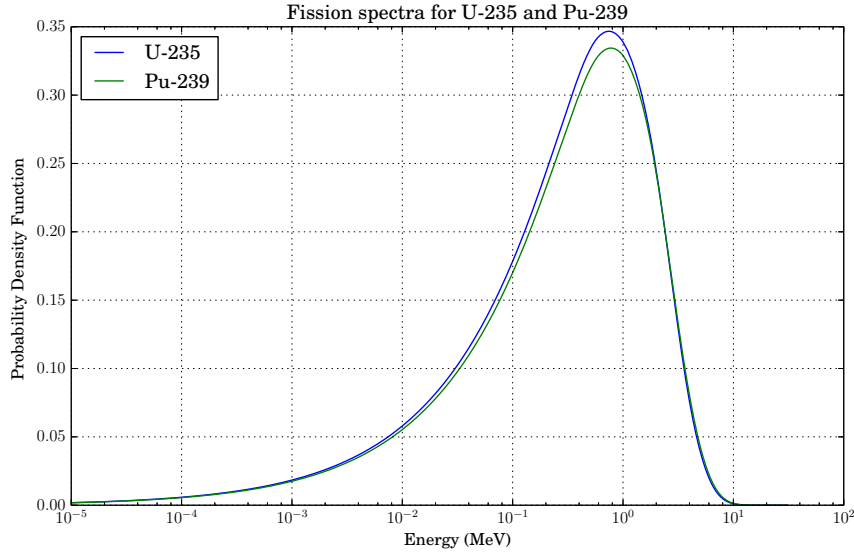
Figure 2.12: The prompt fission spectrum for uranium-235 and plutonium-239.

$$\frac{\chi(E)}{4\pi} \int_0^\infty \int_{4\pi} \nu(E')\Sigma_f(\vec{r}, E')\psi(\vec{r}, \hat{\mathbf{\Omega}}', E', t)d\mathbf{\Omega}'dE' = \frac{\chi(E)}{4\pi} \int_0^\infty \nu(E')\Sigma_f(\vec{r}, E')\phi(\vec{r}, E', t)dE'$$

(2.33)

Streaming

So far all that has been touched on is how neutrons within a volume move in angle and energy or are removed outright. Now we will go over how they move in space, and to do this we consider a surface S around our differential volume. We want to find an expression for the net number of neutrons passing through this surface and this will be our net leakage term (incoming minus outgoing). The angular neutron flux describes the number of neutrons crossing a differential surface at angle $\hat{\mathbf{\Omega}}$, but in order to perform any vector operations on it, we must multiply it by the unit vector. We can then write and expression for the leakage by performing a surface integral over the current and surface normal's dot product, shown in (2.34).

$$\text{Leakage} = \int_S ds \cdot \hat{\mathbf{\Omega}}\psi(\vec{r}, \hat{\mathbf{\Omega}}, E, t)$$

(2.34)

We can turn this into a volume integral by applying the divergence theorem, shown in (2.35). If the volume of interest is shrunk to an infinitesimal volume and we switch the order of the dot product, we come to an expression that can be used in the balance equation. This

expression, shown in (2.36), is often called the *streaming* term, since it describes the net movement of neutrons into the differential volume due to their physical movement.

$$\int_S ds \cdot \hat{\boldsymbol{\Omega}}\psi(\vec{r}, \hat{\boldsymbol{\Omega}}, E, t) = \int_V dV \nabla \cdot \hat{\boldsymbol{\Omega}}\psi(\vec{r}, \hat{\boldsymbol{\Omega}}, E, t) \tag{2.35}$$

$$\lim_{V \to dV} \int_V dV \nabla \cdot \hat{\boldsymbol{\Omega}}\psi(\vec{r}, \hat{\boldsymbol{\Omega}}, E, t) = \hat{\boldsymbol{\Omega}} \cdot \nabla \psi(\vec{r}, \hat{\boldsymbol{\Omega}}, E, t) \tag{2.36}$$

Neutron Transport Equation

Now all the terms of (2.18) have been explicitly defined. Substituting them all in yields (2.37), the *neutron transport equation*, written here with the neutron sinks on the left side and the neutron sources on the right side. An additional external source term, S_{external}, has been added to account for any sources not induced by the neutron flux itself, i.e. external sources.

