2-1-2016

A Novel Neutron Recoil Spectrometer Concept Utilizing Heavy-Ion Recoils and Time- and Spatially-Resolved Sensor

Joel Long

Candidate

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

, Chairperson

A Novel Neutron Recoil Spectrometer Concept Utilizing Heavy-Ion Recoils and Time- and Spatially-Resolved Sensor

by

Joel Long

B.S., University of New Mexico, 2012

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Nuclear Engineering

The University of New Mexico

Albuquerque, New Mexico

December, 2015

Dedication

To my parents, who always encouraged my curiosity and desire to improve things. To my wife, who not only tolerated this long journey, but was a constant source of encouragement, And to the Lord, the great Engineer.

Acknowledgments

I would like to thank my advisor, Dr. Gary Cooper, and the neutron diagnostics and laser diagnostics teams at Sandia National Labs, who provided a great deal of the background knowledge that makes this idea feasible. Thanks also to Dr. Ray Leeper, Dr. Brent Jones, and Dr. Ray Leeper who have helped support my education to this point. I would like to thank my parents, who worked to give me every opportunity at a great education, sometimes in spite of myself, and always encouraged the critical and analytical thinking at the heart of engineering. I owe a great debt to my wife, who has stood by me on the long journey I have taken to get here, sharing the seemingly endless trail of triumphs and setbacks. And finally, thanks to the Lord, who gave me what gifts and abilities I possess.

A Novel Neutron Recoil Spectrometer Concept Utilizing Heavy-Ion Recoils and Time- and Spatially-Resolved Sensor

by

Joel Long

B.S., University of New Mexico, 2012

M.S., Nuclear Engineering, University of New Mexico, 2015

Abstract

This thesis describes a heavy-ion recoil neutron spectrometer as an alternative to existing magnetic proton recoil neutron spectrometers for inertial confinement fusion experiments. Whereas existing designs elastically scatter neutrons off protons or deuterons, which are then magnetically analyzed to infer energy, this device works by elastically scattering neutrons in a nanometer-scale film of a higher-Z material. The resulting heavy ions are then measured on a time- and space-resolved semiconductor detector. Using this time and location information, coupled with the equations governing the mechanics of elastic collisions, one can determine the energy of the neutron that scattered the ion. This thesis considers the many influential design factors, including cross sections, ion straggling, timing, and foil and sensor geometries with respect to the resulting viability and performance. Finally, two reference designs are presented for comparison with the magnetic proton recoil neutron spectrometer currently fielded at the National Ignition Facility. For a monoenergetic 14-MeV neutron source, the minimum achieved full-width-at-half-maximum is 0.33 MeV,

compared with 0.69 MeV for the currently fielded detector. The maximum efficiency is found to be 2.94x10^{-10}, compared with a current maximum of 8.48x10^{-11}.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Neutron Spectroscopy for Intertial Confinement Fusion

It is difficult to measure the energy distribution of a neutron source. High-energy neutrons have much lower cross-sections than charged particles (typically less than 2 barns at 10 MeV), so they barely interact in a moderately-sized detector. Interactions deposit only a stochastically-determined fraction of their energy, so even if a neutron is detected, ascertaining its energy is a challenge.

In many cases, however, information about the energy spectrum of neutrons from an experiment is one of the best ways to assess what is happening. In inertial confinement fusion experiments, huge amounts of energy are packed into a tiny pellet of deuterium or mixed deuterium and tritium. This energy very briefly compresses the pellets to incredible densities, resulting in a variety of exotic physical processes, fusion amongst them. In the space of picoseconds to nanoseconds (depending on the system), the compressed pellet blows apart in a wave of gamma, neutron, and

charged particle radiation. The neutrons that come out of this reaction are one of the best known ways to assess what happens during that extremely brief period. The number of neutrons that comes out of the reaction is called the yield, and is related to how efficiently energy was channeled into the pellet. Many neutrons lose energy escaping the highly-condensed pellet, and the resulting down-scattered spectrum can determine the compression achieved, termed the ρR. Finally, the energy spectrum shows whether or not the fusions that occurred were from thermonuclear fusion and what the ion temperature was. These provide ample motivation to determine the energy spectrum of neutrons from these types of reactions.

Two of the most successful approaches to neutron spectroscopy are time-of-flight spectrometers and proton recoil magnetic neutron spectrometers. The former use large scintillation detectors to look at the total energy deposition versus time and uses coincidence counting and sophisticated mathematics to unfold the energy spectrum of neutrons that passed through. The latter, which this thesis builds on, places a micrometers-thin foil of a hydrogen- or deuteron-rich material in the neutron path. When neutrons scatter in the foil, some of the resulting ions escape. The spectrometer collimates theses ions down to just those which scattered nearly straight forward, since their energy is related to the energy of the originating ions by

$$E_{ion} = \frac{4 * m_{neutron} M_{ion}}{(m_{neutron} + M_{ion})^2} \tag{1.1}$$

These ions then pass through a strong magnetic field. Since the radius of curvature of a charged particle in a magnetic field is a function of its velocity (and consequently energy), the ions' energy distribution is spatially resolved by the magnetic field. The particles are then measured by spatially-sensitive detectors. Coupled with the energy relationship above, the neutron spectrum is resolved. This type of system has been successfully implemented at several facilities, including the Joint European Torus [10] and the National Ignition Facility [4].

The spectrometer fielded at the National Ignition Facility is a good reference design to illustrate the limitations of recoil spectrometers, so it is worthwhile to consider it in more detail. Starting from the point closest to the implosion, the nearest component is a 1.57 millimeter tantalum plate, serving as a blast shield. Next, 5 millimeters further (a total of 26 cm) from the implosion, is the foil, which can be made from CH_2 or CD_2, depending on the energy range of interest. The foil is the most flexible component of the spectrometer, and different foils can be interchanged for different efficiency/resolution requirements. In all configurations it has an area of 12.8 square cm^2 and thicknesses ranging from 47 um for the maximal resolution to 259 um for the best efficiency.

Protons (or deuterons, in CD_2 configurations) that escape the foil then reach the magnet, 570 cm from the foil. The aperature to enter the magnet is only 20 cm^2, arranged as a 2 by 10 cm slot. This represents an average accceptance angle of only 0.22°. Protons scattered outside this angle — the large majority — are rejected. The magnet is a neodymium iron boron permanent magnet in a wedge shape, with a peak field of 0.9 tesla. This magnet separates the beam according to energy before it arrives at the CR39 detector array.

The detector array consists of nine blocks of CR39 positioned to collect a wide range of ions coming out of the magnet. Since the magnet has altered the trajectory of the ion according to its energy, the location where the ion strikes the CR39 determines the ion's energy. The ion leaves a trail of molecular damage in the CR39 which can be scanned later to get a count of how many ions of each energy struck the CR39. Since the forward-scattered ion energies are correlated to the originating neutron energy, this also determines the neutron spectrum. CR39 is immune to damage from gamma rays, which do not leave the type of molecular damage that ions do; however, it is sensitive to neutrons, which elastically scatter in the plastic to produce secondary ions and accompanying damage trails. To minimize this source of noise,

the whole assembly is surrounded with roughly 6000 pounds of polyethylene shielding to attenuate the primary neutron signal.

This design has several inherent shortcomings. Most notably, a very low efficiency, on the order of 10^{-11}. The powerful magnet is also expensive and inflexible. Other designs use electromagnets to support a wider range of energy configurations, but this introduces problems with field consistency and calibration. The resulting system can be adjusted primarily by changing the foil thickness and using deuterium or hydrogen. This gives it a slightly adjustable efficiency and resolution, and two settings for the neutron energy range it can measure. The heavy-ion recoil neutron spectrometer concept presented here works to address some of these limitations.

1.2 Theory of Operation

The layout of the heavy-ion recoil neutron spectrometer is shown below in Figure 1. Neutrons are created by a short-lived point source, typically an inertial confinement fusion experiment such as those carried out on the Z-Machine or National Ignition Facility. A short distance away is a foil of some relatively heavy material. Some of the neutrons elastically scatter, creating heavy ions with one-to-one energy-angle relationship imposed by the laws of mechanics. After some straggling as it escapes the foil and passing through the intervening vacuum, the ion deposits its energy on the space- and time-resolved sensor.

If the neutron source can be approximated as a delta source in time, then knowledge of when and where each ion arrives at the sensor, along with the geometry of the detector, is sufficient to determine the energy spectrum of source neutrons. The location on the sensor at which the energy is deposited resolves the angle at which the neutron was scattered, as well as the distance traveled to reach the detector. This distance, combined with the time at which the energy is deposited, determines

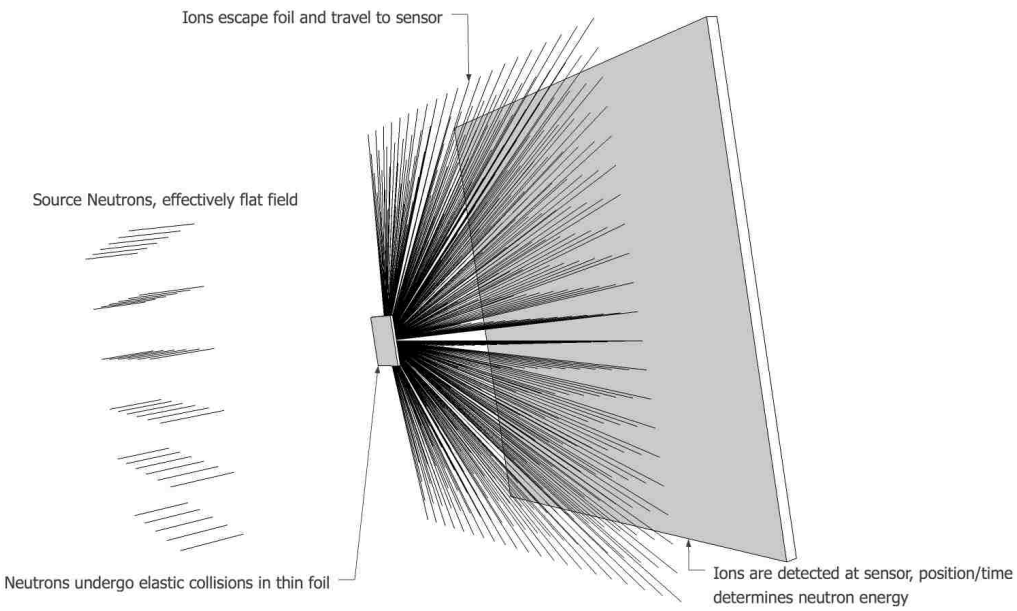

Figure 1.1: A schematic of the components of the heavy-ion recoil neutron spectrometer.

the energy of the ion. This energy, combined with the angle, determines the energy of the scattered neutron. More precisely, suppose we have a neutron source which is effectively flat-field over the solid angle of the detector. For simplicity, we will also assume that the detector subtends a small enough solid angle that the neutrons may be treated as perpendicular to the foil. This source is described as:

$$f(x, y, E) = f(E) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{1.2}$$

where

$$\sigma = E_{peak} + 0.4697881\ T^{\frac{2}{3}} + 0.37570337\ T \tag{1.3}$$

$$\mu = \sqrt{1.0018756\ \sigma\ T} \tag{1.4}$$

This is a gaussian approximation of the energy distribution of neutrons from a thermonuclear fusion source. In thermonuclear fusion, the energy spread of the neutrons is determined by the temperature of the ions in the plasma, which is traditionally expressed in units of energy. In this equation, T is the temperature, in keV. Existing libraries [5] contain the differential cross sections for the foil material, and mechanics yields the equation relating the scattered angle to the ion energy,

$$E_{ion} = \frac{4M_{ion}m_{neutron}}{(m_{neutron} + M_{ion})^2} sin(\frac{\pi - 2\theta}{2})^2 E_{neutron} \tag{1.5}$$

Where E_{ion} is the ion energy, M_{ion} is the ion mass, $m_{neutron}$ is the neutron mass, θ is the ion scattering angle, and $E_{neutron}$ is the pre-scatter energy of the neutron. All values in this equation are in the lab frame, and it is assumed that the ion is non-relativistic (since there is no relevant situation in which a neutrons would elastically scatter to create relativistic ions). This information is sufficient to determine the time and location at which the ion will reach the detector:

$$t_{arrival} = \frac{d\sqrt{1 + tan^2(\theta)}}{\sqrt{\frac{2E_{ion}}{M_{ion}}}} \tag{1.6}$$

$$r_{arrival} = d\ tan(\theta) \tag{1.7}$$

Note that d is the shortest distance to the sensor from the foil. Because each of these relationships is one-to-one, it is reversible: knowing the time and position is sufficient to back out the original neutron energy. Combing Equation 1.5 with the definition of nonrelativistic kinetic energy and solving for the neutron energy:

$$E_n = \frac{(M_{ion} + m_{neutron})^2(r_{arrival}^2 + d^2)}{8m_{neutron}t_{arrival}} csc(\frac{\pi - 2\theta}{2})^2 \tag{1.8}$$

Finally, given the detector efficiency as a function of energy — since the cross section contributes to the efficiency, and varies with energy, the efficiency is a func-

tion of energy as well — the measured spectrum is determined from the measured spectrum:

$$F(E_n)_{measured} = \frac{F(E_n)_{detected}}{\eta(E_n)} \tag{1.9}$$

Where $F(E_n)_{measured}$ is the actual number of neutrons at energy E_n correlating to the measured number of neutrons $F(E_n)_{detected}$ given the detector efficiency η at that energy.

The design is in keeping with the desired improvements of existing recoil spectrometers. It has a much larger acceptance angle (no collimation of the scattered ions is necessary), which is expected to facilitate higher efficiencies. No large-volume, high-precision magnetic field is required, which simplifies constraints. Finally, and most importantly, because the ions are much heavier and therefore slower than the neutrons (or the protons scattered in a traditional scheme), they arrive at the detector much later in time (potentially hundreds of nanoseconds later) than the rest of the "clutter" from an ICF implosion. This improves the signal-to-noise on the detector, potentially allowing for a clearer spectrum.

As with any system, there are sources of uncertainty, and tradeoffs to be made between accuracy and efficiency. The following chapter will focus on the variables inherent to each component and their impact on the overall performance of the detector. Each component of the detector will be considered in the next chapter with reference to its significance to the complete design.

Chapter 2

Design Considerations

2.1 Thin Foil Target

2.1.1 Mass/Energy Relationship

As the atomic mass of the isotope that comprises the thin foil increases, the maximal possible energy transfer to the ion goes as:

$$E_{max} = \frac{4M_{ion}m_{neutron}}{(m_{neutron} + M_{ion})^2} E_{neutron} \tag{2.1}$$

Assuming 14-MeV neutrons and maximum energy transfer (that is, a head-on collision), this results in a range of ion energies as shown in Figure 2.1 as a function of the ion mass.

Assuming a 5 cm separation between the foil and the detector, for an ion scattered directly forward, the corresponding time of arrival verses mass is shown in Figure 2.2.

The time of arrival, assuming maximal energy transfer from a neutron of energy

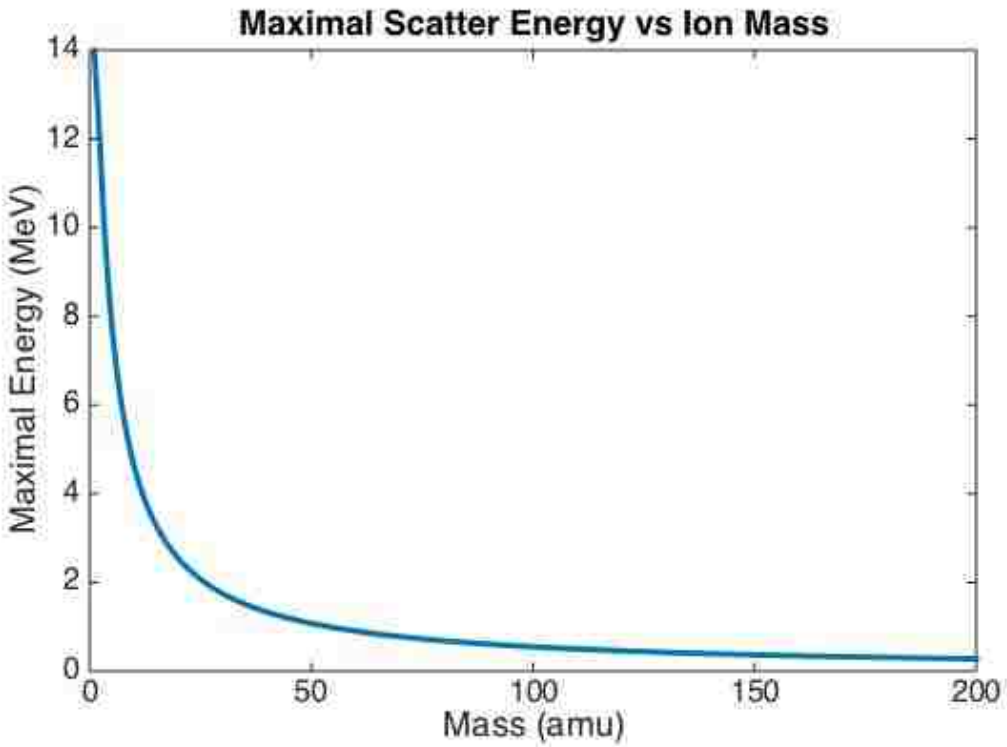

Figure 2.1: The relationship between ion mass and the maximum energy when scattered by a 14-MeV neutron.

E_N, over a distance d, is described by Equation 1.6. A later arrival time for the ion is advantageous: it puts more time between the arrival of the ion and that of the primary neutrons and burst of bremsstrahlung that accompany ICF experiments.

It is also useful to consider the time differential resulting from a small difference in ion energy. Figure 2.3 shows difference in arrival time for ions with a 5 keV difference in energy, as a function of ion mass.

This relationship, in combination with the other parameters, informs the selection of a foil material appropriate to the context.

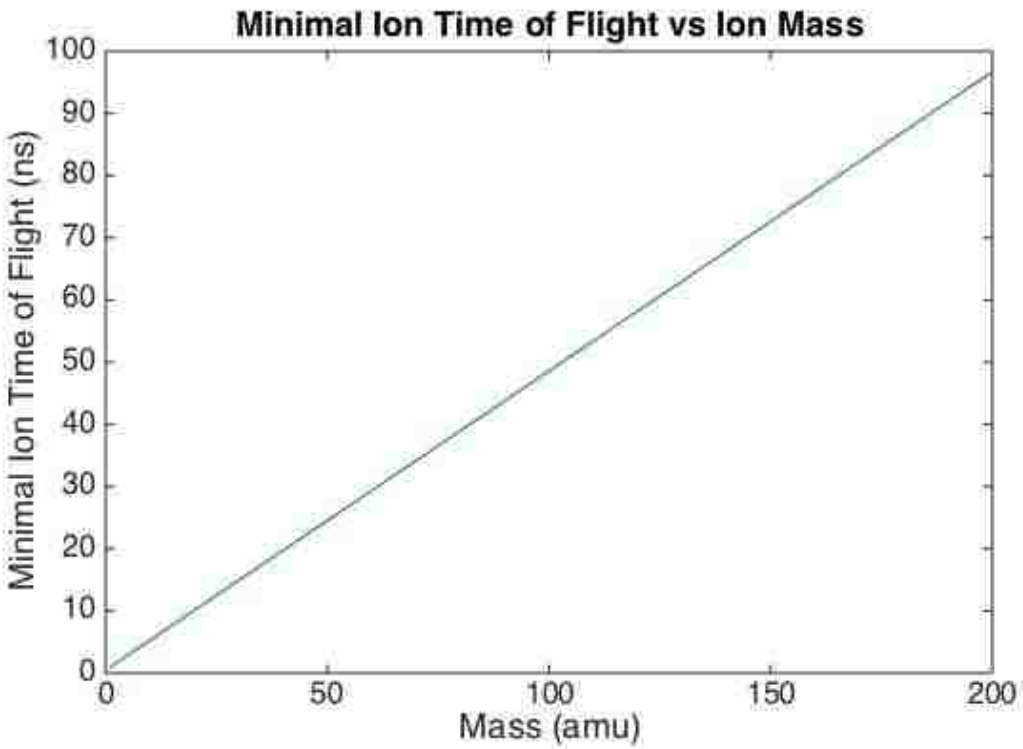

Figure 2.2: The relationship between ion mass and minimum ion time of flight over a distance of 5 cm.

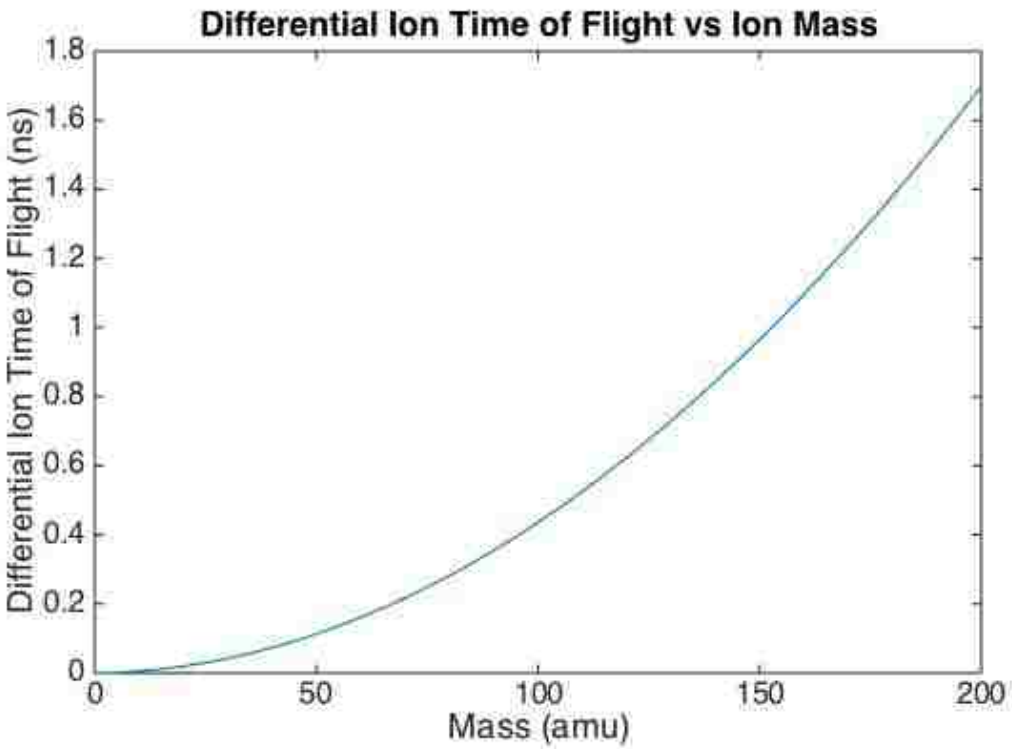

Figure 2.3: The difference in time of flight over a distance of 5 cm for two ions whose energies differ by 5 keV, as a function of ion mass.

