Security Applications of Novel Neutron Sources

Simon Albright

Masters of Physics (Lancaster University)

University of HUDDERSFIELD
Inspiring tomorrow's professionals

Physics

International Institute for Accelerator Applications

University of Huddersfield

January 4, 2016

A thesis submitted to the University of Huddersfield for the degree of

Doctor of Philosophy in the School of Applied Sciences

Abstract

The smuggling of illicit goods poses a significant threat to the safety, security and economy of all nations. Undeclared black market goods, illegal narcotics and weapons are all threats that could ideally be prevented from crossing national borders. At present cargo interrogation is primarily performed using X-rays, which can be defeated by effective shielding and disguising of objects. Neutron interrogation offers an additional line of defence against smuggling, and there are a number of techniques available, which are discussed in this thesis.

In this thesis a review of the limitations of current cargo interrogation technology is given. Current technology has limitations, and these are considered. In preparation of this thesis Monte-Carlo transport codes MCNPX and Geant4 were used as well as nuclear inventory code EASY-II, and a description of their key features is given.

The possible methods of interrogating cargo with neutrons is discussed. Cargo can be interrogated with a range of neutron spectra, and either the neutrons or the produced gammas can be used. The use of techniques based on detecting neutrons or gammas is discussed, and simulations of gamma production by fast inelastic neutron scattering are presented. This is followed by a review of the principles

of compound nucleus based neutron sources. The produced neutron spectra and the decay isotopes are both important considerations, and the results of possible combinations of target and projectile are given.

Use of deuterons to produce neutrons through compound nucleus reactions has potential, due to the high Q of some reactions. If deuterons are used there is also a possibility of dissociation, if kinetic energies above the binding energy are used. At present deuteron dissociation cannot be simulated in Geant4 or MCNPX. Two new models of deuteron dissociation, one high and one low precision, have been developed for inclusion in Geant4. The physics and operation of these models is discussed and comparison with experimental data is presented.

When interrogating cargo with neutrons it is unavoidable that some level of activation will occur. In particular the activation of food is of significant concern due to the exposure caused by ingestion. To date there has been little investigation of the activation of cargo under neutron interrogation. By using up to date nuclear data libraries and numerical techniques it was possible to extend early work in this field. In addition it is claimed in literature that ^{24}Na is the only isotope of concern, this is shown to only be valid for certain combinations of food composition and irradiating energy.

For Louise

Acknowledgements

I would like to thank:

- Louise for her patience, encouragement and reassuring me that there is a light at the end of the tunnel when the inside could get no blacker
- My family for giving me a life time of support and motivation
- Professor R. Seviour for giving me this opportunity, for challenging me and helping me push my limits
- All of my colleagues in the International Institute for Accelerator Applications for their support and valuable discussion, advice and wisdom
- The Science and Technology Facilities Council for supporting my PhD under grant number ST/I00598X/1 and Siemens PLC for their support

Declaration

This thesis is my own work and no portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification at this or any other institute of learning.

"I have no special talent. I am only passionately curious"

- Albert Einstein

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

There is increasing international interest in the use of neutrons for cargo interrogation. This thesis examines the implications of neutron interrogation in the context of activating materials as well as methods of producing neutrons.

Approximately 90%[1] to 95%[2] of all freight is transported by sea. A large port like Felixstowe can have 3 million or more containers pass through each year, where each container will be approximately 2.4 m wide, 2.6 m high and either 6.1 m or 12.2 m long. The interrogation of containers is essential; for comparison of declared and actual contents, to prevent black-market goods crossing national borders, and to search for dangerous contraband such as narcotics and explosives. There are a variety of ways currently available to provide neutron beams, these are discussed in section 1.3. One study showed that whilst 95% of cargo is transported by sea only 10% of this is inspected to ensure the declared and actual contents are the same [2], demonstrating the need for fast, cost effective techniques for

Figure 1.1: The Sovereign Maersk container ship, capable of carrying approximately 8000 Twenty Foot Equivalent Units[3].

interrogating cargo containers. Figure 1.1 shows the Maersk Line container ship capable of carrying approximately 8000 Twenty Foot Equivalent Units with a total weight of approximately $100,000$ tons.

1.2 Security

Identification of threat materials in containers currently relies on a combination of intelligence, X-ray interrogation and manual searches. Single energy X-ray interrogation measures the attenuation of a beam of X-rays along an axis of the interrogated volume. The signal strength is given by

$$ln\left(\frac{I_c}{I_0}\right) \propto \mu_c t_c, \tag{1.1}$$

where I_c is the detected flux, I_0 is the flux through an equivalent thickness of air, μ_c is the attenuation coefficient of the cargo and t_c is the thickness of the cargo [4].

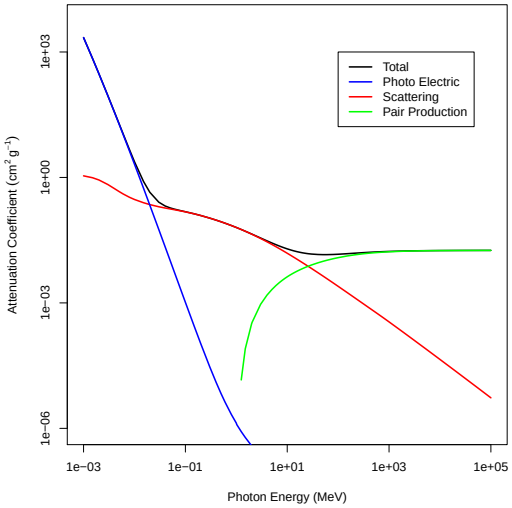
The signal strength calculated by equation 1.1 provides a measurement of the line integral of the attenuation by cargo between source and detector. Variations in attenuation along the line of integration due to smuggled items results in a change in the signal given by

$$ln\left(\frac{I_{c+t}}{I_0}\right) \propto \mu_c(t_c - t_t) + \mu_t t_t, \tag{1.2}$$

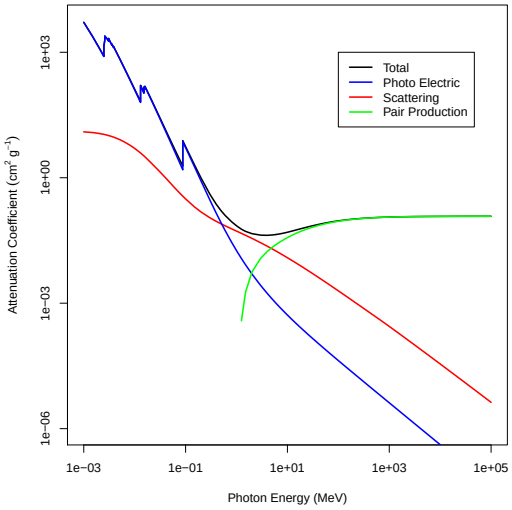
where I_{c+t} is the detected flux after cargo partially replaced by contraband, t_t is the thickness of the contraband and μ_t is the attenuation coefficient of the contraband [4].

The image provided by single energy X-rays is two dimensional and objects can initially be identified by image recognition software; however, it is required for a human operative to make the final decision about whether an item is contraband [5]. Due to the simplicity of single energy X-ray technology it cannot be used to distinguish between a small high density object (μ_c high, t_c low) and a large low density object (μ_c low, t_c high) making it easier to shield or disguise contraband.

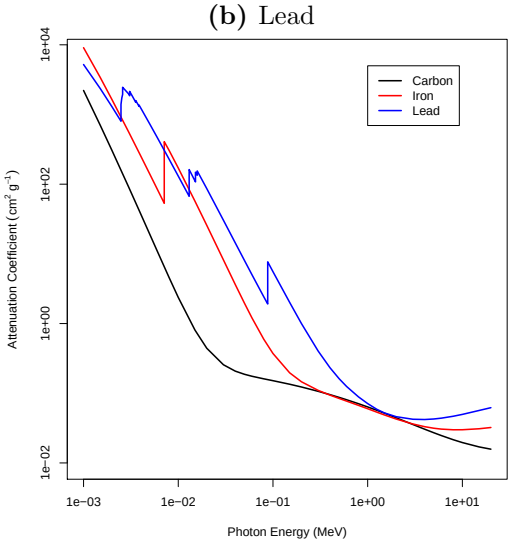
The threat detection capability of X-ray interrogation can be enhanced through dual-energy interrogation. Figures 1.2a and 1.2b show the photon attenuation coefficients for carbon and lead respectively, with the main causes of attenuation shown. At low energies, 0.1 MeV in carbon and 1 MeV in lead, the attenuation is dominated by the photo-electric effect. At high energies, above about 10 MeV, the attenuation is dominated by pair production. In the intermediate region Compton scattering dominates the attenuation. Comparisons of the attenuation coefficients of carbon, iron and lead normalised by density are shown in figure 1.2c.



(a) Carbon



(b) Lead



(c) Group

Figure 1.2: Causes of γ attenuation for carbon and lead and attenuation coefficients for carbon, iron and lead [6].

From figure 1.2c it can be seen that any two photon energies sufficiently far apart will have significantly different attenuation ratios for different materials. When using dual energy interrogation the ratios of the attenuation coefficients is given by

$$R = \frac{\mu_2}{\mu_1} = \frac{ln\left(\frac{I_{2c}}{I_2}\right)}{ln\left(\frac{I_{1c}}{I_1}\right)}, \tag{1.3}$$

R is the ratio of the attenuation coefficients, μ_i is the attenuation coefficient at photon energy i, I_{ic} is the detected flux transmitted through cargo at energy i and I_i is the detected flux through air at energy i [7]. From equation 1.3 it can be seen that for a given material, R can be known independently of the composition and thickness of the volume. For example carbon, iron and lead irradiated at 1 MeV and 10 MeV would give R values of 3.24, 2.00 and 1.42 respectively.

In pure elements μ_i is proportional to Z and so measurements of R make it possible to infer Z. In the case of compounds μ is proportional to the effective Z (Z_{eff}), which can then be used to infer the composition of an unknown material. For a given compound comprised of n elements Z_{eff} can be calculated by

$$Z_{eff} = (\Sigma_{i=1}^{n} a_i Z_i^p)^{1/p}, \tag{1.4}$$

where a_i is the fractional number of electrons per gram of element i, Z_i is the atomic number of element i and p is an empirical constant with a strong energy dependence [4]. Having measured R, and therefore Z_{eff}, for the interrogated volume it is possible to infer if an object is predominantly organic, inorganic or metallic. For a selection of benign and threat materials the density, Z_{eff}, and

compositions are shown in table 1.1. In the case of table 1.1 the value of Z_{eff} is given at energies where attenuation is dominated by the photo-electric effect.

Material	Density ($\frac{g}{cm^3}$)	Z_{eff}	% H	% C	% N	% O	% Other
Tovex	1.2	8.2	9	12	31	44	4
Semtex-H	1.5	7.4	28	16	20	36	0
RDX	1.8	7.3	29	24	29	29	0
TNT	1.7	7.1	2	33	14	29	0
PETN	1.8	7.4	28	17	14	41	0
Machine Parts	7.9	26	0	0	0	0	100
Fresh Fish	1.0	6.6	61	25	2	12	0
Flowers	0.25	7.0	6	44	0	49	0
Clothing	1.0	22	47	29	5	16	0
Magazines	0.8	7.4	51	26	0	23	0
Electronics	0.2	35	34	26	2	3	35

Table 1.1: The density, Z_{eff} and composition of 5 high explosives and 5 benign materials [4].

Figure 1.3 shows a comparison of monochromatic and dual energy X-ray interrogation of a recreation of the radio used in the Lockerbie bombing. The image on the left shows a traditional monochromatic X-rays image, with Dual Energy X-rays used to produce the right hand image. The explosive component in figure 1.3 cannot be identified in the monochromatic image; however, the false colour used to represent R in the dual energy system enables it to be identified. From figure 1.3 it is clear that Dual Energy X-rays can provide greatly enhanced threat detection; however, metallic bodies are still able to shield contraband and it is still possible to disguise threat objects.

In some instances Dual Energy X-rays are not able to identify illicit items. X-ray backscatter imaging uses the X-rays scattered from the surface and subsurface of an interrogated volume to image the first few millimetres. Compton scattered X-rays have a possibility of returning approximately towards their

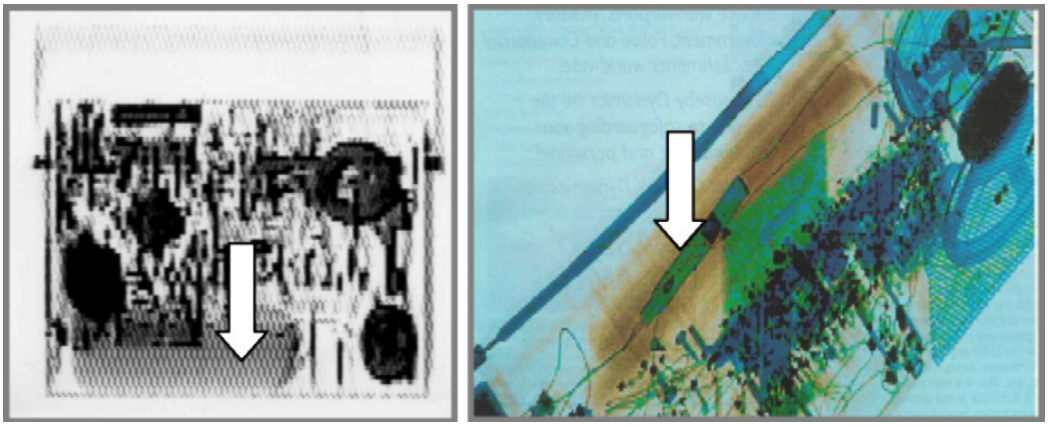


Figure 1.3: Comparison of monochromatic and dual energy X-ray interrogation of a recreation of the radio used in the Lockerbie bombing. In the monochromatic X-ray (left) the explosive, indicated by an arrow, is not identifiable however in the false colour of the dual energy image (right) it is [8].

source with cross section proportional to the electron density [9]. Irradiating a volume with a pencil beam of X-rays enables the back scattered fraction to be measured, for high Z materials this fraction will be greater, enabling an image of the sub-surface to be constructed. An example where Dual Energy systems may not be able to detect threat objects, whereas back scattering systems would succeed, is shown in figure 1.4, the image on the left is a Dual Energy X-ray scan, the right hand image is produced with backscattered X-rays.

Whilst Dual Energy and backscatter X-ray techniques can provide good detection of illicit materials they are still limited. Due to the limitations of X-ray techniques there is an increasing interest in the use of neutrons. The attenuation of neutrons has a highly non-linear dependence on Z, unlike the near linear dependence of X-rays. In addition neutrons are able to propagate through very high-Z materials such as Pb which would traditionally be used to shield against X-rays. The use of neutrons for security is discussed extensively in Chapter 3.

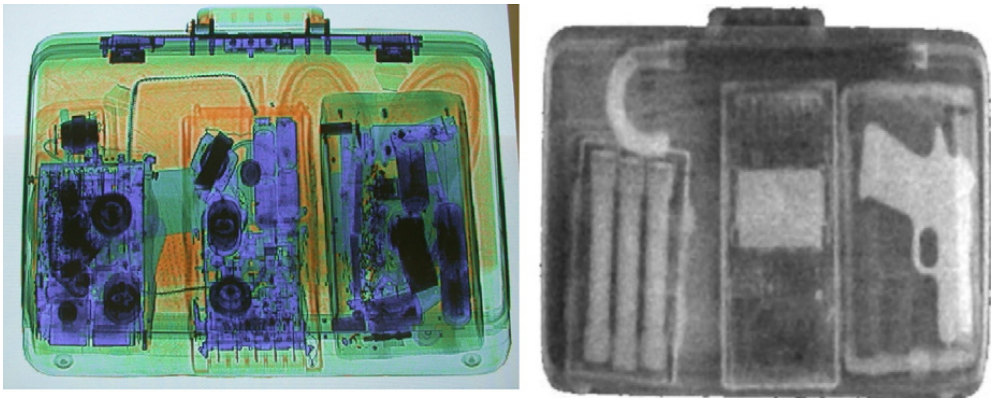


Figure 1.4: Comparison of dual energy X-ray and backscattered X-ray of a bag containing multiple threat objects [8].

Along with conventional explosives and weapons in carry on luggage and containers there is also a need to identify Special Nuclear Material (SNM). The focus of this thesis is on conventional explosives and narcotics; however, the same techniques will detect SNM and many of the issues are the same, therefore a brief discussion of them is suitable. As with conventional explosives it is possible to detect SNM with standard X-ray imaging; however, shielding issues still apply. The use of high energy X-rays can stimulate photo-nuclear interactions, particularly photo-fission, which has characteristic emissions for different fissionable and fissile nuclei [10].

Due to the photo-fission component when irradiating SNM with high energy photons there will tend to be a higher neutron yield than for non-fissionable materials. Jones et al [11] showed that under 10 MeV Bremsstrahlung photon irradiation SNM yielded substantially more neutrons and photons than normal material as a result of the photo-fission component.

In order for a cargo interrogation system to be used several criteria must be met [12]:

- Efficiency

  The ability of any cargo interrogation technique to detect illicit items is not sufficient to make it a viable system, it must exceed the detection capabilities of current technology, a requirement neutron interrogation has been shown to meet [13].

- Rapidity

  Highly effective threat detection is only of use if the flow of goods is not significantly interrupted. Neutron interrogation can take a significant amount of time; the EURITRACK system requires 10 minutes to interrogate each region of interest [14]. The potential for long interrogation times will not prevent the use of neutron interrogation as it is already standard to use techniques of gradually increasing complexity when items are identified as suspicious [15]. If neutron interrogation requires an extended time period using it as the last in a chain of interrogation techniques would ensure it was still a viable option.

- Ease of maintenance and operation

  Any system used for cargo interrogation will be intended for use and maintenance by technicians in non-laboratory environments and so must be as simple as possible both to operate and maintain.

- Safe

  Protecting operators and the public from side-effects of cargo interrogation is essential, and is more difficult with neutrons than X-rays. In potential industrial systems there has been research performed to identify ways to

design neutron interrogation systems so that exposure to radiation is minimised [16].

- Cost effective

  The cost of any interrogation system is inevitably a significant factor. Whilst concerns over contraband detection are significant it is still necessary to minimise the cost of a system, and if it is not cost effective compared to current systems it will not be used.

These criteria have particular implications for neutron interrogation systems. Neutrons require significantly different shielding to X-rays and γs potentially necessitating more complex shielding. The source must be as simple as possible, which would imply a fission source; however, the threat presented by the long-lived fission products would make a fission source unsuitable. Sealed tube fusion source using deuteron beams on Tritiated or Deuterated targets are very simple; however, the presence or production of tritium results in very stringent legislation [17, 18], which when combined with the relatively short life time prevents them being viable for mass deployment.

1.3 Neutron Sources

There are a variety of ways of producing neutrons, which have applications in different areas depending on the neutron flux required and the type of infrastructure available. The main reactions can be broadly grouped into nuclear decay, high energy hadronic interactions, low energy hadronic interactions, and photo-nuclear interactions, the most significant reactions are detailed in this section.

Nuclear decay by spontaneous fission is typically associated with the emission of one or more neutrons over a range of energies. A commonly used commercial fission source is ^{252}Cf. This isotope is favoured due to the relatively high spontaneous fission branching ratio of approximately 3% coupled with a practical half life of approximately 2.6 years. The spontaneous fission branching ratio of ^{254}Cf is approximately 99.7% but the half-life is only 60 days making it unsuitable for most applications.

As part of the fission process neutrons are released covering a broad spectrum of energies. In the case of ^{252}Cf the spectrum has a maximum energy of 13 MeV with a mean energy of 2.5 MeV and a modal energy of 1 MeV [19].

As fission sources use nuclear decay they cannot be controlled but provide a near uniform flux, with known half-life. Being unable to turn off the neutron source necessitates heavy shielding to minimise unwanted neutron emission. After the source has reached the end of its useful life there will be large amounts of long-lived fission products remaining, which will necessitate stringent radiation controls. Other radioisotope sources can also be used, one example is $AmBe$, which uses αs from americium to stimulate neutron emission from beryllium.

Spallation neutron sources use high energy hadrons, typically protons, with energy on the order of 1 GeV to fragment nuclei, causing the emission of nucleons. Spallation can in principle occur on any target and with any projectile of sufficient energy to stimulate a nucleon to escape the nucleus; however, high-Z targets are preferred as they can typically take greater beam power and produce more neutrons.

Spallation of a nucleus involves three stages, first an intra-nuclear cascade transfers energy from the projectile to individual nucleons, followed by the tran-

sition stage where energy is distributed throughout the nucleus and finally an evaporation stage where the excitation energy leaves [20]. In all three stages nucleons are ejected from the nucleus, and in a spallation target the ejectiles may strike other nuclei and spallate them in an internuclear cascade.

A target bombarded by low energy hadrons can form a Compound Nucleus (CN). A CN is an excited state produced by the capture of another particle, which could be a proton, neutron, photon or any nucleus with $A > 1$. The highly excited state will decay through the emission of one or more nucleons or photons. CN reactions take the form shown in equation 1.5, typically expressed as shown in equation 1.6.

$$A + x \rightarrow y + B \tag{1.5}$$

$$A(x, y)B \tag{1.6}$$

In equations 1.5 and 1.6 A is the target nucleus, x is the projectile, y is (are) the ejectile(s) and B is the decay nucleus. A typical example of a CN reaction is the fusion of deuterium and tritium, resulting in a helium nucleus and the emission of a neutron, expressed as shown in equation 1.7

$$T(d, n)^4He. \tag{1.7}$$

For a neutron source the main parameters of interest for a given reaction are the cross-section (σ) and the Q. The Q is the mass difference of the initial

and final states and can be easily calculated for any reaction, the example of $T(d,n)^4He$ is shown in equation 1.8

$$D(1875.6MeV)+T(2808.9MeV) \rightarrow \alpha(3727.4MeV)+n(939.6MeV)+17.6MeV, \quad (1.8)$$

where the mass energy of each component is given and the remaining 17.6 MeV is the Q. The energy released in the reaction, the Q, is divided between the α and the neutron giving them 3.5 MeV and 14.1 MeV respectively. In principle any projectile incident on any target could stimulate the emission of any ejectile(s) if it is energetically allowed.

CN neutron sources have been both proposed and used for a range of applications. In medicine they can be used for fast neutron therapy [21] and Boron Neutron Capture Therapy [22]. In astrophysics they can produce neutrons with characteristics similar to those responsible for the S-process in stellar nucleosynthesis [23]. For neutron scattering experiments using cold neutrons there may be situations that would benefit from a compact source [24]. The continued development of fusion power plants necessitates a more detailed understanding of neutron reaction cross-sections than is currently available, a need that can be met by CN neutron sources [25]. The detection and clearing of land mines and unexploded ordinance in combat zones using fast neutron irradiation can be effective [26]. There is also significant interest in the use of neutrons for cargo interrogation [27, 28, 29].

1.4 Neutron Induced Activation of Food

When irradiating material with neutrons a level of activation is unavoidable. Activation of food is a particular concern as it will be ingested, and some molecules may persist in the body for an extended period of time. Chapters 6 and 7 consider the activation of irradiated foods and the isotopes produced.

Some previous work has been done to begin understanding the level of activation that could be seen in foods, most notably by Findlay et al [30]. The work presented in this thesis extends that in [30] in a number of ways.

The authors of [30] did not have the ability to calculate neutron spectra for every combination of food and energy used, instead a single spectrum for each energy was used and then scaled to suit the food under irradiation. In this work the neutron transport was performed for each food separately and the spectrum then used to calculate the nuclear inventory. In addition the inventory calculations in [30] used an older nuclear data library (EAF-2) and considered the spectrum after 10 cm of transport, whereas 90 cm was used in this thesis.