$$
\begin{aligned}
&\frac{1}{v(E)}\frac{\partial}{\partial t}\psi(\vec{r}, \hat{\boldsymbol{\Omega}}, E, t)+ \\
&\quad \hat{\boldsymbol{\Omega}} \cdot \nabla \psi(\vec{r}, \hat{\boldsymbol{\Omega}}, E, t)+ \\
&\quad \Sigma_t(\vec{r}, E)\psi(\vec{r}, \hat{\boldsymbol{\Omega}}, E, t) \\
&\qquad\qquad = \\
&\qquad \int_0^\infty \int_{4\pi} \Sigma_s(E' \to E, \hat{\boldsymbol{\Omega}}' \to \hat{\boldsymbol{\Omega}})\psi(\vec{r}, \hat{\boldsymbol{\Omega}}', E', t)d\boldsymbol{\Omega}'dE' \\
&\qquad + \frac{\chi(E)}{4\pi}\int_0^\infty \int_{4\pi} \nu(E')\Sigma_f(\vec{r}, E')\psi(\vec{r}, \hat{\boldsymbol{\Omega}}', E', t)d\boldsymbol{\Omega}'dE' \\
&\qquad + S_{\text{external}}
\end{aligned}
\tag{2.37}
$$

The neutron transport equation in this form is an integro-differential equation since it has both derivates and integrals in it. It's spatial parts are differential, whereas its angular and energy parts are integral. It is linear and relatively easy to solve for simple geometries and reaction parameters. The complexity in solving it comes from the complex energy dependence of the reaction parameters and the large and complex domains over which it must be solved in order to capture all the relevant physics. The reaction parameters can span more then 12 orders of magnitude, from 1×10^{-11} to 1×10^1 MeV and above, and the geometries involved can include millions of individual material regions with many different mixtures of material within them.

Time Independent Neutron Transport Equation

In most situations, the equilibrium state of the neutron population is of interest. By setting all time derivates to zero, (2.37) becomes the time-independent neutron transport

equation shown in (2.38). Another parameter has been introduced, the effective multiplication factor, k_{eff}, which was described earlier using the seven factor formula. It is introduced here since by forcing the time derivates to be zero, it is implied that the multiplication factor must be equal to one. The material and geometric properties of a problem may say otherwise, however, and failing to divide the fission source by the effective multiplication factor could make time independent equation inconsistent. Of course, there will be no inequality if the source and sink terms are actually balanced and k_{eff} calculated from (2.37) is one.

$$
\begin{aligned}
\hat{\boldsymbol{\Omega}} \cdot \nabla \psi(\boldsymbol{\vec{r}}, \hat{\boldsymbol{\Omega}}, E, t) + & \\
\Sigma_t(\boldsymbol{\vec{r}}, E)\psi(\boldsymbol{\vec{r}}, \hat{\boldsymbol{\Omega}}, E, t) & \\
& = \\
& \int_0^\infty \int_{4\pi} \Sigma_s(E' \to E, \hat{\boldsymbol{\Omega}}' \to \hat{\boldsymbol{\Omega}})\psi(\boldsymbol{\vec{r}}, \hat{\boldsymbol{\Omega}}', E', t)d\boldsymbol{\Omega}'dE' \\
& + \frac{1}{k_{\text{eff}}}\frac{\chi(E)}{4\pi}\int_0^\infty \int_{4\pi} \nu(E')\Sigma_f(\boldsymbol{\vec{r}}, E')\psi(\boldsymbol{\vec{r}}, \hat{\boldsymbol{\Omega}}', E', t)d\boldsymbol{\Omega}'dE' \\
& + S_{\text{external}}
\end{aligned}
\tag{2.38}
$$

2.2 Discrete Methods

The usual method in solving continuous differential and integral equations on a computer is to discretize the domain over which it is being solved. This can be applied to the neutron transport equation as well. Since energy is involved in integrals in the neutron transport equation, it can be discretized by integrating over set boundaries and treating the reaction parameters as constant within them. The same method can be applied to them angular dependence. The simplest way of discretizing the geometry is to voxelize the domain, laying down a uniform grid in every dimension and treating the material within each voxel as being uniform. Discrete geometry representation is and area where an lot of error can occur since it is three dimensional and scales can range from millimeters to hundred of meters.

derivations? moments, diffusion, SN?