2.1.2 Cross Sections

This encompasses two variables of interest: the total cross section in the relevant energy range, and the distribution of scattering angle. Neutron elastic scattering cross sections range from 0.25 to 3 barns at 14.1 MeV, substantially impacting overall efficiency. For 2.45-MeV neutrons, the elastic scattering cross section tends to be somewhat higher, with much more structure. Structure in the cross-sections is not inherently a problem, but any error in resolving that structure will translate to artificial structure in the measured spectrum, since the spectrum must be corrected for the changing cross section over energy. Figure 2.4 below shows the microscopic scattering cross sections for all the monoisotopic elements, from 1 MeV to 16 MeV.

While these differences are nontrivial, in practice they are subsumed by the difficulty of working with some materials. For example, depositing a submicron-thick layer of sodium on a surface, with no coating, is impractical from a chemical standpoint, regardless of cross sections. Constructing a pure fluorine "foil" would be impossible.

Foils with multiple isotopes could potentially function, but the difference in cross-section and distributions of scattering angles (the latter particularly significant at low mass) would reduce performance. It is also possible to get isotopically pure samples of non-monoisotopic elements, in which case a large number of isotopes could be considered, far beyond the scope of this work. Fortunately, the monoisotopic elements span a wide range of masses, and have reasonable elastic scattering cross-sections, so a selection of monoisotopic elements is sufficient for demonstrating this detector concept. For the remainder of the chapter, six elemental materials will be considered as plausible examples, to give a sense of the range of performance that is achievable. These materials are carbon, aluminum, cobalt, arsenic, indium, and gold. Figure 2.4 is repeated below with just these isotopes.

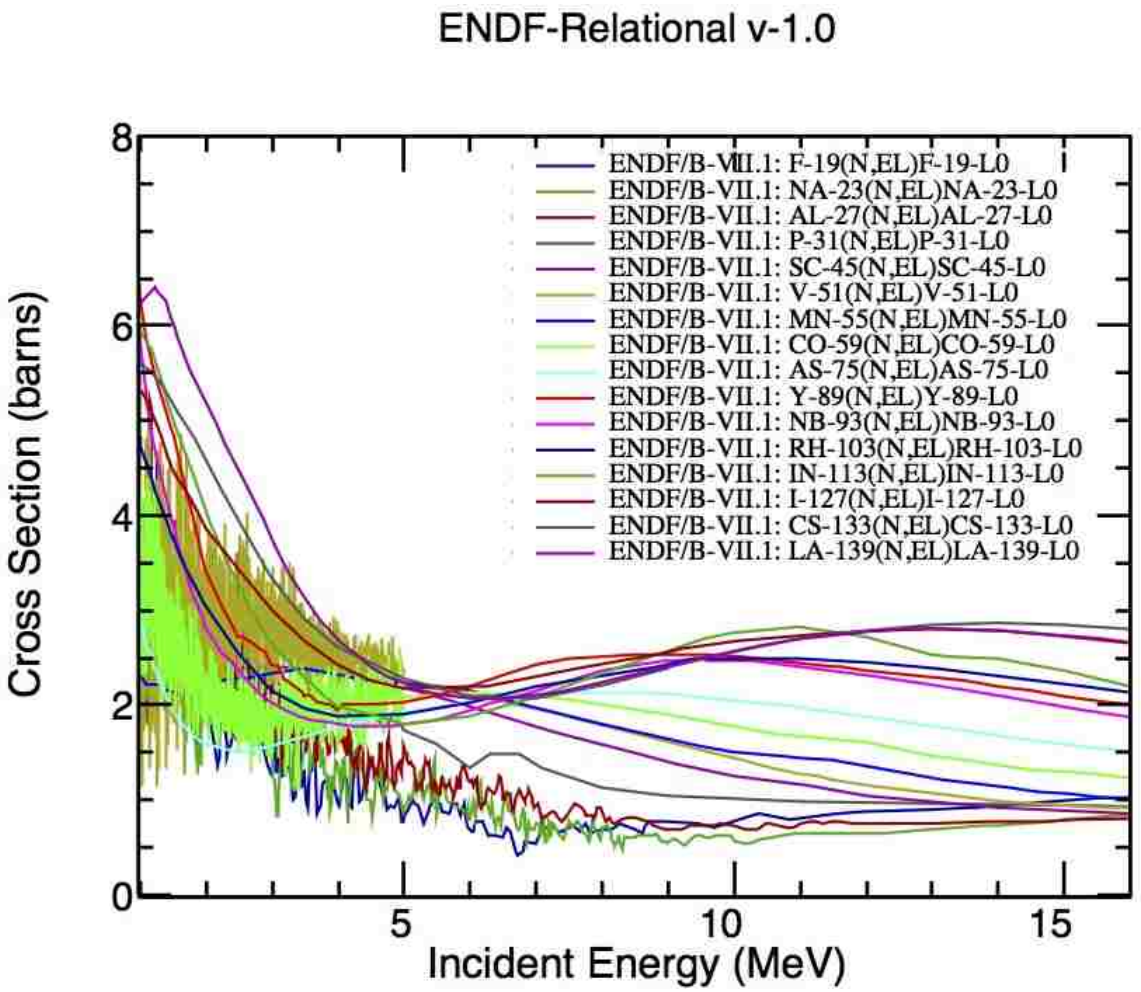

Figure 2.4: The microscopic elastic scattering cross sections of all the monoisotopic elements.

The feasibility of these materials, and other materials more generally, will be discussed in Chapter 3. Here they will be discussed purely from a performance standpoint on the assumption that it is feasible to work with them. Obviously, the microscopic cross section is much less important than the macroscopic, which is shown for the room temperature elemental form of each material in Figure 2.6.

Obviously, the greater the cross section, the better efficiency the detector will have, other factors being equal. Of course, other factors are rarely equal.

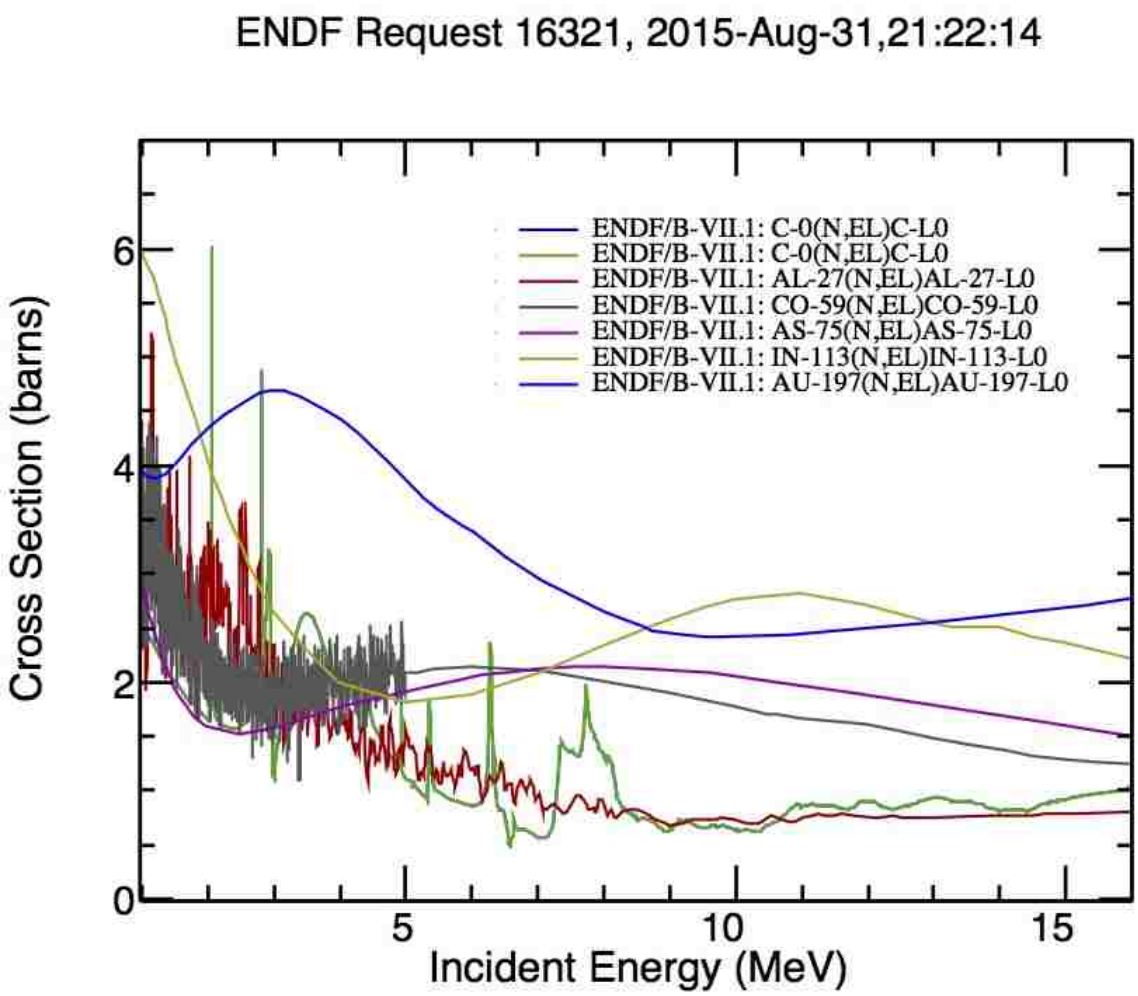

Figure 2.5: The microscopic elastic scattering cross sections of selected isotopes.

Another factor to consider is the angular dispersion of the scattered ions. Figure 2.7 shows the differential cross section verses angle for the representative isotopes at 14 MeV.

From a theoretical standpoint, the detector could have an arbitrarily wide angle of acceptance (i.e. the largest scattering angle which is "counted" at the detector); however, sensors cannot be arbitrarily large, and it is inefficient to look at large angles that have a low probability — they are effectively wasted space. Consequently, it is

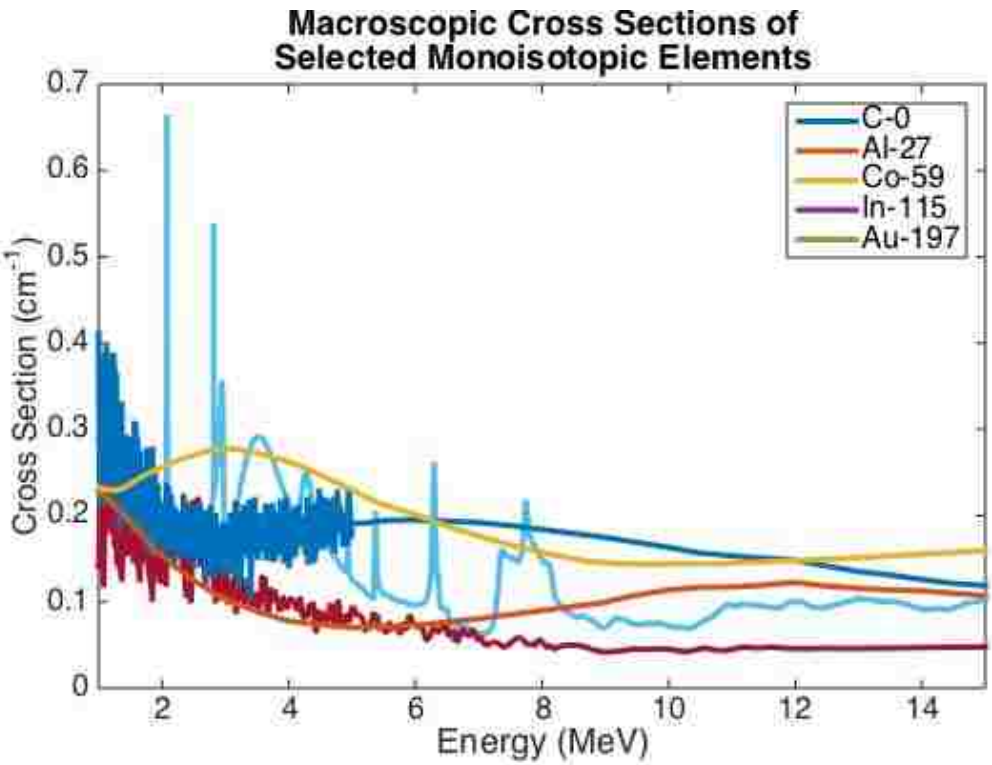

Figure 2.6: The macroscopic elastic scattering cross sections of selected isotopes.

preferable to choose materials with narrow scattering profiles, which is to say heavy materials, so that more data can be collected per unit area of sensor.

To give a more concrete picture, Figure 2.8 shows the fraction of scattered ions which would be scattered within 15 and 30 degree angles of acceptance for the representative materials and 14-MeV neutrons.

In most circumstances, a detector would actually consist of the setup described in Figure 1.2 repeated as cells in a grid over a larger area (see Figure 4.2 for an example configuration), to reach useable efficiencies. In this circumstance, ions scattered outside the angle of acceptance potentially contribute noise to adjacent setups. That is to say, the complementary values to those in Figure 2.7, above, are all contributing to "background" for adjacent "cells". This can be alleviated by "shielding" adjacent

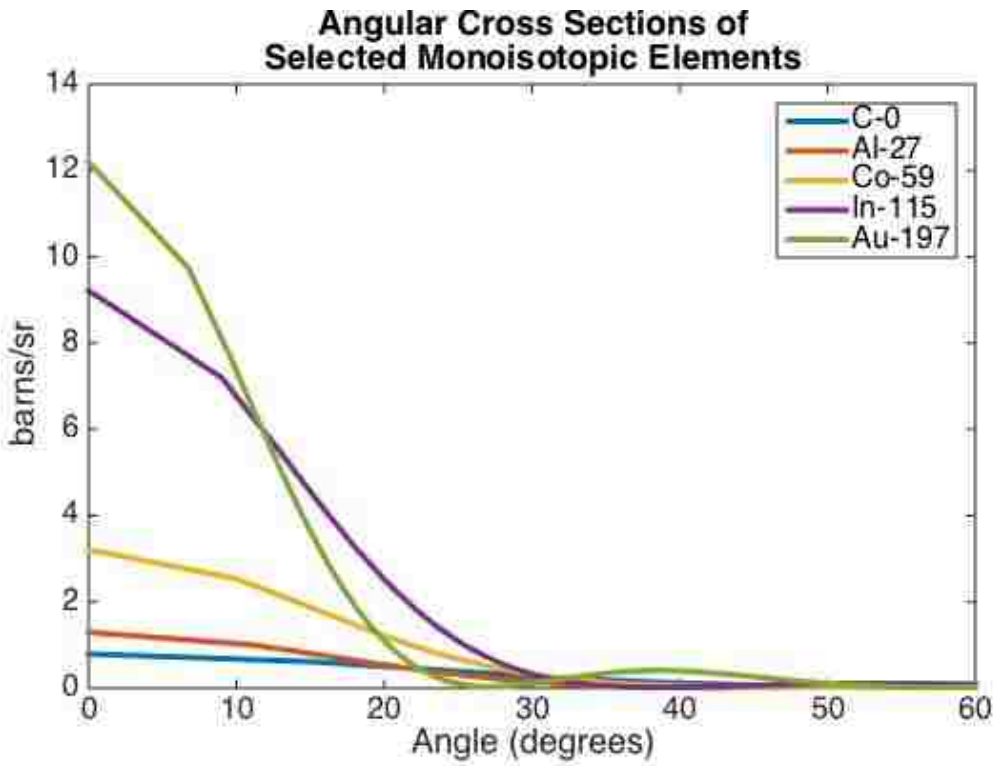

Figure 2.7: The differential elastic scattering cross sections of selected isotopes.

cells from one another.

Combining the effects from the previous two sections, a heavy foil material like gold or indium seems like an obvious choice. Unfortunately, their benefits are counterweighted by the problematic phenomenon of straggling, discussed in the next section.

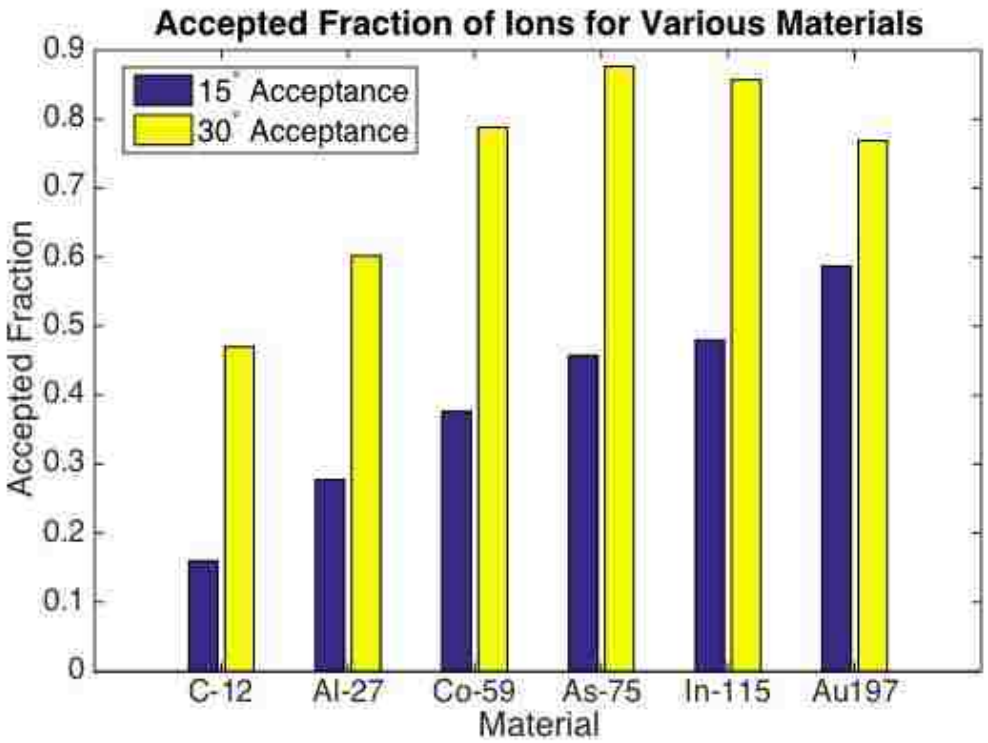

Figure 2.8: The accepted fraction for select materials with 15 and 30 degree angles of acceptance.

2.1.3 Foil Thickness and Straggling

As the stream of scattered ions pass through the foil, several things happen. Most obvious is the steady loss of energy through collisions with electrons in the foil — as the foil thickness increases, the average energy of escaping ions rapidly declines. More subtly, there is a small probability of the ion colliding with a nucleus in the foil, resulting in a larger change in energy and an accompanying change in direction. The thicker the foil, the more of these collisions occur, resulting in a process called straggling. As a result, as the thickness of the foil increases, the width of the energy distribution about the mean will increase, and more ions will be scattered off their initial course. This phenomena has large implications for the design of the heavy ion

recoil neutron spectrometer.

First, consider the steady energy loss. Since lower energy — and consequently slower — ions are to the detector's benefit, the intuitive conclusion is to find the thickest foil from which nearly all ions could escape. Even ignoring potential straggling effects, this approach is foiled by the neutrons, which are equally likely to scatter at any depth in the foil. As a result, for a thick foil, some neutrons will scatter at the front of the foil, producing ions that lose nearly all their energy before escaping, while other neutrons will scatter at the back of the foil, producing ions who escape with all their energy intact.

On the other hand, the thicker the foil, the more neutrons interact, and the higher the efficiency of the detector. Consequently, the appropriate foil thickness depends on the relative importance of efficiency and energy resolution for a given situation.

Further complicating matters is the relationship between ion/material charge and energy loss. The semiclassical formula for stopping power [11] gives a sense of how energy loss will change with higher-z materials:

$$-\frac{dE}{dx} = \frac{4\pi k_0^2 z^2 e^4 n}{mV^2} ln(\frac{mV^2}{hf}) \tag{2.2}$$

Where $-\frac{dE}{dx}$ is the stopping power, k_0 is coulomb's constant, z is the charge of the ion, e is the electron charge, n is the electron density in the material (in this case, equal to z), mV^2 is the ion's kinetic energy, and f is the orbital frequency. At a given ion energy, it suggests the stopping power increases with the cube of the atomic number of the foil material. That is to say that, number densities being equal, lower Z materials can be made thicker — leading to better efficiency — than high Z materials and retain comparable energy loss.

This leaves the phenomena of straggling. Energy straggling, fortunately, is irrelevant: the energy difference between ions created at the front and back of the

foil exceed the straggling losses in the foil, so the straggling losses add no measurable effect. Angular straggling is another matter. An ion scattering even a few degrees from its original direction will introduce a substantial error, resulting in an incorrectly calculated neutron energy.

It is useful to consider some specific examples to give a better sense of the impact. Suppose a beam of 14-MeV neutrons is striking a foil with atomic mass M. If we consider ions which are scattered straight forward, they will have an initial angle of zero degrees and an energy dictated by Equation 1.5. Without straggling, on a detector with infinite spatial and time resolution, every ion would be detected as coming from a 14-MeV neutron.

First, consider the impact of energy loss. Since the neutron is equally likely to scatter anywhere in the foil, the escaping ions will have a uniform energy distribution between their initial scattered energy $E_0 = \frac{4m_{neutron}M_{ion}}{(m_{neutron}+M_{ion})^2}14MeV$ and $E = E_0 - E_{loss}$, where E_{loss} is the average energy loss of an ion passing through the full thickness of the foil. This range can be characterized as a percentage of the initial energy. Combining this with Equation 1.8 gives Figure 2.9, which shows the measured spectrum of the uniform 14 MeV beam at foil thicknesses chosen to give a minimum ion energy of 90%, 95%, 98%, and 99% of the initial energy.

Note that Figure 2.9 assumes all ions are scattered straight forward. When the actual angular distribution is considered, the impact is greater: if the neutron-ion collision scatters the ion to an angle of 30 degrees, the ion will have more material to traverse, leading to greater energy loss and angular straggling.