1.5 Overview

The chapters of this thesis cover the use and production of neutrons in security and the potential hazards posed by cargo activation. Chapter 2 gives a description of the numerical models used to undertake the work presented in this thesis. Chapter 3 covers the range of neutron interrogation techniques available and how they can be used to identify contraband. Chapter 4 is an in depth discussion of how compound nucleus reactions can be used to provide neutron beams. Chapter

5 describes a model of deuteron dissociation, which has been written for incorporation into the Geant4 simulation package. Chapter 6 presents work investigating the activity and ingestion doses of a range of foods under neutron interrogation. Chapter 7 is an analysis of the isotopic inventory of a selection of foods after neutron irradiation and covers the energy dependence of their production. Chapter 8 concludes the thesis with a discussion of significant points and future work.

Bibliography

[1] A. Bergantino, E. Musso, and F. Porcelli. Port management performance and contectual variables: Which relationship? methodological and empirical issues. *Research in Transportation Business and Management*, 8:39–49, 2013.

[2] G. Boghen, A. Donzella, V. Filippini, A. Fontana, M. Lunardon, S. Moretto, S. Pesente, and A. Zenoni. MCNP calculations for container inspection with tagged neutrons. *Nuclear Instruments and Methods in Physics Research B*, 241:831–834, 2005.

[3] Sovereign Maersk. `url:http://www.shipprofiles.com/design/Sovereign_Maersk.jpg`. Retrieved: 16/02/2014.

[4] R. Runkle, T. White, E. Miller, J. Caggiano, and B. Collins. Photon and neutron interrogation techniques for chemical explosives detection in air cargo: A critical review. *Nuclear Instruments and Methods in Physics Research A*, 603:510–528, 2009.

[5] S. Michel, M. Mendes, J. Ruiter, and G. Koomen. Increasing X-ray image interpretation competence of cargo security screeners. *International Journal of Industrial Ergonomics*, 44:551–560, 2014.

[6] J. Hubbell and S. Seltzer. Tables of X-ray mass attenuation coefficients and mass energy-absorption coefficients (version 1.4). url: http://physics.nist.gov/xaamdi, 2004. Retrieved: 12/12/2014, Originally published as NISTIR 5632, National Institute of Standards and Technology, Gaithersburg, MD (1995).

[7] Y. Liu, B. Sowerby, and J. Tickner. Comparison of neutron and high-energy x-ray dual-beam radiography for air cargo inspection. *Applied Radiation and Isotopes*, 66:463–473, 2008.

[8] H. Vogel. Search by x-rays applied technology. *European Journal of Radiology*, 63:227 – 236, 2007.

[9] G. F. Knoll. *Radiation Detection and Measurement*, chapter 2. John Wiley and Sons, inc., 4 edition.

[10] R. C. Runkle, D. L. Chichester, and S. J. Thompson. Rattling nucleons: New developments in active interrogation of special nuclear material. *Nuclear Instruments and Methods in Physics Research A*, (663):75–95, 2012.

[11] J. L. Jones et al. Detection of shielded nuclear material in a cargo container. *Nuclear Intruments and Methods in Physics Research A*, (562):1085–1088, 2006.

[12] A. Buffler and J. Tickner. Detecting contraband using neutrons: Challenges and future directions. *Radiation Measurements*, (45):1186–1192, 2010.

[13] A. Buffler. Contraband detection with fast neutrons. *Radiation Physics and Chemistry*, (71):853–861, 2004.

[14] C. Carasco et al. In-field tests of the euritrack tagged neutron inspection system. *Nuclear Instruments and Methods in Physics Research A*, (588):397–405, 2008.

[15] S. Singh and M. Singh. Explosives detection systems (EDS) for aviation security. *Signal Processing*, 83:31–55, 2003.

[16] A. Donzella, G. Bonomi, E. Giroletti, and A. Zenoni. Biological shielding assessment and dose rate calculation for a neutron inspection portal. *Radiation Physics and Chemistry*, (81):414–420, 2012.

[17] Ionising radiations regulations 1999. `http://www.legislation.gov.uk/uksi/1999/3232/contents/made`. Retrieved: 02/12/2015.

[18] High-activity sealed radioactive sources and orphan sources regulations 2005. `http://www.legislation.gov.uk/uksi/2005/2686/contents/made`. Retrieved: 02/12/2015.

[19] J. F. Dicello, W. Gross, and U. Kraljevic. Radiation quality of californium-252. *Physics in Medicine and Biology*, 17(3):345 – 355, 1972.

[20] G. S. Bauer. Physics and technology of spallation neutron sources. *Nuclear Instruments and Methods in Physics Research A*, 463:505 – 543, 2001.

[21] V. N. Kononov et al. Accelerator-based fast neutron sources for neutron therapy. *Nuclear Instruments and Methods in Physics Research A*, (564):525–531, 2006.

[22] H. Koivunoro et al. Bnct dose distribution in liver with epithermal dd and dt fusion-based neutron beams. *Applied Radiation and Isotopes*, (61):853859, 2004.

[23] M. Friedman et al. Simulation of the neutron spectrum from the ^{7}li(p,n) reaction with a liquid-lithium target at soreq applied research accelerator facility. *Nuclear Instruments and Methods in Physics Research A*, (698):117–126, 2013.

[24] F. Hiraga, T. Okazaki, and Y. Kiyanagi. Neutronic design on a small accelerator-based ^{7}li(p,n) neutron source for neutron scattering experiments. *Physics Procedia*, (26):97–107, 2012.

[25] M. Pillon, M. Angelone, M. Martone, and V. Rado. Characterization of the source neutrons produced by the frascati neutron generator. *Fusion Engineering and Design*, (28):683–688, 1995.

[26] M. Maucec and C. Rigollet. Monte carlo simulations to advance characterisation of landmines by pulsed fast/thermal neutron analysis. *Applied Radiation and Isotopes*, (61):35–42, 2004.

[27] J. M. Hall et al. The nuclear car wash: Neutron interrogation of cargo containers to detect hidden snm. *Nuclear Instruments and Methods in Physics Research B*, 261:337–340, 2007.

[28] E. L. Reber et al. Idaho explosives detection system. *Nuclear Instruments and Methods in Physics Research B*, (241):738–742, 2005.

[29] K. Bergaoui et al. Monte carlo simulation of explosive detection system based on a deuterium-deuterium (D-D) neutron generator. *Applied Radiation and Isotopes*, (94):118–124, 2014.

[30] D.J.S. Findlay, R.A. Forrest, and G.M. Smith. *Neutron-induced activation of food*. AEA Technology Industrial Technology, 1993. AEA-InTec-1051.

Chapter 2

Simulation Software

2.1 Introduction

This chapter introduces and explains the numerical simulations used to undertake the work presented this thesis. Two Monte-Carlo codes, Geant4 [1] and MCNPX [2], were used for particle transport and the nuclear inventory code Fispact-II was used for activation studies.

The Monte-Carlo technique is a method of finding solutions to complex probabilistic or deterministic problems [3]. The transport of particles through a geometry, and the subsequent interactions, are handled as a probabilistic system, typically within particle transport Monte-Carlo codes, e.g. MCNPX and Geant4.

2.2 Geant4

Geant4 was first developed for use in High Energy Physics but has now been extended for use in a broad range of environments including medical and space

physics. Geant4 provides the user with a C++ library with which simulations are constructed from a variety of default particles, geometries and interactions covering the majority of user requirements. Geant4 is designed to allow additional code to be readily incorporated by the user, such as new physics interactions, should they be required [4].

Geant4 transports particles along a track composed of a series of steps. Starting at a given location L_0 the first step takes a particle over step S_1 to the location of the next interaction L_1. Based on the interaction at L_1 a new step S_2 is defined and the process repeats. The length of a track is determined by the mean free path, or interaction length, λ, of a particle, given by equation 2.1;

$$\lambda = \left(\sum_{1}^{i} \left[n_i \times \sigma \left(Z_i, E \right) \right] \right)^{-1} \tag{2.1}$$

n_i is the amount of isotope i, $\sigma(Z_i, E)$ is the cross section for a given nuclide (Z_i) at the current energy (E) [5]. Having calculated the mean-free-path we then determine the length of step S_n, giving the location of the next interaction point L_n. The number of mean-free-paths (n_λ) travelled in step S_n is given by $n_\lambda = -ln(\eta)$ Where η is chosen from a uniformly distributed series of random numbers in the range $[0 \rightarrow 1]$.

There are a range of interactions that can happen along a step, such as an unstable particle decaying, and at the end of a step, such as elastic scattering. The interactions along a step, and at the end of a step, are also sampled probabilisticly from data tables and numerical models where applicable. When an interaction produces a new particle this will also be tracked through the geometry such that the last particle to be created is the first one to be tracked.

Calculations of steps and interactions proceed from the creation of a source particle until all particles have either escaped the geometry, lost too much energy to need tracking or have been lost through an interaction. All particles are tracked in this way and the interactions available are defined by the user. For most applications standard sets of interactions are available and included in the simulation as required. If no suitable interaction is available in the Geant4 code the user must create their own models.

The primary use of Geant4 covered in this thesis was the implementation of a numerical model of deuteron dissociation. Due to the nature of Geant4 it is ideally suited to the implementation of additional physics processes and interactions. In this thesis the addition of a new hadronic physics model, deuteron dissociation was required. Section 3.5 of reference [4] covers inclusion of new models in detail, with the necessary code shown in Appendix B.1. The operation of the deuteron dissociation code is described in detail in chapter 5.

2.3 MCNPX

The Monte-Carlo N-Particle eXtended (MCNPX) code is an extension of the MCNP code, the only significant difference being the ability to track a very large variety of particles, as opposed to the Neutrons, Photons and Electrons available in MCNP. MCNPX is Fortran-based and is run with the use of a single input file, called a deck, in which each line is referred to as a card, a legacy reference to the use of punch cards in the earliest versions of the code.

MCNPX has not been designed to have additional code incorporated as easily as Geant4; however, it requires less user input to start using. Each simulation requires a single input file used in a pre-compiled program, simplifying the development of a simulation but limiting the potential for altering the function.

The principle behind MCNPX is essentially the same as Geant4 and other Monte-Carlo codes. Particles are tracked through a geometry and random number generators combined with transport and interaction equations determines what happens[6].

In MCNPX the distance between collisions is given by

$$l = \frac{1}{\Sigma_t} ln(\xi), \tag{2.2}$$

where l is the distance travelled, Σ_t is the macroscopic cross-section for the current material and ξ is selected from a uniform random number distribution from $[0 \rightarrow 1]$. At the end of each step, as in Geant4, the interaction is selected from either data tables or numerical models using random number distributions.

2.4 Fispact-II

Fispact-II is a nuclear inventory code, within the EASY-II package [7], designed to enable simulation of the production and decay of radioisotopes in a given system. Fispact-II solves the rate equation for the production of isotope i from reactions involving isotope j,

$$\frac{dN_i}{dt} = \Sigma_j \left(\lambda_i^j + \sigma_i^j \phi^{int}(t) \right) N_j, \tag{2.3}$$

in which N_i is the amount of isotope i, λ_i^j is the rate of production of i from decay of j, σ_i^j is the cross-section for production of i from j, ϕ^{int} is the source particle flux and N_j is the number of nuclei of isotope j. A special case of equation 2.3 is when $i = j$, in this case λ_j^j is the total loss of i/j through all decay channels and σ_j^j is the total loss of i/j through all production channels.

A significant time saving is enabled in Fispact-II through the use of cross-section collapse. Cross-section collapse is the process of taking a weighted mean of a cross-section, where the weighting factor is given by the irradiating spectrum. The equation used for cross-section collapse is given by equation 2.4

$$\sigma_i^j = \frac{\Sigma_k \sigma_i^j(E_k)\phi_n(E_k)}{\phi_n(E_k)} \tag{2.4}$$

$\sigma_i^j(E_k)$ is the cross section to produce i from a reaction with j at energy bin E_k, and $\phi_n(E_k)$ is the flux of irradiating particles in energy bin E_k.

The production and subsequent decay of nuclei in Fispact-II is dependent upon the use of accurate values for the cross-section in each energy bin. Where it is available, evaluated nuclear data, such as that found in the ENDF and the JEFF files is used, for example ENDF/BVII.1 [8] or JEFF 3.1.1 [9]. There are many $A(n, x)B$ reactions without accurate cross-section measurements. In cases without available evaluated data the Talys Evaluated Nuclear Data Library (TENDL) is used [10].

The TENDL libraries enable up to 2424 isotopes to be simulated [7]. The cross-sections in TENDL are calculated using the Talys code [11] and therefore are reliant upon nuclear models rather than experimental data. The Talys code is predominantly very reliable and well benchmarked but there are instances where

the results are less accurate. For example comparing the TENDL $^7Li(p,n)$ cross-section with measured data shows a large discrepancy, which can be seen in figure 2.1

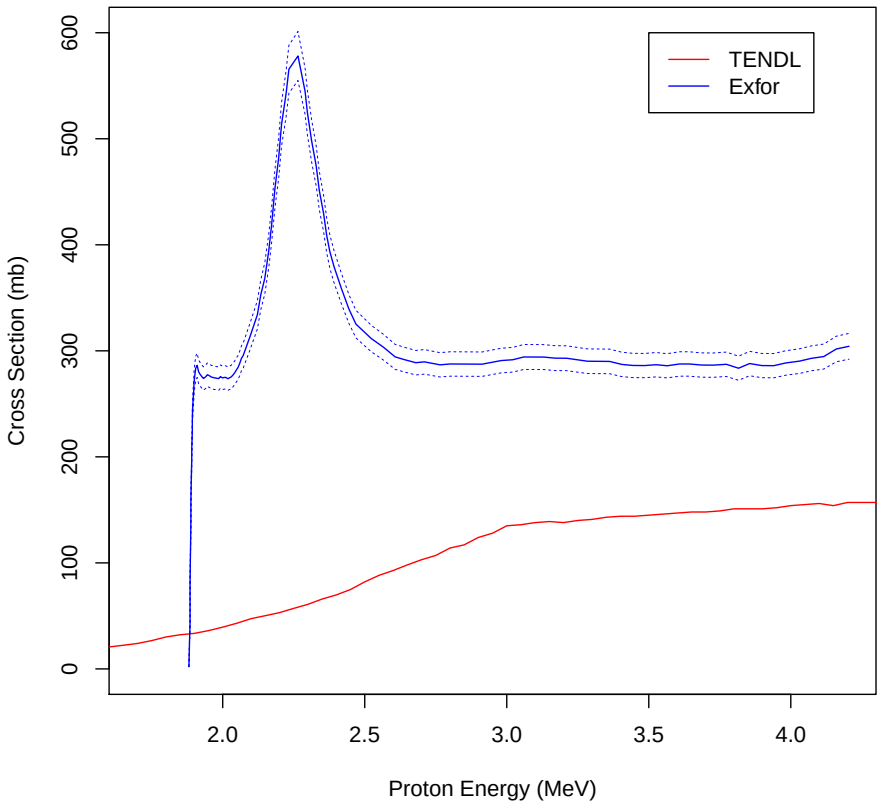


Figure 2.1: Cross-section of the $^7Li(p,n)$ reaction as given by TENDL [7] (red) and taken from the Exfor data base [12] (blue), the dashed lines indicate the experimental uncertainties.

Bibliography

[1] J. Allison et al. Geant4 developments and applications. *Nuclear Science, IEEE Transactions on*, 53(1):270–278, Feb 2006.

[2] Los Alamos National Laboratory. *MCNPX^{TM} User's Manual*, 2.6.0 edition.

[3] J. M. Hammersliey and D. C. Handscomb. *Monte Carlo Methods*, chapter 1. Methuen and Co Ltd, 1975.

[4] Geant4 Collaboration. *Geant4 User's Guide for Toolkit Developers*, version 9.6.0 edition, November 2012.

[5] Geant4 Collaboration. *Physics Reference Manual*, version 9.6.0 edition, November 2012.

[6] Los Alamos National Laboratory. *MCNP^{TM} A General Monte Carlo NParticle Transport Code*, 4c edition.

[7] J-C. Sublet, J. Eastwood, G. Morgan, A. Koning, and D. Rochman. EASY-II(12): a system for modelling of n, d, p, γ, α activation and transmutation processes. *Progress in Nuclear Science and Technology*, 4:349–353, 2014.

[8] M. B. Chadwick et al. ENDF/B-VII.1: Nuclear data for science and technology: Cross sections, covariances, fission product yields and decay data. *Nuclear Data Sheets*, 112(12):2887 – 2996.

[9] K-H. Schmidt, B. Jurado, and Charlotte Amouroux. *The JEFF-3.1.1 Nuclear Data Library*. OECD-NEA. NEA/DB/DOC(2014)1.

[10] A. J. Koning and D. Rochman. Modern nuclear data evaluation with the TALYS code system.

[11] A.J. Koning, S. Hilaire, and M. C. Duijvestijn. Talys: Comprehensive nuclear reaction modeling. In *Proceedings of the Inernational Conference on Nuclear Data for Science and Technology - ND2004*, volume 1154, 2004.

[12] K. K. Sekharan, H. Laumer, B. D. Kern, and F. Gabbard. A neutron detector for measurement of total neutron production cross sections. *Nuclear Instruments and Methods in Physics Research*, 133:253, 1976.

Chapter 3

Interrogation Techniques

3.1 Introduction

In Chapter 1 the principles of X-ray interrogation were discussed, and its inherent weaknesses due to the ease with which contraband can be disguised. The low inspection rate currently used at ports and border crossings, coupled with the relatively poor threat detection available with X-rays, motivates the search for alternative, better, methods of interrogation [1]. This chapter presents an overview of neutron security techniques, which have the potential to meet this need.

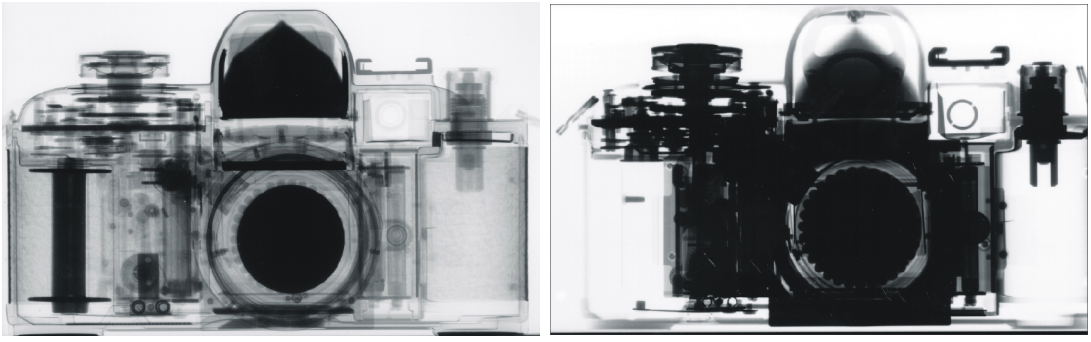
Neutron interrogation can be broadly categorised into neutron in/neutron out (NiNo) and neutron in/photon out (NiPo) techniques. NiNo techniques rely on measuring changes in the irradiating neutron beam, whereas NiPo techniques rely on measurements of the photons produced as a result of neutron interrogation. NiNo and NiPo techniques can be combined, with a range of potential bene-

fits over the two individually, into neutron in/neutron and photon out (NiNPo) techniques.

3.2 Neutron in/Neutron out

3.2.1 Neutron Transmission Imaging

Neutron Transmission Radiography (NTR), much like single-energy X-ray interrogation, gives a line integral of the attenuation between source and detector. As neutron attenuation is highly non-linear with atomic mass (A) and atomic number (Z), whereas X-ray attenuation is nearly linear with Z, an NTR image gives a complementary image to X-rays, which can be used to better identify threats. Figure 3.1 shows an SLR camera imaged with neutrons (3.1a) and X-rays (3.1b), the metallic components cause strong attenuation in the X-ray image whereas the organic cause strong attenuation of the neutrons.



(a) Neutron radiograph **(b)** X-ray radiograph

Figure 3.1: X-ray and neutron transmission images of an SLR camera [2].

A system using NTR and γ interrogation has been tested, and shown to be effective [3]. Combining the attenuation of neutrons by organics with the attenuation of γs by metals enables material identification. Using transmitted neutrons, rather than the stimulated γ emission, reduces the necessary flux and scan time compared to NiPo techniques.

The neutron interaction cross-section is strongly dependent on both the neutron energy and the target isotope. Irradiating a container with a broad energy source and looking for characteristic troughs in the transmission spectrum, corresponding with peaks in the interaction cross-section, can enable material identification [4]. The total interaction cross-sections of Hydrogen, Carbon, Nitrogen and Oxygen are shown in figure 3.2.

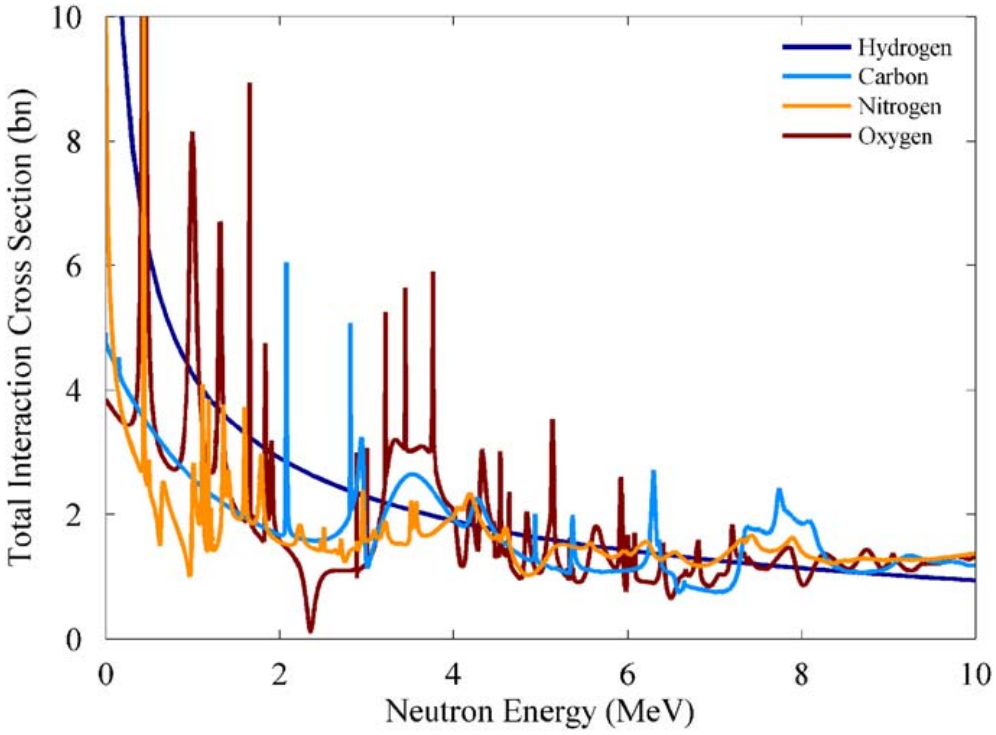


Figure 3.2: The energy dependence of the neutron interaction cross-section for Hydrogen, Carbon, Nitrogen and Oxygen [4].

3.2.2 Fast Neutron Scattering

Neutrons can scatter elastically from all nuclei, and inelastically from all nuclei except ^1H as it has no excited states, where inelastic scattering refers to a billiard ball like collision where some energy is used to excite the target nucleus. The energy lost by the neutron in both elastic and inelastic scattering is unique to the scattering nucleus, as is the cross-section at a given angle. The detection of neutrons at different scattering angles, combined with their Time-of-Flight (ToF) or another form of spectroscopy, can be used to identify materials [5].