Although discrete methods are readily mapped to computer hardware and execute quite fast, there are drawbacks and limitations in using them. The biggest drawback is that they're accuracy greatly depends and the discretization methods used. MORE.

2.3 Monte Carlo

The Monte Carlo method takes a different approach in mapping the transport problem to a computer. Instead of the neutron population being continuous and the spatial, angular, and energy dimensions being discretized, it leaves the dimensions continuous and discretizes the neutron population! Quantities of interested are then determined by integrating over the

neutron population! This way of integrating the neutron transport equation can be thought of as integrating it "sideways." Deterministic approaches integrate it by discretizing the dimensions and ensuring consistency of the transport equation between the discrete points. This is the "standard" way of integrating the transport equation. In the Monte Carlo method, the discretized neutron "shoots through" the transport equation many times, sweeping out the entire phase space, effectively integrating it when a sum is done over the individual neutron realizations, or *histories*. It is important to note that the Monte Carlo method does not actually solve the transport equation, per se, but rather is able to *estimate* the solution very well.

There are advantages in using the Monte Carlo method for neutron transport, some of which have been mentioned. The first and foremost is that it that very few assumptions must be done to the physics and the geometry of the problem. Paraphrasing Forrest Brown when he spoke at the Supercomputing in Nuclear Applications and Monte Carlo conference on October 30th, 2013 in Paris, the only reason people use Monte Carlo methods is that they are able to produce physically accurate results and are trusted to so so. Anyone developing Monte Carlo code should not abuse this trust and must put accuracy first and foremost in their implementations.

There are considerable drawbacks to using Monte Carlo as well; primarily convergence rates and silent inaccuracies. Since it is a statistical calculation, it is bound by statistical laws which dictate that the error in Monte Carlo calculations reduces as $1/\sqrt{N}$, where N is the number of histories performed. Also, since geometries can be large, very small but very influencing volumes can be missed by the neutron random walk if the number of histories is not large enough. The estimated statistical error may be low at N histories, but there is a chance the entire phase space has not yet been sufficiently sampled to produce accurate results. Deterministic methods would not have this problem since there will be a discretized point in this small volume, and it will contribute to the overall calculation. The slow convergence rate can be combated by using parallel computing, however. Requiring that neutrons do not interact with each other was an assumption that went into deriving the neutron transport equation, and can be exploited in Monte Carlo simulations. Since each neutron history is completely independent, histories can be run completely in parallel without communication until the results are combined at the end of the simulation. This leads to very good scaling of parallel calculations, which can be used to reduced run times of large problems to acceptable values.

Statistics

To understand how the Monte Carlo method performs this integration, simple equations and derivations and such as calculating π can be done. First, given a continuous random variable x that follows the probability distribution P, the mean value can be calculated by taking the first moment as was shown in (2.20) and is reproduced in (2.39). But we are interested in determining the mean from a set measurements rather than the underlying

Figure 2.13: The Monte Carlo random walk process.

distribution. If we have N independent measurements of quantity x, this is simply taking the arithmetic mean, as shown in (2.40).

$$\mu = \int_{-\infty}^{\infty} xP(x)dx \qquad (2.39)$$

$$\bar{X}_N = \frac{1}{N} \sum_i^N x_i \qquad (2.40)$$

The shape of the distribution about the mean is important as well, since it quantifies the uncertainty. The variance can be computed by performing the central second moment on P and estimated by Bessel's Correction on the standard variance estimator (2.42)[]. It is useful that the sample mean can be removed from the sum of the sample squares since this

means that only the sample sum and the sum of sample squares need be stored. The entire sample set does not need to be stored and used to compute the variance at the end when the sample mean is known.