Continuing on to angular straggling, which will be defined as the polar angle between the ion's trajectory when it is initially scattered and its trajectory when it escapes the foil. Extending the scenario described above, where only ions scattered at 0 degrees were considered, Figure 2.10 shows the measured energy spectrum of a

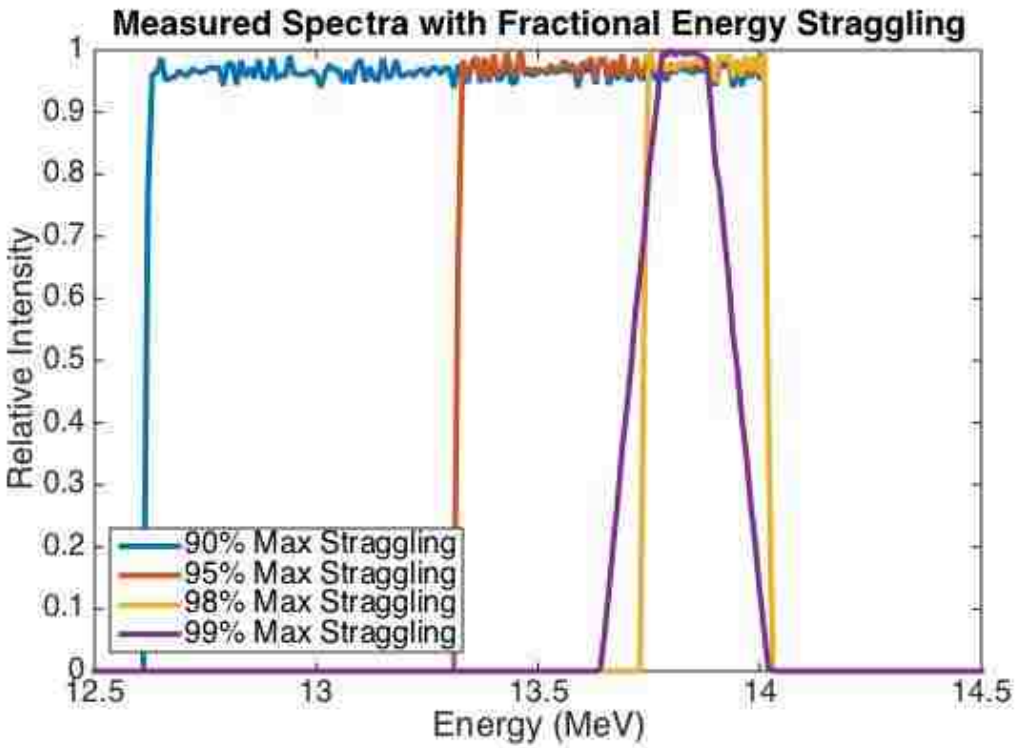

Figure 2.9: Measured spectra with various foil thicknesses, accounting for the distribution of energies of ions escaping the foil. For each case, the thickness was chosen so that an ion scattered directly forward at the front of the detector would, on average, escape the foil with the labeled percentage of its initial energy.

foil thickness chosen to give an average straggling of 1°, 2°, 5°, an 10°, still assuming the initial ions are all scattered directly forward (i.e. an initial angle of 0°).

Consider Equation 1.8 and Equation 1.6. The angle of scattering is coupled to the energy of the ion by the laws of mechanics, so increasing the angle implies decreasing the energy and consequently increasing $t_{arrival}$. Angular straggling is equivalent to changing θ in Equation 1.8 without changing $t_{arrival}$. In the situation considered above, the initial angle is zero degrees, so any change must be an increase in angle. Without the accompanying change in the time of arrival, the calculated energy will be higher than the actual energy.

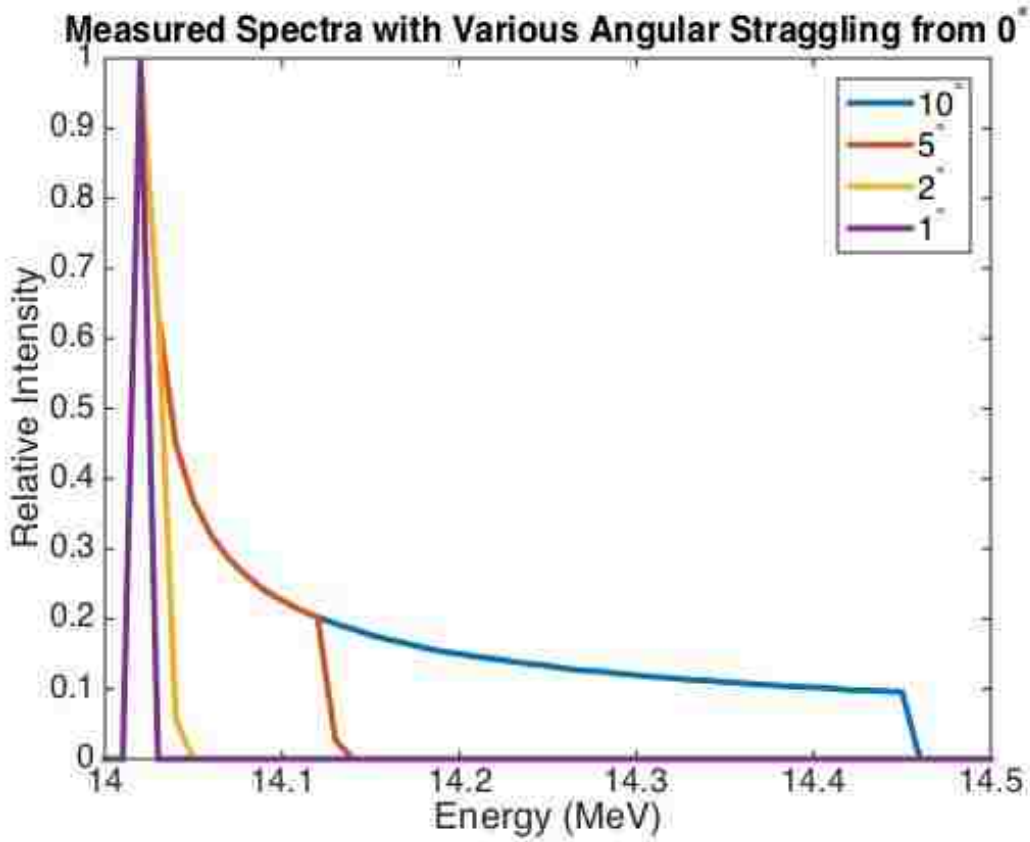

Figure 2.10: Measured spectra with angular straggling with all ions initially scattered into a 0° angle.

Angular straggling has a much greater impact on measured energy if (for example) it is a change from 20 degrees to 30 degrees than if it is a change from 0 degrees to 10 degrees. Figure 2.11 shows the spectra resulting from the straggling described above, but with an initial scattering angle of 20°. For consistency with the previous scenario, the change in angle is always an increase, rather than being distributed between increase and decrease as would physically happen.

The simplest way to put all of this together is with a simulation. Specifically, Geant4 was used to simulate scattering of 14-MeV neutrons in cobalt 59. The foil thickness of 6.106 nm was selected for an average outcoming energy of 98% of the

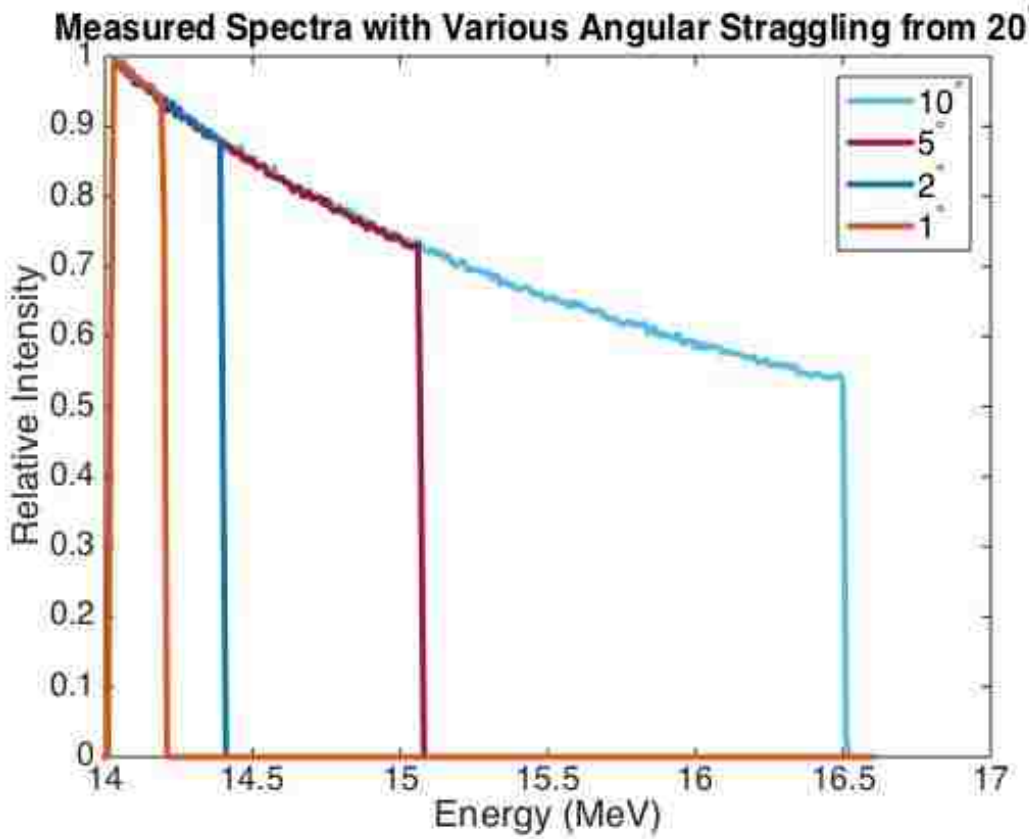

Figure 2.11: Measured spectra with angular straggling, with all ions initially scattered into a 20° angle.

initial scattered energy. Perfect spatial and time sensitivity was assumed, with an acceptance angle of 30 degrees. For a more detailed discussion of the methodology, see Chapter 4 as well as Appendix A. The detected spectrum, for $1.0 \text{x} 10^7$ scatters, is shown in Figure 2.12.

The full-width-at-half-max (FWHM) measured is 0.476 MeV, with a mean of 13.953 MeV. This suggests that the 98% energy point is fairly reasonable thickness to use. The down-shift in the average energy is due to the energy losses in the foil — on average, the ions are created in the center of the foil and experiences some energy loss before escaping the foil, translating to a low measured average energy.

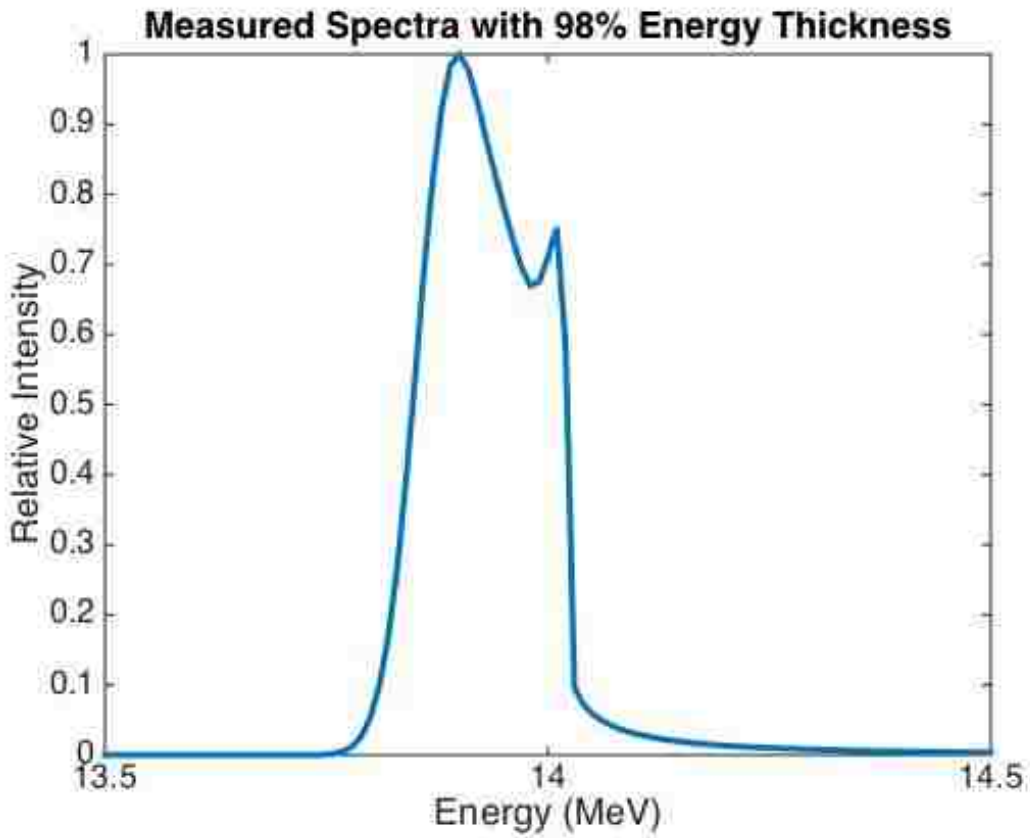

Figure 2.12: Measured spectra with 6.106 nm Co foil.

With these results in mind, Table 2.1.3 shows the 95%, 98% and 99% energy foil thicknesses for the isotopes under study, along with relevant peripheral information about the materials.

Table 2.1: Shows the thickness for which the average outcoming energy is 99%, 98%, and 95% of the initial energy for several materials. All thicknesses are in nanometers. Source: SRIM [12]

Material	Atom Density (cm^{-3})	σ at 14 MeV (barns)	Thickness for 95% Energy	Thickness for 98% Energy	Thickness for 99% Energy
C	1.13E+23	0.82	117.72	45.65	21.62
Al	6.02E+22	0.78	58.27	23.31	11.65
Co	9.08E+22	1.39	15.30	6.10	3.05
As	4.60E+22	1.70	14.79	5.92	2.96
In	3.83E+22	2.94	5.79	2.31	1.16
Au	5.90E+22	2.63	3.21	1.29	0.65

2.1.4 Foil Area and Distance

The previous section discussed the implications of foil thickness to detector performance. This section considers two further spatial design parameters: the area of the foil, and the distance from the foil to the detector.

The implicit assumption in Equation 1.8 is that the ions scattered by the neutrons are a point source. All of the calculations of distance, angle, and ultimately energy, assume that the ion was scattered at the exact center of the foil. In reality, of course, the ions are scattered randomly in the finite area of the foil. In fact, since the efficiency of the detector scales linearly with the area of the foil, it is preferable that it be a relatively large area. As with every other aspect of the detector's design, foil area is a tradeoff between efficiency and energy resolution.

This tradeoff is mitigated by distance. The loss in energy resolution correlates not to the absolute area of the foil, but to the solid angle subtended by the foil at the closest point on the sensor. That is to say, a 1 mm^2 foil 2 cm from the sensor is equivalent to a 2.5 mm^2 foil 5 cm from the sensor, for purposes of uncertainty.

Increasing the distance also increases the effective resolution. For a given absolute sensor resolution, the further away the sensor is from the foil, the smaller the solid angle of each detectable point, which means that each point/time combination aligns with a narrower range of neutron energies, improving the measured energy resolution. Finally, increasing the distance increases the time interval between when the neutrons and gammas strike the sensor and when scattered ions strike the sensor.

As with all choices, moving the detector further away has substantial downsides. Firstly, for a given angle of acceptance, the sensor area goes up with the square of the distance. For most sensor technologies, large sensors are expensive and often impractical. Secondly, greater spacing reduces the number of detector "units" that can work in parallel, reducing overall efficiency. (See Chapter 3 for further discussion of this problem.)

For a concrete picture of the implications, consider a square foil 20 mm by 20 mm (unreasonably large, but useful for demonstration) placed 10 cm from a sensor and struck by 14-MeV neutrons. Once again, suppose cobalt is the foil material. Two neutrons scatter off the foil, producing ions: one scatters in the exact center of the foil, the other scatters at its outer edge, but both ions are scattered toward the exact center of the sensor. The situation is shown in Figure 2.13.

Both scattered neutrons had the same energy, 14 MeV, but scattered ions at different angles. Equation 1.5 yields respective ion energies of 0.917778 and 0.917686 MeV. The ion from the center of the foil will arrive at the detector 57.718 ns after scattering, while the ion from the edge will arrive 59.295 ns after scattering, 1.577 nanoseconds later. This is effectively changing $t_{arrival}$ in Equation 1.8, resulting in a change in the measured energy. Specifically, the ion that came from the center of the foil will be correctly correlated to a 14-MeV neutron, while the second ion, which came from the foil edge, will be calculated to have come from a 13.724 MeV neutron. This error is exacerbated further from the center of the sensor: if both ions

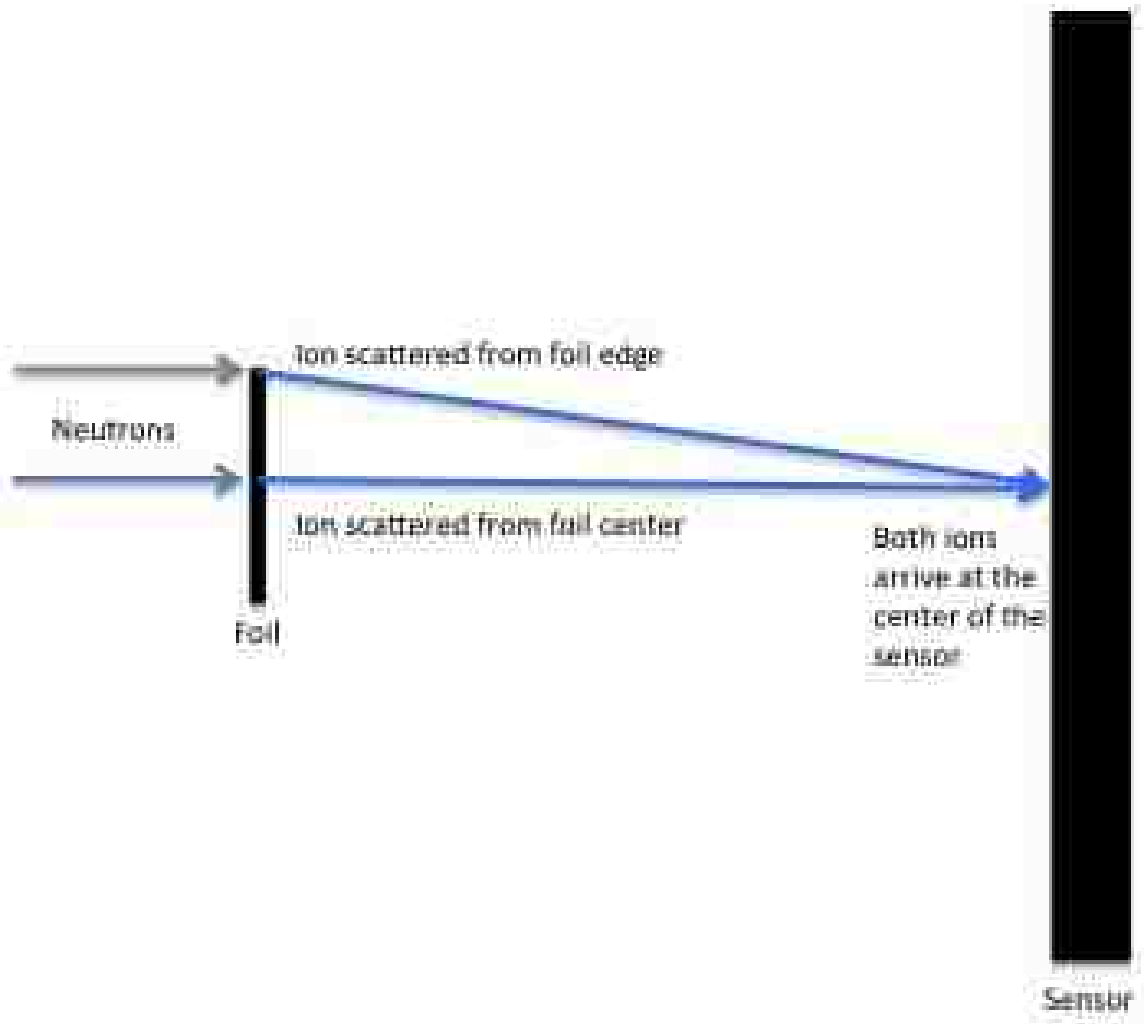

Figure 2.13: Diagram showing scattering from different points on the foil to the same point on the sensor. Not to scale.

were scattered to a point 5.77 centimeters from the center of the sensor (a 30 degree scatter coming from the center of the foil), the measured energies would have been 14 and 16.508 MeV.

Figure 2.14 gives a broader picture. It shows several spectra, from the configuration discussed above but with several different foil sizes.

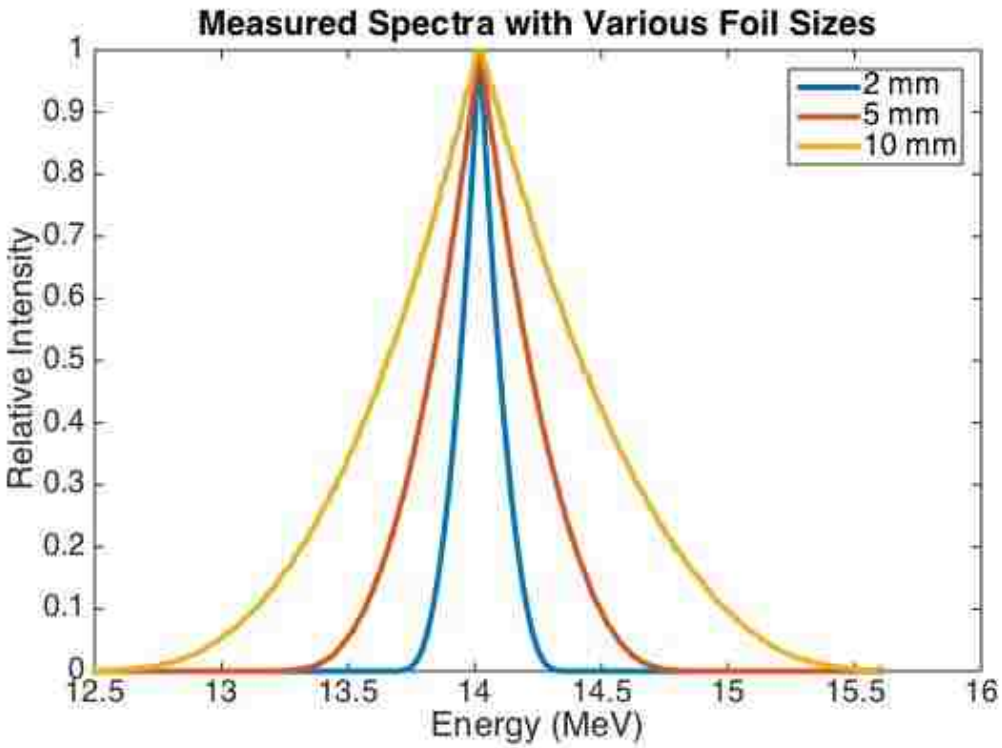

Figure 2.14: Measured spectra for various foil sizes, ignoring straggling losses.

Clearly, the detector resolution is less sensitive to this parameter than to most others discussed in this chapter, but in high resolution contexts it becomes quite significant. The measured FWHM ranges from 0.205 MeV with the 2 mm foil to 1.025 MeV for the 10 mm foil.

2.1.5 Noise

As discussed, the purpose of selecting a heavy material for the foil is so that the scattered ions arrive at the sensor much later than the primary neutrons and bremsstrahlung, to insure that they do not "white out" the useful signal. Unfortunately, the neutrons and bremsstrahlung produce secondary radiation lasting microseconds after

the primary burst.