Figure 3.3 shows the characteristic scattering from Hydrogen, Carbon, Nitrogen and Oxygen under 7.5 MeV neutron irradiation [4]. High energy final state neutrons correspond to elastic scattering whereas low energy neutrons are from inelastic scattering.

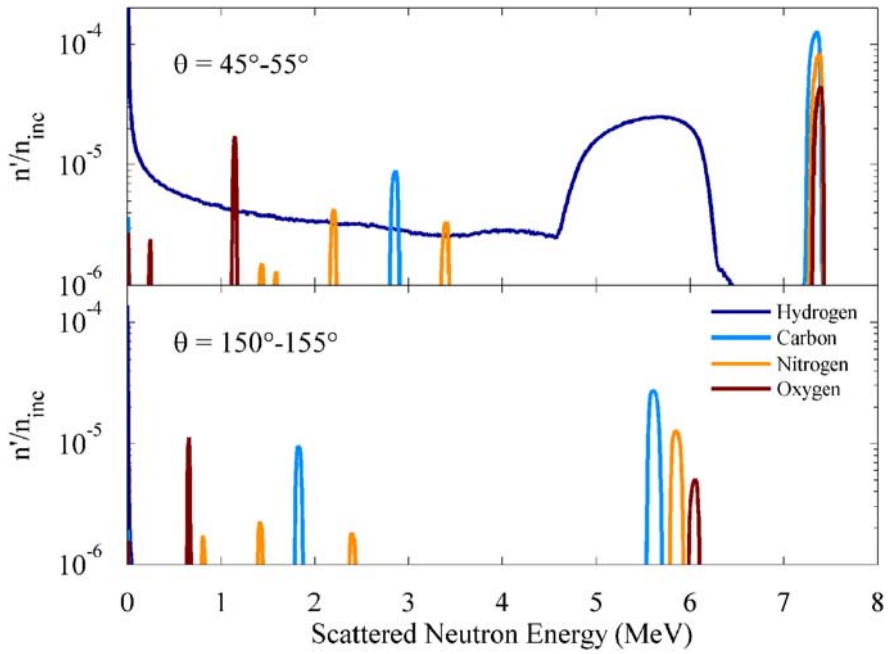


Figure 3.3: The energy dependence of the neutron interraction cross-section for Hydrogen, Carbon, Nitrogen and Oxygen [4].

3.3 Neutron in/Photon out

3.3.1 Thermal Neutron Capture

Low energy neutrons impinging on a target can be used for elemental identification through neutron activation. Suited to near-surface objects, neutron capture techniques use the photons emitted through neutron capture and subsequent decay of the daughter isotopes for material recognition [4]. The energies of the γs emitted in neutron capture are unique to the element interrogated allowing direct correlation between the γ spectrum and the composition.

Due to the use of thermal neutrons, this technique is best suited for near surface interrogation. One area where it has potential to be highly beneficial is detection of buried land mines [6]. Thermal Neutron Capture techniques would not be well-suited to large volume cargo interrogation due to the large volumes that need to be imaged.

3.3.2 Inelastic Fast Neutron Scattering

An alternative use of fast neutron scattering is as a NiPo technique. Fast neutrons stimulate the emission of prompt γs from materials with the photon energy unique to the element. When a fast neutron inelastically scatters from a nucleus a fraction of the energy is transferred to the nucleus, placing it in an excited state. For example figure 3.4 shows the allowable excited states of the ^{12}C nucleus. After being struck by a neutron with sufficient energy the nucleus may be excited to one of these energy levels before relaxing by γ emission. Many nuclei have more

complex energy levels than ^{12}C and may relax from a given excited state via the emission of one or more γs, transitioning through multiple excited states.

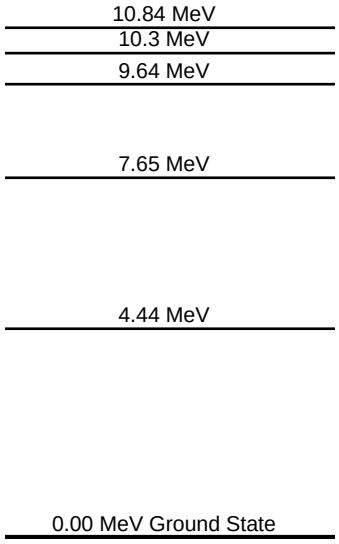


Figure 3.4: The first five excitation levels of ^{12}C [7].

Since the energy levels are unique to the isotope, detecting a specific energy, e.g. the 4.44 MeV excitation of ^{12}C, identifies the presence of that nucleus. As well as the energy of each state being unique to the target isotope the branching ratio of each γ is unique along with the cross-section for exciting a nucleus to the necessary excited state. Irradiating a material with sufficiently energetic neutrons will populate the available excited states of its constituent nuclei and detecting the resultant γs will enable the composition to be determined.

In particular the technique Pulsed Fast Neutron Analysis (PFNA) is growing in popularity and has been demonstrated to be effective [8]. Pulsing the neutron source enables Time-of-Flight (ToF) information to be included in the produced data. ToF allows depth information to be given therefore enabling a 3D breakdown of the container into voxels. Combining the ToF information with

the characteristic prompt γs therefore enables the materials within each voxel to be identified.

3.4 Neutron in/Neutron and Photon out

Combining NiNo and NiPo techniques into NiNPo techniques will be more effective than either one individually. The material identification of NiPo fast neutron techniques is ideal for identifying threats; however, NTR is faster and will identify volumes shielded against neutron interrogation more readily.

Preliminary research into NiNPo based on fast neutron scattering has been performed by Lehnert [9] using simulations of NiNPo with fast neutrons. Fast neutrons will scatter with characteristic energy and angular distributions, as discussed in 3.2.2, and stimulate the emission of characteristic γs, as discussed in section 3.3.2. Based on a highly simplified geometry of a very large sphere, 93.5 cm in radius, within cargo a number of *flags* were identified by Lehnert in the scattered neutron angle and energy distribution and the characteristic γ spectra potentially enabling threat detection.

3.5 Simulations

The Monte-Carlo code MCNPX was used to simulate neutrons with $E = 14$ MeV propagating through a selection of elements and compounds. The γ spectra produced by 14 MeV neutron irradiation of pure samples of C, N, O and Cl are shown in figure 3.5. The 4.44 MeV excitation level of ^{12}C is clearly visible in the spectra in figure 3.5.

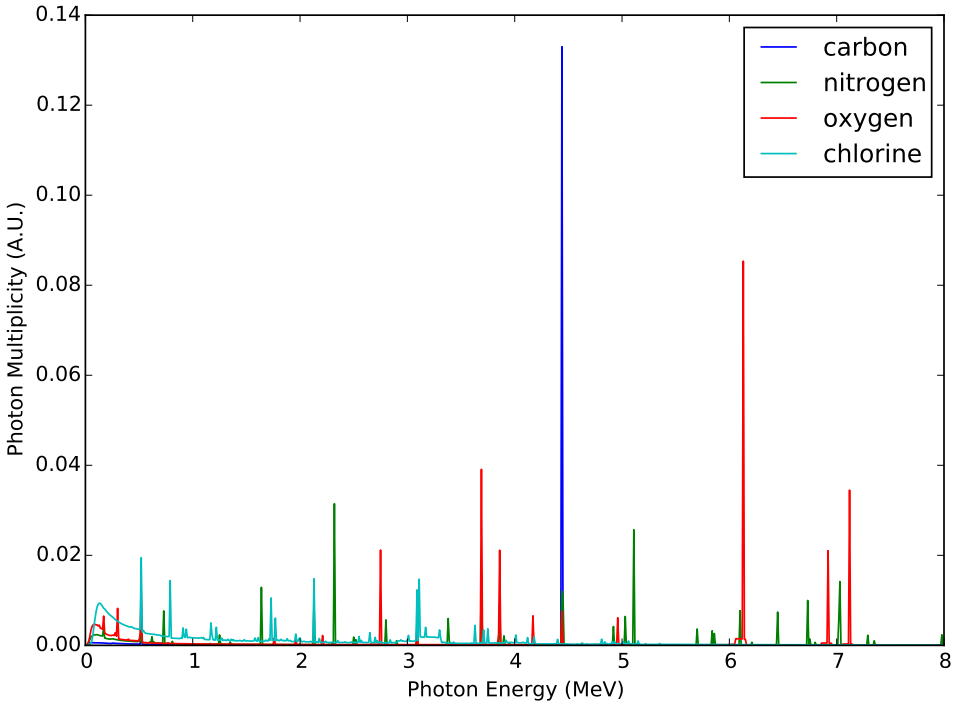


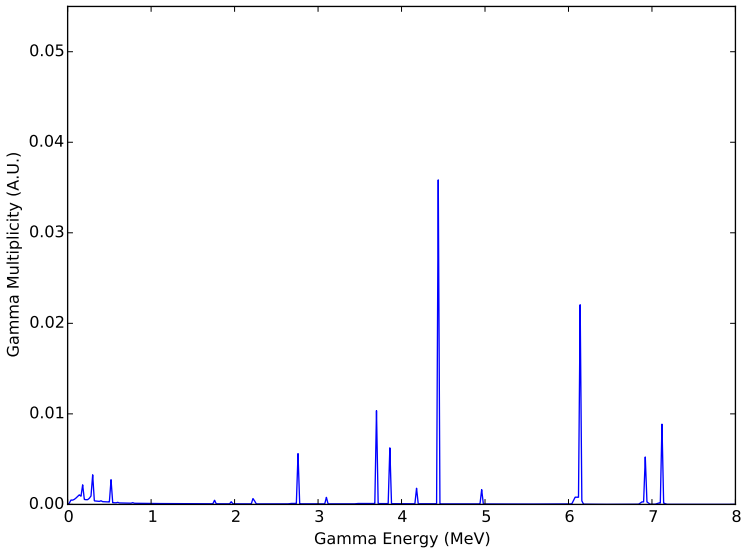
Figure 3.5: Simulated γ spectrum of C, N, O, Cl under 14 MeV neutron irradiation.

	Absolute molar composition					
	Hydrogen	Carbon	Nitrogen	Oxygen	Sodium	Chlorine
Cellulose	10	6	0	5	0	0
Latex	3	3	1	0	0	0
Nylon	6	4	1	2	0	0
Surfactant	29	18	0	3	1	1
Ethanol	6	2	0	1	0	0
Cocaine	21	17	1	4	0	0
Heroin	23	21	1	5	0	0
RDX	1	3	6	6	0	0
TNT	5	7	3	6	0	0
	Fractional molar composition					
Cellulose	0.48	0.29	0	0.24	0	0
Latex	0.43	0.43	0.14	0	0	0
Nylon	0.46	0.31	0.08	0.15	0	0
Surfactant	0.56	0.35	0	0.058	0.019	0.019
Ethanol	0.67	0.22	0	0.11	0	0
Cocaine	0.49	0.40	0.023	0.093	0	0
Heroin	0.46	0.42	0.02	0.1	0	0
RDX	0.063	0.19	0.38	0.38	0	0
TNT	0.24	0.33	0.14	0.29	0	0

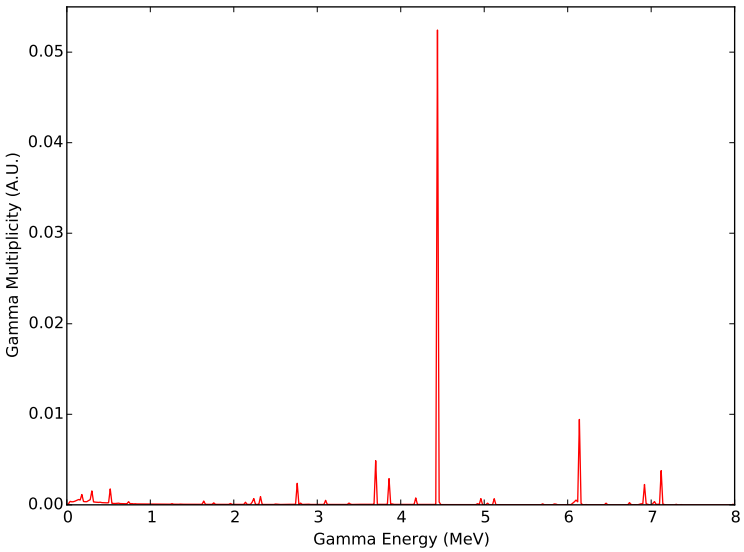
Table 3.1: The absolute and fractional molar composition of simulated irradiated samples.

The simulations were repeated for 5 benign materials and 4 illicit ones. The benign materials used were Cellulose, Latex, Nylon, Surfactant and Ethano; the illicit ones were Cocaine, Heroin, RDX and TNT. The compositions used in the simulations is shown in table 3.1. The total molar fraction is given in the top half and the normalised fraction, to two significant figures, in the bottom half.

The spectra of pure cellulose and cocaine are shown in figures 3.6a and 3.6b respectively. Comparison of figures 3.6a and 3.6b with figure 3.5 shows that they both contain C and O; however, comparison of the heights of the peaks shows the ratios are very different.



(a) Cellulose spectrum



(b) Cocaine spectrum

Figure 3.6: Simulated emitted γ spectrum of Cellulose and Cocaine under 14 MeV neutron irradiation

To identify a compound from a given γ spectra the relative heights of the characteristic peaks is used. A simple technique to do this is taking the ratio of a given peak from the compound spectrum and from the pure element spectrum. Using the ratios of spectral peaks from samples with compositions given in table 3.1 gives the compositions shown in figure 3.7.

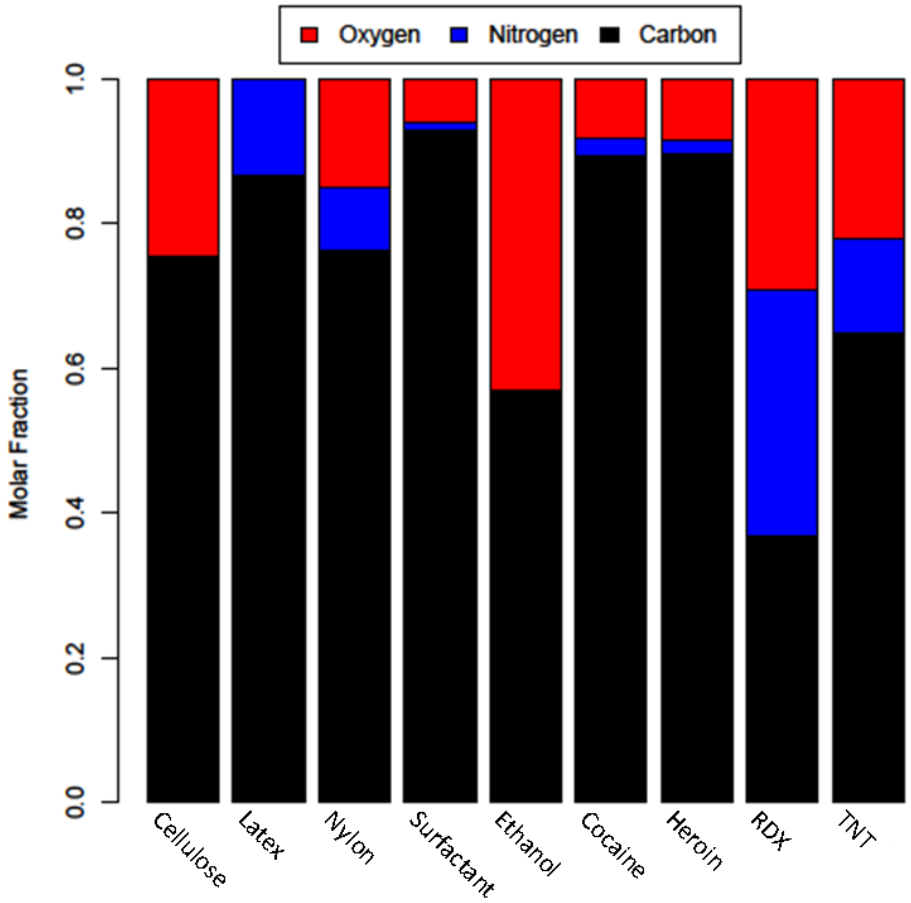


Figure 3.7: Calculated elemental composition of nine materials based on the ratios of the characteristic peaks in their γ spectra produced under 14 MeV neutron irradiation in MCNPX simulations.

The results for the surfactant in figure 3.7 show a small amount of nitrogen; however, there is no N in the surfactant simulated. The N shown in the surfactant is a small background caused by the presence of Cl. Using more spectral lines and comparing the relative height within an element to correct for background will improve the accuracy of the measurement.

3.6 Conclusion

There are a variety of neutron interrogation techniques that can be applied to cargo containers. As the neutrons pass through a container they will cause the emission of γs, and also be deflected and absorbed. Changes to the neutron beam can be used to measure attenuation along the flight path making it possible to identify shapes, and changes in composition. The γs emitted as a result of neutron interactions can also be measured. If this is combined with Time-of-Flight information it is possible to build a 3-Dimensional reconstruction of an interrogated volume with materials within identified.

Measurement of γ spectra is likely to require longer scan times than neutron transmission techniques, and potentially higher fluences; however, the increased threat detection potential may make this justified. One way to maximise the efficacy of neutron interrogation would be to combine neutron transmission with measurements of γ spectra. Combining the two would enable discrimination between voids and shielded regions, both would have minimal γ emission, but voids would not attenuate a neutron beam.

Neutron interrogation is dependent upon the use of a suitable neutron source. Chapters 4 and 5 discuss possible ways of producing neutrons.

Bibliography

[1] B. D. Sowerby and J. R. Tickner. Recent advances in fast neutron radiography for cargo inspection. *Nuclear Instruments and Methods in Physics Research A*, (580):799–802, 2007.

[2] url: http://mnrc.ucdavis.edu/radiography.html. retrieved: 16/02/15.

[3] J. E. Eberhardt, S. Rainey, R. J. Stevens, B. D. Sowerby, and J. R. Tickner. Fast neutron radiography scanner for the detection of contraband in air cargo containers. *Applied Radiation and Isotopes*, (63):179–188, 2005.

[4] R. Runkle, T. White, E. Miller, J. Caggiano, and B. Collins. Photon and neutron interrogation techniques for chemical explosives detection in air cargo: A critical review. *Nuclear Instruments and Methods in Physics Research A*, 603:510–528, 2009.

[5] A. Buffler, F.D. Brooks, M. S. Allie, K. Bharuth-Ram, and M. R. Nchodu. Material classificiation by fast neutron scattering. *Nuclear Instruments and Methods in Physics Research B*, (173):483–502, 2001.

[6] J. E. McFee, A. A. Faust, H. R. Andrews, E. T. H. Clifford, and C. M. Mosquera. Performance of an improved thermal neutron activation detector for buried bulk explosives. *Nuclear Instruments and Methods in Physics Research A*, (712):93101, 2013.

[7] E. Browne et al. *F.* John Wiley & Sons inc., 7th edition.

[8] J.Obhodas et al. Analysis of containerized cargo in the ship container terminal. *Nuclear Instruments and Methods in Physics Research A*, 619:460–466, 2010.

[9] A. L. Lehnert and K. J. Kearfott. Preliminary identification of flags for a novel algorithm-based approach for explosives detection using neutron interrogation for a simulated idealized cargo container scanerio. *Nuclear Instruments and Methods in Physics Research A*, (638):201–205, 2011.

Chapter 4

Compound Nucleus Sources

4.1 Introduction

As discussed in Chapter 1 there are a variety of ways to produce neutrons. On small scale, when trying to avoid the difficulties associated with isolated fission sources, Compound Nucleus (CN) reactions are ideal. This chapter discusses the physics behind CN reactions and how they can operate as a neutron source.

A CN is a highly excited state formed when a nucleus merges with another nucleus, a nucleon, or is excited by a γ. The typical life-time of a compound nucleus is of order 10^{-16}s-10^{-18}s [1] after which it will decay either by emission of a γ or at least one nucleon. CN reactions can provide high fluxes of neutrons with a narrow spectrum ideally suited for use in security. A CN neutron source can also be designed such that there are no significant long lived isotopes produced and they can also be turned off and on at will, significant advantages over fission sources.

The energy spectrum of a CN neutron source is dependent upon the energy released in the reaction, which comes from the Q of the reaction and the kinetic energy of the projectile. The neutron energy can be increased either by changing the target or projectile to give a higher Q, or by increasing the projectile kinetic energy. A simple approximation can be derived, shown in equation 4.3, to approximate the neutron energy from a given reaction.

At low projectile energy it is reasonable to ignore relativistic effects and momentum conservation, therefore it can be assumed that the excitation energy of the compound nucleus is given by the sum of the projectile kinetic energy and the Q of the reaction. Assuming a 2 particle final state with the decay nucleus in its ground state gives the entirety of the excitation energy going into the kinetic energy of the ejectiles, giving the equality in equation 4.1

$$Q + E_k = E_n + E_{DN}, \tag{4.1}$$

where E_k, E_n and E_{DN} are the projectile, neutron and decay nucleus kinetic energies respectively. No mass terms need to be included as the Q of the reaction contains the mass of all components. By ignoring relativistic effects the ratio of E_n and E_{DN} can be equated to the inverse ratio of their masses

$$\frac{M_n}{M_{DN}} = \frac{E_{DN}}{E_n}. \tag{4.2}$$

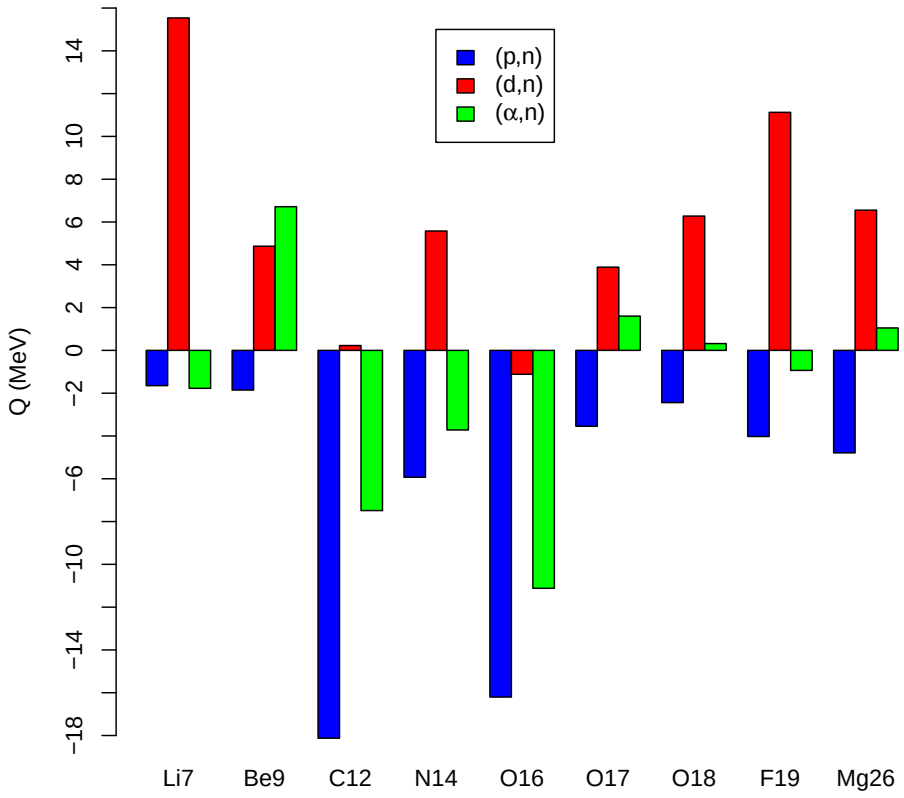


Figure 4.1: Q-values for (p, n), (d, n) and (α, n) reactions with a selection of light isotopes.