$$\sigma^2 = \int_{-\infty}^{\infty} x^2 P(x) dx - \mu^2 \tag{2.41}$$

$$\sigma_N^2 = \frac{1}{N-1} \sum_i^N (x_i - \mu)^2 = \frac{1}{N-1} \left(\sum_i^N x_i^2 - N\bar{X}_N^2 \right) \tag{2.42}$$

The law of large number states that (2.40), the sample mean, will converge to the true mean (2.39) in the limit where $N \to \infty$.

$$Pr\left(\lim_{N \to \infty} \bar{X}_N = \mu \right) = 0 \tag{2.43}$$

The rate at which the estimate (2.40) converges to the true mean (2.39) comes from the central limit theorem. Shown in (2.44), it states that the distribution of the means of taken from a large set of independent random variables will be normally distributed.

$$\sqrt{N} \left(\left(\frac{1}{N} \sum_i^N x_i \right) - \mu \right) \xrightarrow{d} \mathcal{N}(0, \sigma^2) \tag{2.44}$$

In this case, if a large number of measurements are taken, we would like to know how close the mean of a completely independent set measurements will be to the true mean. In other words, the variance of the mean is desired. Luckily, this can be estimated with only one measurement instead of directly computing the variance with many measurements. The Bienaymé formula, shown in (2.45), states that the variance of a sum of uncorrelated random variables is equal to the sum of the variances of the variables.

$$Var\left(\sum_{i=0}^N x_i \right) = \sum_{i=0}^N Var(x_i) \tag{2.45}$$

Applying the Bienaymé formula and the variance relation show in (2.46) to the sample mean results in an expression for the variance of the mean, $Var(\bar{X}_N)$, in terms of the variance of the sample, σ_N^2. This expression is shown in (2.48).

$$Var\left(aX \right) = a^2 Var\left(X \right) \tag{2.46}$$

$$Var(\bar{X}_i) = Var\left(\frac{1}{N} \sum_{i=0}^N x_i \right) = \frac{1}{N^2} \sum_{i=0}^N Var(x_i) = \frac{N\sigma_N^2}{N^2} = \frac{\sigma_N^2}{N} \tag{2.47}$$

$$\frac{\sigma_N^2}{N} = \frac{1}{N(N-1)} \left(\sum_i^N x_i^2 - N\bar{X}_N^2 \right) = \frac{1}{(N-1)} \left(\frac{1}{N} \sum_i^N x_i^2 - \left(\frac{1}{N} \sum_i^N x_i \right) \right) \tag{2.48}$$

It should be noted that (2.47) implies that the standard deviation of the mean always scales as $1/\sqrt{N}$. This is a blessing and a burden in that it means any calculation will variance of the mean will go to zero, or the sample mean will converge to the true mean, as long it is run long enough and enough samples are collected, but that it will converge as $1/\sqrt{N}$, which is slow.

Since the central limit theorem states that it is normally distributed, we can make use of the well-know properties of the normal distribution, namely the confidence interval. The normal distribution has well-defined confidence intervals: 68% of the population will lie within a single standard deviation, σ, and 95.5% will lie within 2σ. From this knowledge, an expression for the realtive error can be written. The expression for the relative error shown in (2.49) is for the 68% confidence level, and simply needs to be doubled for the 95% level [**MCNP**][4].

$$\text{Rel.Err.} = \frac{\sigma}{\bar{X}_N} = \frac{1}{\bar{X}_N}\sqrt{\frac{1}{(N-1)}\left(\frac{1}{N}\sum_i^N x_i^2 - \bar{X}_N^2\right)} \tag{2.49}$$

Sampling Schemes

In order to actually transport a neutron across the geometry in a Monte Carlo simulation, a the probability distributions of the reaction types must be sampled in order to produce accurate distributions when the histories are aggregated. There are two methods that are commonly used to do so; the direct inversion method and the rejection method.