There does not appear to be a detailed experimental time history examination of the radiation environment at a suitable ICF facility — most sources cut off after a few tens of nanoseconds or less. In any case, it would be strongly geometry-dependent, so comparison between such a source and the anticipated environment for this detector would be questionable. This makes a detailed assessment of the impact on detector performance difficult, but some useful generalizations are helpful.

First, consider the time frame of interest for the detector. Table 2.2 shows the peak time for a forward scattered ion to arrive at a sensor for the sample foil materials at several foil to sensor distances, for both DD and DT sources, with a 5 cm foil to sensor separation.

Table 2.2: Shows the most probable time of arrival (measured from when the scatter occurs) for forward scattered ions (both DD and DT) to a sensor 5 cm away for several materials.

Material	Peak Energy (DD, MeV)	Arrival Time (ns)	Peak Energy (DT, MeV)	Arrival Time (ns)
C	0.70	14.95	4.00	6.23
Al	0.34	32.19	1.94	13.42
Co	0.16	68.99	0.92	28.76
As	0.13	87.38	0.73	36.42
In	0.09	131.07	0.49	54.64
Au	0.05	227.65	0.28	94.90

On these short time scales, noise resulting from activation products can be largely discounted. Few single-neutron activations have half-lives short enough to occur in the time periods of interest. On the other hand, the (n,γ) and (n,Xn') reactions are too fast: the gammas or neutrons produced at the foil will arrive well before the elastically scattered ions. This leaves two probable sources of noise: heavy ions produced by neutron scatter in non-foil materials, and secondary gammas an x-rays

being scattered back into the detector.

Since heavy ions have very little penetrating power, only materials in the detector itself are a concern — any ions from outside the detector will not be able to reach the sensor. Of primary concern are the materials backing the foil: as thin as the foil has to be, it is impractical for it to be freely suspended; rather, it will have to be coated or attached to a larger quantity of some backing material. A more detailed discussion of the feasibility of this can be found in section 3.2; for now the primary question is whether such a backing material will produce too much noise for the detector to function; after all, neutrons are as likely to scatter in this backing material as in the foil.

Selecting a backing material with a very different atomic mass from the foil material can largely mitigate this problem. For example, if the foil is indium, and the backing material is polypropylene, the average ion (scattered by a 14-MeV neutron) will arrive at a sensor five centimeters away at 58.26 nanoseconds, while the average hydrogen from the polypropylene arrives at 1.02 nanoseconds and the average carbon arrives at 6.64 nanoseconds. There is plenty of time between the backing ions and the foil ions to distinguish between them.

Of course, there are some few backing ions that will be straggled to exactly the right energy to hit the sensor at the same time as the foil ions. Calculating their contribution is difficult, and unnecessary, as their is an easier solution. The problem can also be completely eliminated if a comparatively light material is chosen for the foil. In this case, a much heavier material can be used for the backing (for example, aluminum foil with lead backing). In this case, the foil ions arrive substantially before even the highest-energy backing ions, so that the backing ions can easily be gated out. For example, if cobalt is chosen for the foil material, and lead for the backing material, then the average ion arrives at 28.86 nanoseconds, while the average backing ion arrives at 200.09 nanoseconds. More importantly, for any reasonable angle of

acceptance, it is physically impossible for a neutron to impart sufficient energy to the lead ion to arrive before or during the cobalt arrival duration, so this source of noise is eliminated.

The second source of noise to consider is gammas scattered back into the detector. Here, shielding is highly beneficial. Substantial shielding would be placed around the periphery of the detector, such that very few gammas scattered out of the detector would be able to get back in. In any case, the time intervals are large for light. Table 2.3 is a duplicate of Table 2.2, but with the times replaced by distances, in meters, that light would travel in those times.

Table 2.3: Shows the distance traveled by light in the same time intervals shown in Table 2.2.

Material	Peak Energy (DD, MeV)	Light Distance (m)	Peak Energy (DT, MeV)	Light Distance (m)
C	0.70	4.48	4.00	1.87
Al	0.34	9.65	1.94	4.02
Co	0.16	20.68	0.92	8.62
As	0.13	26.20	0.73	10.92
In	0.09	39.29	0.49	16.38
Au	0.05	68.25	0.28	28.45

For heavier foil materials, or for configurations with large separation between the foil and the sensor, it seems safe to presume most of the gamma signal will be gone. For light foil materials, or short, distances, it could be a substantial concern, though peripheral shielding could eliminate much of the problem.

This conclusion is largely supported by simulations done in support of an x-ray streak camera designed for placement near Laser Megajoule facility in France [3]. This facility is designed for ICF experiments similar to those undergone at the National Ignition Facility, with implosions producing in excess of $1x10^{15}$ neutrons. In

assessing the dose to a CCD sensor roughly 3.4 meters from the implosion, the paper suggests substantial gamma dose occurs for tens of nanoseconds after the primary pulse, due to the diffusion of the bremsstrahlung and secondary gammas. This is consistent with the above estimations. Given that this detector would generally be placed closer than 3 meters, it is probably a generous estimate.

In summary, while the noise from the high radiation environment is certainly not negligible, and would merit substantial study before a detector was constructed, there is no reason to believe it fatal to this detector design.

2.2 Sensor

2.2.1 Timing

Sensor timing limitations are probably the greatest bottleneck in the performance of this type of spectrometer. Even if extremely heavy materials are selected for the foil, the time between the slowest and fastest ions arrival at the detector is still only tens of nanoseconds. This means that time bins have to be extremely narrow — preferably sub-nanosecond — and having more time bins dramatically improves resolution of the detector. On the other hand, the useful time resolution is inherently limited by uncertainty: if the neutron source uncertainty is in the 10 nanosecond range, 100 picosecond timing resolution gives no useful benefit over 1 nanosecond resolution.

The chapter on feasibility will discuss what sort of timing is plausible with current technology. For the present, it suffices to analytically assess how timing characteristics of the detector design impact overall performance.

It is best to start by considering the timing variation for different materials. Table 2.4 shows the time 90% of ions will arrive after, the most probable time of arrival,

and the time 90% of ions will arrive before. The table assumes a 14-MeV neutron source with a foil 5 cm from the sensor.

Table 2.4: For several materials, shows the time after which 90% of ions will arrive, the average time of arrival, and the time before which 90% of ions will arrive.

Material	10% Arrive Before (ns)	50% Arrive Before (ns)	90% Arrive Before (ns)
C	6.47	8.22	inf
Al	13.71	15.89	41904.57
Co	29.19	31.92	516.53
As	36.86	39.41	51.91
In	55.24	58.82	145.61
Au	95.60	99.96	197.00

The larger time is somewhat deceptive; ions arriving at such late times are necessarily scattered at large angles, which means they are unlikely to be detected for any reasonable angle of acceptance.The 10% to 50%, however, gives a good picture of the span necessary. It is, of course, possible to increase the distance between the foil and the sensor, with all the implications discussed above, which acts to double the time spread. For a reasonable range of distances, however, the time will vary from hundreds of picoseconds to tens of nanoseconds.

Several time bins are necessary for reasonable resolution. Figure 2.3.1.1 shows 4 spectra, created with the same configurations but different number of time bins: 5 bins, 20 bins, 100 bins, and 200 bins, in each case spanning the 90% to 90% time interval. This simulation was done with 1 mm square cobalt foil 10 cm from the sensor. Results are summarized in Figure 2.15.

The results range from a FWHM of 0.107 MeV with 200 bins to 1.163 MeV for 5 bins, with clear signs of diminishing returns — 100 bins and 200 bins are almost indistinguishable, indicating that the lower limit for uncertainty with 1 mm foil is approaching. In general, however, the more bins the better. Uncertainty in time is

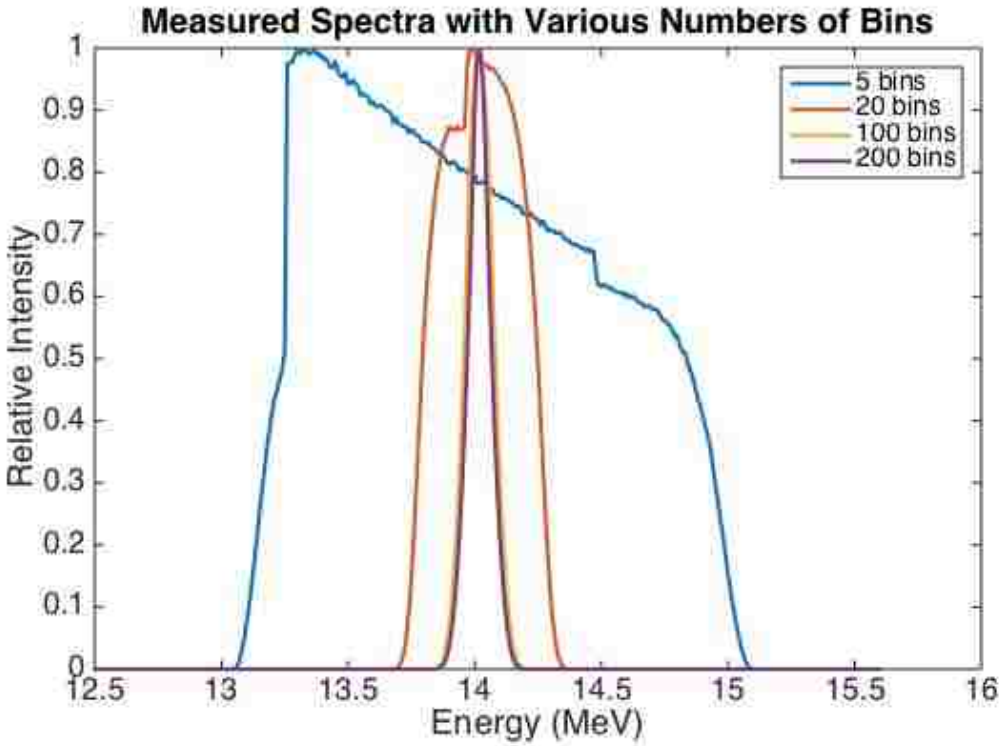

Figure 2.15: Measured spectra for different numbers of bins, neglecting straggling.

a significant complication. Uncertainty comes from any number of sources: duration of the neutron source, jitter in timing equipment, and so on. Figure 2.16 shows four spectra 20 (timing) bins wide. Each spectrum arises from a different (uniform) timing uncertainty: 0 ps, 100 ps, and 500 ps, and 1000 ps.

Timing uncertainty decreases the accuracy, but not to an unreasonable degree. Even with 2 nanoseconds of uncertainty, the resulting FWHM is only 0.683 MeV. To get a sense of how number of bins impacts matters, Figure 2.17 shows four more spectra, calculated using 100 bins, with the same uncertainties.

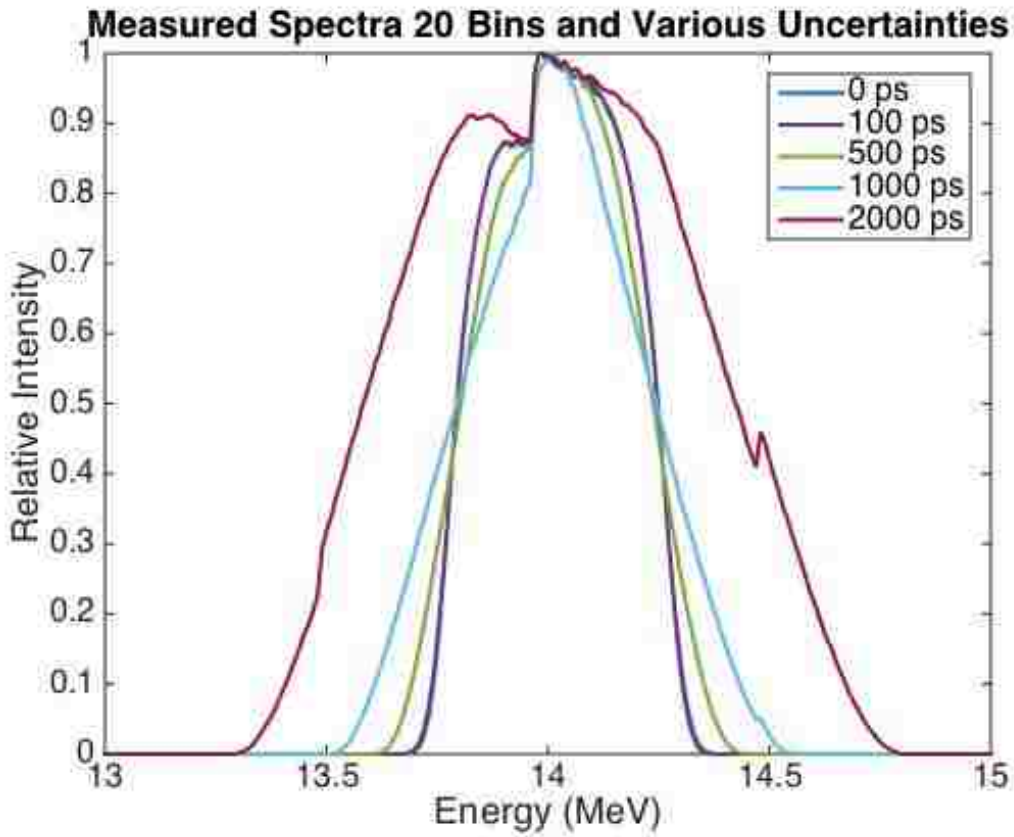

Figure 2.16: Measured spectra with 20 bins and various timing uncertainties, neglecting straggling.

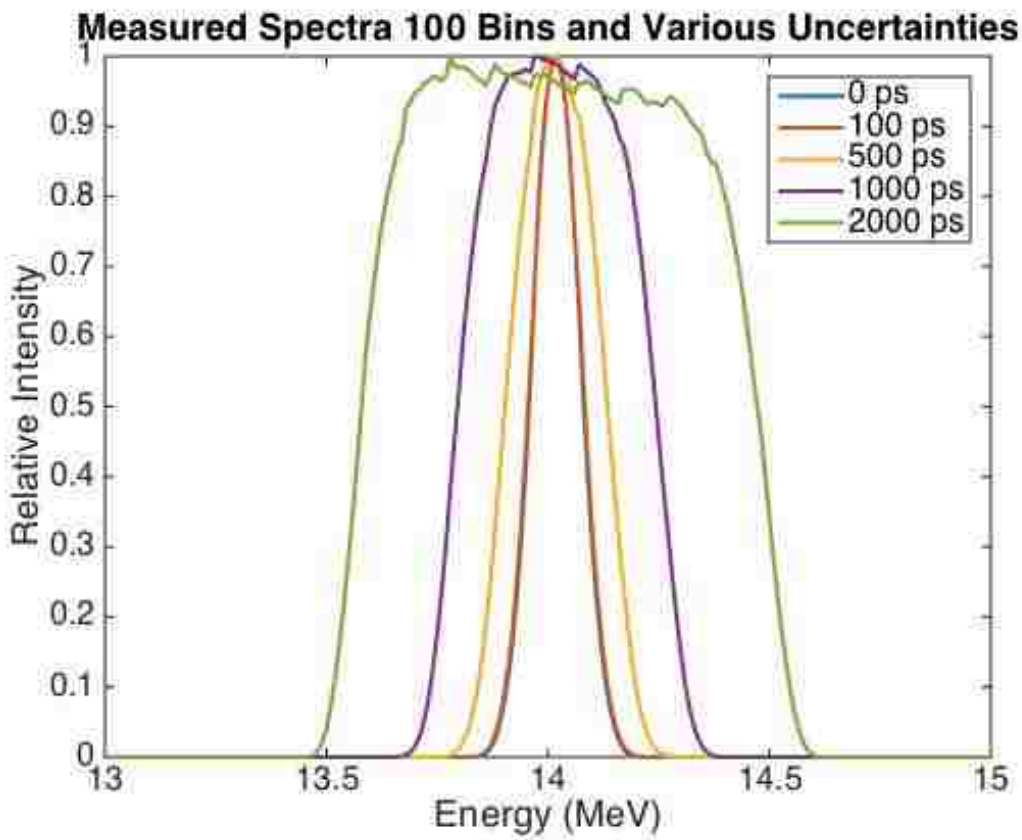

Figure 2.17: Measured spectra with 100 bins and various timing uncertainties, neglecting straggling.

2.2.2 Pixel Size and Coverage

The next parameter to consider is the spatial resolution and coverage of the sensor. If the sensor is a camera, the spatial resolution would be the pixel pitch, and the coverage is simply the fraction of the sensor surface which is sensitive. Alternatively, the sensor could be a diode array, in which case the spatial resolution is the spacing between diodes, and the coverage is the area of the sensor which is sensitive diode divided by the total area.

The significance of the coverage is intuitive: the efficiency of the detector is proportional to the coverage, since every ion scattered into a non-sensitive region is lost. Furthermore, dead space between sensitive regions also decreases the spatial resolution. Obviously, it is desirable to maximize coverage.

On the other hand, detector performance is insensitive to a wide range of spatial resolutions. This is because of solid angle. Any reasonable design will have a relatively large foil — typically at least 1 mm^2. With a camera, spatial resolution will almost always be in the um range, which means that spatial uncertainty due to spatial resolution is overshadowed by uncertainty from foil size. Even if an infinitely narrow foil were plausible, angular straggling contributes more spatial uncertainty than the resolution: 1 degree of scatter, at 5cm, equates to 873 um difference in where the ion impacts the sensor. In the case that diodes are used (see the feasibility section for more on the distinction), larger pixels are probable, but will still be similar to or smaller than the foil size.

To isolate spatial resolution, consider a hypothetical configuration with a point-foil, so that the foil contributes no spatial uncertainty, and an infinitely thin foil, so there is no straggling — in short the scattered ions are coming from a point source. The configuration will use cobalt foil, 10 cm from the sensor. Figure 2.18 shows the resulting spectra, with several different spatial resolutions.

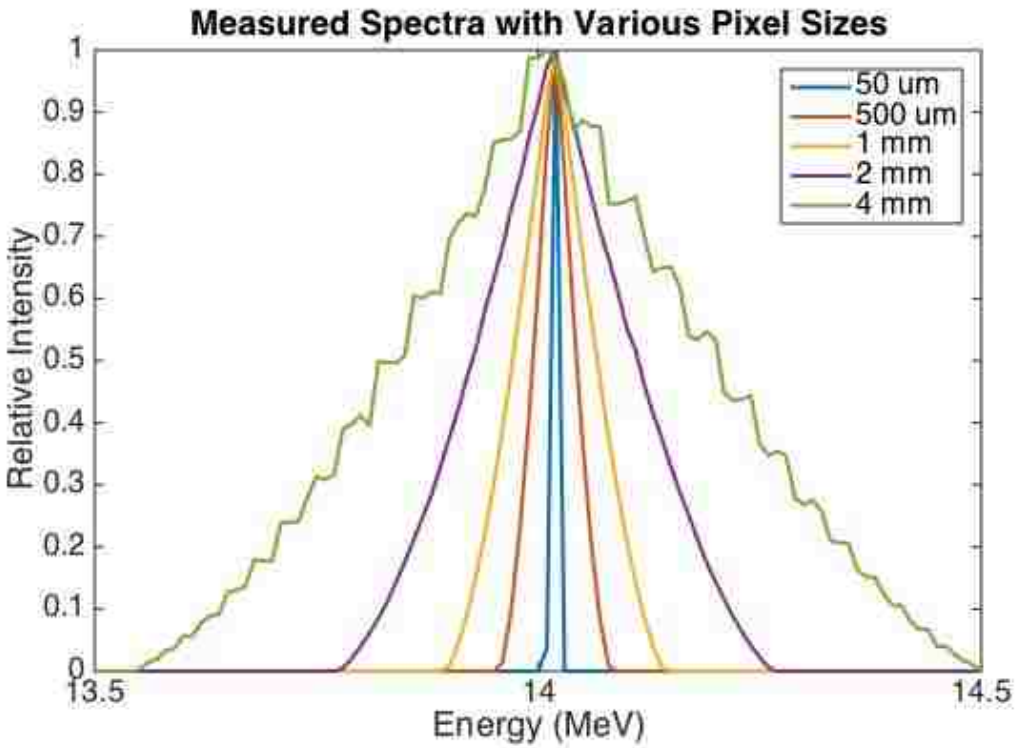

Figure 2.18: Measured spectra with various pixel sizes, neglecting other uncertainties.

With 1 mm pixels, the FWHM is only 0.103 MeV, quite small relative to other factors, and at 4 mm it is 0.406 MeV, still quite comparable. This indicates that pixel size can be selected as appropriate to the sensor technology without seriously impacting detector performance.

Chapter 3

Feasibility

3.1 Fielding Environment

Any discussion of feasibility must start by considering useful cases. This detector is only useful for very short pulses of neutrons — it would be useless at the Joint European Torus. It will also have a very low efficiency, so it needs to be fielded with a large neutron source. With these considerations, two prime options for fielding are the Z Machine, at Sandia National Laboratories, and the National Ignition Facility, at Livermore National Laboratories.

The Z Machine has a record neutron yield of $4\text{x}10^{13}$, while the National Ignition Facility has a record neutron yield of $9\text{x}10^{15}$, so a feasible detector must have a high enough efficiency to work somewhere in this range. In both cases, experiments can potentially fielded roughly 30 cm from the implosion. Being closer improves efficiency, so the detector needs to be able to survive placement near the implosion. Finally, neither facility produces delta sources of neutrons. The Z Machine produces neutron bursts averaging tens of nanoseconds long, while the National Ignition Facility implosions are generally less than one nanosecond. With this range of

fielding environments as a baseline, this chapter discusses the feasibility of operating a heavy-ion recoil neutron spectrometer.

3.2 Thin Foil Target

For a wide range of material/backing pairs, the requirements of this detector are well within normal parameters. For this type of small-area thin-film, the most probable technology to use would be ablation deposition. In this case, the surface to be coated (the substrate) and a block of the coating material are placed in a vacuum chamber, a short distance apart. The coating material is ablated, generally using a laser. The resulting plasma expands, depositing a thin layer on the nearby substrate.

Because the process consists of ablating material which is then deposited, thin coatings are generally discussed in terms of required mass per unit area, typically micrograms per square centimeter. For the isotopes discussed in the previous sections, assuming a foil thickness between 1 and 10 nm thick, the equivalent coating mass is in the range of 1 to 10 micrograms per square centimeter. This is well within the normal achievable range, and does not present a problem [9].

In terms of material selection and viability, there are three primary factors to consider: how amenable the coating material is to ablation, the amount of surface area to be coated, and the permeability of the substrate. Most of the materials under consideration are metals, and ablate well. Carbon is a nonmetal, but also works fine for these types of thin coatings [9]. The detector requires a small surface area to be covered, not more than one square centimeter for most plausible designs; this is significant because the resulting small solid angle increases the achievable uniformity of the coating. Finally, appropriate substrates can easily be chosen: lead is likely to work, copper is a common choice, and many others are feasible.

In short, the thin coatings requirements are a non-issue.