Rearranging equation 4.1 with equation 4.2 will then give the approximate value of the neutron kinetic energy as shown in 4.3

$$E_n \approx \frac{E_k + Q}{1 + \frac{M_n}{M_{DN}}}. \tag{4.3}$$

The full derivation of equation 4.3 is given in appendix A.1.

Figure 4.1 shows the Q-values of (p, n), (d, n) and (α, n) reactions on a selection of light isotopes. Using these calculated Q-values it is possible to predict the energies for a range of neutron producing reactions. Using a selection of targets would enable different energies of beam to be produced with a single energy accelerator.

In the compound nucleus rest frame the neutron emission is approximately isotropic with a Maxwellian distribution in energy [2]. The distribution of angles and energies in the rest frame prevent a CN source being perfectly monochromatic. Equation 4.3 ignores momentum conservation; however, in reality there will be a transfer of momentum from the projectile to the target nucleus, as a result the compound nucleus is not stationary in the lab frame, which will result in an increase in the neutron yield in the forwards direction.

Neutron sources based upon CN reactions are currently used in several environments and with a range of technologies, three examples are shown in table 4.1. The Frascati Neutron Generator [3] is a materials testing and nuclear physics research facility using the $T(d, n)$ reaction to produce 14 MeV neutrons. The source described in [4] uses 6 MeV deuterons with a $D(d, n)$ reaction to produce 8.5 MeV neutrons for cargo interrogation. The authors of [5] propose using an $^{7}Li(p, n)$ reaction as a compact source for neutron scattering experiments.

Reaction	Projectile energy	average current	neutron flux	Spectrum	reference
$T(d,n)$	300 keV	3 mA	$5 \times 10^{11}\ s^{-1}$	Quasi-monochromatic 14 MeV	[3]
$D(d,n)$	6 MeV	1.5 mA	$> 1 \times 10^{7}\ \frac{n}{cm^2 s}$	$\leq$ 8.5 MeV	[4]
$^7Li(p,n)$	2 MeV	1 mA	$1 \times 10^{12}\ s^{-1}$	$\leq$ 800 keV	[5]

Table 4.1: Examples of functioning or proposed compound nucleus neutron sources for research and industrial applications with proton or deuteron projectiles.

4.2 Simulations

For applications in which neutron beams are desired, in this instance security, a detailed understanding of the neutron beam characteristics is essential. Monte-Carlo tracking codes, such as MCNPX [6], can be used to simulate some $A(x,n)Y$ reactions and subsequent neutron propagation.

The neutron spectrum, angular distribution and multiplicity are controlled by the target material, beam energy and target thickness. For a given energy of beam, above the threshold energy for a reaction, the neutron multiplicity will increase with increased target thickness. Once the target is thick enough that the energy loss in the beam takes the projectiles below the threshold of the reaction the neutron multiplicity will stop increasing. With a thin target and increasing beam energy the multiplicity is not guaranteed to increase but instead will approximately track the cross-section for the reaction.

4.2.1 Proton Induced Reactions

Using equation 4.3 the neutron energy produced in $^{7}Li(p,n)$ reactions was calculated for a range of proton energies. Figure 4.2 shows the predicted energies (vertical dashed lines) against simulated energies for a range of incident proton energies. From figure 4.2 it is clear that equation 4.3 gives reasonable agreement, and is sufficient to give a first approximation of neutron energy.

The $X(p,n)Y$ reactions were simulated for a selection of materials with either constant target thickness or constant beam energy.

The results in figure 4.3 show surface plots of neutron energies and fluences produced for a range of bombarding proton energies on ^{9}Be and ^{26}Mg. From

equation 4.3 it would be expected that increasing the beam energy also increases the neutron energy available from the reaction, this is demonstrated in figure 4.3. The maximum neutron energy produced at each proton energy is the result of all available energy being divided between the emitted neutron and a decay nucleus in its ground state. The lower energy emissions visible, especially in ^{9}Be in figure 4.3a at approximately 2 MeV proton energy and above, are the result of other decay channels and/or decay nuclei in excited states.

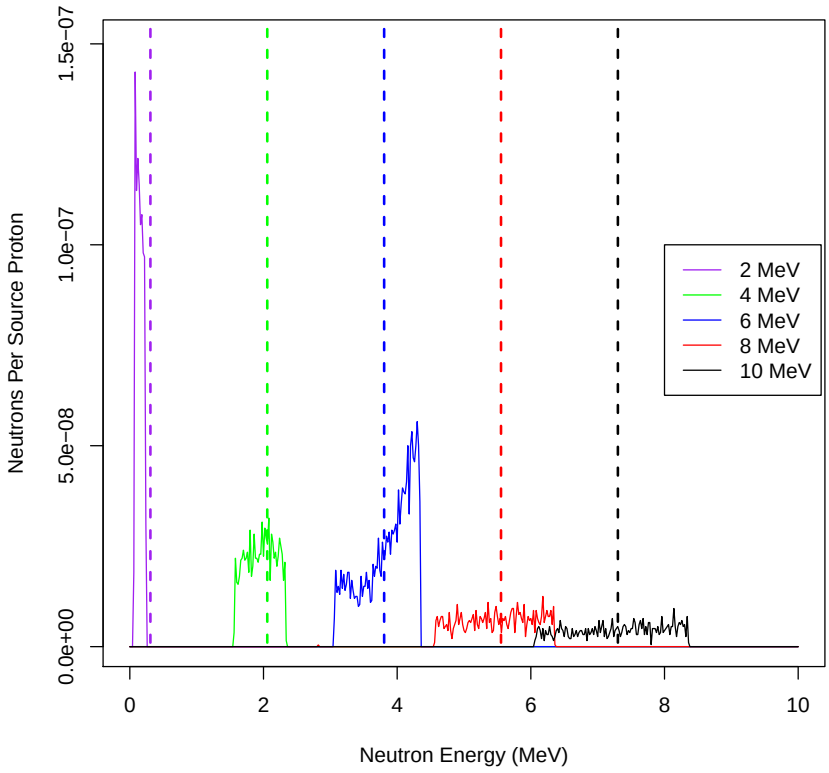
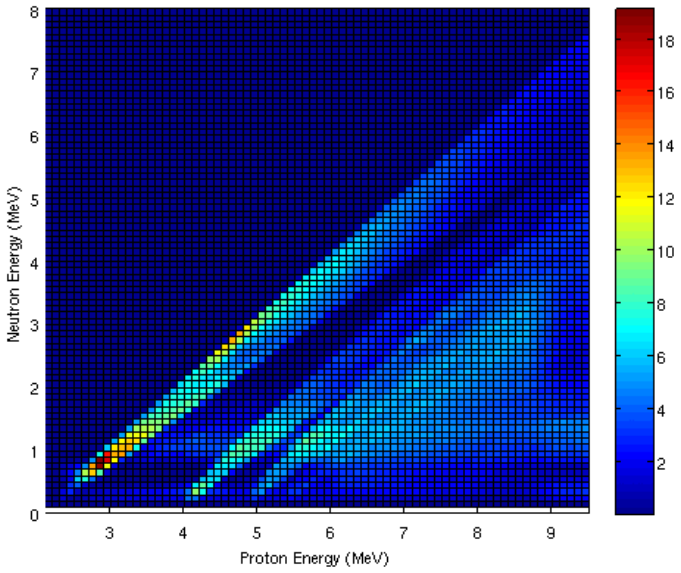


Figure 4.2: $^{7}Li(p,n)$ neutron spectra as produced by MCNPX simulations for a range of incident proton energies. The vertical dashed line gives the energy predicted by equation 4.3.

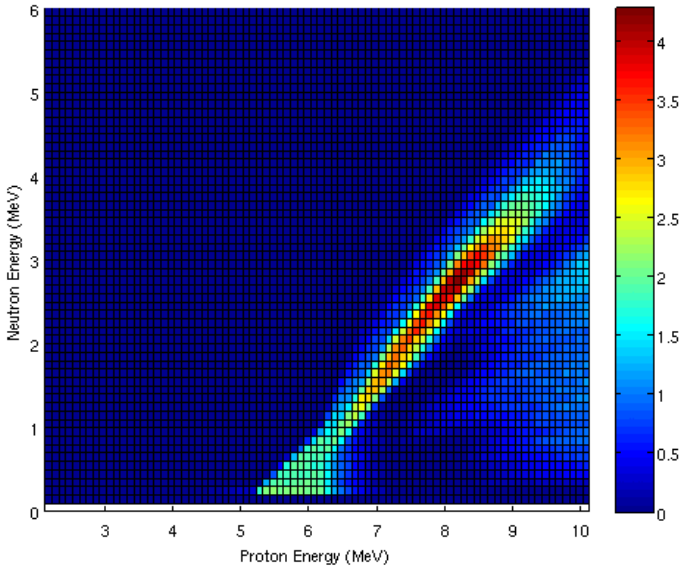
The spectra of $^{7}Li(p,n)$ (figure 4.2), $^{9}Be(p,n)$ (figure 4.3a) and $^{26}Mg(p,n)$ (figure 4.3b) show a very strong correlation between increasing beam energy and increasing peak neutron energy. The spectrum of ^{9}Be has a broad range of energies, the multiple peaks that can be seen, especially at higher beam energies, are due to additional decay channels and/or meta-stable states in the decay nucleus. In addition the integral of the $^{7}Li(p,n)$ spectrum shows an important effect as the neutron multiplicity can be clearly seen to decrease with increasing proton energy. The decreased multiplicity with increasing energy of $^{7}Li(p,n)$ is an inevitable product of the cross-section decreasing over the energies used combined with a thin target.

Whilst increasing the irradiating proton energy increases the maximum emitted neutron energy it is also possible to vary the target thickness. For a given source energy, above the (p,n) reaction threshold, an increased target thickness will provide a greater neutron yield. In conjunction with the increased yield the thicker target will also broaden the neutron spectrum for two reasons. As the projectiles penetrate a target they will lose energy, as a result at the point of interaction they will have less energy resulting in a reduced neutron energy. When a neutron is produced there will be a small possibility of interaction with any target material in its path, which will also reduce the neutron energy.

Figure 4.4 shows the neutron spectra and total yield from (p,n) reactions on ^{9}Be and ^{26}Mg of increasing thickness under 6 MeV proton irradiation. The ^{26}Mg results in figure 4.4b show the increase in both spectral width and total yield, the increase from 50 μm (black line) to 200 μm (red line) results in an approximately doubling of both the total yield and the FWHM of the spectrum. The effect is less pronounced in the $^{9}Be(p,n)$ results as there are a number of spectral peaks



(a) $^{7}Be(p,n)$



(b) $^{26}Mg(p,n)$

Figure 4.3: Neutron energy spectra of $^{7}Be(p,n)$ and $^{26}Mg(p,n)$ reactions over a range of irradiating proton energies. The color indicates the number of neutrons emitted per 10^{7} irradiating protons.

present under 6 MeV irradiation and they have already merged with a 50 μm target.

4.2.2 Deuteron Induced Reactions

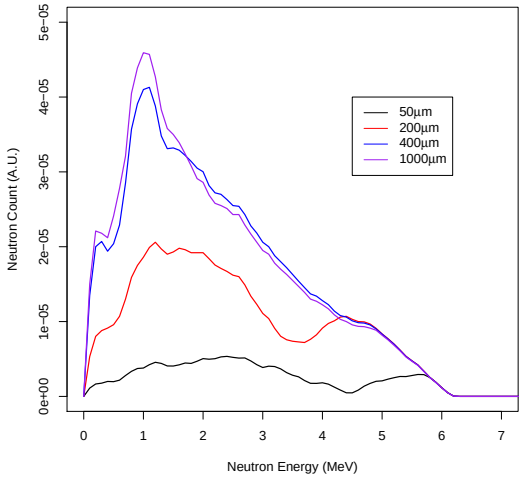
At present Geant4 and MCNPX are not able to simulate deuteron interactions accurately at the energies of interest. Whilst Monte-Carlo simulations are not possible some approximate ideas of the outcome of (d, n) reactions is possible. Based on equation 4.3 a 2 MeV deuteron beam would enable the following reactions and neutron energies to be provided:

- $^{7}Li(d,n)^{8}Be$, 15 MeV
- $^{19}F(d,n)^{20}Ne$, 12 MeV
- $^{9}Be(d,n)^{10}B$, 5.7 MeV
- $^{16}O(d,n)^{17}F$, 0.4 MeV

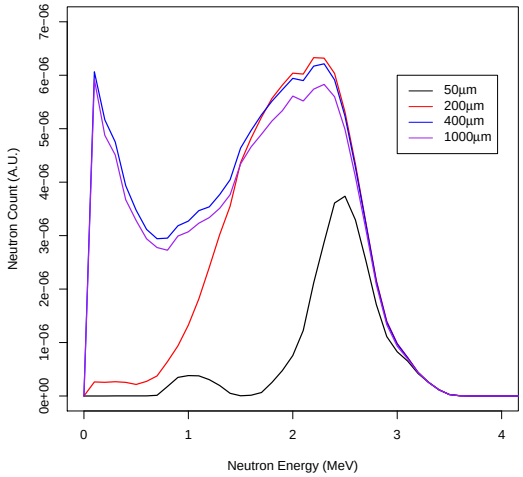
Whilst the values predicted will be a reasonable approximation of the maximum neutron energy produced by each of these reactions, as in the case of $^{7}Li(p,n)$ (figure 4.2), there may be multiple energies produced by any of them limiting the value of such a prediction.

An idealised model of compound nucleus reactions has been written for inclusion within Geant4 allowing an approximate spectra from $X(d,n)Y$, and other, compound nucleus reactions to be produced. For this model the kinetic energy of the neutron in the Center of Momentum frame (CoM) is given by equation 4.4

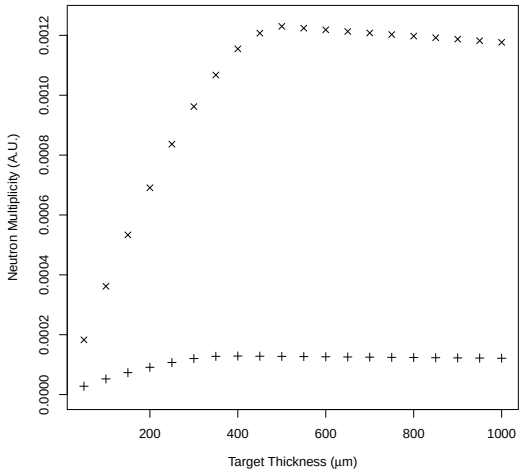
$$E_{k_n} = \frac{(E_p + E_T) - (M_n + M_D)}{1 + \frac{M_n}{M_D}}, \tag{4.4}$$



(a) ^{9}Be



(b) ^{26}Mg



(c) Total yield

Figure 4.4: The emitted neutron spectra from a range of ^{9}Be ($\times$) and ^{26}Mg ($+$) target thicknesses and the total neutron yield under 8 MeV proton irradiation.

where E_{k_n} is the neutron kinetic energy, E_p and E_T are the total projectile and target energies, and M_n and M_D are the neutron and decay nucleus rest mass. Equation 4.4 can be solved for any combination of target and projectile and a neutron of appropriate energy is then emitted over an isotropic distribution in the CoM, which can then be Lorentz boosted to the lab frame for further tracking. There are fewer approximations used in this model than in equation 4.3; however, no excited states or multi-particle decay channels are included and it is therefore unable to produce the multi-peaked spectra seen in reactions such as $^{9}Be(p,n)$.

The idealised model has been used to simulate the (d,n) reactions listed above. 5 μm targets of ^{7}Li and ^{9}Be at room-temperature density and ^{19}F and ^{16}O at boiling-point density were used. The spectra produced by the four targets with all cross-sections set as 200 mb are shown in figure 4.5, in each case the data have been normalised to the peak value for each spectrum. 200 mb was chosen as the actual cross-sections were unknown, therefore a value was chosen that is high enough to allow reasonable statistics, but low enough to represent the physics.

The energy of neutron produced is not the only concern when choosing a projectile/target combination. The decay nucleus produced by a given combination must also be considered, as well as what effect undesired decay channels may have.

4.3 Decay Products

As well as a thorough understanding of the neutron production properties of a target it is vital that the isotope production be well characterised. Under constant bombardment the target nuclei will transmute which may lead to a build up of

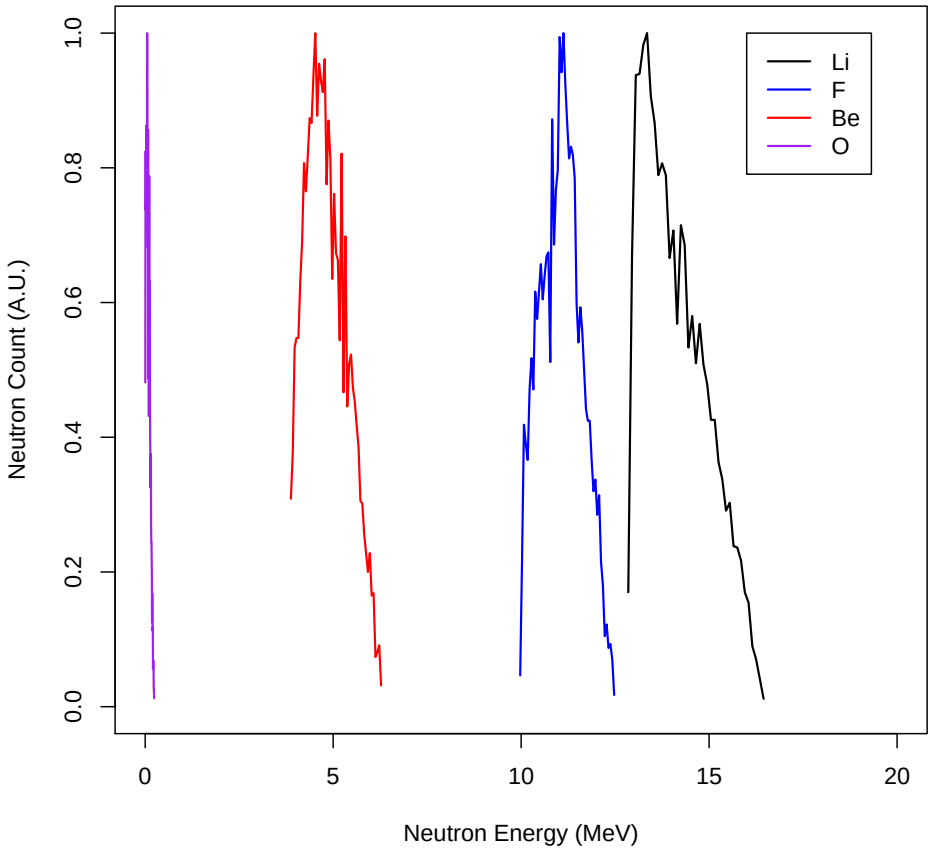


Figure 4.5: Neutron energy spectra for $X(d,n)Y$ reactions on ^{7}Li, ^{19}F, ^{10}B and ^{16}O produced by a 7 MeV deuteron beam.

activity. Ideally any neutron source to be used in a commercial environment will be chosen to ensure no significant activity builds up within the system.

Reactions considered in previous sections are:

- $^{7}Li(p,n)^{7}Be_{(t_{\frac{1}{2}}\approx 53d)} \rightarrow^{7} Li$
- $^{9}Be(p,n)^{9}B_{(t_{\frac{1}{2}}\approx 8\times 10^{-19}s)} \rightarrow^{8} B_{(t_{\frac{1}{2}}\approx 770ms)} \rightarrow 2^{4}He$
- $^{26}Mg(p,n)^{26}Al_{(t_{\frac{1}{2}}\approx 7.17\times 10^{5}years)} \rightarrow^{26} Mg$
- $^{7}Li(d,n)^{8}Be_{(t_{\frac{1}{2}}\approx 6.7\times 10^{-17}s)} \rightarrow 2^{4}He$
- $^{19}F(d,n)^{20}Ne$ **Stable**
- $^{9}Be(d,n)^{10}B$ **Stable**
- $^{16}O(d,n)^{17}F_{(t_{\frac{1}{2}}\approx 64.5s)} \rightarrow^{17} O$

Half-lives on the order of seconds and minutes, or thousands of years, do not pose a threat. Short half-lives will allow all activation products to decay very quickly, the activity of ^{17}F will be approximately 0.1 % of it's maximal value an hour after irradiation. Very long half-life isotopes such as ^{26}Al are also unlikely to pose a threat due to the very low decay rate. Of the listed reactions the only isotope that may pose a threat is ^{7}Be, produced by $^{7}Li(p,n)$, but despite the potential concerns, the decay mode is electron capture and so the only by-product will be a 0.477 MeV γ ray in some instances.

Alternative reactions should also be considered, along with the effect of neutrons on the target material. Where d,n reactions are used there is also a possibility of (d,p) reactions. The (d,p) component of $D(d,n)$ fusion sources causes the build up of tritium, making them unsuited to mass deployment.

- $^{7}Li(d,p)^{8}Li_{(t_{\frac{1}{2}}\approx 840ms)} \rightarrow^{8} Be_{(t_{\frac{1}{2}}\approx 6.7\times 10^{-17}s)} \rightarrow 2^{4}He$
- $^{19}F(d,p)^{20}F_{(t_{\frac{1}{2}}\approx 11s)} \rightarrow^{20} Ne$
- $^{9}Be(d,p)^{10}Be_{(t_{\frac{1}{2}}\approx 10^{6}years)} \rightarrow^{10} B$
- $^{16}O(d,p)^{17}O$ **Stable**

None of the above $X(d,p)Y$ reactions would be of concern in a security environment, again due to very short or long half-lives. Of greater concern are multi-nucleon final states, both due to the decay nuclei produced and also the ejectiles. For low Z nuclei there are very few intermediate life radio-isotopes; however, there is also a possibility of (d,t) reactions, where the ejectile is tritium. The Exfor database has cross-sections for $X(d,t)Y$ reactions on a number of targets. For low Z targets at low energies the cross-sections are mostly very low; however, both $^{9}Be(d,t)$ [7] and $^{7}Li(d,t)$ [8] have cross-sections on the order of 100 mb at deuteron kinetic energies of less than 5 MeV.

Along with the possibility of unintended reactions the behaviour of the target material under neutron irradiation should be considered. Under high energy neutron irradiation ^{7}Li can produce tritium through the $^{7}Li(n,n+t)^{4}He$ reactions. With a sufficiently thin target this may not pose a significant threat but should be considered as a potential risk when designing a target for industrial purposes.

4.4 Conclusion

At present cargo interrogation research is primarily performed using $T(d,n)$ and $D(d,n)$ fusion sources. Due to the use or production of tritium these sources are

not suitable for mass deployment and so alternative sources need to be found. The spectral characteristics of a source are very important for cargo interrogation and every combination of target and projectile will have a different spectrum.

Protons, deuterons and αs are all possible projectile options. Hypothetically any atomic nucleus could be used, though anything heavier than an α would be impractical. In principle any isotope could be used as a target, but using a combination of target and projectile with a positive Q would be preferable as it would enable fast neutrons to be produced with a lower energy accelerator.

Deuterons are very lightly bound, if they were used with kinetic energy greater than the binding energy (2.22 MeV) there would inevitably be a number of neutrons produced through deuteron dissociation. The next chapter discusses C++ models of deuteron dissociation designed for inclusion in Geant4.

Bibliography

[1] K. S. Krane. *Introductory Nuclear Physics*, chapter 11. John Wiley and Sons.

[2] K. Heyde. *Basic Ideas and Concepts in Nuclear Physics: An Introductory Approach*, chapter 1. Institute of Physics Publishing, 3 edition.

[3] M. Martone, M. Angelone, and M. Pillon. The 14 MeV Frascati neutron generator. *Journal of Nuclear Materials*, (212-215):1661–1664, 1994.