The *direct inversion* method relies and being able to analytically integrate the probability distribution function (PDF) to create a cumulative distribution function (CDF) and being able to invert it. The first step is to integrate the PDF to find the CDF, as shown in (2.50).

$$CDF(x) = \int_0^x PDF(x')dx' \tag{2.50}$$

Since a PDF must be positive and normalized by definition, the CDF must range from 0 to 1 and increase monotonically. Setting the CDF equal to a uniformly distributed random number ξ and solving for the value to be sampled yields the sampling scheme. Figure 2.3 shows this graphically. The CDF describes the probability that a value of a random number x obeying the PDF will be below the value x. Taking an interval gives the probability x will be in the interval, as shown in (2.51). Since the width of inverse CDF is directly proportional to the PDF at the same value of x, as shown in (2.52), and if ξ is uniformly distributed on [0,1), the interval $\Delta\xi$ will contain a fraction of samples on [0,1) equal to $P(x_1 < x_i < x_2)$. As the interval is reduced to 0, $P(x_i) = PDF(x)dx = d\xi$ (again, since ξ is uniformly distributed on [0,1), leading to the sampling scheme shown in (2.53). This method is only useful if the CDF is simple enough to be inverted analytically.

$$CDF(x_2) - CDF(x_1) = P(x_1 < x_i < x_2) = \int_{x_1}^{x_2} PDF(x)dx \tag{2.51}$$

$$CDF(x_2) - CDF(x_1) = \Delta\xi = \int_{x_1}^{x_2} PDF(x)dx \qquad (2.52)$$

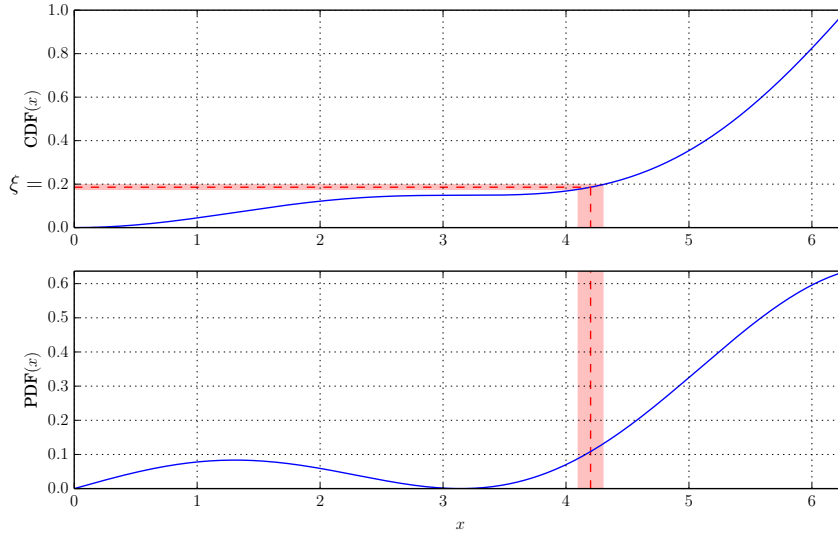$$x_i = CDF^{-1}(\xi_i) \qquad (2.53)$$

Figure 2.14: Sampling $PDF(x) = \frac{1}{2\pi^2}(x \cos x - x)$ by the direct inversion method on the domain $(0, 2\pi)$.

When the CDF is not invertible, the *rejection sampling* method can be used. It uses a secondary PDF, fx, that is greater at all points than the PDF to be sampled from and whose CDF is easily inverted. Since the PDF is normalized, $f(x)$ now corresponds to probabilities greater than one. This is simply a scaling problem, and the random numbers used to directly sample from it must be uniform on $[0, \int_{\text{domain}} f(x))$ instead of $[0, 1)$. Then two random numbers are generated. The first, ξ_1 is sampled from a $f(x)$ using a scaled random number as mentioned, and the second, ξ_2 is uniformly distributed on $[0, f(\xi_1))$. If $\xi_2 < PDF(\xi_1)$, the sampled value (ξ_1), is added to the population, else it is *rejected*, or discarded from the population. The set of accepted values of ξ_2 will then follow $PDF(x)$. Figure 2.3 shows an illustration of rejection sampling.