3.3 Sensor

3.3.1 Technology

The sensor has to be space and time resolved, and take multiple readings over the span of a few nanoseconds. There are relatively few technologies that can hope to meet this mix of criteria, but two are under development at Sandia National Laboratories: a sub-nanosecond multi-frame camera, UXI [6], and a fast diode array system based on the PSEC4 chip [8].

The camera has been developed for low energy x-ray sensitivity, but the technology for ion sensitivity is fundamentally the same. Furthermore, since this camera was developed with the expectation that it would be fielded in a neutron environment it is expected to perform relatively well in the intense conditions the detector will need to endure.

The fast diode array concept is not as well developed, but shows great promise. The premise is that a large number of fast diodes are placed in a grid on a chip, making the "sensor". These diodes are then connected to an array of PSEC4 chips, which act as "scopes on a chip", allowing for a large number of reads in a short time.

3.3.2 Time Resolution

The current iteration of UXI can collect four frames. Each frame has a one nanosecond integration time, and is separated from the next from by a minimum one nanosecond recovery time. To get a reasonable number of time bins, multiple foil/sensor pairs

would have to be fielded, but that is necessary for efficiency anyway. The minimum resolution (that is, the minimum integration time) is enough, barely, for some useful data; however, for good resolution it would have to be improved. The goal of the project is to reach sub-nanosecond timing. As the achievable integration time gets shorter, the camera will quickly become a more appealing option.

The PSEC4-based diode array has much better timing characteristics: 256 bins 70 ps wide. This spans enough time (18 ns) to cover the whole pulse for reasonable diode distance and foil material, and still gives time bins down at or below the width of the time uncertainty. From this perspective, the diode array is obviously the superior choice.

The problem comes from trying to synchronize a large number of diodes on a large number of PSEC4 chips. UXI, as a camera, is designed from the ground up so that all pixels have the same timing alignment; in spite of this it has an end-to-end timing uncertainty in the vicinity of 100 ps. Getting that kind of timing alignment out of a large number of PSEC4 chips, all at the same once, will be a difficult task.

3.3.3 Spatial Resolution and Coverage

From a spatial resolution perspective, the UXI camera is perfect, if not overkill. With 35 um pixels, it is well below the uncertainty level from the foil and the angular straggling. The full coverage is excellent, and means no ions are lost "in the cracks". A board could be designed with multiple UXI sensors edge-to-edge, for a larger effective sensor.

The diode array is trickier. Depending on the details of the design, the diodes can reasonably have a fill factor in the vicinity of 80%, which is not perfect, but is a fairly small contribution to efficiency relative to other factors. The main challenge comes from the sheer number of diodes required, combined with the small diodes necessary

for good performance. The time response of a diode is proportional to the diode's area. At 2 mm, which is a fairly large diode, the sensor will have 25 diodes per square centimeter. Since the smallest conceivable geometry will have several square centimeters of sensor, and a plausibly efficient design will repeat this arrangement dozens of times, the result is thousands of diodes, with a PSEC4 chip, or something comparable, for every few diodes. While there is no hard limit on how many diodes is feasible, at some point it becomes infeasible from a design standpoint. At that point, the practical solution is to reduce the fill factor, spacing out the diodes over the area rather than packing it. In this case, effective fill factor might be reduced dramatically, with a corresponding reduction in efficiency.

3.3.4 Radiation Hardness and Survivability

This is the most difficult area to assess, for the simple reason that very little testing has been done to assess the performance of either the UXI camera or most diodes in a high-gamma, high-neutron environment. Until there is empirical data, the best that can be done is consider these types of devices in general.

While there is limited data on fielding silicon detectors at the National Ignition Facility or the Z Machine, considerable simulation work has been done in preparation for fielding a high-speed CMOS streak camera for x-ray diagnostics on the Laser Megajoule facility in France [3]. This facility is similar to the National Ignition Facility, designed for ICF experiments producing in excess of $1x10^{15}$ neutrons. Rather than attempting to design a CMOS detector capable of withstanding these extreme conditions, this team focused on simulated shielding requirements. While their sensor was placed relatively far from the implosion (3.4 meters), their shielding requirements give a sense of what would be needed here. To shield the x-rays and gammas down to acceptable levels, they required 11.5 mm of lead in front of their detector, and

3.5 mm of lead surrounding the rest of the detector. Thicker layers would be needed for the present case; however, as the particle of interest is neutrons, not x-rays, a much thicker layer of lead can be used before the neutron signal is unacceptably attenuated.

Alternatively, the detector itself can be designed with radiation hardness in mind. The large hadron collider uses a large number of silicon detectors, which must survive even harsher environments than those of NIF. The design goals there are silicon detectors that can survive fluxes of 10^{16} $n_{1MeV equivalent}/cm^2$ over long periods of time. A foil at 30 cm from a record-setting implosion on NIF would receive little more than 10^{12} n/cm^2, suggesting that silicon diodes that can survive pulsed power experiments are definitely achievable. Li discusses several techniques which have allowed diodes to reach the desired hardness to neutron radiation [7]. While data on the gamma pulse from NIF is less clear, the same research teams for the LHC have developed oxygenated silicon didoes that are nearly immune to radiation damage from gamma rays [7].

The controlling electronics are also potentially sensitive to radiation damage, and are likely to need replaced periodically; however, most problems from radiation exposure are transient effects immediately upon radiation. Since the useful signal arrives substantially after the primary pulse, these transient effects are not expected to cause major problems on individual experiments.

Overall, while it is unclear whether hardware available right now would be effective in the extreme conditions necessary for the spectrometer, and it is likely that some components of the detector would need to be replaced frequently, the sensor technology to survive fielding on ICF experiments is within reach.

Chapter 4

Anticipated Performance

4.1 Introduction

This chapter puts together all of the details and considerations of the previous sections into two feasible designs and discusses their simulated performance. The presented designs are not proposed detectors to be constructed; rather, they are "sketches" of designs, meant to diffuse the many variables, possibilities, and limitations discussed into two designs, aimed at high efficiency and high resolution. This enables discussion of overall performance for feasible designs.

In choosing parameters for the designs to discuss, there are many areas, as discussed in the previous chapter, where the current limits on what is possible are not known — there simply is not enough data available. In general, the chosen constraints are based on what is reasonably considered achievable with moderate effort, rather than what could be fielded tomorrow. This is in keeping with the purposes of this paper, which is to illustrate the potential of a novel design.

As a reference for feasible size and placement, both designs are intended roughly

as "drop-in" replacements for the magnetic recoil spectrometer at the national ignition facility, which is described in detail in [4]. Based on this requirement, both designs will fit in roughly a 30 cm by 30 cm by 100 cm volume, roughly 30 cm from the implosion. In keeping with the configurations presented in [4], the two setups discussed are aimed, respectively, at maximizing resolution, while maintaining a feasible efficiency, and maximizing efficiency, while maintaining a useful resolution.

4.2 Methodology

The performance of the configurations was simulated using Geant4 and Matlab. Geant4 is a set of open source libraries for Monte Carlo simulation, implemented in C++ [1]. Matlab is a commercial numerical integrated development environment and computing language, used under an academic license.

Geant4 supports arbitrarily complex geometries, with source particles ranging from neutrons and protons to optical photons to exotic high-energy hadrons. While it originated at CERN for use in High Energy Physics research, it has been adopted and expanded by research groups in a wide range of fields, ranging from medical physics to semiconductor design. Different groups have extended the physics models as appropriate to their various fields. For this project, most of the physics were provided by the Livermore Low Energy Electromagnetic Models for charged particles, which are well tested and validated [2].

While Geant4 also provides high-precision neutron libraries based on the ENDF-B/VII data, the low elastic scattering cross section made doing a large number of tests and simulations impractical. Instead, ions of the appropriate type were produced directly as primary particles. For each foil material, the angular scattering distribution at 14 MeV and 2.45 MeV was inputted into the code — over the narrow energy variations relevant to these experiments, the angular distribution is insensi-

tive. In a real detector, the slight structure in angular distribution as a function of energy would be corrected in the same way as energy variations in cross sections. By producing ions directly, both of these corrections are skipped. In a real system, both would contribute to the overall uncertainty of the detector; however, this thesis does not attempt to assess the uncertainty of performance, since to many variables are unknown and impossible to meaningfully estimate. Consequently, producing the ions directly does not impact the simulated performance of the detector.

For a given configuration, the ions are randomly generated within the space of the foil. The polar angle is sampled from the angular distribution, while the azimuthal angle is randomly sampled between 0 and 2π. The energy corresponding to the sampled polar angle is calculated, and the particle is sent into the simulation. There is also a provision to set the start time of the particle, to simulate a neutron source of finite duration. This setup allows for force conditions to be implemented, to isolate the source of different behaviors: it can easily be changed to a point source, rather than a spatial source, to see what difference the foil makes, or the foil can be replaced with vacuum, to test the impact of straggling, and so on.

The detector is a rectangular prism, parallel to the foil, at a distance determined by the configuration. Geant4 is configured to tally at what point each ion enters the detector, and at what time. This information is recorded for every simulated ion and written to a file at the end of the simulation.

This data is then read into Matlab, where it is binned based on the defined specifications of the configuration. Detector response, or non-ideal factors, could also be implemented at this stage; however, for purposes of these tests, it is assumed that the detector responds ideally, or could be corrected to ideal response. In this way, several detector resolutions, in both time and space, could be tested without re-simulating the physics.

Appendix A contains a partial code listing for more implementation detail.

4.3 High Resolution Configuration

4.3.1 Setup

Based on earlier chapters, optimizing resolution requires minimizing foil and pixel size while using a large number of time bins with short duration relative to the spread of time-of-arrival for the ions. Furthermore, thin foil is preferable to minimize ion straggling. Balancing these requirements and the need for a feasible efficiency, the chosen configuration uses a 2 mm by 2mm by 2 nm cobalt foil, 10 cm from a diode array sensor. With a narrow, 15° angle of acceptance, the sensor consists of 729 diodes, each 2 mm by 2mm in size, forming a grid 5.4 cm on a side. This is more than could reasonably be accommodated by existing digitizing chips, but could plausibly be accommodated with an expanded version of the technology in PSEC4 chips. Based on existing hardware, it will be assumed that each diode has a limit of 70 ps samples, with 256 samples. While traditional diodes of this size could not achieve these speeds due to capacitance, it is achievable with three-dimensional diodes similar to those developed at CERN. The arrangement is shown in Figure4.1.

This configuration will form as a single "cell". Over the available volume, it is repeated in a grid to achieve usable efficiencies. It will be assumed that each "cell" is isolated from its neighbors (e.g. by thin partial walls extending from the borders), so that there is no appreciable crosstalk between cells. Each cell is geometrically identical, but the timing configuration for each is adjusted based on its location relative to the source. In reality, each cell would also be angled slightly relative to the source to maintain perpendicularity to the source; but this is not reflected in Figure 4.2, showing the configuration.

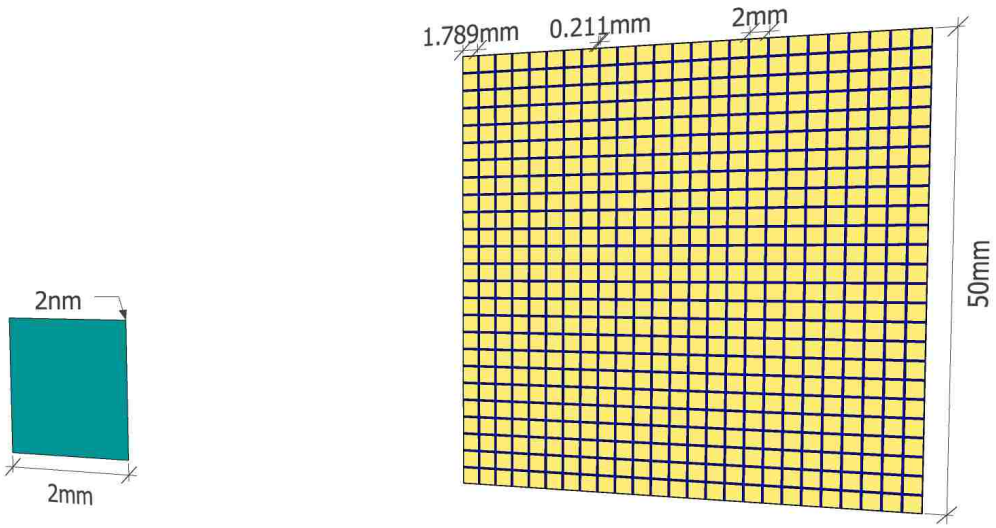

Figure 4.1: A single cell of the high resolution configuration.

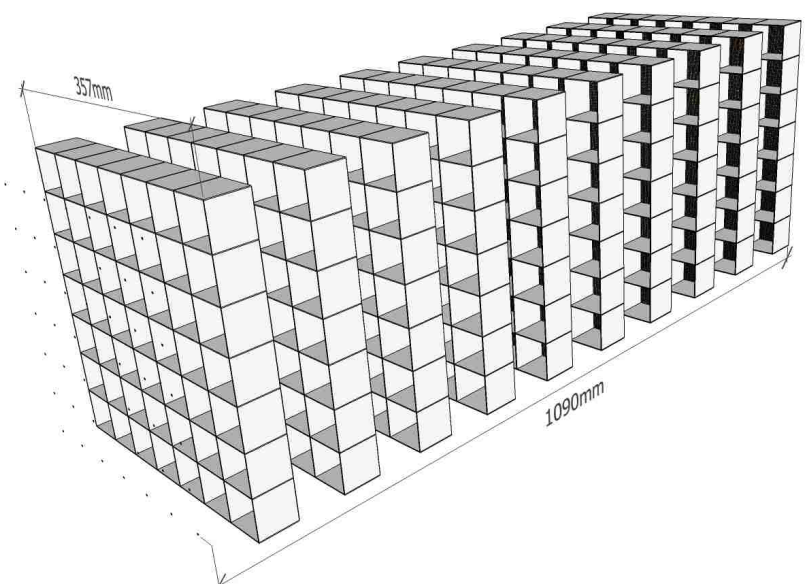

Figure 4.2: Full view of the high resolution configuration.

A flat-field gaussian fusion source provides the neutrons. Tests were done with both DD and DT sources, at 5 keV and 10 keV. Each of these was then done with 0 ps, 500 ps, 1000 ps, and 2000 ps of uniform uncertainty in source time. While most real sources are probably closer to gaussian in time, this is intended to also roughly account for timing uncertainty in circuitry and other sources. This results in a total of 12 tests for this configuration, and a thorough picture of performance.

4.3.2 Performance

While there are lots of graphs and pictures to consider, it is desirable to begin with a concrete description of performance relative to a known standard. This is provided by direct comparison to the existing spectrometer at the National Ignition Facility, as described in table I of [4]. The key data for the high resolution performance configuration are reproduced in Table 4.1, along with the simulated performance of the proposed setup.

Table 4.1: A comparison of the performance of NIF Proton Recoil Neutron Spectrometer in its high resolution configuration and the propose heavy-ion recoil spectrometer in its high resolution configuration.

Parameters	NIF High Resolution Configuration	Heavy-Ion Recoil High Resolution Configuration
Foil Area (cm^2)	12.8 ± 0.1	19.6
Foil Distance (cm)	26 ± 0.3	81.1 (average)
Foil Thickness	47 ± 1 um	2 nm
Foil Number Density ($\frac{d}{cm^3}x10^{22}$)	7.9 ± 0.1	9.1
Efficiency at 14 MeV	$1.56x10^{-11}$	$3.89x10^{-11}$
δE_{MRS} FWHM at 14 MeV (MeV)	0.69	0.33

The high-resolution configuration performs well. It achieves twice the resolution of the existing detector with more than double the efficiency. This is achievable, in

spite of foil 4 orders of magnitude thinner than the existing NIF detector, because of the much wider acceptance angle. While this is a comparison of a fully designed and functional detector to a sketch of a detector concept, it indicates that the heavy-ion Recoil spectrometer could operate in the same league as existing designs, and likely at least incrementally better.

While clear and pleasantly quantifiable, this assessment says little about performance in measuring a real spectrum, arriving over a much wider time interval than a perfect, 14 MeV beam. Since timing is key for this detector, it is appropriate to consider more realistic assessments. For example, a key performance question is how much timing uncertainty the detector can accommodate and still provide useful results. Figure 4.3 shows several spectra: the actual, generated DT spectra, along with the spectra as measured with the various timing spreads discussed above.

Visual inspection is encouraging. It appears to be a good fit, and can weather considerable timing uncertainty before performance is dramatically impacted. More concretely, with 500 ps uncertainty the measured mean is 14.087 MeV with a full width at half max (FWHM) of 0.847 MeV, an error of 7.98% in the FWHM and 0.20% in the mean. Most of the error in the FWHM is due to the systematically down-shifted spectrum at lower energies. The systematically low mean is due to the straggling energy loss in the foil, which causes the calculated neutron energy to be slightly low; this effect will be more pronounced at the lower energies of DD ions. Overall, performance is consonant with what would be expected from the 14 MeV performance in table 4.1.

Since DD neutrons are of much lower energy, there is a wider time spread for DD neutrons on a given detector geometry, so for DD measurements on the cobalt foil, the duration of the time bins was increased to 120 ps to fully span the range of arrival times. Measurements of slow DD ions are less impacted by time uncertainty, so dramatically better performance with large time spreads is to be expected. This

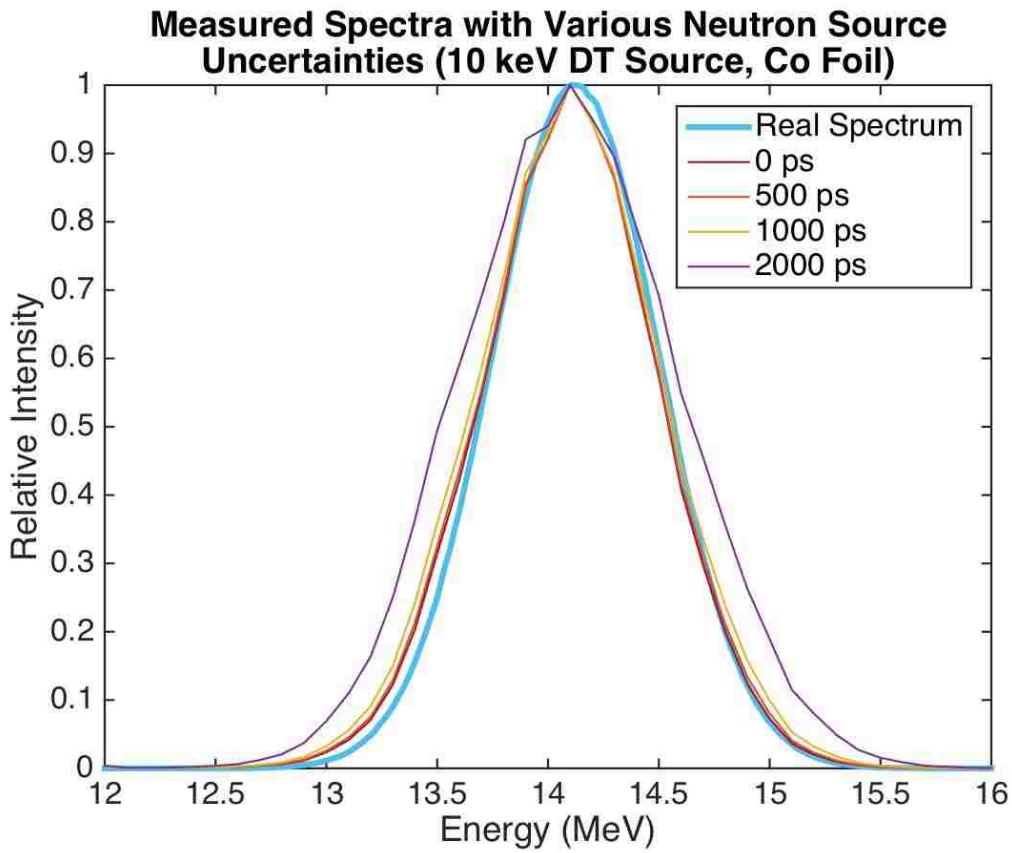

Figure 4.3: Measured spectra from 10 keV DT source on Co foil, with high resolution sensor and uniform random variation in source neutron timing.

behavior are confirmed in Figure 4.4.

At a glance, the results for DD actually appear better than those for DT; however, this impression is largely due to the hugely reduced sensitivity to time uncertainty, to the point that performance is not statistically different for any of the measured spectra. Once again using the 500 ps uncertainty spectra as a reference, the measured mean is 2.448 MeV, while the FWHM is 0.366 MeV, which represent errors of 1.11% and 0.14%, respectively. That is to say that the results are better than for DT, which is counter intuitive. Given the lower energy, it would be reasonable to expect reduced performance due to greater straggling losses; however, at the lower energy the

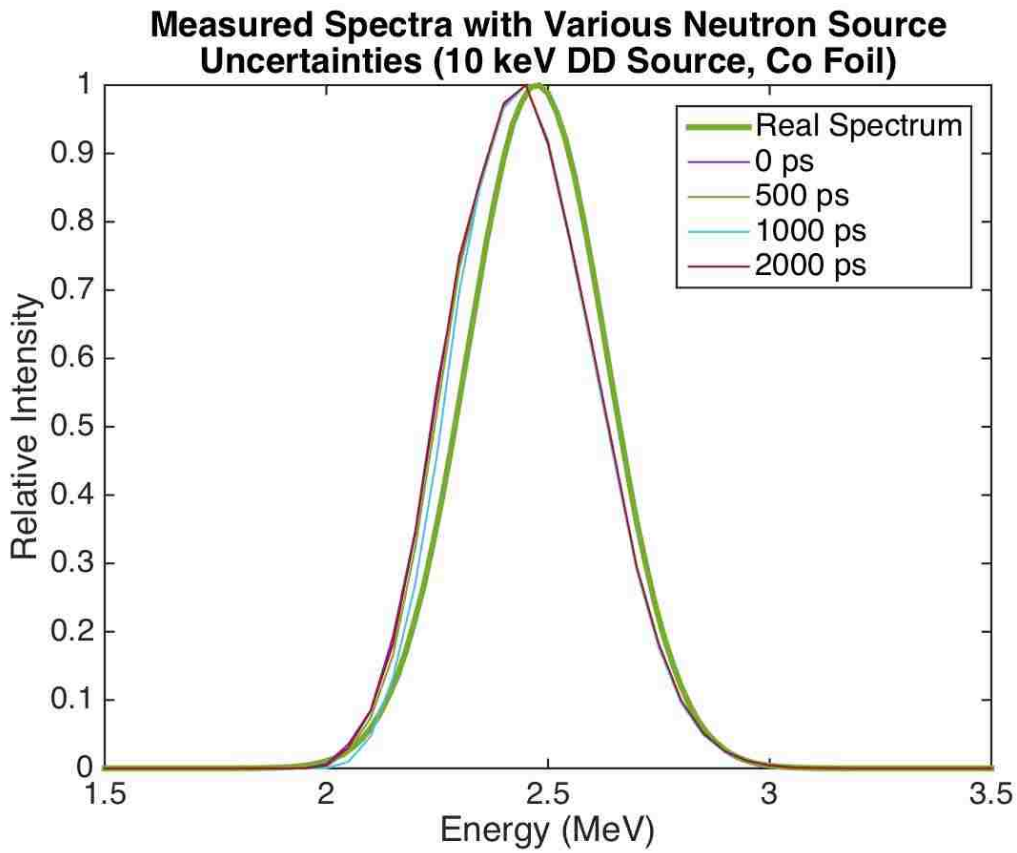

Figure 4.4: Measured spectra from 10 keV DD source on Co foil, with diode sensor and uniform variation in source neutron timing.

straggling losses are more uniform across the relevant energy spectrum. As a result, rather than a downshifted low end, the DD data are almost uniformly downshifted, leading to a more accurate FWHM measurement.