[4] B. A. Ludewigt et al. Accelerator-driven neutron source for cargo screening. *Nuclear Instruments and Methods in Physics Research B*, 261:303–306, 2007.

[5] F. Hiraga, T. Okazaki, and Y. Kiyanagi. Neutronic design on a small accelerator-based ^{7}li(p,n) neutron source for neutron scattering experiments. *Physics Procedia*, (26):97–107, 2012.

[6] R. Forster, L. Cox, R. Barrett, T. Booth, J. Briesmeister, F. Brown, J. Bull, G. Geisler, J. Goorley, R. Mosteller, S. Post, R. Prael, E. Selcow, and A. Sood.

Mcnp version 5. *Nuclear Instruments and Methods in Physics Research B*, 213:82–86, 2004.

[7] R. E. Heft and W. F. Libby. Absolute cross sections for deuterons on beryllium. *Physical Review*, 100(3):799–813, 1955.

[8] R. L. Macklin and H. E. Banta. Tritium production from lithium by deuteron bombardment. *Physical Review*, 97(3):753–757, 1955.

Chapter 5

Deuteron Dissociation

The previous chapter showed that deuteron induced compound nucleus reactions have the potential to provide the neutron beams required for cargo interrogation. Deuteron beams will not only provide neutrons through compound nucleus reactions, there will also be a component from deuteron dissociation [1]. At present deuteron dissociation cannot be simulated in either MCNPX or Geant4, this chapter presents the development of a numerical model of deuteron dissociation written for inclusion in Geant4.

There has been significant work to construct theoretical models of deuteron dissociation, see for example references [2, 3, 4]. There have also been experimental measurements of ejected neutron and/or proton spectra, see for example references [5, 6]. A firm understanding of deuteron dissociation is beneficial in applied physics due to its applicability as a neutron source [5] and due to the effect of dissociation on activation cross-sections [7].

5.1 Physics

At high deuteron energies the Fermi momentum of the nucleons is significantly smaller than the relativistic momentum of the deuteron and so can be ignored. At high energies the nucleons can be approximated as travelling in the same direction as the deuteron with energy

$$E_N = \frac{E_D - |\varepsilon_B|}{2}, \tag{5.1}$$

Where E_N is the emitted nucleon kinetic energy, E_D is the deuteron kinetic energy and ε_B is the deuteron binding energy. The models presented here have been developed for use at energies where the Fermi momentum is not insignificant and therefore equation 5.1 will not be sufficiently accurate.

The models presented in this chapter treat the deuteron as two separate nucleons trapped in a potential well with momentum $\overrightarrow{P}$. When low precision solutions can be accepted, such as under high energies, or when sending deuterons into a thick target, certain approximations can be made. When high precision is required some of these approximations are not acceptable. Two models have been produced, one high precision and one low precision to allow for these two cases.

5.1.1 Low Precision

The author of [8] gives the Hulthen function in the form shown in equation 5.2. Equation 5.2 gives the probability of a nucleon having a given momentum as,

$$N(P) \propto P^2 \left(\frac{1}{\alpha^2 + P^2} - \frac{1}{\beta^2 + P^2} \right)^2, \tag{5.2}$$

In which $N(P)$ is the number of nucleons with momentum P and α and β are constants with values 45.7 MeV/c and 320 MeV/c, respectively, according to the analysis of [8]. The Hulthen function describes a Lorentzian distribution of nucleon momenta shown in figure 5.1.

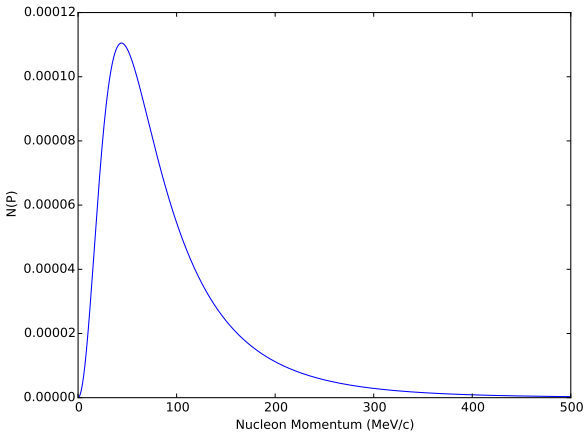


Figure 5.1: $N(P)$ versus nucleon momentum (P) calculated with equation 5.2.

Equation 5.2 gives a probability distribution for a nucleon to have absolute momentum P. The momenta are distributed isotropically within the deuteron rest frame, with $\overrightarrow{P_p} = -\overrightarrow{P_n}$. A Lorentz boost from the deuteron rest frame can then be used to give the lab frame momenta as $\overrightarrow{P_p'}$ and $\overrightarrow{P_n'}$

The low precision model calculates $\overrightarrow{P_p'}$ and $\overrightarrow{P_n'}$ with this method. The interactions between the deuteron and target nucleus are not included and the influence of the deuteron potential energy is ignored preventing energy from being conserved. The approximations make the model less computationally expensive but also limit accuracy, it is therefore necessary to consider the suitability for a given application carefully.

5.1.2 High Precision

Unlike the low precision model the high precision model includes the effects of both the deuteron potential energy and coulomb scattering from the target nucleus. To include coulomb interactions the proton is treated as undergoing Rutherford scattering with a finite mass target causing it to receive a momentum kick ($\overrightarrow{P_k}$) and scatter through angle θ_k. The scattering angle is calculated using the Rutherford scattering formula given in equation 5.3

$$\theta_k = 2cot^{-1}\left(\frac{b\mu u^2}{qQ} \times 4\pi\varepsilon_0\right), \tag{5.3}$$

In which b is the impact factor, μ is the reduced mass, u is the proton velocity in the lab frame and q and Q are the proton and target charge respectively. Using the reduced mass in place of the projectile mass incorporates finite target mass. The impact factor would be the distance of closest approach if there was no coulomb repulsion, with $b = 0$ for a head-on collision.

θ_k can be used to calculate the components and magnitude of $\overrightarrow{P_k}$ relative to the direction of $\overrightarrow{P_p}$ with the equations

$$\begin{aligned} P_{k\perp} =& \mu u sin\theta_k, \\ P_{k\parallel} =& \mu u(1 - cos\theta_k), \\ P_k =& \mu u(2 - 2cos\theta_k)^{\frac{1}{2}}. \end{aligned} \tag{5.4}$$

The value of P_k gives the change in momentum experienced by the proton as it travels from $-\infty$ to $+\infty$. If the proton gains a sufficient momentum increase it will be able to overcome the deuteron potential energy and the deuteron will

dissociate. After the kick is applied the magnitude of the proton and neutron momenta in their center of momentum frame (CoM), which is distinct from the deuteron rest frame, can be calculated by

$$P_{CoM} = \left(\left(\frac{E_T^2 - M_p^2 + M_n^2}{2E_T} \right)^2 - M_n^2 \right)^{\frac{1}{2}}, \tag{5.5}$$

Where P_{CoM} is the magnitude of the proton and neutron momenta in the CoM, E_T is the total energy, and M_p and M_n are the proton and neutron rest mass respectively. E_T can be calculated by

$$E_T = (P_p^2 + M_p^2)^{\frac{1}{2}} + (P_n^2 + M_n^2)^{\frac{1}{2}} + |\nu|, \tag{5.6}$$

In which P_p and P_n are the proton and neutron momenta and ν is the deuteron potential energy in the CoM. Once P_{CoM} is calculated the magnitude of the proton and neutron momenta in the CoM can be set to P_{CoM}, maintaining their original direction, and then returned to the lab frame.

The High Precision model is significantly more detailed than the Low Precision model; however, there are still approximations used. The momentum kick is assumed to be applied instantaneously at the point of closest approach, and dissociation is assumed to happen at this point. In actuality it is likely that dissociation will occur anywhere within a sphere surrounding the nucleus, causing a range of values of $\overrightarrow{P_k}$ for a given impact factor. There is also no strong interaction used, this will only matter when the deuteron energy is comparable to the coulomb barrier; however, for fast deuterons or low Z nuclei this may have an effect. The final approximation is that the target nucleus is assumed to remain

in the ground state at all times. In reality there will be a possibility of the target being put into an excited state, reducing the energy available for dissociation.

5.2 Numerical Models

The numerical models have been written in C++ for inclusion with Geant4. The code is split into two sections, high and low precision, to suit different requirements.

5.2.1 Low Precision

In order to numerically calculate the momentum distribution of nucleons it is necessary to first integrate equation 5.2 giving equation 5.7

$$\begin{aligned}\int N(P)dP =& \frac{4\alpha\beta(\alpha+\beta)}{\pi(\alpha-\beta)} \times \Bigg[\\ & \left(\frac{1}{2\alpha} + \frac{2\alpha}{\alpha^2\beta^2}\right) tan^{-1}\frac{P}{\alpha} + \\ & \left(\frac{1}{2\beta} - \frac{2\beta}{\alpha^2\beta^2}\right) tan^{-1}\frac{P}{\beta} - \\ & \frac{1}{2}\left(\frac{P^2}{P^2+\alpha^2} + \frac{P^2}{P^2+\beta^2}\right)\Bigg],\end{aligned} \tag{5.7}$$

The full integration is given in Appendix A.2. Equations 5.7 and 5.2 can then be used with Newton's method to identify the momentum associated with values selected from a uniform random number distribution varied between 0 and 1.

The Hulthen function is used to calculate P_n, which is isotropically distributed in the deuteron rest frame. The direction of individual neutrons is calculated

using a random distribution such that $cos(\theta)$ varies from 0 to 1 and φ varies from 0 to 2π. Since the proton and neutron momenta must be equal and opposite $\overrightarrow{P_p}$ can be set equal to $-\overrightarrow{P_n}$. The final step in the low precision code is to boost $\overrightarrow{P_p}$ and $\overrightarrow{P_n}$ into the lab frame where they continue to be tracked by Geant4.

Due to the high momentum tail of the Hulthen function there will be a small number of cases where momentum and energy are not conserved; however, at sufficiently high deuteron velocities the discrepancy is not sufficient to be of concern.

5.2.2 High Precision

The high precision model is designed to incorporate the effect of Coulomb repulsion of the deuteron and also prevent violation of momentum and energy conservation. The high precision model takes $\overrightarrow{P_p}$ and $\overrightarrow{P_n}$, as calculated by the low precision model, and continues from there.

From equations 5.3 and 5.4 it can be seen that $P \propto \theta \propto \frac{1}{b}$. To generate a distribution of momenta and angles it is necessary to sample b from the range $0 \rightarrow B$ where B is the maximum value of b. The value of B is that at which ΔP is sufficient to overcome the potential energy of the deuteron and dissociate the nucleons, where ΔP is given by equation 5.8

$$\Delta P = \left((P_p + P_k)^2 - P_p^2\right)^{\frac{1}{2}}. \tag{5.8}$$

To calculate B it is necessary to find the minimum value of ΔP for a given Q, m_t and u that will allow the deuteron to dissociate.

To calculate the minimum value of ΔP we must consider the neutron/proton system in which the momentum kick is applied entirely to the proton. The minimum value of ΔP can be calculated with equation 5.9

$$\Delta P_{min} = -P_p + \left[(E_i - E_B)^2 - m_p^2\right]^{\frac{1}{2}}, \tag{5.9}$$

In which E_i is the total initial proton energy and m_p is the proton rest mass. The minimum scattering angle is when $P_k = \Delta P_{min}$, as given in equations 5.4, from this the minimum value of θ can be calculated with equation 5.10

$$\theta_{min} = cos^{-1}\left[1 - \frac{1}{2}\left(\frac{\Delta P_{min}}{\mu u}\right)^2\right], \tag{5.10}$$

Which can then be used with equation 5.3, rearranged for b, to calculate B for a given θ_{min}. Having calculated B it is possible to calculate b using equation 5.11

$$b = B\sqrt{N}, \tag{5.11}$$

In which N is sampled from a distribution of random numbers such that $0 \leq N \leq 1$. Having kicked the proton the deuteron potential energy, calculated before the proton was kicked, is subtracted from the proton and neutron such that their momenta remain equal and opposite in the frame co-moving with the neutron/proton CoM frame after the kick.

The resulting proton and neutron momenta are then boosted to the lab frame and returned to Geant4 for further tracking along with the target nucleus, which has momentum $-\overrightarrow{P_k}$.

5.3 Comparison With Data

There is currently very limited data on deuteron dissociation in the literature, especially at the energies of interest to this model. Additionally separating the break up and compound nucleus component of the emitted neutron spectrum is very difficult, and not necessarily possible in experiment.

5.3.1 Energy

Bleuel [5] measured the emitted neutron spectrum produced by deuterons at 20 MeV and 29 MeV incident on Ti and Ta targets with area density 17.0 mg/cm^2 and 42.2 mg/cm^2 respectively. The facility used is shown diagrammatically in figure 5.2. Accurate recreation of the facility in simulation was not possible as not all dimensions are available, including the size of the detector; however, an approximate recreation is possible.

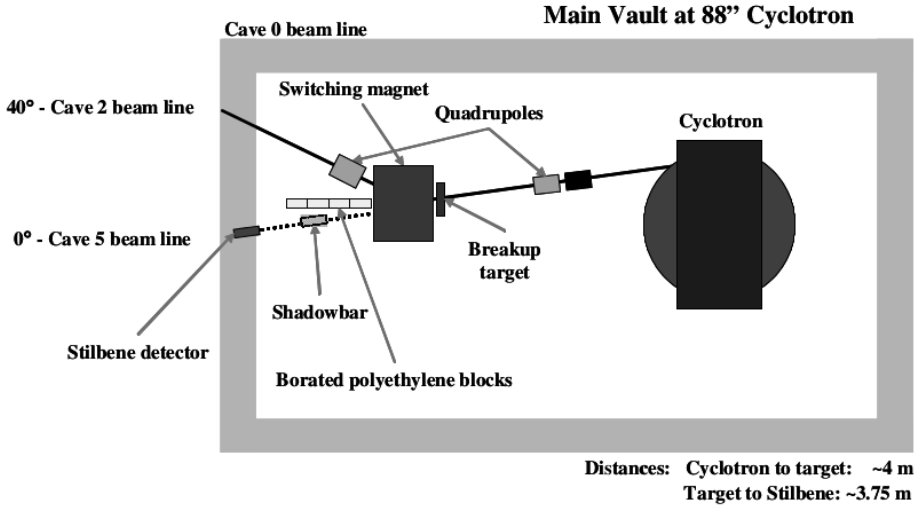


Figure 5.2: Facility used to measure neutron spectra by the authors of [5].

Figure 5.3 compares the emitted neutron spectrum at 20 (blue) and 29 (red) MeV produced by deuterons on Ti in simulation (lines) and experiment (crosses) using the low precision model. The simulated data is noticeably broader than the experimental data for two reasons. The low energy component is due to the neutrons being backwards emitted in the CoM more easily in the model than in reality, therefore when viewed in the lab frame more low energy neutrons are detected. The high energy component is due to energy not being conserved resulting in unrealistically high energies being emitted. Other approximations will also have an effect, but they will be insignificant in the case of the low precision model.

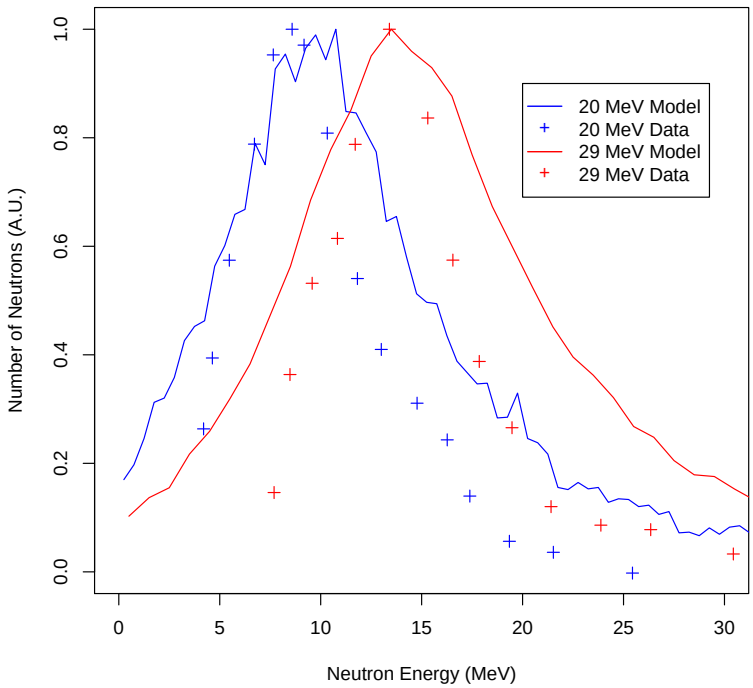


Figure 5.3: Normalised neutron spectra for 20 and 29 MeV deuterons incident on a 17.0 mg/cm^2 Ti target simulated with the low precision model compared with data from [5].

Despite the obvious discrepancy that can be seen in figure 5.3, where both the high and low energy data are too broad in spectrum, the modal energy is in good agreement. The high energy component of the simulated data extends beyond the limits of the plot with small numbers of high energy neutrons caused by the lack of energy conservation in the low precision model.

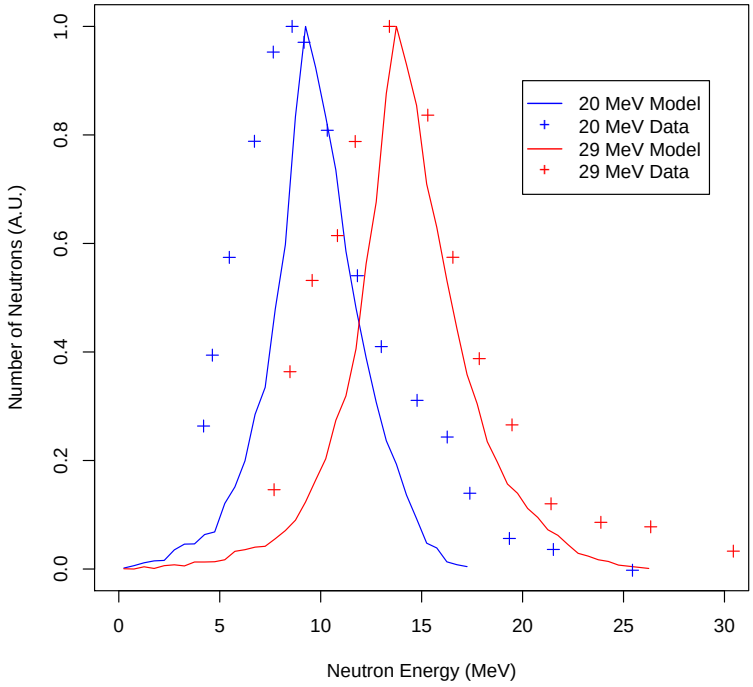


Figure 5.4: Neutron spectra for 20 and 29 MeV deuterons incident on a 17.0 mg/cm^2 Ti target simulated with the high precision model compared with data from [5].

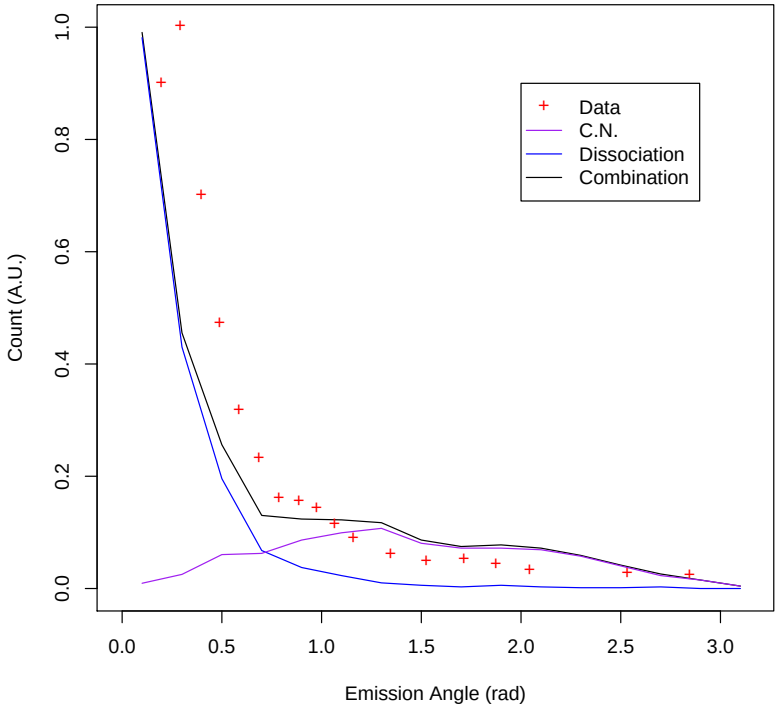
The high precision model does conserve energy, making it more accurate but also more computationally expensive, than the low precision model. The result in figure 5.4 are the same as figure 5.3 but with the high precision model used. In this instance the results are far more accurate, the FWHM is slightly reduced

compared to data but the maximum emitted energy is within the bounds of the data and does not have a high energy component violating energy conservation.

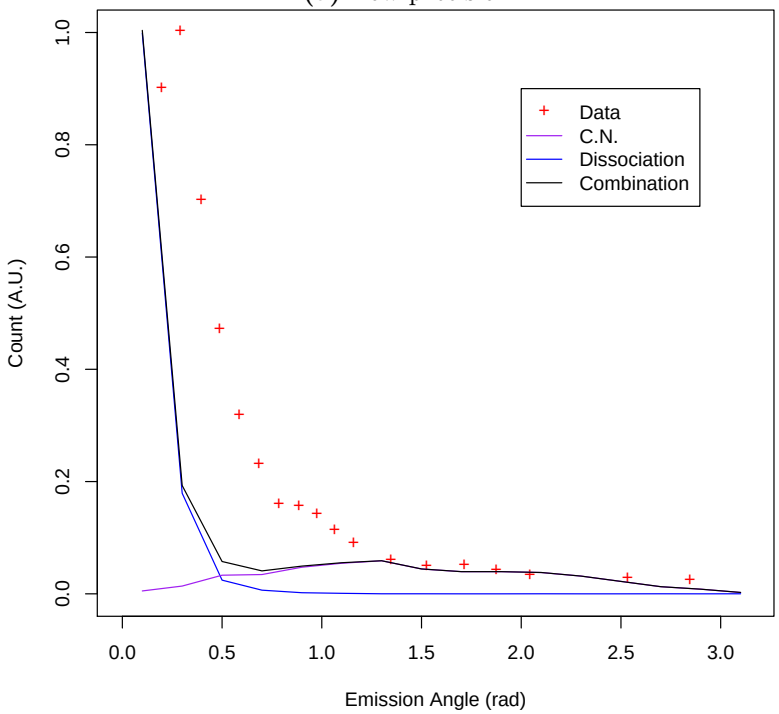
The cause of the inaccuracy in the high precision model will be due to a combination of multiple effects, which will have affected the low precision model but insignificantly compared to other effects. The deuteron beam simulated was perfectly mono-energetic at 20 and 29 MeV. In reality there will have been a spread of energies resulting in a slight broadening of the emitted neutron spectrum. The model assumes that P_k is applied at the point of closest approach, and that dissociation occurs at this point, with no loss of energy before, which is not realistic but is a beneficial approximation.

5.3.2 Angular Distribution

An initial comparison of the emitted angular distribution with the results of [9] showed an extremely poor similarity. In all cases the large forward emitted component was matched, with varying degrees of accuracy, but the component at higher angles was not. Combining the neutrons from the deuteron dissociation model with the idealised compound nucleus model discussed at the end of Chapter 4 produced a much greater similarity. The results for both the high and low precision models, combined with the compound nucleus models, with 16 MeV deuterons incident on a ^9Be target with area density 1.85 mg/cm^2 are shown in figure 5.5.