Using these two methods of sampling from probability distributions, expressions can be found for every reaction type and transport phenomenon that a neutron undergoes in it's random walk through matter. The specific schemes are described in the following subsections.
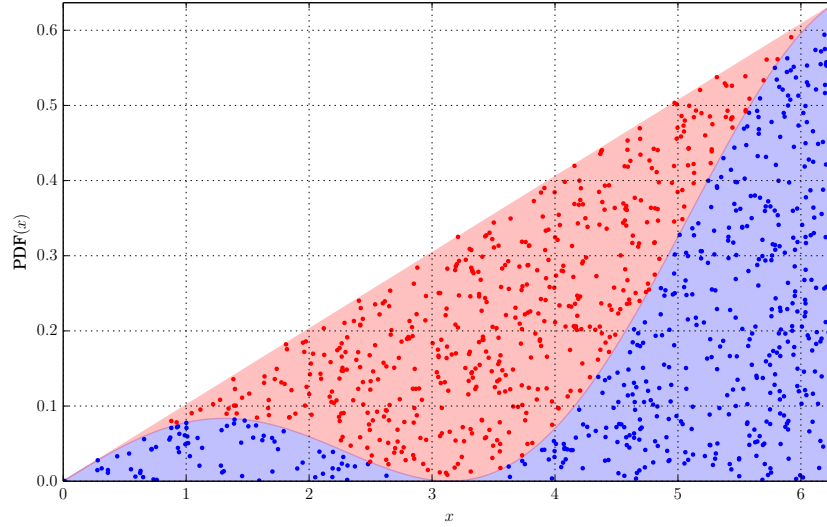
Figure 2.15: Sampling $PDF(x) = \frac{1}{2\pi^2}(x\cos x - x)$ by the rejection method on the domain $(0, 2\pi)$. The auxiliary function $f(x) = x/\pi$ and the random number used to sample from it must be on $[0, 2\pi)$ instead of $[0, 1)$.

Distance To Interaction

The first step in stochastically transporting a neutron through matter is finding the distance to the next interaction. The expression for the probability of non-interaction was shown in (2.3). The probability of interaction in dx around x is $\Sigma_t dx$. Therefore the probability of not interacting from 0 to x and then interacting at dx around x is simply the product. This expression, shown in (2.55), is commonly referred to as the *first collision probability*.

$$P(x) = \Sigma_t e^{-\Sigma_t x} \tag{2.54}$$

This expression is simple and can be inverted for direct sampling. The sampling scheme is shown in (2.56). The expression $1-\xi$ can be replaced with ξ since it is identically distributed.

$$CDF(x) = \int_0^x \Sigma_t e^{-\Sigma_t x'} dx' = 1 - e^{-\Sigma_t x} \tag{2.55}$$

$$x_i = CDF^{-1}(\xi) \qquad \Rightarrow \qquad x_i = \frac{-\ln(1-\xi)}{\Sigma_t} = \frac{-\ln(\xi)}{\Sigma_t} \tag{2.56}$$

The total cross section Σ_t is the total macroscopic cross section of the material the neutron is traveling through, i.e. it is the sum of the total macroscopic cross sections of its constituent isotopes as was shown in (2.7).

Isotope and Reaction Selection

Now that the neutron has been determined to interact, it must be decided with which isotope it interacts with. The discrete PDF of this process is shown in (2.57), where there are i isotopes in the material each having a total macroscopic cross section $\Sigma_{t,i}$. The distribution is sampled by generating a uniform random number, ξ, on $[0, 1)$ and performing a running sum over the individual isotope's total macroscopic cross sections. When $CDF_i > \xi$, the neutron collision is sampled to happened in isotope i.

$$PDF_i = \frac{\Sigma_{t,i}}{\Sigma_t} \qquad \Rightarrow \qquad CDF_i = \frac{1}{\Sigma_t} \sum_{n=1}^{i} \Sigma_{t,n} \tag{2.57}$$

Determining the reaction type is done in a similar fashion, except the number density of the material is no longer a concern since the isotope has already been selected. Therefore, the running sum of the CDF is done over the isotope i's microscopic cross sections. Another random number is generated, and when $CDF_k > \xi$, the neutron is sampled to undergo reaction k in isotope i.

$$PDF_k = \frac{\sigma_k}{\sigma_t} \qquad \Rightarrow \qquad CDF_k = \frac{1}{\sigma_t} \sum_{n=1}^{k} \sigma_n \tag{2.58}$$

Free Gas Treatment

This method introduces another type of sampling method which is based on mathematical first principles instead of integration of a PDF.