Finally, the pragmatic question from a fusion energy perspective: can this detector distinguish spectra for different fusion temperatures. The answer, fortunately, is yes, even with fairly substantial uncertainties, but identifying the temperatures is somewhat harder than differentiating them. Figures 4.5 and 4.6 show measured 5 and 10 keV spectra with 500 ps timing uncertainties for DT and DD neutrons and cobalt foil.

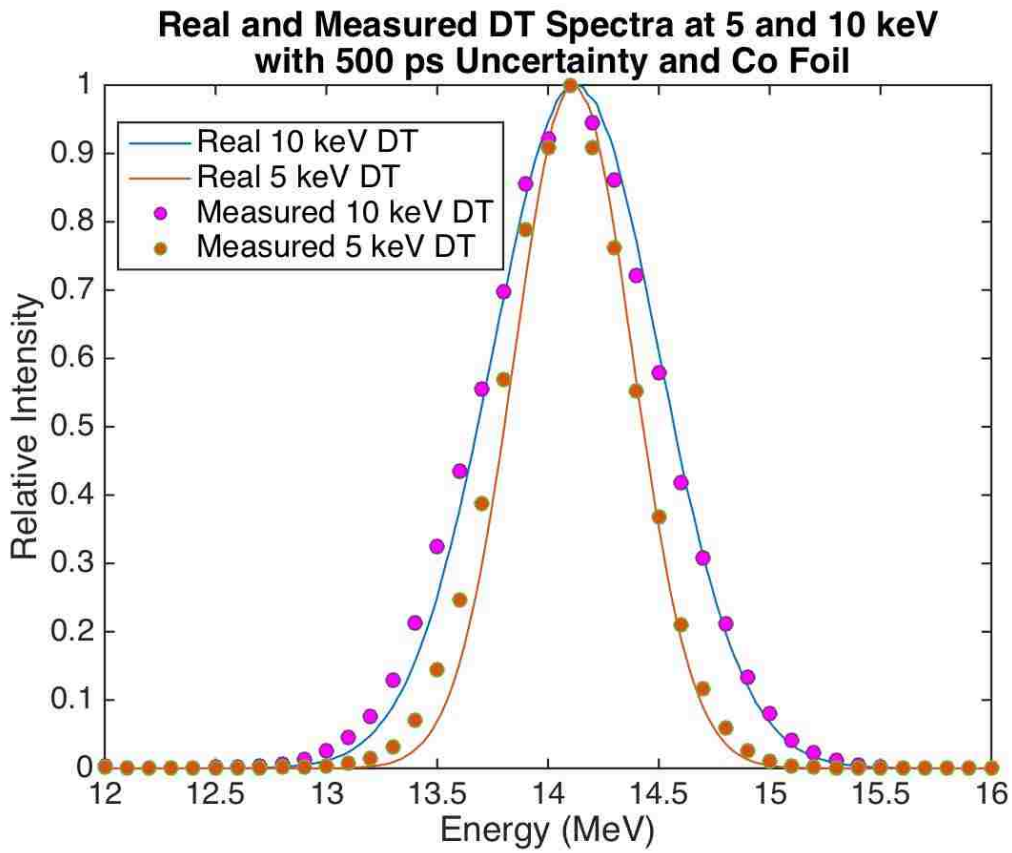

Figure 4.5: Real and measured DT spectra at 5 and 10 keV on Co foil, with diode sensor and 500 ps of variation in neutron source timing.

The DT results look good, and are objectively solid. The measured temperatures are 6.69 keV and 11.69 keV, for respective errors of 33.8% and 16.9%. While these are not incredibly accurate, the difference in temperature is measured exactly correctly. The DD results are not, visually, as impressive, primarily due to a more prominently shifted low end. The analytic results, however, are comparable, with measured temperatures of 7.35 keV and 10.07 keV, for errors of 47% and 0.7%. It is unclear why the 10 keV DD results are so much better than any others. Overall, it is an imperfect, but not useless, tool for measuring absolute temperature, and an excellent tool for measuring relative temperature.

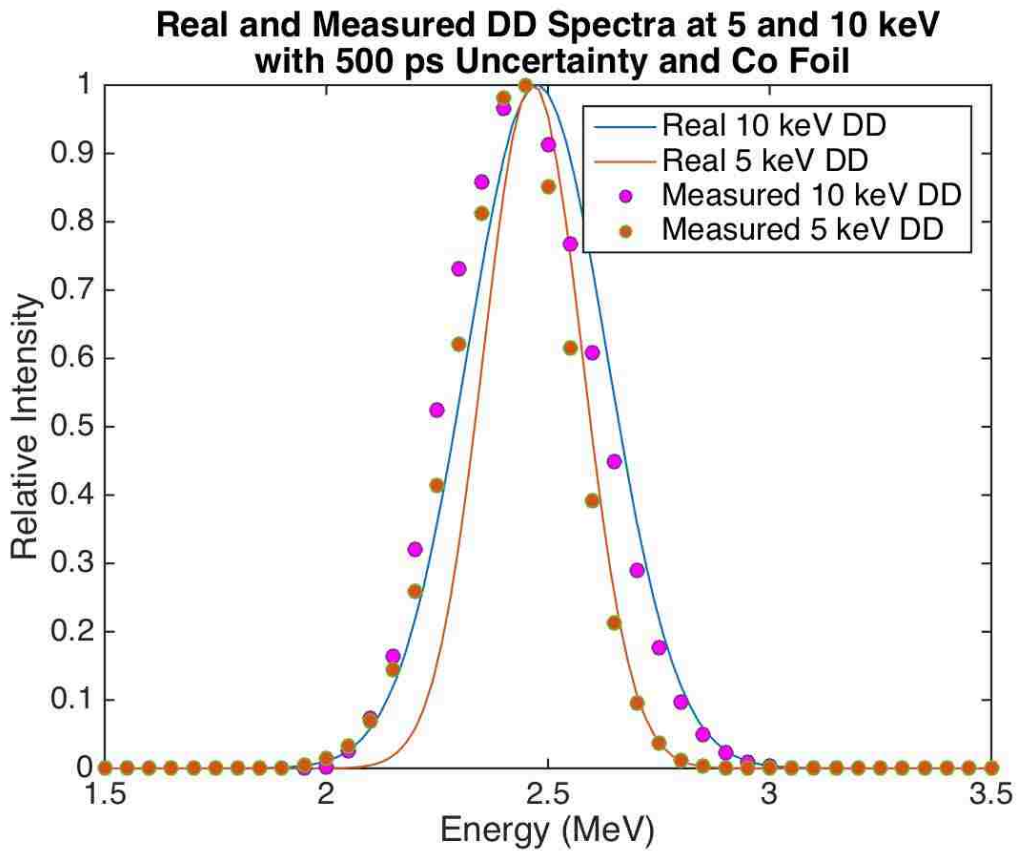

Figure 4.6: Real and measured DD spectra at 5 and 10 keV on Co, with diode sensor and 500 ps of variation in neutron source timing.

4.4 High Efficiency Configuration

4.4.1 Design

This configuration faces the inverse problem of the high resolution configuration: getting the efficiency as high as possible while keeping moderately useful resolution. While nearly every design parameter impacts efficiency to some degree, the single greatest factor is foil dimensions. Efficiency increases as the cube of the foil volume, so increasing a high efficiency design is primarily about making the foil as large and

thick as possible without compromising too much performance. With this in mind, a lighter foil material is obviously desirable. The resulting design uses a 10 mm by 10 mm by 25 nm carbon "foil" 10 cm from a diode detector. The diode detector will be configured exactly the same as in the high efficiency detector — meaning that the two foils could be easily switched out to go from "high efficiency" to "high resolution" modes. Likewise, the same sources are used as above. The resulting cell geometry and complete arrangement are shown in Figures 4.7 and 4.8.

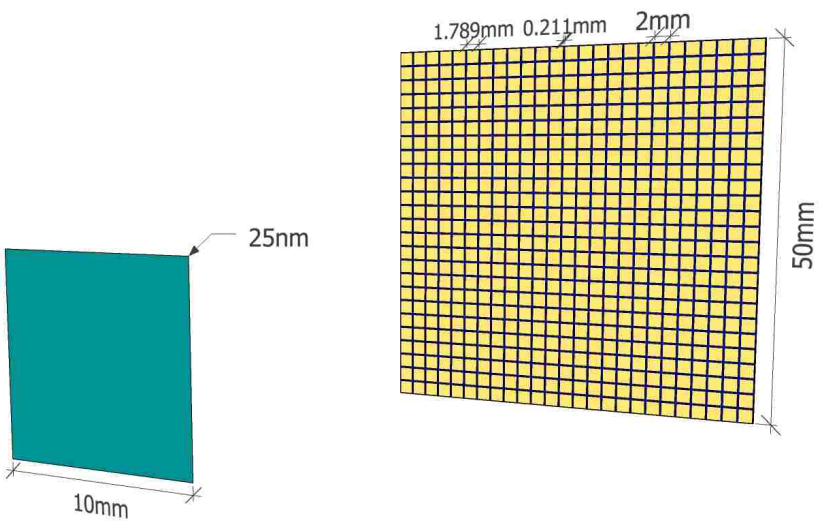

Figure 4.7: A single cell of the high efficiency configuration.

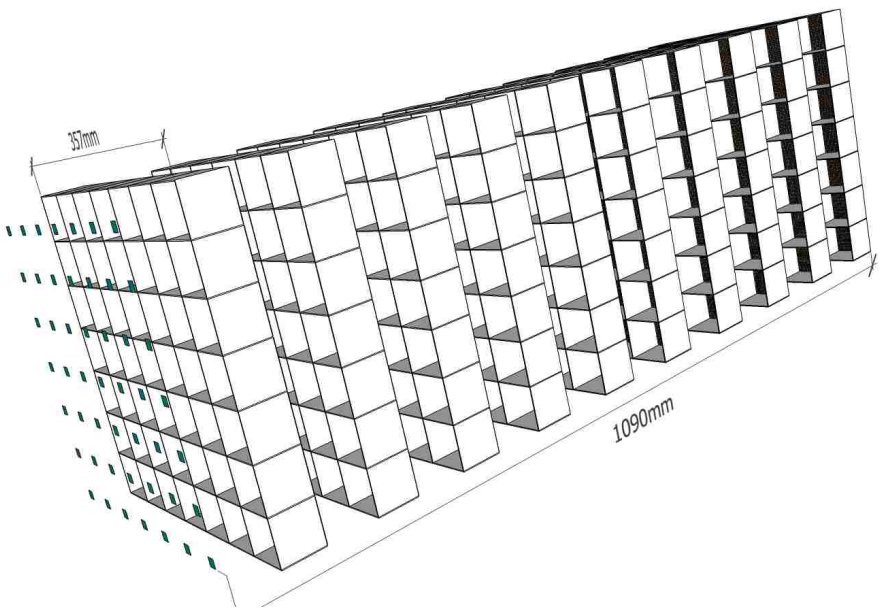

Figure 4.8: Full view of the high efficiency configuration.

4.4.2 Performance

Since this is the high efficiency configuration, it is appropriate to begin by comparing performance to the high efficiency configuration of the spectrometer at NIF, in Table 4.2.

The high efficiency configurations compare very similarly to the high resolution configurations: the heavy-ion spectrometer has better efficiency (by a factor of 3.5), and a little over double the resolution. While these numbers are impressive, the FWHM at 14 MeV is not reflective of how well the spectrometer does with a full spectrum.

The main performance struggle for this detector comes from the lighter material. The carbon ions, being five times lighter than the cobalt, receive much more energy in elastic collisions, and go much faster, as shown in the discussion of arrival times

Table 4.2: A comparison of the performance of NIF Proton Recoil Neutron Spectrometer in its high efficiency configuration and the propose heavy-ion recoil spectrometer in its high efficiency configuration.

Parameters	NIF High Efficiency Configuration	Heavy-Ion Recoil High Efficiency Configuration
Foil Area (cm^2)	12.8 ± 0.1	490
Foil Distance (cm)	26 ± 0.3	81.1 (average)
Foil Thickness	259 ± 1 um	25 nm
Foil Number Density ($\frac{d}{cm^3}x10^{22}$)	7.7 ± 0.1	10.8
Efficiency at 14 MeV	8.48x10^{-11}	2.94x10^{-10}
δE_{MRS} FWHM at 14 MeV (MeV)	1.6	0.67

in the second chapter. This means that even with a bin duration of 70 ps, the bins are too wide for good resolution. Consequently, in comparison to the beautifully accurate spectra of the high resolution configuration, the spectra in Figure 4.9 look extremely rough.

Since the carbon ions are so much faster, and consequently have a much narrower spread in arrival time, timing uncertainty has a proportionately greater impact — the 2 ns spread was not even worth plotting. Clearly, this configuration is less than desirable for the detailed measurement of a spectrum, but it does reasonably well at measuring the most important aspects. The measured mean without time uncertainty is 14.6032 MeV, while the FWHM is 1.1662 MeV, which correlates to errors of 3.381% and 33.04%, respectively. Much worse than the high resolution configuration, but a reasonable degradation relative to the improvement in efficiency.

The detector should work much better for a DD source, with its much lower energy range. This is born out in Figure 4.10.

The slower ions largely negate the timing spread, resulting in spectra that look much more comparable to the DD performance of the high resolution configuration.

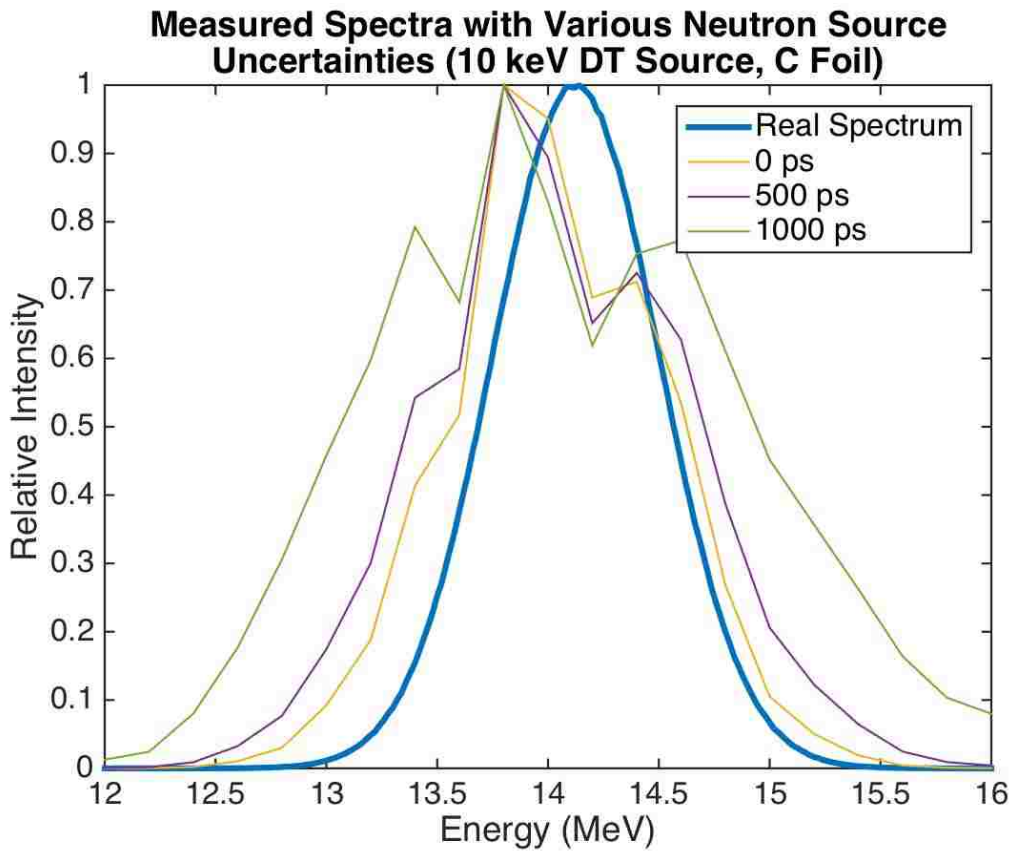

Figure 4.9: Measured spectra from 10 keV DT source on C foil, with diode sensor and uniform random variation in source neutron timing.

The spectra are not as tightly bunched with increased timing variation, but it shows the same downshift do to increased straggling of the lower-energy ions. Analytic assessment bears out the visual impression: with 500 ps of uncertainty the error in the mean is only 1.683%, while the error in FWHM is 5.680%. This is an impressively accurate spectrum.

It remains to assess how this configuration performs at distinguishing temperatures in fusion spectra. The spectra are shown in Figures 4.11 and 4.12.

For DT, the results are rather poor, but given the quality of the 10 keV spectra

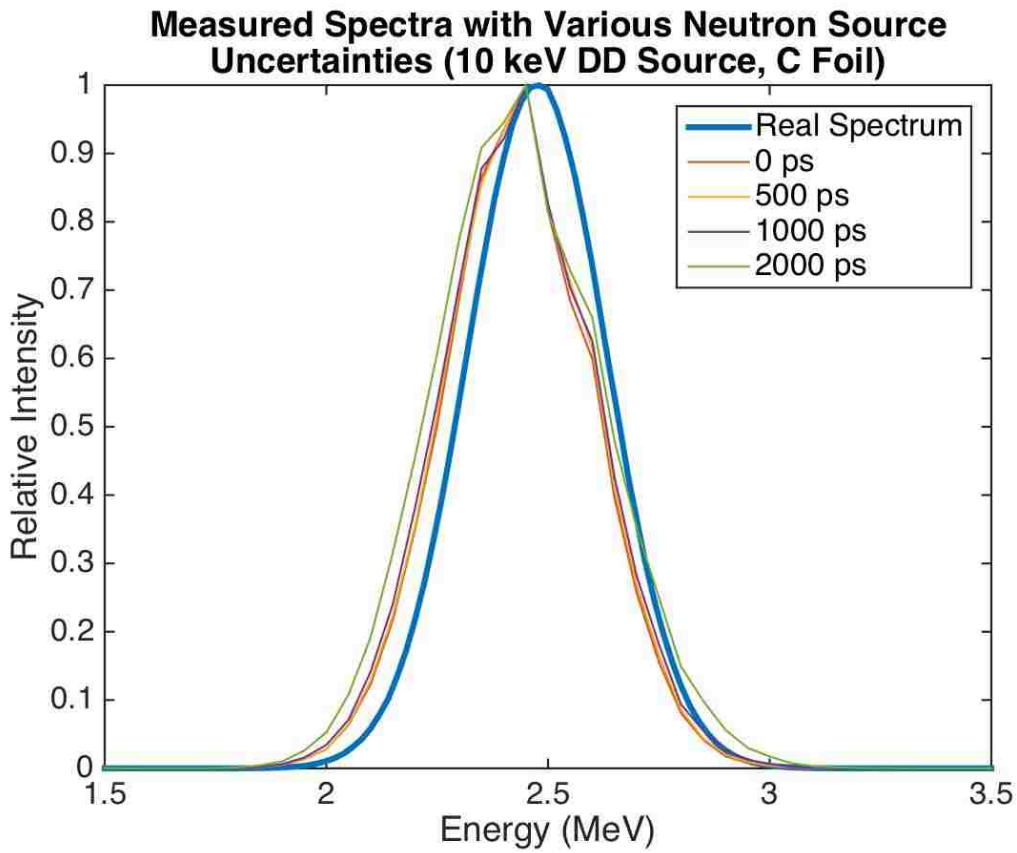

Figure 4.10: Measured spectra from 10 keV DD source on C foil, with diode sensor and uniform variation in source neutron timing.

discussed above, this is not surprising. The two temperatures are distinguishable, but without knowing the difference between the spectra was temperature, it would not necessarily be obvious. The measured temperatures are 11.15 keV and 16.18 keV, respectively. It is worth noting that while the temperatures are inaccurate, the measured difference in temperatures, is off by only 0.52%. In other words, it remains a reasonable tool for measuring relative temperature. The DD results look substantially better, again due to the slower-moving ions. The analytic results are also moderately better, with measured temperatures of 6.37 keV and 11.36 keV. These results are almost identical to those for the high resolution configuration with

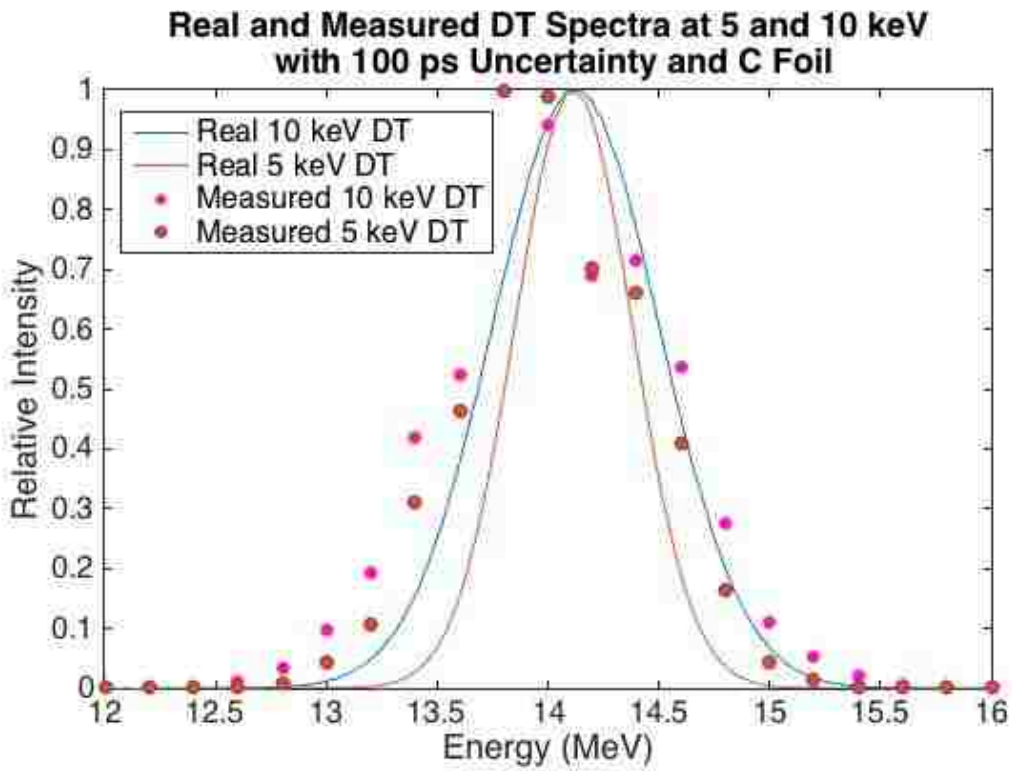

Figure 4.11: Real and measured DT spectra at 5 and 10 keV on C foil, with diode sensor and 100 ps of variation in neutron source timing.

a DT spectrum, and extremely good for the high efficiency design.