(a) Low precision



(b) High precision

Figure 5.5: Simulated angular distribution of emitted neutrons from a combination of deuteron dissocation and compound nucleus reactions of 16 MeV deuterons incident on a 1.85 mg/cm^2 ^9Be target with the break up component from the low and high precision models compared with data from [9].

The results presented in figure 5.5 are the combination of the two models with cross-sections set to 200 mb for both reactions at all energies. 200 mb was chosen because it is comparable to many compound nucleus cross-sections, is high enough to allow reasonable statistics without requiring very large simulation time, and is low enough that some energy loss in the target material is expected. In both figure 5.5a and 5.5b the total distribution is given by the black line, the dissociation component by the blue line and the compound nucleus component by the purple line, the experimental data points are given by red crosses.

From figure 5.5 it can be seen that the low precision model (figure 5.5a) has a broader angular distribution than the high precision model (figure 5.5b). The different angular distributions of the two models are caused by the energy conservation of the high precision model reducing the probability of high emission angles, whereas the low precision model emits isotropically in the deuteron rest frame and the direction is provided by the Lorentz boost to the lab frame.

Comparison with the data for other targets in [9] shows similar levels of agreement, with moderate discrepancies in all cases. The discrepancies in the angular distribution of the high precision model will be due to the same approximations that cause the discrepancies in the energy spectrum.

5.4 Conclusion

In its current form the model presented here meets a requirement that will become increasingly important as interest in low energy accelerator based neutron sources grows. As shown in Chapter 3 neutron sources are of interest in cargo

interrogation, but there are also many other applications as discussed in Chapter 1.

There are further improvements that can be made to the model for situations requiring higher accuracy. At present there is no Z-dependence to the neutron emission, the authors of [5] provide a spectrum for Ta as well as Ti. The Ta results show approximately 5 MeV lower modal energy. The Ta results are not as accurate as the Ti results due to background effects so it is not clear how strong the Z-dependence is in practice but it is expected to be a real effect.

An approximation used in this model is that break up does not occur until the point of closest approach, with no loss of energy to this point, and the entire momentum kick is applied to the proton. The authors of [5] refer to the break up radius, the distance from the center of the nucleus at which the proton and neutron separate, which is non-constant. A more accurate model should include the variable break up radius, and will therefore also include the Z-dependence.

These models simulate only complete dissociation of the deuteron, where both proton and neutron are ejected. In addition to complete break up it is possible for a transfer reaction to occur, where one of the nucleons is left behind to merge with the target nucleus. Extending the current break up models with a model of transfer reactions, in which the proton or neutron is not ejected, would be very beneficial. In particular if the nucleon that is transferred from the deuteron to the nucleus is then treated in a compound nucleus model it may enhance the simulation.

The Z-dependence suggested by current literature needs to be validated, and a model benchmarked against it. In addition there is insufficient data to know the cross-section of the reaction. In the presented work a constant cross-section

was used, independent of irradiating energy and target material, when in reality both Z and energy will influence the cross-section.

Bibliography

[1] S. M. Grimes, P. Grabmayr, R. W. Finlay, S. L. Graham, G. Randers-Pehrson, and J. Rapaport. A technique to correct for backgrounds caused by break-up neutrons from the d(d,n) reaction. *Nuclear Instruments and Methods*, (203):269–272, 1982.

[2] R. C. Johnson and P. J. R. Soper. Contribution of deuteron breakup channels to deuteron stripping and elastic scattering. *Physical Review C*, 1(3):976–990, 1970.

[3] J. A. Tostevin, S. Rugmai, and R. C. Johnson. Coulomb dissociation of light nuclei. *Physical Review C*, 57(6):3225–3236, 1998.

[4] L. P. Kaptari, B. Kämpfer, S. M. Dorkin, and S. S. Semikh. Relativistic description of exclusive deuteron break-up reactions. *Physics Letters B*, (404):8–14, 1997.

[5] D. L. Bleuel et al. Characterization of a tunable quasi-monoenergetic neutron beam from deuteron breakup. *Nuclear Instruments and Methods in Physics Research B*, (261):974–979, 2007.

[6] L. Jarczyk, J. Lang, R. Müller, A. Pollitzer, J. F. Valley, W. Wölfli, and P. Marmier. Deuteron breakup in the field of a heavy nucleus. *Physics Letters*, 39B(2):191–192, 1972.

[7] M. Avrigeanu et al. Analysis of deuteron elastic scattering and induced activation cross-sections of light and medium nuclei for IFMIF EVEDA. *Fusion Engineering and Design*, (84):418–422, 2009.

[8] R. Barlow. *Pion Deuteron Interactions at 21 GeV/C*. PhD thesis, University of Cambridge, 1977.

[9] M. Febbraro et al. Systematic study of (d,n) reactions at $e_d = 16$ mev using a deuterated scintillator array. *INPC 2013 International Nuclear Physics Conference*, 66, 2014.

Chapter 6

Cargo Activation

6.1 Introduction

In Chapter 3 some of the methods of interrogating cargo were discussed. These methods have a range of neutron spectrum requirements, including low energy, high energy and white spectra. Chapters 4 and 5 showed that there are ways of producing neutrons of virtually any energy with a range of spectral characteristics. In this chapter the potential risk of foods being activated under neutron interrogation are discussed, and the influence of neutron energy is considered.

As neutrons propagate through a container some of the contents, including food, will be activated raising the possibility of exposing the general public to radiation. When exposing individuals to radiation it is required to keep the dose As Low As Reasonably Achievable (ALARA). The ALARA principle does not require radiation to be removed altogether, but if a reduction can be made without compromising efficacy and with minimal cost that reduction should be made [1]. In addition to the ALARA principle it is necessary to consider the

Justification Principle, which says that any change in radiation exposure must do no more harm than good [1]. To date there has been little research into the relationship between source neutron energy, food composition and activation.

Tenforde [2] showed that pharmaceuticals and medical devices irradiated by a mix of fast (8.5 MeV with narrow distribution) and thermal (Maxwellian distribution extending up to 0.1 MeV) neutrons would not produce effective doses above a recommended safe limit of 1 mSv per year. The results given by Tenforde [2] suggest that for an 8.5 MeV source irradiating various pharmaceuticals the production of Na^{24} in milk of magnesia (MgOH) is likely to produce the highest absorbed dose. Assuming 10g of Mg ingested (recommended dose 2.7 gd^{-1}) the absorbed dose for a 50 kg person would be 6.84×10^{-8} mSv, far less than the recommended dose limit of 1 mSv used by [2].

Due to the conclusions of Tenforde [2] that only ^{24}Na need be considered for pharmaceuticals and medical devices, Tenforde [3] considered only ^{24}Na production when analysing the results of neutron irradiation on food. As with pharmaceuticals the production of ^{24}Na by an 8.5 MeV neutron source was below safe levels.

The food irradiation studies of Giroletti [4] agreed with those of Tenforde [3] and showed that no significant production of ^{24}Na would be seen. Nelson[5] looked at the activation of various common cargo items ranging from jars of pasta sauce to sheets of aluminium. By measuring the time taken for irradiated goods to return to background Nelson [5] showed that the activation would reduce to safe levels within the typical storage time of transported goods.

The exclusive consideration of ^{24}Na by Tenforde [3] and Giroletti [4] was based upon a study of pharmaceuticals and medical devices under 8.5 MeV neutron

irradiation. As Giroletti [4] uses a 14 MeV source additional activation channels not considered by Tenforde [2] may have become available. Additionally it can be argued that the composition of pharmaceuticals and medical devices are not adequately representative to identify all possible hazards that might be produced in foodstuffs, which have highly varied composition.

The experimental method of Nelson [5] was to use an unmoderated 14 MeV neutron beam incident on a single layer of material. In reality as a neutron beam passes through a container the spectrum will become moderated. The moderation will produce a significant thermal tail in the spectrum, which will have an impact on activation. Despite the variety of the previous research the limitations prevent firm conclusions from being drawn.

Experimentally studying the relationship between neutron energy and cargo activation is complicated by the need for a variable energy, mono-chromatic, neutron source. An in-depth study is essential as there are multiple possible reactions a nuclide can undergo e.g. (n, α), (n, p), (n, γ), which may include multiple cross section resonances. This chapter covers simulations designed to explore the relationship between food composition and neutron energy with the induced activity and ingestion dose.

The foods chosen for this chapter were; Almond, Banana, Brie, Cocoa Powder, Corn, Potato and Rice. Whilst the list of foods simulated is far from exhaustive the variation in composition is broad and will show whether food composition plays a significant role in activation. In addition to the varied compositions all the foods are commonly imported and exported by a range of countries necessitating interrogation. The induced γ activity and ingestion dose, and the time required for samples to return to pre-irradiation ingestion dose are considered.

6.2 Simulations

The results presented in this chapter were produced with a combination of radiation transport and nuclear inventory simulations. The radiation transport was performed in MCNPX[6] and the nuclear inventory in Fispact-II[7], which are described in detail in Chapter 2.

Neutron interaction cross-sections are strongly energy dependent, typically with multiple resonances. MCNPX was used to provide the neutron spectrum, which was then passed to Fispact-II to compute the nuclear inventory. For each food a 1 m^3 cube was simulated to ensure realistic levels of neutron spectrum moderation. A pencil beam of mono-chromatic neutrons was directed into one side of the volume and the spectrum recorded after 90 cm,shown schematically in figure 6.1. Leaving 10 cm of food between the surface where the spectrum was measured and the end of the volume to ensure any reflected or scattered neutrons are included in the spectrum passed to Fispact-II. Tracking the spectrum through the simulated volume showed that the thermal component is rapidly populated and the distribution remains approximately constant throughout with only the total number of neutrons reducing with depth.

The Fispact-II simulations were run with a flux of 10^8 n/cm^2/s and a fluence of 10^9 n/cm^2. The fluence used is comparable to that used by the authors of [4]. The fluence will be the dominant factor in the level of activation, with flux only influencing the number of very short lived isotopes left at the end of irradiation. We do not seek to identify an optimum fluence, and the level of activation will be approximately linear with fluence, therefore the differences between foods and energies will be approximately flux and fluence independent.

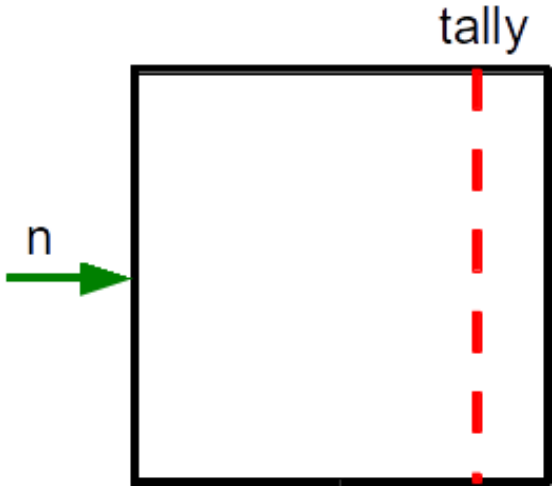


Figure 6.1: A schematic representation of the model used in MCNPX for neutron tracking. The neutrons enter from the left (green arrow) in a monochromatic pencil beam and the energy is recorded through the red dashed line on the right.

The masses of trace elements in foodstuffs along with the water and protein mass are well known and many sources are available, the values in this thesis were taken from [8]. Two important approximations were included, the mass not included in trace elements, water or protein was simulated as cellulose and the sodium was assumed to be in the form of sodium chloride.

To calculate the mass of H, C, N, O and S the mass of water, protein and cellulose were used. In water the mass ratio of $H : O$ was assumed as $1 : 8$, cellulose has the chemical composition $C_6H_{10}O_5$ giving $C : H : O$ mass ratios of $7.2 : 1 : 8$. There are a variety of proteins found in nature; however, Torabizadeh [9] calculated a generic formula $C_nH_{1.85n}N_{0.28n}O_{0.3n}S_{0.01n}$ which gave mass ratios of $C : H : N : O : S$ in protein as $12 : 1.85 : 3.92 : 4.8 : 0.32$.

For each of the foods simulated; Almond, Banana, Brie, Cocoa, Corn, Potato and Rice the relative mass per 100 g of food of each element is given in table 6.1, these foods were chosen because they cover a range of compositions and are

commonly containerised. The foods were simulated at a density of 1 g/cc, the results are given and discussed in the following section.

6.3 Results

To understand the relationship between activation, neutron energy and composition three figures of merit have been considered: Time to Background (TtB), activity and ingestion dose. In many respects the ingestion dose is most important. In an extreme example a low-activity α emitter could be more harmful than a high-activity γ emitter when ingested. The activity and TtB are important when considering the effect of irradiation on those handling goods.

Immediately after irradiation the γ activity of goods is the highest threat as any α and β activity will be blocked by container walls and packaging. If the activity is too high it may necessitate storage in a radiation controlled area and it is important to know how long a container may need such measures and if this time can be reduced.

The results in figure 6.2 show the decay in γ activity in Bq/kg for Almond, Brie, Cocoa and Potato. The other foods fit within the range covered by these four. The percentage uncertainty in the activity for the four foods shown peaked at 12.6%, 18%, 11.8% and 15% but this included contributions from all decay modes, not just the γ activity shown in figure 6.2. Figure 6.2 shows that food composition can have a substantial effect, in this case the γ activity varies by more than 2 orders of magnitude approximately 10 hours after irradiation.

According to the analysis of Tenforde [3] and Giroletti [4] high salt foods undergoing $^{23}Na(n,\gamma)^{24}Na$ reactions are the primary activation threat for foods.

	Almonds	Banana	Brie	Cocoa	Corn	Potato	Rice
H	6.47×10^{0}	9.85×10^{0}	8.59×10^{0}	6.26×10^{0}	6.72×10^{0}	1.01×10^{1}	7.16×10^{0}
C	4.34×10^{1}	1.11×10^{1}	2.38×10^{1}	4.34×10^{1}	4.03×10^{1}	9.61×10^{0}	3.66×10^{1}
N	3.67×10^{0}	1.90×10^{-1}	3.59×10^{0}	2.93×10^{0}	1.63×10^{0}	1.30×10^{0}	2.55×10^{0}
O	4.44×10^{1}	7.85×10^{1}	6.16×10^{1}	4.42×10^{1}	5.05×10^{1}	7.83×10^{1}	5.21×10^{1}
F	0	2.20×10^{-6}	0	0	0	0	0
Na	1.00×10^{-3}	1.00×10^{-3}	6.30×10^{-1}	2.10×10^{-2}	3.50×10^{-2}	6.00×10^{-3}	1.30×10^{-2}
Mg	2.70×10^{-1}	2.70×10^{-2}	2.00×10^{-2}	5.00×10^{-1}	1.30×10^{-1}	2.30×10^{-2}	2.65×10^{-1}
P	4.80×10^{-1}	2.20×10^{-2}	1.90×10^{-1}	7.30×10^{-1}	2.10×10^{-1}	5.70×10^{-2}	6.16×10^{-1}
S	3.00×10^{-1}	1.60×10^{-2}	3.00×10^{-1}	2.40×10^{-1}	1.30×10^{-1}	1.10×10^{-1}	2.08×10^{-1}
Cl	1.50×10^{-3}	1.50×10^{-3}	9.70×10^{-1}	3.20×10^{-2}	5.40×10^{-2}	9.20×10^{-3}	2.00×10^{-2}
K	7.10×10^{-1}	3.60×10^{-1}	1.50×10^{-1}	1.52×10^{0}	2.90×10^{-1}	4.20×10^{-1}	$4.12 \times 10{-}1$
Ca	2.60×10^{-1}	5.00×10^{-3}	1.80×10^{-1}	1.30×10^{-1}	7.00×10^{-3}	1.20×10^{-2}	4.26×10^{-2}
Mn	2.30×10^{-3}	3.00×10^{-4}	0	3.80×10^{-3}	5.00×10^{-4}	2.00×10^{-4}	6.90×10^{-3}
Fe	3.70×10^{-3}	3.00×10^{-4}	5.00×10^{-4}	1.40×10^{-2}	2.70×10^{-3}	8.00×10^{-4}	2.7×10^{-3}
Cu	1.00×10^{-3}	1.00×10^{-4}	0	3.80×10^{-3}	3.00×10^{-4}	1.00×10^{-4}	5.00×10^{-4}
Zn	3.10×10^{-3}	2.00×10^{-4}	2.40×10^{-3}	6.80×10^{-3}	2.00×10^{-3}	3.00×10^{-4}	3.70×10^{-3}
Se	2.50×10^{-6}	1.00×10^{-5}	1.45×10^{-4}	1.43×10^{-4}	1.55×10^{-4}	3.00×10^{-7}	4.33×10^{-5}

Table 6.1: The elemental composition of the foods simulated. The relative mass per 100 g of food for each element (3 s.f.) is given [8]

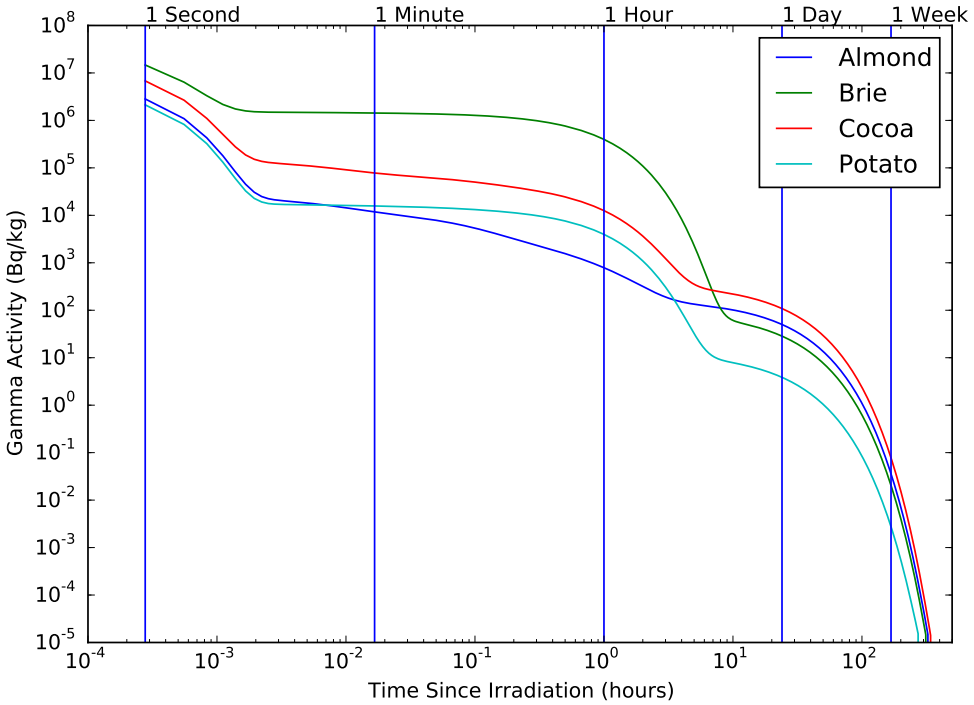


Figure 6.2: Decay in γ activity with time starting immediately after irradiation and continuing to 83 hours (3.5 days) after irradiation. The γ activity for Almond, Brie, Cocoa and Potato irradiated by a 14 MeV neutron source is shown in (Bq/kg).

The sodium content of Brie is 629 mg per 100 g of cheese and so $^{23}Na(n,\gamma)^{24}Na$ reactions will contribute significantly to the activation. In foods with low levels of salt and magnesium the production of ^{24}Na by $^{23}Na(n,\gamma)^{24}Na$ and $^{24}Mg(p,n)^{24}Na$ reactions is never a dominant process. In the case of Bananas and Potatoes the activity from ^{42}K is higher than that of ^{24}Na on the time scales where the activity of ^{24}Na is dominant in other foods.

Figure 6.3 shows the energy dependence of γ activity immediately after irradiation for Almond, Brie, Cocoa, and Corn. The results of the other foods fall below the data of Cocoa also with low energy dependence. The energy dependence of Almond, Cocoa and Corn is very small and may not be experimentally measurable; however, Brie shows a very strong energy dependence of nearly an order of magnitude across the energy range. As with the results shown in figure 6.2 the uncertainties are only available for the total activity; however, in this case they vary with energy, low energy results have higher uncertainties peaking at 24.7%, in Brie at 1 MeV, and dropping to 10.7%, in Cocoa, at 20 MeV. The energy dependence shown in figure 6.3 is representative of all samples and continues with time but decreases and becomes inconsequential after the first 1 to 2 hours.

The results in figure 6.4 show the decay in ingestion dose of Almond, Brie, Cocoa and Corn. As with the γ activity shown in figure 6.2 the ingestion dose shown in figure 6.4 indicates a strong food dependence. The uncertainties are shown by the faint lines bracketing the thicker lines.

Along with the magnitude of the activity and the ingestion dose discussed previously the time required for a sample to return to background should be considered. Figure 6.5 shows the time required for the ingestion dose of Brie, Cocoa, Corn and Rice samples to return to within 5% of pre-irradiation levels,

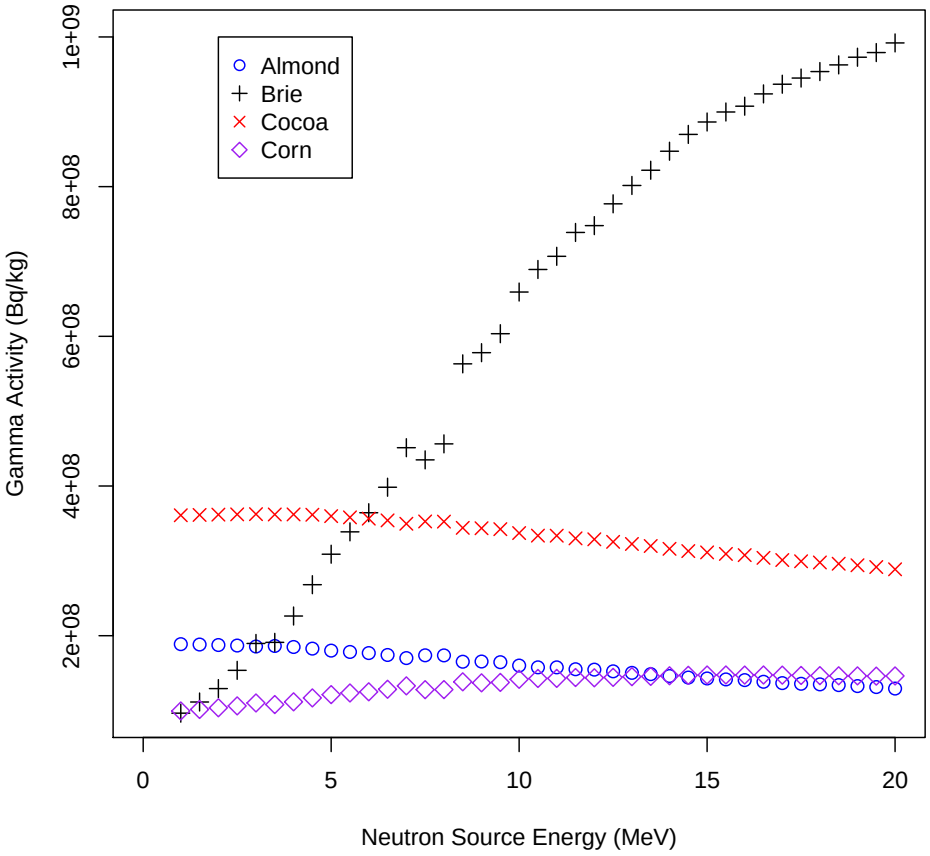


Figure 6.3: The dependence of the γ activity of Almond, Brie, Cocoa and Corn on the energy of the irradiating neutron source, neutron energy ranging from 1 MeV to 20 MeV. Almond, Cocoa and Corn show insignificant dependence but Brie shows approximately an order of magnitude variation.