Elastic Scattering

Inelastic Reactions

Fission

fission dist
source pop

Interpolation

"Continuous Energy" = interpolation

2.4 GPUs

Part of the reason GPUs are able to perform efficiently is do to their reliance on single instruction, multiple data (SIMD) execution. This execution method uses the same instructions carried out over multiple pieces of data at the same time. This reduces the amount

of power used in control and therefore more math can be done per watt [citation definitely needed]. The GPU programming model abstracts SIMD execution by using threads, which can be thought of in the traditional sense . There are some tradeoffs, however, including limited on-board memory (currently 12 GB), limited cache and control space, and requiring data parallelism for full utilization.

Since all threads in a GPU thread block must execute the same instructions, having conditional statements based on random numbers can cause threads to be serialized, leading to resource under-utilization. For example, in particle transport where a particle history is assigned to each thread, thread divergence mainly arises through particles undergoing different reactions based on the results of cross-section sampling. Also, GPUs have even higher global memory latency than CPUs, up to 800 clock cycles on older cards, compared to about 50 cycles on a typical CPU. It is also important to remember than GPUs are clocked much slower than CPUs, so the memory latency performance issue is even greater [5, 7].

another brief introduction.

advantages

pitfalls, weaknesses

Supercomputing

how and why they came to be used for supercomputing.

Architecture

SIMD, threads

Memory

speed, levels, etc. limitations

CUDA

kernels

2.5 Previous Works

talk about the work that other people did

2.6 Preliminary Studies

all my prelim work to motivate this project

2D Scattering Game

Ray Tracing with OptiX

2.7   Scope in Detail

Chapter 3

GPU Implementation

typical task-based MC. data-parallel MC - wide but single task, homework grading, weaving!

3.1 Data Layout

brief introduction, not using S(a,b) or unresolved resonance parameters at this point

unionized

unionized glory, why its good and bad [4].

linked lists

how to resolve the data heterogeny problem

Embedded Python

PyNE
amazing Python C API

Pseudo-Random Numbers

3.2 Tasks

breakdown of tasks into sub-tasks, actually going through how data is stored, accessed, etc

Grid Search

binary search here? more logN? quad tree?

Geometry Representation

stuff about geom and "where am I?"

OptiX

Acceleration Structures

Importance of logN

Point-in-Polygon / Gauss's Law / Divergence Thoerem

Roll the where am I into trace
cellular geometry as opposed to combinatorial

Scaling

optix scaling toy problem - need to have very large sets in order to hide overhead

Macroscopic

material

Microscopic

nuclide

Interactions

Scattering

Rodriguez rotation

Secondary-Producing

Disappearance

Concurrent Kernels

Parallel Operations

CUDPP, reductions, sums, and sorts - sorting by reaction to keep warps full, memory implications

Fixed source subcritical multiplication

necessary to be subcritical.
pop can pop source particles in to keep population constant until NPS exceeded...

Criticality

cannot simply pop, unless generation number is associated, then data can be generated and post-processed?

need to do generations since the source update is generation-dependent, can this be changed?

Chapter 4

Results

Chapter 5

Conclusions and Future Work

replace geom, parallel routines with handwritten, specific ones - better geom performance with woodcock maybe - dynamica parallelism can be implemented and kernel launch overhead minimized.

Appendix A

Appendix

Table A.1: ENDF MT numbers and the reactions they stand for.

Sum and Elastic Cross Sections

MT	Reaction	Description
1	(n,total)	Neutron total cross section.
2	(z,z0)	Elastic scattering cross section.
3	(z,nonelastic)	Nonelastic cross section.
4	(z,n)	Production of one neutron in the exit channel. Sum of MT=50-91.
5	(z,anything)	Sum of all reactions not given explicitly in another MT number.
10	(z,continuum)	Total continuum reaction; exclues all discrete reactions.
27	(z,abs)	Absorption. Sum of MT=18 and MT=102-117.
101	(z,disap)	Disappearance. Sum of MT=102-117.