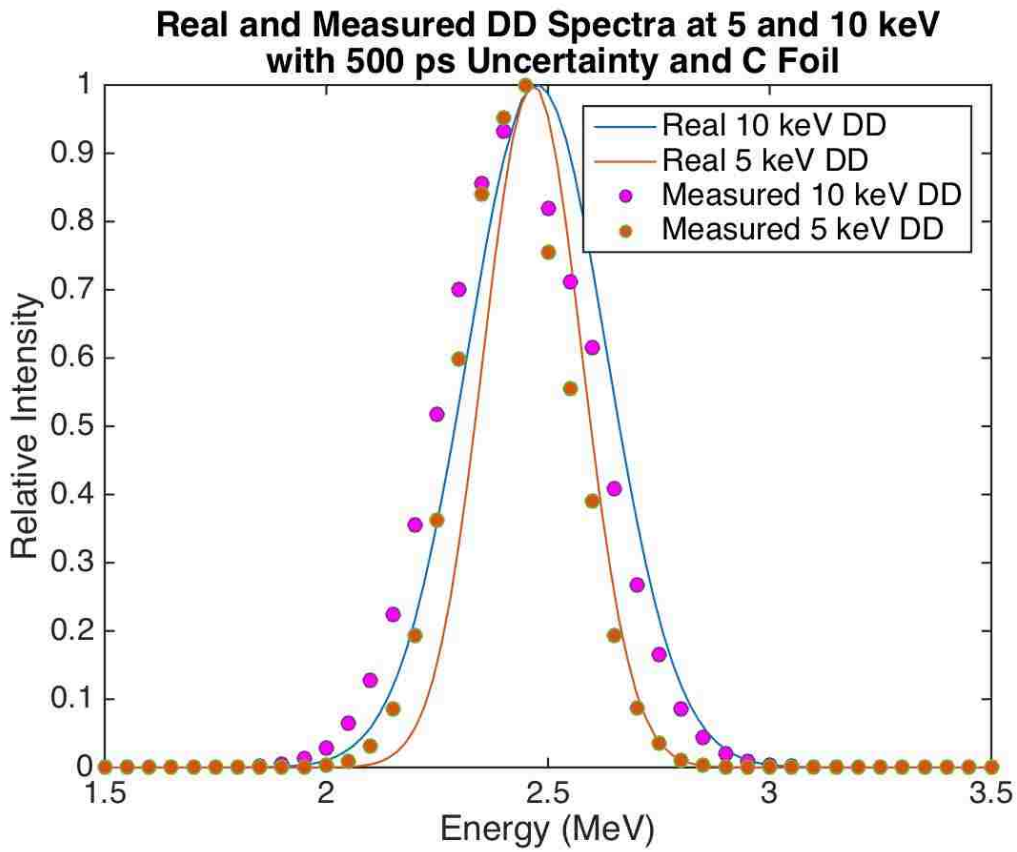

Figure 4.12: Real and measured DD spectra at 5 and 10 keV on C foil, with diode sensor and 500 ps of variation in neutron source timing.

Chapter 5

Conclusion and Future Work

The heavy-ion recoil spectrometer promises to meet or exceed the performance of existing proton recoil magnetic spectrometers. There are uncertainties and unknowns, but only testing and experimentation can change them to certainties, and all of simulated results make a compelling case that the concept is worth pursuing.

This work has been broad, surveying the range of options and the dynamics of how they might impact performance. The next step is to delve into details, selecting a location and design constraints, and assembling a detailed design that can be subjected to extensive and meticulous simulation. This process would uncover the engineering challenges — electronics performance, unforeseen sources of noise, detailed shielding requirements — that are always a part of any new design concept.

Perhaps there will be significant problems. Perhaps the electronics will be too sensitive or too fragile. What makes this concept exciting is that it depends on technology that is improving by leaps and bounds. Existing recoil spectrometers rely on established technologies: CR39 and electromagnets. They are extremely reliable, but have little room to improve. The heavy-ion recoil spectrometer depends on semiconductor detector technology and high-speed electronics. It would be difficult

to name two more active fields — and each increase in the bandwidth of silicon detectors, each increase improvement in radiation hardening, each breakthrough in high speed electronics timing, directly improves the performance of this detector concept. The potential for this detector concept will only improve with time.

Appendix A

Sample Geant4 Simulation

A.1 Introduction

The simulations from which all the results in Chapter 4, and several sections elsewhere, originated, are in two parts. Geant4 is used to simulate the physics involved, and then Matlab is used to post-process the ion energy deposition in the sensor to simulate what the detector would have "seen" for the simulation. This offers the benefit of exploring the parameter space of the sensor configuration (e.g. changing the angle of acceptance, the spatial and time binning, etc) without rerunning the simulation. The following sections list most of the code used for both parts of the simulation, with summary description of the role of each file

A.2 Geant4 Simulation

A.2.1 Main

The main file, in keeping with C++ standards, is the "head" of the simulation. In it the physics, geometry, particle source, and detection/data logging objects are all initialized, and the macro that runs the simulation is called. The main file and the macro function are included here. Most of these objects have separate classes, described below; however, the physics is built using preconfigured modular subunits within the main function, rather than being assembled in a separate class.

```
1  #ifdef G4MULTITHREADED
2  #include "G4MTRunManager.hh"
3  #else
4  #include "G4RunManager.hh"
5  #endif
6
7  #include "G4UImanager.hh"
8  #include "G4UIterminal.hh"
9  #include "G4UItcsh.hh"
10 #include "G4UIterminal.hh"
11 #include "G4UIExecutive.hh"
12
13 #include "ActionInitialization.hh"
14 #include "DetectorConstruction.hh"
15 #include "PhysicsList.hh"
16 #include "G4VModularPhysicsList.hh"
17 #include "G4EmStandardPhysics_option4.hh"
18 #include "G4EmStandardPhysics_option3.hh"
```

```
19 #include "G4EmStandardPhysics_option1.hh"
20 #include "G4EmLivermorePhysics.hh"
21 #include "G4StepLimiterPhysics.hh"
22 #include "G4HadronPhysicsFTFP_BERT_HP.hh"
23 #include "G4HadronElasticPhysics.hh"
24 #include "G4IonPhysics.hh"
25 #include "G4GenericIon.hh"
26 #include "G4ProcessManager.hh"
27 #include "G4EmLivermorePhysics.hh"
28
29 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo....
30
31 int main(int argc,char** argv)
32 {
33     // Choose the Random engine
34
35     G4Random::setTheEngine(new CLHEP::RanecuEngine);
36
37     // Construct the default run manager
38
39 #ifdef G4MULTITHREADED
40     G4MTRunManager* runManager = new G4MTRunManager;
41     runManager->SetNumberOfThreads(8); // Is equal to 2 by
           default
42 #else
43     G4RunManager* runManager = new G4RunManager;
44 #endif
```

```
45
46      // Set mandatory user initialization classes
47      DetectorConstruction* detector = new
            DetectorConstruction;
48      runManager->SetUserInitialization(detector);
49
50      //      runManager->SetUserInitialization(new PhysicsList)
            ;
51      G4VModularPhysicsList* physicsList = new
            G4VModularPhysicsList();
52      physicsList->SetVerboseLevel(0);
53      physicsList->RegisterPhysics(new G4IonPhysics);
54      physicsList->RegisterPhysics(new G4HadronElasticPhysics)
            ;
55      G4EmLivermorePhysics* emphys = new G4EmLivermorePhysics
            (0);
56      physicsList->RegisterPhysics(emphys);
57      physicsList->RegisterPhysics(new G4StepLimiterPhysics())
            ;
58      physicsList->SetDefaultCutValue(1*nm);
59      physicsList->SetCuts();
60      runManager->SetUserInitialization(physicsList);
61
62      // User action initialization
63      runManager->SetUserInitialization(new
            ActionInitialization(detector));
64
65      // Initialize G4 kernel
```

```
66      runManager->Initialize();
67
68      // Get the pointer to the User Interface manager
69      G4UImanager* UImanager = G4UImanager::GetUIpointer();
70
71      G4UIterminal * ui = new G4UIterminal(new G4UItcsh());
72      ui->SessionStart();
73      UImanager->ApplyCommand("/control/execute recoilspec.mac
            ");
74
75      delete ui;
76
77      delete runManager;
78
79      return 0;
80  }
```

```
1  # /control/execute vis.mac
2  /process/em/spline false
3  /process/msc/StepLimit Minimal
4  /process/msc/RangeFactor 0.000001
5  /process/eLoss/minKinEnergy 0.1 eV
6  /process/msc/LateralDisplacement true
7  /process/msc/Skin 5.0
8  /process/msc/GeomFactor 10
9  /process/em/spline false
10 /process/em/fluo true
11 /process/em/auger true
12 /process/em/pixe true
```

```
#/process/em/deexcitation world true true true
/run/printProgress 500000
/tracking/verbose 0
/process/verbose 0
#/gun/particle ion
#/gun/ion 20 48 15
/gun/particle gamma
/histogram/Set_X_bin 0.01 um
/histogram/Set_Y_bin 0.01 um
/histogram/Set_Z_bin 0.1 um

/gun/energy 2.0 keV
/histogram/Empty_Histogram
/histogram/Set_Output_Path /users/encharitimone/
    C12_25nmFoil_10mmEdge_500psUncert_10cmSep_1E7Particles_DD5keV
    .csv
/run/initialize
/run/beamOn 10000000
#/run/beamOn 100
/histogram/Write_Histogram

/gun/energy 3.0 keV
/histogram/Empty_Histogram
/histogram/Set_Output_Path /users/encharitimone/dummy.csv
/run/initialize
/run/beamOn 20
/histogram/Write_Histogram
```

A.2.2 Geometry

The next main component is, in Geant4 parlance, the detector construction, which describes the geometry and materials involved in the simulation. Here the material for the foil is determined (including replacing it with vacuum to neglect straggling), and foil thickness is chosen. While the foil material is chosen here, the ion material must be correctly defined in the source as well — the correct ions are not automatically used.

```
1  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo....
2
3  #ifndef DetectorConstruction_h
4  #define DetectorConstruction_h 1
5
6  #include "G4VUserDetectorConstruction.hh"
7  #include "G4VPhysicalVolume.hh"
8  #include "G4LogicalVolume.hh"
9  #include "G4Box.hh"
10 #include "G4Sphere.hh"
11 #include "G4Material.hh"
12 #include "G4NistManager.hh"
13 #include "G4PVPlacement.hh"
14 #include "G4UserLimits.hh"
15 #include "G4VisAttributes.hh"
16
17 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo....
18
```

```
19 class G4Region;
20
21 class DetectorConstruction : public
      G4VUserDetectorConstruction
22 {
23     public:
24
25     DetectorConstruction();
26
27     ~DetectorConstruction();
28
29     G4VPhysicalVolume* Construct();
30
31     G4Region* GetTargetRegion()  {return fRegion;}
32
33     private:
34
35     G4double             fWorldSizeX;
36     G4double             fWorldSizeY;
37     G4double             fWorldSizeZ;
38
39     G4VPhysicalVolume* fPhysiWorld;
40     G4LogicalVolume*   fLogicWorld;
41     G4Box*             fSolidWorld;
42
43     G4Material*        fFoilMaterial;
44     G4Material*        fBackingMaterial;
45     G4Material*        fVacMaterial;
```

```
46      G4Region*              fRegion;
47
48      void DefineMaterials();
49
50      G4VPhysicalVolume* ConstructDetector();
51  };
52  #endif
```

```
 1  #include "DetectorConstruction.hh"
 2  #include "G4SystemOfUnits.hh"
 3  #include "G4Region.hh"
 4  #include "G4ProductionCuts.hh"
 5  #include "G4UserLimits.hh"
 6
 7  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
        ........oooOO0OOooo....
 8
 9  DetectorConstruction::DetectorConstruction()
10  :fPhysiWorld(NULL), fLogicWorld(NULL), fSolidWorld(NULL)
11  {}
12
13  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
        ........oooOO0OOooo....
14
15  DetectorConstruction::~DetectorConstruction()
16  {}
17
18  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
        ........oooOO0OOooo....
```

```
19
20 G4VPhysicalVolume* DetectorConstruction::Construct()
21
22 {
23     DefineMaterials();
24     return ConstructDetector();
25 }
26
27 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo....
28
29 void DetectorConstruction::DefineMaterials()
30 {
31
32 }
33
34 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo....
35 G4VPhysicalVolume* DetectorConstruction::ConstructDetector()
36 {
37
38     //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
           ........oooOO0OOooo....
39
40     // Possible foil materials
41     G4NistManager * man = G4NistManager::Instance();
42     G4Material * C = man->FindOrBuildMaterial("G4_C");
43     G4Material * Au = man->FindOrBuildMaterial("G4_Au");
```

```
44      G4Material * In = man->FindOrBuildMaterial("G4_In");
45      G4Material * As = man->FindOrBuildMaterial("G4_As");
46      G4Material * Co = man->FindOrBuildMaterial("G4_Co");
47      G4Material * Al = man->FindOrBuildMaterial("G4_Al");
48      //     G4Material * poly = man->FindOrBuildMaterial("
          G4_POLYETHYLENE");
49
50      //Materials used in model
51      fVacMaterial = man->FindOrBuildMaterial("G4_Galactic");
52      fBackingMaterial = fVacMaterial;
53      fFoilMaterial = C;
54
55      // WORLD VOLUME
56      fWorldSizeX  = 50*cm;
57      fWorldSizeY  = 50*cm;
58      fWorldSizeZ  = 50*cm;
59
60      fSolidWorld = new G4Box("World",
                        //its name
                                fWorldSizeX/2,fWorldSizeY/2,
61                                 fWorldSizeZ/2);  //its size
62
63
64      fLogicWorld = new G4LogicalVolume(fSolidWorld,        //
          its solid
65                                        fVacMaterial,
                                                  //its
                                            material
```

```
66                                                    "World");               //
                                                          its name
67
68      fPhysiWorld = new G4PVPlacement(0,                                    //no
            rotation
69                                                    G4ThreeVector(),        //at
                                                          (0,0,0)
70                                                    "World",                //
                                                          its name
71                                                    fLogicWorld,
                                                                  //its logical
                                                          volume
72                                                    0,                      //
                                                          its mother volume
73                                                    false,
                                                                  //no boolean
                                                          operation
74                                                    0);                     //
                                                          copy number
75
76      //Foil Volume
77      G4double foilSideSize = 10*mm; // length of foil on one
            side
78      G4double foilSizeZ =  25*mm; // thickness of the foil
79
80      G4Box* foilSolid = new G4Box("Foil",
                                  //its name
81                                      foilSideSize/2,foilSideSize
```

```
                                        /2,foilSizeZ/2);    //its
                                        size
82
83
84      G4LogicalVolume* logicFoil = new G4LogicalVolume(
            foilSolid,          //its solid
85                                                          fFoilMaterial
                                                              ,
                                                              //its
                                                              material
86                                                          "Foil")
                                                              ;
                                                              //its
                                                               name
87
88
89      new G4PVPlacement(0,                //no rotation
90                        G4ThreeVector(0.0,0.0,-foilSizeZ/2),
91                        "Foil",           //its name
92                        logicFoil,        //its logical volume
93                        fPhysiWorld,              //its mother
                              volume
94                        false,            //no boolean
                              operation
```

```
95                              1);                              //copy
                                    number
96
97
98      //Foil Backing Volume
99      G4double backingSizeZ = 2*mm;
100
101     G4Box* backingSolid = new G4Box("Backing",
102                                     fWorldSizeX/2,
                                        fWorldSizeY/2,
                                        backingSizeZ/2);
103
104     G4LogicalVolume* logicBacking = new G4LogicalVolume(
            backingSolid,
105                                                         fBackingMaterial
                                                            ,
106                                                         "
                                                            Backing
                                                            ")
                                                            ;
107
108     new G4PVPlacement(0,
109                       G4ThreeVector(0.0,0.0,-backingSizeZ/2
                              - foilSizeZ/2),
110                       "Backing",
111                       logicBacking,
112                       fPhysiWorld,
113                       false,
```

```
114                             2);
115
116         //Sensor Volume
117         G4Box* detSolid = new G4Box("detector",
                                        //its name
118                                         fWorldSizeX/2,fWorldSizeY
                                              /2,0.1*mm);     //its size
119
120
121         G4LogicalVolume* logicDet = new G4LogicalVolume(detSolid
                ,          //its solid
122                                                          fVacMaterial
                                                             ,
                                                             //its
                                                             material
123                                                          "
                                                             Detector
                                                             ");
                                                             //its
                                                             name
124
125         new G4PVPlacement(0,                             //no
                rotation
126                           G4ThreeVector(0,0,10*cm),      //
```

```
                                    Defines space between foil and
                                    sensor
127                                 "TargetFront",                  //
                                    its name
128                                 logicDet,                       //its
                                    logical volume
129                                 fPhysiWorld,                    //its
                                    mother volume
130                                 false,                          //no
                                    boolean operation
131                                 4);                             //copy
                                    number
132
133         // Visualization attributes
134         G4VisAttributes* worldVisAtt= new G4VisAttributes(
                G4Colour(1.0,1.0,1.0)); //White
135         worldVisAtt->SetVisibility(false);
136         fLogicWorld->SetVisAttributes(worldVisAtt);
137
138         G4VisAttributes* worldVisAtt1 = new G4VisAttributes(
                G4Colour(1.0,0.0,0.0));
139         worldVisAtt1->SetVisibility(true);
140         logicFoil->SetVisAttributes(worldVisAtt1);
141
142         // Create Target G4Region and add logical volume
143
144         fRegion = new G4Region("Foil");
145
```

```
146       G4ProductionCuts* cuts = new G4ProductionCuts();
147
148       G4double defCut = 1*nanometer;
149       cuts->SetProductionCut(defCut,"gamma");
150       cuts->SetProductionCut(defCut,"e-");
151       cuts->SetProductionCut(defCut,"e+");
152       cuts->SetProductionCut(defCut,"proton");
153
154       fRegion->SetProductionCuts(cuts);
155       fRegion->AddRootLogicalVolume(logicFoil);
156
157       return fPhysiWorld;
158 }
```