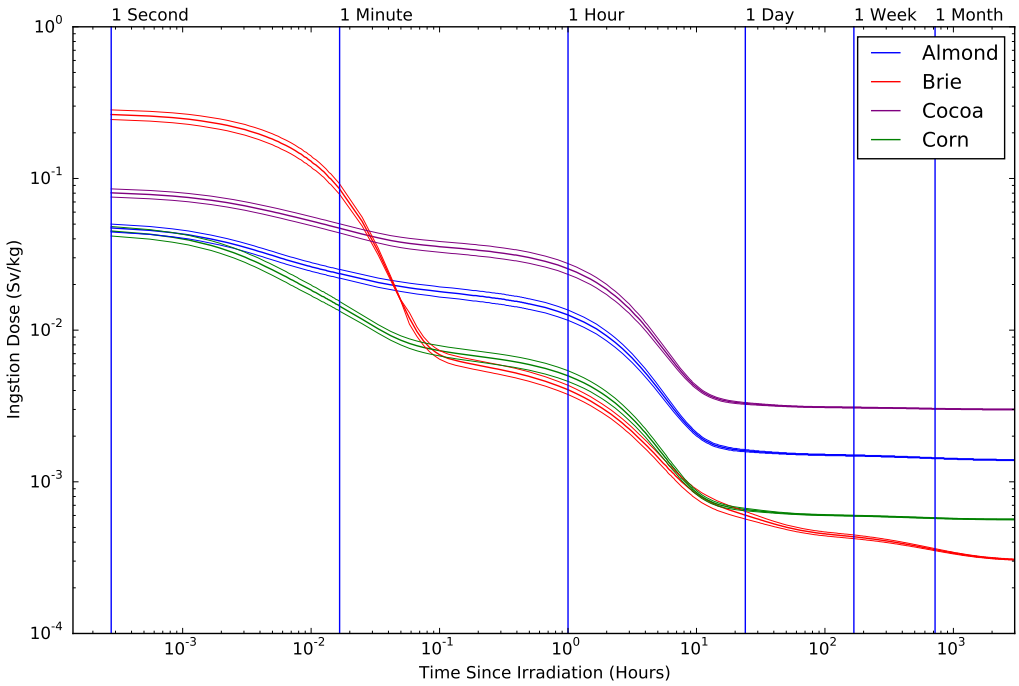


Figure 6.4: Decay with time of the ingestion dose induced in Almond, Brie, Cocoa and Corn under 14 MeV neutron irradiation shown in Sv/kg. The decay is shown starting 10^{-5} days (0.8 s) after irradiation through to 30 days (1 month) after. The uncertainties are given by the faint lines which bracket each main line.

the other foods show minimal energy dependence and fall below the level of Rice. The ingestion TtB shows a strong food dependence and a weak energy dependence. In descending order the means of the ingestion TtBs are: Brie, 106 days; Almond, 30 days; Rice, 26 days; Corn, 18 days; Cocoa, 4 days; Potato, 4 days; Banana, 1 Day varying by approximately 10% across the energy range used. The TtB for activity of each sample was comparable to that of the ingestion dose but the energy dependence was weaker.

The uncertainties in the simulations are dominated by the cross sections used in Fispact-II. For these simulations the 616 group EAF-2010 neutron activation cross sections [10] were used. Reactions with unknown cross sections are calculated numerically resulting in some isotopes having very large uncertainties. Figure 6.6 shows the experimental cross section for the $^{14}N(n,p)$ reaction from the ENDF and TENDL libraries overlaid with the simulated neutron spectrum produced by a 14 MeV beam passing through 90 cm of Brie. Where the ENDF data is available the two libraries are in agreement as the TENDL library combines experimental data with calculated where available. Where there is no experimental data available, above approximately 7 MeV, the calculated cross-section must be used, which does not always represent reality leading to greater uncertainty in the results.

The molecular structure in which the produced radioisotopes are found is not considered in the Fispact-II simulations when calculating ingestion dose. The molecule a radioisotope is part of can have a significant impact on the biological half-life, and therefore the radiotoxicology [11]. The biological half-life may have a dramatic effect on the results presented; however, significant research would need to be performed to determine if that was the case.

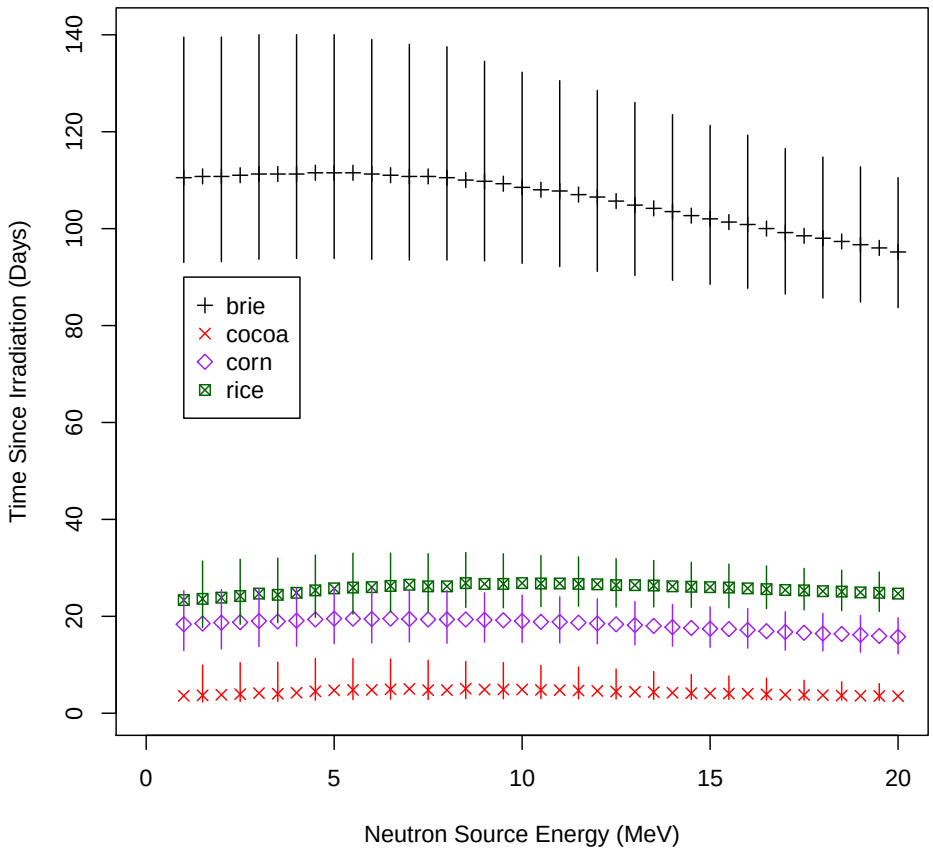


Figure 6.5: The time required for the ingestion dose of four of the irradiated samples to return to background after irradiation by a neutron beams ranging from 1 MeV to 20 MeV. Background is taken as the ingestion dose prior to irradiation and the plot shows time in days required to reach background+5%. The large uncertainty in the Brie result is due to the presence of isotopes with large uncertainties in their production cross-section.

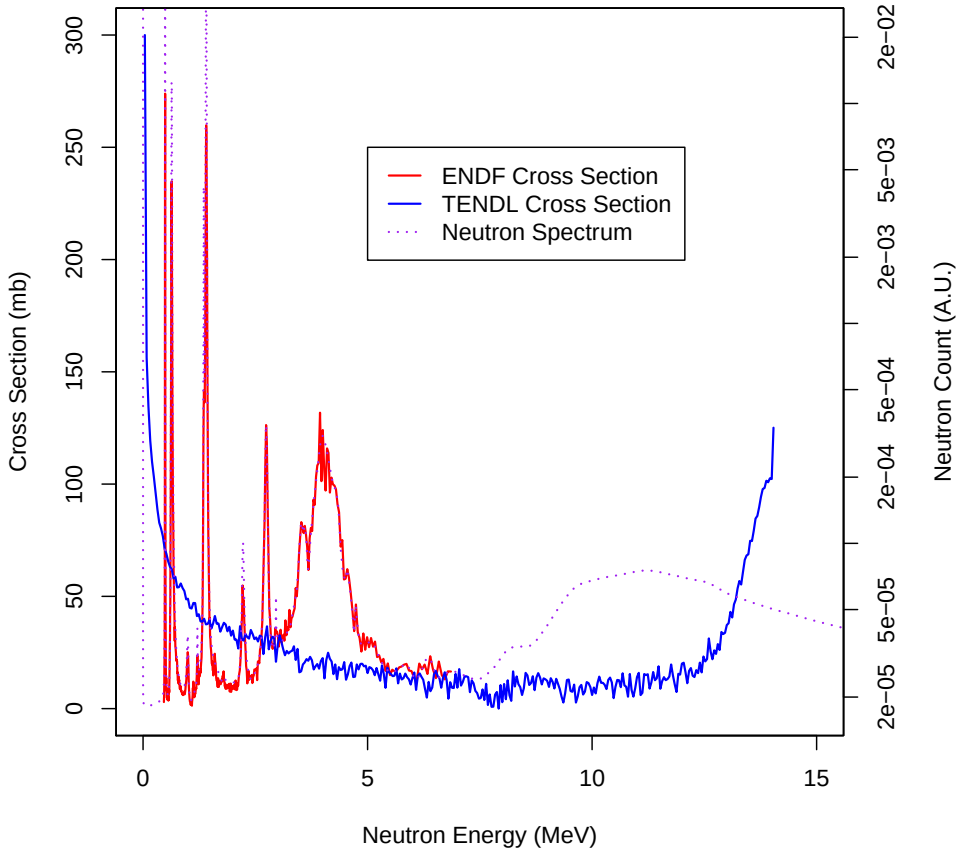


Figure 6.6: The $^{14}N(n,p)$ cross section provided by the ENDF and TENDL libraries overlaid with the neutron spectrum after propagation through 90 *cm* of Brie.

6.4 Conclusion

This chapter has shown that the composition of food has a strong effect on three relevant metrics; Time to Background (TtB), ingestion dose and activity, and that the energy of the irradiating neutrons can also have an effect. The precise effect of the neutron irradiation is strongly dependent upon both the food being irradiated and the time since irradiation.

Previous research [2, 4] has claimed that the production of ^{24}Na through $^{23}Na(n,\gamma)^{24}Na$ and $^{24}Mg(n,p)^{24}Na$ reactions in irradiated food is the only concern. In the case of Banana and Potato, which have a very low salt content, the gamma activity cannot be explained through ^{24}Na but is in fact due to ^{42}K predominantly from $^{41}K(n,\gamma)^{42}K$ reactions, though others are also contributors.

The activation of food is more strongly dependent on composition than neutron energy; however, the radiological risk may be reduced in some cases with careful selection of irradiation energy. A variable-energy neutron source could enable safer security scanning, with neutron energies tailored to have the lowest possible impact on a given food.

Although the results in this chapter show that varying neutron energy has an impact on the three figures of merit considered (Time to Background, Activity and Ingestion Dose) it is clear that the type of food being irradiated has a much bigger impact.

Due to the accuracy limitations of the numerical models, in particular those used in Fispact-II, it is essential that experimental work is performed to verify the results in this chapter. In addition to the limitations of the numerical models for production of radionuclides, the resulting ingestion dose is also not a triv-

ial problem. Puncher [12] did a detailed analysis of the uncertainties associated with inhalation and ingestion doses. This chapter has shown that there is an energy dependence for activation of food however all other containerised goods will potentially be irradiated by neutrons as part of a security system. An understanding of how other goods, e.g. clothes, computers and machinery are affected by neutrons, and what role the energy plays in that effect would be beneficial. In addition it is vital that experimental work be used to extend the results presented here. There are two directions that future work needs to take, the first is to increase the availability of neutron activation cross-sections, allowing improved simulations. Experimental verification of simulations is also vital and must be done before any decision can be made on the safety of neutron interrogation.

Bibliography

[1] J. Valentin, editor. *The 2007 Recommendations of the International Commission on Radiological Protection.* Elsevier, 2007.

[2] T. Tenforde. *Letter report on radiation protection advice for pulsed fast neutron analysis system used in security surveillance.* Presidential Report, `http://www.dhs.gov/xlibrary/assets/nepa/Mgmt_NEPA_NCRP_Report_1.pdf`, 2002.

[3] T. Tenforde. *Presidential report on radiation protection advice for the pulsed fast neutron system used in security surveillance: Part II. The ALARA principle and related issues.* Presidential Report, `http://www.ncrponline.org/Publications/Pres_Reports/SC%201-11%28II%29.PDF`, 2003.

[4] E. Giroletti, G. Bonomi, A. Donzella, G. Viesti, and A. Zenomi. Radiological risks from irradiation of cargo contents with EURITRACK neutron inspection systems. *Radiation Physics and Chemistry*, 81:758–765, 2012.

[5] M. Nelson. *Investigation into the Feasibility of Highly Enriched Uranium Detection by External Neutron Stimulation (Expanded Study).* De-

fense Threat Reduction Agency, http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA452712, 2006.

[6] R. Forster, L.Cox, R. Barrett, T. Booth, J. Briesmeister, F. Brown, and J. Bull. MCNP version 5. *Nuclear Instruments and Methods in Physics Research B*, 213:82–86, 2004.

[7] J. Sublet, J. Eastwood, and J. Morgan. *The Fispact-II User Manual*, 2012.

[8] *Composition of Foods Raw, Processed, Prepared USDA National Nutrient Database for Standard Reference, Release 21.* U.S. Department of Agriculture, http://www.ars.usda.gov/News/docs.htm?docid=18880, 2008.

[9] H. Torabizadeh. All proteins have a basic molecular formula. *World Academy of Science, Engineering and Technology*, 54:961–965, 2011.

[10] J. Sublet, L. Packer, J. Kopecky, R. Forrest, A. Koning, and D. Rochman. *The European Activation File: EAF-2010 Neutron-Induced Cross Section Library.* https://t2.lanl.gov/nis/data/jeff/EAF2010.pdf, 2010.

[11] H. Takeda. Comparative metabolism of tritium in rat after single ingestion of some tritiated organic compounds versus tritiated water. *Journal of Radiation Research*, 23:345–357, 1982.

[12] M. Puncher. An assessment of the reliability of dose coefficients for intakes of radionuclides by members of the public. *Journal of Radiological Protection*, 34:625–643, 2014.

Chapter 7

Isotopic Analysis

7.1 Introduction

Chapter 6 showed that the composition of irradiated foods can strongly affect the resulting activity and ingestion dose. This chapter covers the cause for this variation, which is the radioisotopes produced under neutron irradiation.

The authors of [1] showed that under 8.5 MeV neutron irradiation the dominant threat isotope in pharmaceuticals and medical devices was ^{24}Na. The method used in [1] was to calculate the induced activity based on a spectrum with fast and thermal components rather than a full Monte-Carlo approach.

The results of [1] were extended in [2] to include ^{24}Na production in food. As with [1] the conclusion of [2] was that no unacceptable level of activation would be seen. To determine if the induced activity would pose a problem the authors of [1, 2] calculated the ingestion dose for irradiated goods. Tenforde [2] set the acceptable dose received by the public as a result of irradiation at 1 mSv/year,

the greatest dose calculated was 1 μ
Sv/year.

The threat isotopes considered by [2] may not be readily applied to all foodstuffs as the compositions of foods and pharmaceuticals are not necessarily equivalent. Additionally the target elements considered by [1] have omissions which may not matter for pharmaceuticals but may be significant for some foods. Activation reactions based on $Ca(n, X)$ were not included in [1], but Ca is found in significant quantities in a variety of foods, including dairy and tofu.

The results of [2] were further extended by the authors of [3] who considered ^{24}Na production by 14 MeV neutron irradiation. As with [2] only ^{24}Na production was considered; however, the increased neutron energy may enable additional reactions and the applicability of pharmaceuticals as an analogue of food is still to be verified.

This chapter shows that the induced activity and ingestion dose are caused by a variety of isotopes. Furthermore it is shown that the conclusion that ^{24}Na is the dominant threat isotope is only valid under certain conditions. Finally it is shown that some of the produced isotopes have an energy dependence, which may justify a detailed investigation into the optimal source energy for different irradiated goods.

7.2 Simulations

This chapter expands upon the work reported in Chapter 6, and the simulations used were the same. The results obtained from these simulations are specific to the flux and fluence used. Changing the flux and/or fluence will influence the

levels of activation after irradiation. For example at higher flux the interrogation time will be reduced meaning that short lived isotopes will be at higher levels immediately after irradiation.

The foods used were Almond, Banana, Brie, Cocoa, Corn, Potato and Rice, which were chosen as they cover a broad variety of compositions and are commonly containerised for import/export. The elemental ratios used in these simulations was chosen to approximate the average composition of distributed foods. The elemental composition will vary with country of origin and cellulose is not the only organic component; however, these approximations are sufficient to highlight any significant effects.

7.3 Results

The results presented here consider only the non-natural isotopes in each food. The contribution from naturally occurring radioisotopes is not considered, for example ^{40}K, as only activation above background is a concern for public health. The produced isotopes dominating both the total activity and the total ingestion dose under 14 MeV irradiation is shown as well as how the production of these isotopes varies with energy. As in chapter 6 the applied flux and fluence were 10^8 n/cm^2/s and 10^9 n/cm^2 respectively.

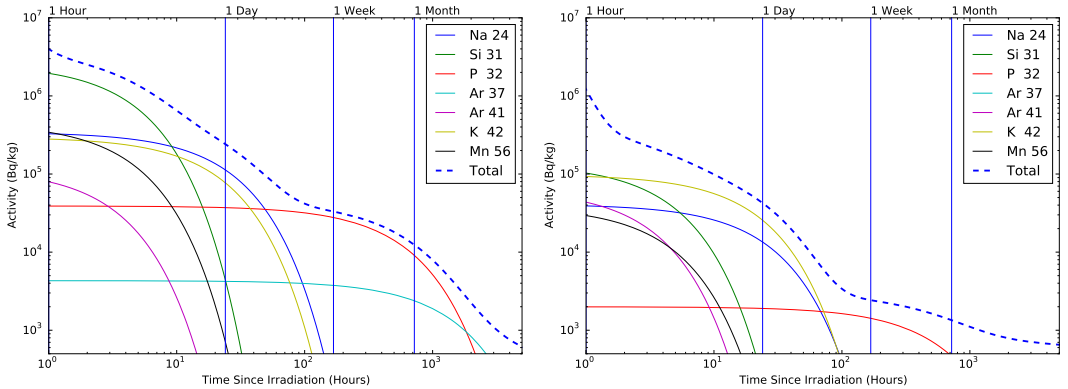
The production of ^{24}Na through $^{23}Na(n,\gamma)^{24}Na$ and $^{24}Mg(n,p)^{24}Na$ reactions were identified by [3, 2] as the greatest threat to health when irradiating food for security. In this study the highest Na content was found in Brie with 629 mg per 100 g and the highest Mg content was in Cocoa Powder with 499 mg

per 100 g. The Na and Mg content of Banana is very low at 1 mg and 27 mg per 100 g respectively.

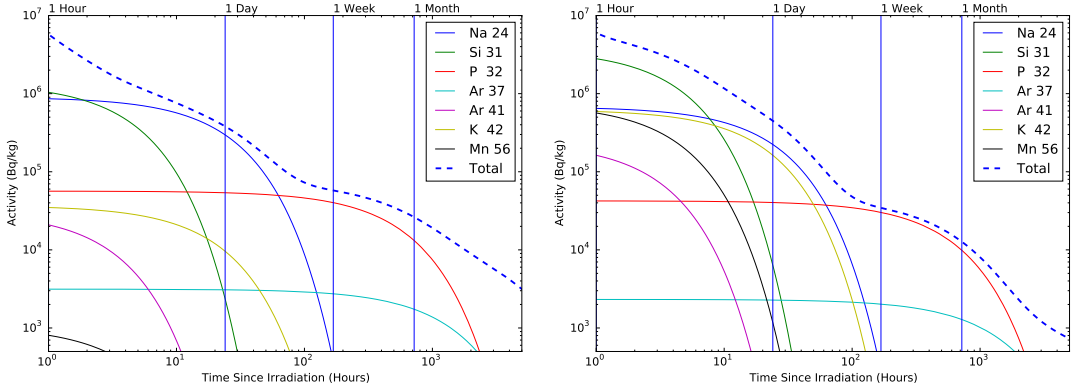
Figures 7.1 and 7.2 show the dominant contributors to the induced activity from 1 hour after irradiation for the 7 foods. Low activity and short lived isotopes are omitted to allow the most significant isotopes to be seen. The omission of ^{15}N causes a large spike in the initial activity in figures 7.1b and figure 7.1c. The Brie results (figure 7.1c) show strong ^{24}Na dominance from approximately 5 to 75 hours after irradiation, with other isotopes dominating outside this range. Corn (figure 7.2a) and Rice (figure 7.2c) also show a ^{24}Na dominance, for slightly less time than Brie. Finally Almond (figure 7.1a), Banana (figure 7.1b), Cocoa (figure 7.1d) and Potato (7.2b) show a very weak ^{24}Na dominance, or no dominance at all, with ^{42}K having comparable or greater activity for a significant time.

The induced activities show that, as previously claimed in [1], ^{24}Na can be the most important isotope; however, this only applies to a narrow time window and for foods high in Na and/or Mg. In addition whilst ^{24}Na is dominant in a lot of cases there is very little difference between it and ^{42}K. Whilst the activity is an important consideration, and a useful parameter to measure, the most important consideration for foods is the ingestion dose. As there is no direct relationship between the activity and ingestion dose of a nuclide the ingestion dose was taken from the results of the Fispact-II simulations.

The ingestion dose of the dominant produced radioisotopes is shown for the seven foods in figures 7.3 and 7.4. The ingestion dose results show that whilst ^{24}Na can be very significant for the activity its significance is considerably reduced in the ingestion dose. There is a visible ^{24}Na dominance in Brie (figure 7.3c), Cocoa (figure 7.3d), Corn (figure 7.4a) and Rice 7.4a); however, in all 4

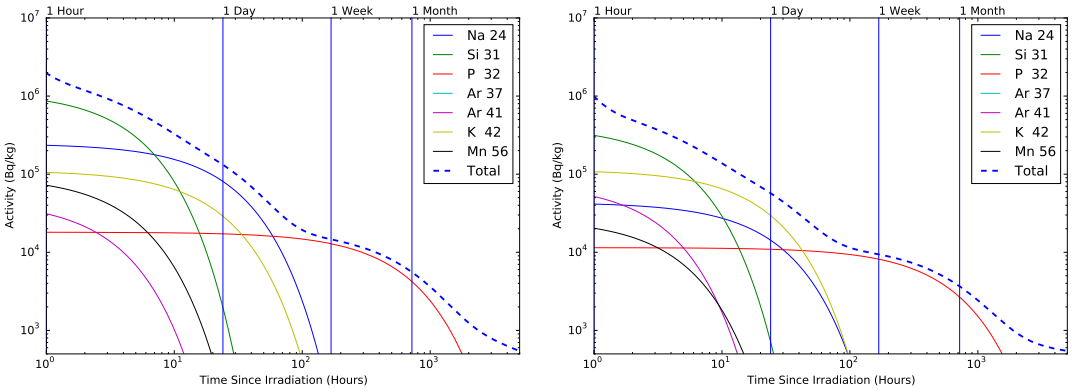


(a) Almond **(b)** Banana



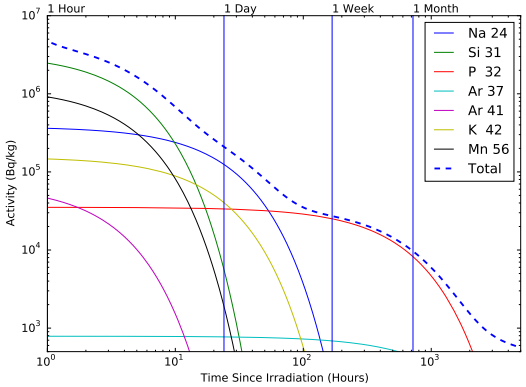
(c) Brie **(d)** Cocoa

Figure 7.1: The activity contribution of ^{24}Na, ^{31}Si, ^{32}P, ^{37}Ar, ^{41}Ar, ^{42}K and ^{56}Mn from 1 hour after irradiation of Almond, Banana, Brie and Cocoa by a 14 MeV neutron source, the total induced activity is also shown. The most significant contributors to the totals are shown, short half-life and low activity isotopes are omitted. The large activity at the start of Banana and Brie is caused by ^{15}N, which is omitted due to its short half-life.