Neutron-Producing & Continuum Reactions

MT	Reaction	Description
11	(z,2nd)	Production of two neutrons and a deuteron, plus a residual.
16	(z,2n)	Production of two neutrons, plus a residual.
17	(z,2n)	Production of three neutrons, plus a residual.
18	(z,fission)	Total fission.
19	(z,f)	First-chance fission.
20	(z,nf)	Second-chance fission.
21	(z,2nf)	Third-chance fission.
22	(z,na)	Production of a neutron and alpha particle, plus a residual.
23	(z,n3a)	Production of a neutron and three alpha particles, plus a residual.
24	(z,2na)	Production of two neutrons and an alpha particle, plus a residual.
25	(z,3na)	Production of three neutrons and an alpha particle, plus a residual.
28	(z,np)	Production of a neutron and a proton, plus a residual.
29	(z,n2a)	Production of a neutron and two alpha particles, plus a residual.
30	(z,2n2a)	Production of two neutrons and two alpha particles, plus a residual.
32	(z,nd)	Production of a neutron and a deuteron, plus a residual.
33	(z,nt)	Production of a neutron and a triton, plus a residual.
34	(z,n3He)	Production of a neutron and a 3He particle, plus a residual.
35	(z,nd2a)	Production of a neutron, a deuteron, and two alpha particles, plus a residual.
36	(z,nt2a)	Production of a neutron, a triton, and two alpha particles, plus a residual.
37	(z,4n)	Production of four neutrons, plus a residual.
38	(z,3nf)	Fourth-change fission.
41	(z,2np)	Production of two neutrons and a proton, plus a residual.
42	(z,3np)	Production of three neutrons and a proton, plus a residual.
44	(z,n2p)	Production of a neutron and two protons, plus a residual.
45	(z,npa)	Production of a neutron, a proton, and an alpha particle, plus a residual.

Neutron-Producing Discrete Reactions

MT	Reaction	Description
51	(z,n1)	Production of a neutron, nucleus in the first excited state.
52	(z,n2)	Production of a neutron, nucleus in the second excited state.
...		
90	(z,n40)	Production of a neutron, nucleus in the 40th excited state.
91	(z,nc)	Production of a neutron in the continuum.

Reactions That Do Not Produce Neutrons

MT	Reaction	Description
102	(z,gamma)	Radiative capture.
103	(z,p)	Production of a proton, plus a residual.
104	(z,d)	Production of a deuteron, plus a residual.
105	(z,t)	Production of a triton, plus a residual.
106	(z,3He)	Production of a He particles, plus a residual.
107	(z,a)	Production of an alpha particle, plus a residual.
108	(z,2a)	Production of two alphas, plus a residual.
109	(z,3a)	Production of three alphas, plus a residual.
111	(z,2p)	Production of two protons, plus a residual.
112	(z,pa)	Production of a proton and an alpha particle, plus a residual.
113	(z,t2a)	Production of a triton and two alphas, plus a residual.
114	(z,d2a)	Production of a deuteron and two alphas, plus a residual.
115	(z,pd)	Production of a proton and a deuteron, plus a residual.
116	(z,pt)	Production of a proton and a triton, plus a residual.
117	(z,da)	Production of a deuteron and an alpha particle, plus a residual.

Table A.2: Reaction number encodings in WARP.

Number	Description
1-117	Reaction described by MT number
888	done flag
999	Resample

Bibliography

[1]   D.E. Cullen and C.R. Weisbin. "Exact Doppler broadening of tabulated cross sections. [SIGMA 1 kernel broadening method]". In: *Nucl. Sci. Eng.* (July 1976).

[2]   James J. Duderstadt and Louis J. Hamilton. *Nuclear Reactor Analysis.* New York, NY: John Wiley & Sons, Inc., 1976.

[3]   M. Holloway and R. Baker. *Note on the Origin of the Term "Barn".* Tech. rep. LANL No. LAMS-523, 13 March 1944.

[4]   JAAKKO LEPPÄNEN. "Development of a New Monte Carlo Reactor Physics Code". D.Sc. Dissertation. Helsinki, Finland: Helsinki Institute of Technology, 2007.