A.2.3 Source Description

This component describes the particle source for the simulation. For these sims, each particle is generated by sampling a gaussian fusion energy distribution for neutrons, as well as sampling the angular scattering distribution for the foil material at the appropriate energy (these distributions are hard-coded into the files, but omitted in the code listing for brevity), to then describe the recoil ion, which is the actual primary particle. This class also assigns the internal "start" time of the particle, which is used to add uniform time uncertainty to the simulation. Here too the effective size of the foil is defined, based on the spread of the source, DD/DT/constant energy neutrons are selected, and any other source parameters that might need changed for different types of simulations.

```
1 #ifndef PrimaryGeneratorAction_h
```

```
 2 #define PrimaryGeneratorAction_h 1
 3
 4 #include "G4VUserPrimaryGeneratorAction.hh"
 5 #include "G4ParticleGun.hh"
 6 #include "globals.hh"
 7 #include "Randomize.hh"
 8 #include "G4MTRunManagerKernel.hh"
 9 #include "G4WorkerThread.hh"
10 #include <map>
11
12 #include <stdio.h>
13
14 class G4ParticleGun;
15 class G4Event;
16 class ScintDetDetectorConstruction;
17
18 class PrimaryGeneratorAction : public
      G4VUserPrimaryGeneratorAction
19 {
20   public:
21     PrimaryGeneratorAction();
22     virtual ~PrimaryGeneratorAction();
23
24     // static access method
25     static const PrimaryGeneratorAction* Instance();
26
27     // method from the base class
28     virtual void GeneratePrimaries(G4Event*);
```

```
29
30      // method to access particle gun
31      const G4ParticleGun* GetParticleGun() const { return
            fParticleGun; }
32
33          CLHEP::RandGauss* energyDistribution;
34
35    private:
36      static PrimaryGeneratorAction* fgInstance;
37      std::vector<double>* angvals;
38      std::vector<double>* angdist;
39
40      FILE* afile;
41
42      CLHEP::HepJamesRandom* engine;
43      G4ParticleGun*  fParticleGun; // pointer a to G4 gun
            class
44  };
45
46  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
        ........oooOO0OOooo......
47
48  #endif
```

```
1  #include "PrimaryGeneratorAction.hh"
2
3  #include "G4RunManager.hh"
4  #include "G4ParticleGun.hh"
5  #include "G4ParticleTable.hh"
```

```
 6 #include "G4ParticleDefinition.hh"
 7 #include "G4IonTable.hh"
 8 #include "Randomize.hh"
 9 #include "CLHEP/Random/RandFlat.h"
10 #include "G4ThreeVector.hh"
11 #include "globals.hh"
12
13 #include "G4Neutron.hh"
14
15
16 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo......
17
18 //binary search function for sampling spectra and other
       distributions
19 G4int getIndexInArray(double val, std::vector<double>*
       anarray, int startind, int endind){
20
21     //     printf("val = %f, start = %f, end = %f\n", val,
           anarray[startind], anarray[endind]);
22     //     fflush(stdout);
23
24     int ind = (endind + startind)/2;
25
26     if(startind == ind || endind == ind){
27         return ind;
28     }else if((*anarray)[ind] > val){
29         return getIndexInArray(val, anarray, startind, ind);
```

```
30      }else{
31          return getIndexInArray(val, anarray, ind, endind);
32      }
33  }
34
35
36  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
        ........oooOO0OOooo......
37
38  const PrimaryGeneratorAction* PrimaryGeneratorAction::
        Instance()
39  {
40      return fgInstance;
41  }
42
43  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
        ........oooOO0OOooo......
44
45  PrimaryGeneratorAction::PrimaryGeneratorAction()
46  : G4VUserPrimaryGeneratorAction(),
47  fParticleGun(0)
48  {
49      G4int n_particle = 1;
50      fParticleGun  = new G4ParticleGun(n_particle);
51
52      // default particle kinematic
53      fParticleGun->SetParticleMomentumDirection(G4ThreeVector
            (0.,0.,1.));
```

```
54      engine = new CLHEP::HepJamesRandom;
55
56      //Angular Distributions
57      //Indium and gold
58      double angbinarray[] = {/*truncated*/}
59      //Carbon DT
60      //      double angdistarray[] = {/*truncated*/}
61      //Cobalt DT
62      //      double angdistarray[] = {/*truncated*/}
63      //indium DT
64      //      double angdistarray[] = {/*truncated*/}
65      //gold DT
66      //      double angdistarray[] = {/*truncated*/}
67
68      //Carbon DD
69      double angdistarray[] = {/*truncated*/}
70      //Cobalt DD
71      //      double angdistarray[] = {/*truncated*/}
72      //Gold DD
73      //      double angdistarray[] = {/*truncated*/}
74
75      angvals = new std::vector<double>(angbinarray,
            angbinarray + 361);
76      angdist = new std::vector<double>(angdistarray,
            angdistarray + 361);
77
78      fgInstance = this;
79  }
```

```
80
81 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo......
82
83 PrimaryGeneratorAction::~PrimaryGeneratorAction()
84 {
85     delete fParticleGun;
86     fgInstance = 0;
87 }
88
89 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo......
90
91 void PrimaryGeneratorAction::GeneratePrimaries(G4Event*
       anEvent)
92 {
93
94     G4ParticleTable* particleTable = G4ParticleTable::
           GetParticleTable();
95     //Mean excitation energies, for reference
96     //carbon = 78
97     //aluminum = 166
98     //cobalt = 297
99     //gold = 790
100    //indium = 488
101    G4ParticleDefinition* particle = particleTable->
           GetIonTable()->GetIon(6, 12, 78*CLHEP::eV);
102    fParticleGun->SetParticleDefinition(particle);
```

```
103
104         eventCount[myID]++;
105         G4double newEnergy = DBL_MAX;
106         G4double newAng = 0;
107
108         G4double samp = CLHEP::RandFlat::shoot(engine, 0.0,1.0);
109
110         //Masses, for reference
111         //carbon
112         G4double mass = 12;
113         //aluminum
114         //      G4double mass = 26.981539;
115         //cobalt
116         //      G4double mass = 59;
117         //indium
118         //      G4double mass = 114.818;
119         //gold
120         //      G4double mass = 196.96657;
121
122         G4int ind = getIndexInArray(samp, angdist, 0,angdist->
                size());
123
124         newAng = (samp - (*angdist)[ind])/((*angdist)[ind+1] -
                (*angdist)[ind])*((*angvals)[ind+1] - (*angvals)[ind])
                 + (*angvals)[ind]; // sample scatter angle
125         double rot_ang = CLHEP::RandFlat::shoot(engine, 0.0, 2*
                CLHEP::pi); //sample particle rotation
126
```

```
127     fParticleGun->SetParticleMomentumDirection(G4ThreeVector
            (cos(rot_ang)*sin(newAng),sin(rot_ang)*sin(newAng),cos
            (newAng))); // Set corresponding direction
128
129     //Neutron spectra, based on the gaussian fusion
            distributions described in the MCNP source code
130     //DD Fusion Spectra
131     double T = 0.005; //T in keV
132     double b = 2.449408+0.4697881*pow(T,2.0/3.0)+0.37570337*
            T;
133     double a = sqrt(1.0018756*b*T);
134
135     //DT Fusion Spectra
136     //      double T = 0.01; //T in keV
137     //      double b = 14.1+0.4697881*pow(T,2.0/3.0)
            +0.37570337*T;
138     //      double a = sqrt(1.0018756*b*T);
139
140     double neut_en = CLHEP::RandGauss::shoot(engine, b*CLHEP
            ::MeV, a*CLHEP::MeV); //sample the active neutron
            source
141     //      double neut_en = 14*CLHEP::MeV; //Or generate a
            fixed-energy neutron
142
143     newEnergy = mass*4*1.00/pow(1.00+mass,2.0)*pow(sin((
            CLHEP::pi-2*newAng)/2.0),2.0)*neut_en; //Calculate ion
             enegy
144
```

```
145     //time uncertainty
146     fParticleGun->SetParticleTime(CLHEP::RandFlat::shoot(
            engine,0.0*CLHEP::ns,0.5*CLHEP::ns)); //adds time
            spread to source
147
148     fParticleGun->SetParticleEnergy(newEnergy); //set the
            ion energy
149     //randomize the ion's initial position within the foil
150     fParticleGun->SetParticlePosition(G4ThreeVector(CLHEP::
            RandFlat::shoot(engine,-5.0*CLHEP::mm,5*CLHEP::mm),
            CLHEP::RandFlat::shoot(engine,-5*CLHEP::mm,5*CLHEP::mm
            ),CLHEP::RandFlat::shoot(engine,-25*CLHEP::nm,0.0*
            CLHEP::nm)));
151     //or fix it at the origin for ideal tests
152     //     fParticleGun->SetParticlePosition(G4ThreeVector
            (0,0,0));
153
154     fParticleGun->GeneratePrimaryVertex(anEvent);
155 }
156 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
        ........oooOO0OOooo......
```

A.2.4 Data Recording

This component keeps track of when particles deposit energy on the sensor, and records where (in two dimensions) the energy is deposited, the time (relative to the particle's assigned start time), and the amount of energy deposited. At the end of the simulation, this data is written out to a user-specified file. No parsing parameters,

such as the pixels size, are decided here — all data is recorded exactly to machine precision.

```
1  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo....
2
3  #ifndef SteppingAction_h
4  #define SteppingAction_h 1
5
6  #include "G4UserSteppingAction.hh"
7  #include "G4ThreeVector.hh"
8  #include "SteppingActionMessenger.hh"
9
10 #include <fstream>
11 #include <ios>
12 #include <map>
13
14 class RunAction;
15 class DetectorConstruction;
16 class PrimaryGeneratorAction;
17
18 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo....
19
20 class SteppingAction : public G4UserSteppingAction
21 {
22 public:
23
24      SteppingAction();
```

```
25      ~SteppingAction();
26
27      void UserSteppingAction(const G4Step*);
28
29      void doWrite();
30      void emptyHistogram();
31      void doMerge();
32
33      void setXBin(G4double);
34      void setYBin(G4double);
35      void setZBin(G4double);
36      void setFilePath(G4String);
37      G4double getXBin();
38      G4double getYBin();
39      G4double getZBin();
40
41      G4int getIndexInArray(double val, std::vector<double>*
           anarray, int startind, int endind);
42
43  private:
44      SteppingActionMessenger * theMessenger;
45      std::fstream file_writer;
46      static std::map<G4ThreeVector,G4double>*
           shared_histogram;
47      std::map<G4ThreeVector,G4double>* histogram;
48      std::vector<double>* xbins;
49      std::vector<double>* ybins;
50      std::vector<double>* zbins;
```

```
51      static bool lock;
52      static int threads_ready;
53      bool top_thread;
54      G4int myID;
55      G4double xbin;
56      G4double ybin;
57      G4double zbin;
58      G4bool file_started;
59      G4int count_correction;
60  };
61  #endif
```

```
1  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo....
2  #include "G4MTRunManagerKernel.hh"
3  #include "G4WorkerThread.hh"
4
5  #include "SteppingAction.hh"
6  #include "RunAction.hh"
7  #include "PrimaryGeneratorAction.hh"
8
9  #include "G4SystemOfUnits.hh"
10 #include "G4SteppingManager.hh"
11 #include "G4VPhysicalVolume.hh"
12 #include "G4GenericIon.hh"
13
14 #include <thread>
15 #include <chrono>
16 #include <map>
```

```
17 #include <vector>
18
19 //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
       ........oooOO0OOooo....
20
21 std::map<G4ThreeVector,G4double>* SteppingAction::
       shared_histogram = new std::map<G4ThreeVector,G4double>();
22 bool SteppingAction::lock = false;
23 int SteppingAction::threads_ready = 0;
24 G4int SteppingAction::e_count = 0;
25 G4double SteppingAction::e_energy_count = 0;
26 G4int SteppingAction::xray_count = 0;
27 G4double SteppingAction::xray_energy_count = 0;
28
29 SteppingAction::SteppingAction() :
30 file_writer(),
31 xbin(25.0*um),
32 ybin(0.5*ns),
33 zbin(0.5*um),
34 count_correction(0),
35 file_started(false)
36 {
37     myID = G4MTRunManagerKernel::GetWorkerThread()->
           GetThreadId();
38     if(myID == 0)
39         top_thread = true;
40     else
41         top_thread = false;
```

```
42
43      while( threads_ready < G4MTRunManagerKernel::
            GetWorkerThread()->GetNumberThreads() ){
44          if(myID == threads_ready){
45              printf("Initializing thread %d, threads = %d\n",
                    myID,threads_ready);
46              fflush(stdout);
47              threads_ready++;
48          }else{
49              if( threads_ready >= G4MTRunManagerKernel::
                    GetWorkerThread()->GetNumberThreads() )
50                  break;
51          }
52      }
53
54      if(top_thread){
55          printf("All threads initialized.\n");
56          fflush(stdout);
57      }
58
59      histogram = new std::map<G4ThreeVector,G4double>();
60
61      if(top_thread){
62          printf("Finished initialization!\n");
63          fflush(stdout);
64      }
65
66      std::this_thread::sleep_for(std::chrono::milliseconds
```

```
        (10));
67
68      threads_ready = 0;
69
70      theMessenger = new SteppingActionMessenger(this);
71  }
72
73  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
        ........oooOO0OOooo....
74
75  SteppingAction::~SteppingAction()
76  {
77      while(lock){}
78      delete theMessenger;
79      file_writer.close();
80  }
81
82  //....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo
        ........oooOO0OOooo....
83
84  void SteppingAction::UserSteppingAction(const G4Step* step)
85  {
86      while(lock){}
87
88      if(step != NULL && step->GetPostStepPoint() != NULL &&
           step->GetPreStepPoint() != NULL && step->
           GetPostStepPoint()->GetPhysicalVolume() != NULL &&
           step->GetPreStepPoint()->GetPhysicalVolume() != NULL
```

```
        && step->GetPostStepPoint()->GetPhysicalVolume()->
        GetCopyNo() == 4 && step->GetPreStepPoint()->
        GetPhysicalVolume()->GetCopyNo() == 0){
89        double tval = step->GetPostStepPoint()->
              GetGlobalTime();
90        double xval = step->GetPostStepPoint()->GetPosition
              ().x();
91        double yval = step->GetPostStepPoint()->GetPosition
              ().y();
92
93        (*histogram)[G4ThreeVector(tval,xval,yval)] += step
              ->GetPostStepPoint()->GetKineticEnergy();
94    }
95 }
96
97 void SteppingAction::doWrite(){
98     doMerge();
99     if(top_thread){
100        if(!lock)
101            lock = true;
102
103        if(!file_started){
104            file_writer<<"time (ns),x (um),y (um),Energy
                  Deposited (eV)"<<G4endl;
105            file_started = true;
106        }
107        printf("We are about to write...\n");
108        fflush(stdout);
```

```
109             for(std::map<G4ThreeVector,G4double>::iterator iter
                    = shared_histogram->begin(); iter !=
                    shared_histogram->end(); iter++){
110                 file_writer<<iter->first.x()/ns<<",";
111                 file_writer<<iter->first.y()/um<<",";
112                 file_writer<<iter->first.z()/um<<",";
113                 file_writer<<iter->second/eV<<G4endl;
114             }
115             file_writer.flush();
116             lock = false;
117             threads_ready = 0;
118         }
119         else
120             while(lock){}
121 }
122
123 void SteppingAction::doMerge(){
124
125         while( threads_ready < G4MTRunManagerKernel::
                GetWorkerThread()->GetNumberThreads() ){
126             if(myID == threads_ready){
127                 for(std::map<G4ThreeVector,G4double>::iterator
                        iter = histogram->begin(); iter != histogram->
                        end(); iter++)
128                     (*shared_histogram)[iter->first] = (*
                            shared_histogram)[iter->first] + iter->
                            second;
129
```

```
130             threads_ready = threads_ready + 1;
131         }else{
132             if(threads_ready >= G4MTRunManagerKernel::
                    GetWorkerThread()->GetNumberThreads() or
                    threads_ready < 0)
133                 break;
134         }
135     }
136     histogram->clear();
137     e_count = 0;
138     e_energy_count = 0.0;
139     xray_count = 0;
140     xray_energy_count = 0.0;
141
142     threads_ready = -1;
143 }
144
145 void SteppingAction::emptyHistogram(){
146
147     while(lock){}
148     if(myID == 0)
149         shared_histogram->clear();
150     histogram->clear();
151     e_count = 0;
152     e_energy_count = 0.0;
153     xray_count = 0;
154     xray_energy_count = 0.0;
155     count_correction = PrimaryGeneratorAction::
```

```
            GetGlobalEventCount();
156         file_started = false;
157
158     }
159
160     void SteppingAction::setXBin(G4double newval){
161         while(lock){}
162         emptyHistogram();
163         xbin = newval;
164     }
165     void SteppingAction::setYBin(G4double newval){
166         while(lock){}
167         emptyHistogram();
168         ybin = newval;
169     }
170     void SteppingAction::setZBin(G4double newval){
171         while(lock){}
172         emptyHistogram();
173         zbin = newval;
174     }
175     void SteppingAction::setFilePath(G4String newval){
176         while(lock){}
177         if(top_thread){
178             file_writer.flush();
179             file_writer.close();
180             file_writer.open(newval.c_str(),std::ios::out|std::
                    ios::app);
181         }
```

```
182 }
183 G4double SteppingAction::getXBin(){
184     return xbin;
185 }
186 G4double SteppingAction::getYBin(){
187     return ybin;
188 }
189 G4double SteppingAction::getZBin(){
190     return zbin;
191 }
192
193 G4int SteppingAction::getIndexInArray(double val, std::
      vector<double>* anarray, int startind, int endind){
194
195     //     printf("val = %f, start = %f, end = %f\n", val,
          anarray[startind], anarray[endind]);
196     //     fflush(stdout);
197
198     int ind = (endind + startind)/2;
199
200     if(startind == ind || endind == ind){
201         return ind;
202     }else if((*anarray)[ind] > val){
203         return getIndexInArray(val, anarray, startind, ind);
204     }else{
205         return getIndexInArray(val, anarray, ind, endind);
206     }
207 }
```

A.3 Post-Processing

This Matlab function takes a path to the output from a Geant4 simulation, along with some basic information about the simulation, such as the distance from the foil to the sensor, and desired information about the sensor, such as the angle of acceptance, the number and duration of time bins, and so on, and uses it to process the Geant4 output into a collection of measured neutron energies, which are returned as a Matlab array for plotting, histogramming, etc.

```
1  function [measured_ens] = ProcessGeantRecoilSim(afile,
       ion_mass_amu, sensor_spacing_m, pixel_size_m,
       angle_of_acceptance_rads, time_bin_s, num_time_bins,
       neut_en)
2
3   samples = importdata(afile);
4   samples = samples.data;
5   %samples(:,1) = samples(:,1)/1E9;
6   %samples(:,2) = samples(:,2)/1E6;
7   %samples(:,3) = samples(:,3)/1E6;
8
9  mass_kg = ion_mass_amu*1.66053892E-27;
10
11 time = samples(:,1);% - 1.0E-9;
12 rad_pos = sqrt(samples(:,2).^2 + samples(:,3).^2);
13
14 time_meas = (floor(time./time_bin_s))*time_bin_s +
       time_bin_s/2;
15 min_time = min(time_meas(find(~isnan(time_meas))));
```

```
16 inds = find(time_meas >= min_time & time_meas <= min_time +
       num_time_bins*time_bin_s);
17
18 pos_meas = (floor(rad_pos ./pixel_size_m))*pixel_size_m +
       pixel_size_m/2;
19 en_meas = 0.5*mass_kg*(sqrt(pos_meas.^2 + sensor_spacing_m
       .^2)./time_meas).^2;
20
21 ang_meas = atan(pos_meas/sensor_spacing_m);
22
23 time_meas = time_meas(inds);
24 pos_meas = pos_meas(inds);
25 en_meas = en_meas(inds);
26 ang_meas = ang_meas(inds);
27
28 inds = find(ang_meas < angle_of_acceptance_rads);
29
30 time_meas = time_meas(inds);
31 numel(unique(time_meas))
32 pos_meas = pos_meas(inds);
33 en_meas = en_meas(inds);
34 ang_meas = ang_meas(inds);
35
36 measured_ens = en_meas .*1./ion_mass_amu.*(1 + ion_mass_amu)
       .^2/4./sin((pi-2*ang_meas)/2).^2;
37 measured_ens = measured_ens * 6.24150934e12;
```

References

[1] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytracek, G. Cooperman, G. Cosmo, P. Degtyarenko, A. Dell'Acqua, G. Depaola, D. Dietrich, R. Enami, A. Feliciello, C. Ferguson, H. Fesefeldt, G. Folger, F. Foppiano, A. Forti, S. Garelli, S. Giani, R. Giannitrapani, D. Gibin, J.J. Gmez Cadenas, I. Gonzlez, G. Gracia Abril, G. Greeniaus, W. Greiner, V. Grichine, A. Grossheim, S. Guatelli, P. Gumplinger, R. Hamatsu, K. Hashimoto, H. Hasui, A. Heikkinen, A. Howard, V. Ivanchenko, A. Johnson, F.W. Jones, J. Kallenbach, N. Kanaya, M. Kawabata, Y. Kawabata, M. Kawaguti, S. Kelner, P. Kent, A. Kimura, T. Kodama, R. Kokoulin, M. Kossov, H. Kurashige, E. Lamanna, T. Lampn, V. Lara, V. Lefebure, F. Lei, M. Liendl, W. Lockman, F. Longo, S. Magni, M. Maire, E. Medernach, K. Minamimoto, P. Mora de Freitas, Y. Morita, K. Murakami, M. Nagamatu, R. Nartallo, P. Nieminen, T. Nishimura, K. Ohtsubo, M. Okamura, S. O'Neale, Y. Oohata, K. Paech, J. Perl, A. Pfeiffer, M.G. Pia, F. Ranjard, A. Rybin, S. Sadilov, E. Di Salvo, G. Santin, T. Sasaki, N. Savvas, Y. Sawada, S. Scherer, S. Sei, V. Sirotenko, D. Smith, N. Starkov, H. Stoecker, J. Sulkimo, M. Takahata, S. Tanaka, E. Tcherniaev, E. Safai Tehrani, M. Tropeano, P. Truscott, H. Uno, L. Urban, P. Urban, M. Verderi, A. Walkden, W. Wander, H. Weber, J.P. Wellisch, T. Wenaus, D.C. Williams, D. Wright, T. Yamada, H. Yoshida, and D. Zschiesche, *Geant4—a simulation toolkit*, Nuclear Instruments and

Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **506** (2003July), 250–303.

[2] J. Apostolakis, S.Giani1, M. Maire, P.Nieminen, M.G. Pia, and L. Urban, *Geant4 low energy electromagnetic models for electrons and photons*, CERN-OPEN-XX **99** (99Aug).

[3] Thierry Beck, Cline Zuber, Damien Aubert, Clment Chollet, Patrick Brunel, Stphan Huelvan, Christophe Brisset, and Henri-Patrick Jacquet, *Recent advances in the development of x-ray cameras inserted inside a pressurized box for lmj plasma diagnostics*, IEEE TRANSACTIONS ON PLASMA SCIENCE **38** (2010Oct), 2867–2872.

[4] D. T. Casey, J. A. Frenje, M. Gatu Johnson, F. H. Seguin, C. K. Li, R. D. Petrasso, V. Yu. Glebov, J. Katz, J. Magoon, D. D. Meyerhofer, T. C. Sangster, M. Shoup, J. Ulreich, R. C. Ashabranner, R. M. Bionta, A. C. Carpenter, B. Felker, H. Y. Khater, S. LePape, A. MacKinnon, M. A. McKernan, M. Moran, J. R. Rygg, R. Zacharias M. F. Yeoman, R. J. Leeper, K. Fletcher, M. Farrell, D. Jasion, J. Kilkenny, and R. Paguio, *The magnetic recoil spectrometer for measurements of the absolute neutron spectrum at omega and the nif*, Review of Scientific Instruments **84** (2013).

[5] M.B. Chadwick, M. Herman, P. Obloinsk, M.E. Dunn, Y. Danon, A.C. Kahler, D.L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, R. Brewer, D.A. Brown, R. Capote, A.D. Carlson, Y.S. Cho, H. Derrien, K. Guber, G.M. Hale, S. Hoblit, S. Holloway, T.D. Johnson, T. Kawano, B.C. Kiedrowski, H. Kim, S. Kunieda, N.M. Larson, L. Leal, J.P. Lestone, R.C. Little, E.A. McCutchan, R.E. MacFarlane, M. MacInnes, C.M. Mattoon, R.D. McKnight, S.F. Mughabghab, G.P.A. Nobre, G. Palmiotti, A. Palumbo, M.T. Pigni, V.G. Pronyaev, R.O. Sayer, A.A. Sonzogni, N.C. Summers, P. Talou, I.J. Thompson, A. Trkov, R.L. Vogt, S.C. van der Marck, A. Wallner, M.C. White, D. Wiarda, and P.G. Young, *Endf/b-vii.1: Nuclear data for science and technology: Cross sections, covariances, fission product yields and decay data*, Nucl. Data Sheets **112** (2011).

[6] L. Claus, L. Fang, R. Kay, M. Kimmel, J. Long, G. Robertson, M. Sanchez, J. Stahoviak, D. Trotter, and J. L. Porter, *An overview of the ultrafast x-ray imager (uxi) program at sandia labs*, Proceedings of the spie, 2015Aug.

[7] Z. Li, *Radiation damage effects in si materials and detectors and rad-hard si detectors for slhc*, Pixel 2008 international workshop, Unknown Month 2008Sep. 23.

[8] E. Oberla, J.-F. Genat, H. Grabas, H. Frisch, K. Nishimura, and G. Varner, *A 15 gsa/s, 1.5 ghz bandwidth waveform digitizing asic*, arXiv:1309.4397.

[9] C. Ruiz, 2015.

[10] E. Andersson Sundn, H. Sjstrand, S. Conroy, G. Ericsson, M. Gatu Johnson, L. Giacomelli, C. Hellesen, A. Hjalmarsson, E. Ronchi, M. Weiszflog, J. Kllne, G. Gorini, M. Tardocchi, A. Combo, N. Cruz, A. Batista, R. Pereira, R. Fortuna, J. Sousa, and S. Popovichev, *The thin-foil magnetic proton recoil neutron spectrometer mpru at jet*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **610** (November 2009), 682–699.

[11] James E. Turner, *Atoms, radiation, and radiation protection*, 3rd ed., WILEY-VHC, 127 Windham Road, Oak Ridge, TN 37830, 2007.

[12] J.F. Ziegler, M.D. Ziegler, and J.P. Biersack, *Srim - the stopping and range of ions in matter (2010)*, Nuclear Instruments and Methods in Physics Research, Section B: Beam Interactions with Materials and Atoms **268** (June 2010), 1818–1823.