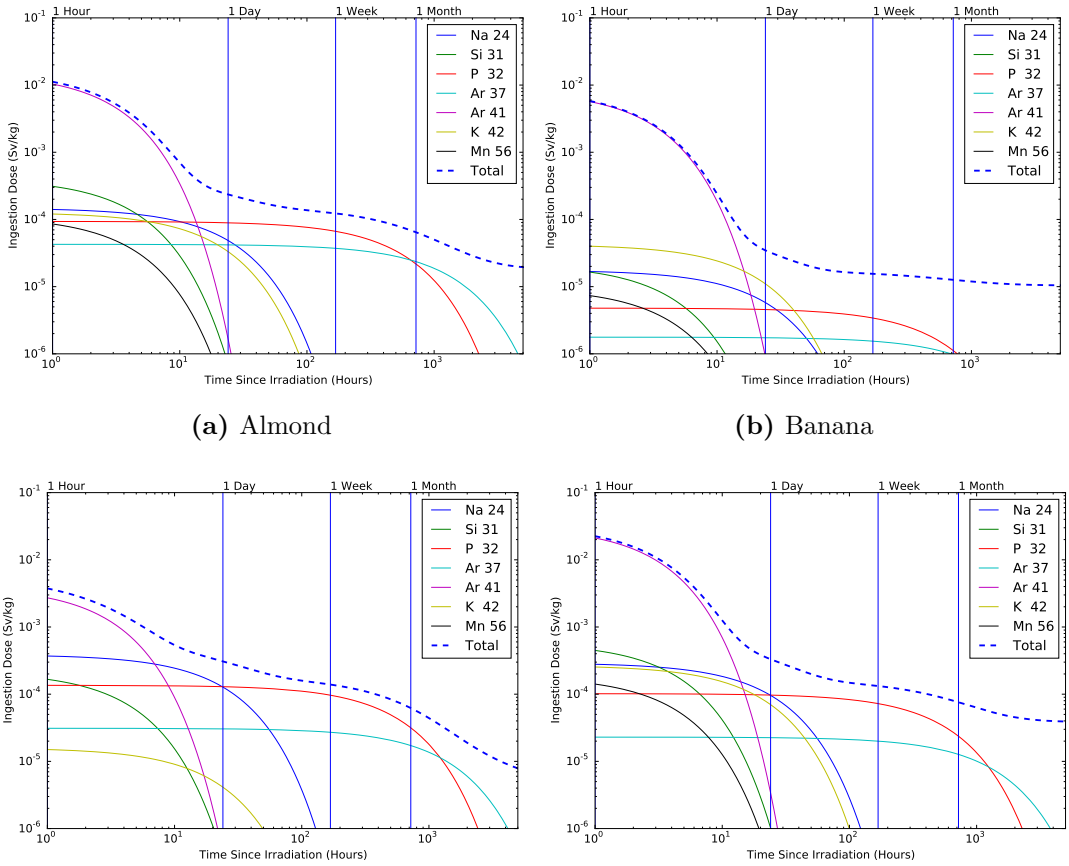
(a) Corn

(b) Potato



(c) Rice

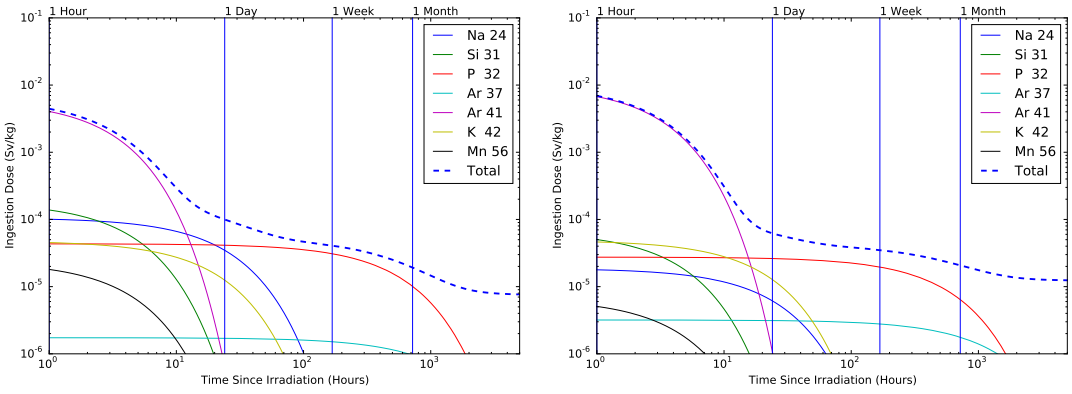
Figure 7.2: The activity contribution of ^{24}Na, ^{31}Si, ^{32}P, ^{37}Ar, ^{41}Ar, ^{42}K and ^{56}Mn from 1 hour after irradiation of Corn, Potato and Rice by a 14 MeV neutron source, the total induced activity is also shown. The most significant contributors to the totals are shown, short half-life and low activity isotopes are omitted.



(a) Almond **(b)** Banana

(c) Brie **(d)** Cocoa

Figure 7.3: The ingestion dose contribution of ^{24}Na, ^{31}Si, ^{32}P, ^{37}Ar, ^{41}Ar, ^{42}K and ^{56}Mn from 1 hour after irradiation of Almond, Banana, Brie and Cocoa by a 14 MeV neutron source, the total induced induced ingestion dose is also shown. The most significant contributors to the totals are shown, short half-life and low ingestion dose isotopes are omitted.



(a) Corn

(b) Potato

(c) Rice

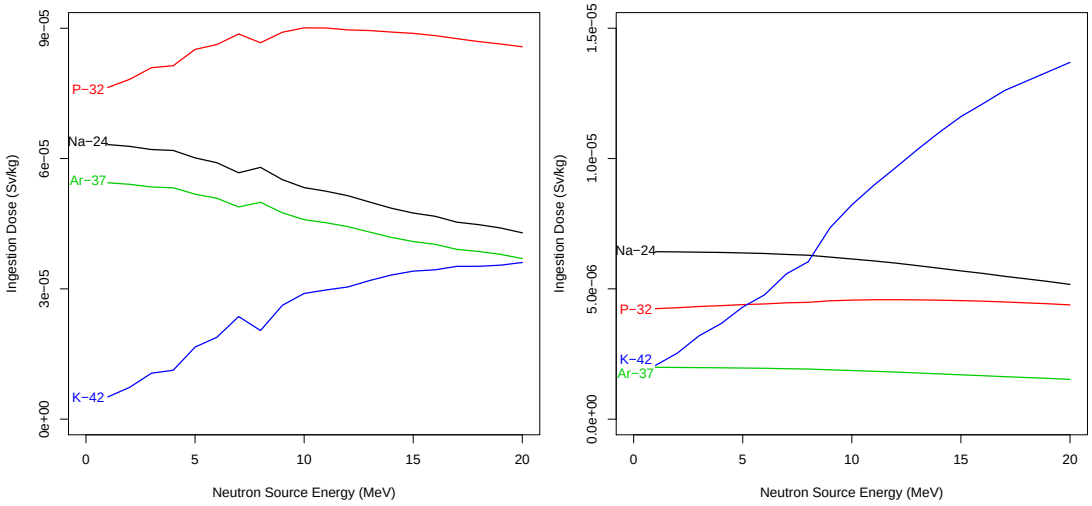
Figure 7.4: The ingestion dose contribution of ^{24}Na, ^{31}Si, ^{32}P, ^{37}Ar, ^{41}Ar, ^{42}K and ^{56}Mn from 1 hour after irradiation of Corn, Potato and Rice by a 14 MeV neutron source, the total induced induced ingestion dose is also shown.The most significant contributors to the totals are shown, short half-life and low ingestion dose isotopes are omitted.

cases the dominance is much weaker and shorter lived than in the case of activity. Particular attention should be paid to the ingestion doses of ^{41}Ar and ^{32}P in comparison to ^{24}Na. Whilst the activity of ^{24}Na is significantly higher the difference in the ingestion doses is much smaller.

The production of ^{41}Ar is primarily through $^{41}K(n,p)$ and $^{44}Ca(n,\alpha)$ reactions, with a small contribution from other interactions. The original calculations used in [1] considered only $^{41}K(n,\gamma)^{42}K$ reactions for K isotopes and did not use any reactions from Ca. As can be seen in figures 7.3 and 7.4 their is a very significant contribution to the ingestion dose from ^{41}Ar for approximately 10 hours after irradiation. The production of ^{41}Ar demonstrates the importance of including all isotopes in the calculations of produced isotopes. The limited considerations of K and Ca reactions may have been reasonable for [1]; however, when considering activation of food, as in [2] and [3], it is evidently problematic.

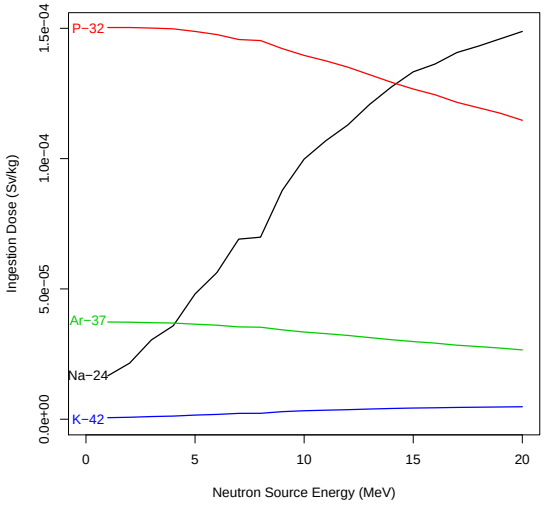
The number of produced isotopes is proportional to the cross-section(s) for the reaction(s) producing that isotope. As a range of energies were simulated it is possible to observe any significant energy dependence in the resulting ingestion dose and activity. The results in figures 7.5 and 7.6 show the energy dependence of the ingestion doses for dominant produced isotopes in the seven foods 24 hours after irradiation. After 24 hours many isotopes have decayed to essentially 0 leaving the longer half-life isotopes, mainly ^{24}Na, ^{32}P, ^{37}Ar and ^{42}K.

The results in figures 7.5 and 7.6 show the energy dependence for the dominant ingestion dose contributing isotopes in the seven foods. The energy dependencies again highlight the effect of composition already demonstrated in figures 7.1, 7.2, 7.3 and 7.4. Banana (figure 7.5b), Corn (figure 7.6a) and Potato (figure 7.6b) show minimal energy dependence in the ingestion dose of ^{24}Na, ^{32}P, ^{37}Ar with

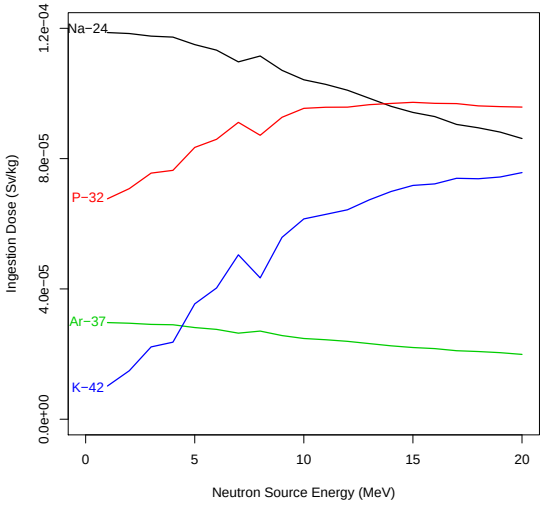


(a) Almond

(b) Banana

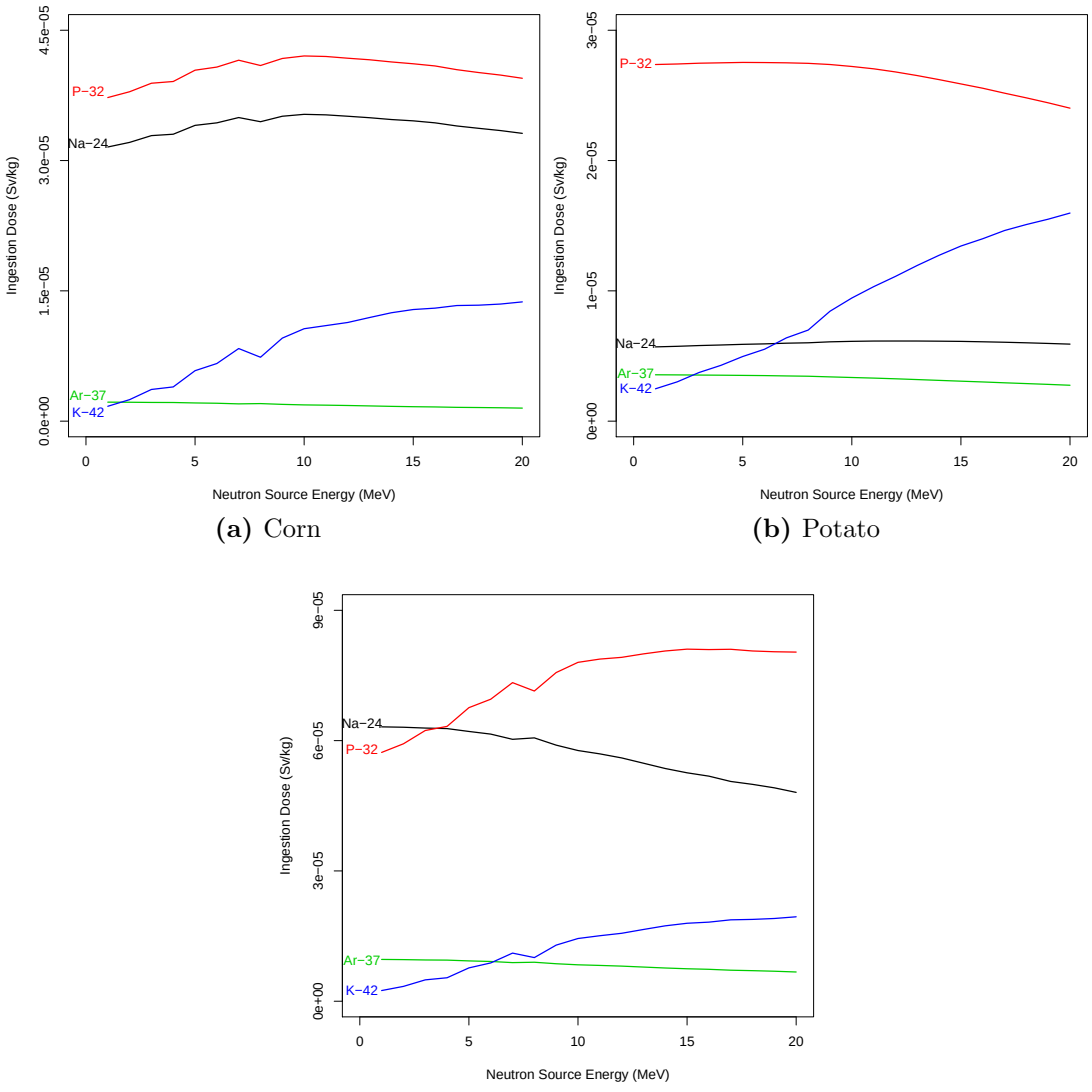


(c) Brie



(d) Cocoa

Figure 7.5: Energy dependence of the ingestion dose of ^{24}Na, ^{32}P, ^{37}Ar and ^{41}K 24 hours after irradiation in Almond, Banana, Brie and Cocoa.



(a) Corn

(b) Potato

(c) Rice

Figure 7.6: Energy dependence of the ingestion dose of ^{24}Na, ^{32}P, ^{37}Ar and ^{41}K 24 hours after irradiation in Corn, Potato and Rice.

an energy dependence for the level of ^{41}K. Almond (figure 7.5a), Brie (figure 7.5c), Cocoa (figure 7.5d) and Rice (figure 7.6c) all show a significant energy dependence in the levels of ^{24}Na and ^{32}P, which in the case of Brie, Cocoa and Rice is sufficient that changing the irradiation energy causes first one to be dominant and then the other.

The differing energy dependences shown in figures 7.5 and 7.6 demonstrate the potential influence of irradiation energy in an interrogation system. Whilst the energy dependence is not universal it does suggest that careful consideration of the source energy may enable reduced exposure of the public to additional radiation.

7.4 Conclusion

The conclusion that ^{24}Na is the primary threat isotope was based on research into induced activity in pharmaceuticals and medical devices under 8.5 MeV irradiation [1]. Food was first considered by Tenforde [2] at 8.5 MeV, and later Giroletti [3] at 14 MeV. The applicability of pharmaceuticals as an analogue of food was not considered, nor was the potential for 14 MeV neutrons to stimulate different reactions, and have different cross-sections, to 8.5 MeV neutrons considered.

This chapter has shown that there are multiple isotopes influencing both the activity and the ingestion dose of irradiated food and that ^{24}Na is only the dominant threat under certain conditions. Some reactions that were omitted from [1], such as $Ca(n, X)$ reactions, can be significant. Excluding Ca reactions may have been justified for pharmaceuticals and medical devices; however, as a

source of ^{41}Ar this has a significant effect on the ingestion dose for approximately 24 hours after irradiation.

The change from 8.5 MeV considered by Tenforde [1] to 14 MeV considered by Giroletti [3] will have an influence on the produced isotopes. The energy dependence of different isotopes, shown 24 hours after irradiation, indicates that a change in source energy necessitates a detailed analysis of the induced activity and ingestion dose.

The compositions of the foods considered has been shown to have a significant effect on the quantities of isotopes produced. Despite the composition of all foods being above 93% H, C and O there is still a significant influence from the trace elements.

Bibliography

[1] T. Tenforde. *Letter report on radiation protection advice for pulsed fast neutron analysis system used in security surveillance.* Presidential Report, `http://www.dhs.gov/xlibrary/assets/nepa/Mgmt_NEPA_NCRP_Report_1.pdf`, 2002.

[2] T. Tenforde. *Presidential report on radiation protection advice for the pulsed fast neutron system used in security surveillance: Part II. The ALARA principle and related issues.* Presidential Report, `http://www.ncrponline.org/Publications/Pres_Reports/SC%201-11%28II%29.PDF`, 2003.

[3] E. Giroletti, G. Bonomi, A. Donzella, G. Viesti, and A. Zenoni. Radiological risks from irradiation of cargo contents with EURITRACK neutron inspection systems. *Radiation Physics and Chemistry*, 81:758–765, 2012.

Chapter 8

Conclusion And Future Work

8.1 Conclusion

This thesis has investigated the production of neutrons for the interrogation of cargo and possible side effects. Neutron producing reactions that don't use or produce tritium have been shown to have potential for security, but more experimental data is needed. The production of radioisotopes has been shown to have a dependence on both the composition of the cargo and the energy of the irradiating beam. By looking at alternative neutron producing reactions the possibility of selecting the energy to suit the cargo is available.

Approximately 90%[1] to 95%[2] of the world's freight is transported by sea each year, with approximately 10% being interrogated. Single energy X-ray systems measure the line integral of the attenuation between source and detector, allowing a 2-dimensional image to be constructed showing the attenuation through the container. For a single energy system potential illicit goods are identified

by shape and variations in density, as a result it is relatively easy to shield or disguise objects.

An improvement on single energy X-rays is to use two separate energies and measure the ratios of the attenuation. Dual energy X-rays are able to infer whether a region of a container is dominated by metallic, organic or inorganic material; however, shielding and disguising objects is still possible. Another possibility is the use of back-scattered X-rays, which can measure the Z of materials in the surface and sub-surface of a container.

Whilst a variety of X-ray techniques exist there are significant limitations due to the nature of X-ray interactions with matter. The attenuation of X-rays has a near linear dependence on Z, simplifying shielding, whereas attenuation of neutrons has a highly non-linear dependence on Z. Further the materials most commonly used to shield against X-rays, such as Pb and Fe, give relatively low attenuation of neutrons.

In Chapter 3 some of the available neutron interrogation techniques were discussed, these give information ranging from 2-dimensional transmission radiography to a full 3-dimension reconstruction with material identification. Neutron interrogation techniques can be broadly divided into Neutron in/Neutron out (NiNo) and Neutron in/Photon out (NiPo) categories, depending upon the detected particle. Both categories have benefits NiNo techniques generally require lower fluxes, whereas NiPo techniques give better elemental identification. NiPo techniques and some NiNo techniques can be used to measure the elemental composition of a cargo, greatly improving threat detection and contents verification. A study on NiNo transmission imaging using a combination of neutron and γ transmission imaging was shown to give effective threat detection [3]. The NiPo

technique has also been demonstrated to be effective by a number of groups, most recently the EURITRACK collaboration [4].

In order for neutron interrogation to be utilised a suitable neutron source must be used. The majority of research so far has been conducted using either $T(d,n)$ (DT) sources or $D(d,n)$ (DD) sources. Both DT and DD sources produce high fluxes relatively efficiently; however, the presence of tritium, by design for DT and as a by-product for DD, makes them unsuitable for mass deployment.

In principle any combination of target and projectile could be used as a neutron source; however, only a small number would give suitable neutron emission. Imaging techniques reliant on fast neutrons and Time-of-Flight, such as Pulsed Fast Neutron Analysis, require a spectrum with the smallest FWHM possible. At present there is insufficient data to know what reactions could be used to replace DT and DD sources for this. Chapter 4 showed that $^{7}Li(d,n)$ or $^{19}F(d,n)$ could be suitable sources, but more experimental data is needed to know if they could be used.

When irradiating a material with deuterons of sufficient energy there will inevitably be a component of the neutron spectrum from deuteron dissociation. At present deuteron dissociation processes cannot be simulated with commonly available Monte-Carlo codes, such as Geant4 or MCNPX. Two models, high and low precision, have been written for inclusion in the Geant4 framework. Chapter 5 discusses the physics of deuteron dissociation and the operation of the numerical models. The models are intended for distribution within the relevant community and publication.

When neutrons interact with matter it is inevitable that some activation will occur. Activation is a concern for all cargo, but most significantly for food.

Since food is ingested it will do significantly more damage to the body than other sources of radiation. The ingestion dose and activation are not directly proportional, a high energy β will have a higher ingestion dose than a γ even if the activity is lower. In Chapter 6 it was shown that both the irradiating energy and the food composition can influence the resulting activity.

Since neutron interactions are strongly energy dependent it is unsurprising that in some cases the activity and ingestion dose of samples also shows an energy dependence. The activity and ingestion dose of a food is the result of the build up of radio-isotopes. In literature it is claimed that ^{24}Na is the dominant threat isotope; however, this is dependent on both energy and composition. Chapter 7 showed that other isotopes such as ^{32}P can be a greater threat due to the emission of a relatively high energy β and a longer half life.

The results presented in chapters 6 and 7 could be used to begin the process of determining acceptable limitations of neutron based interrogation. In particular the influence of target composition and irradiating energy, which have not been adequately explored in literature, should be considered when deciding on the legislation of neutron interrogation.

Both the ingestion dose received by the public, and the exposure of operatives and cargo handlers after interrogation must be considered. A detailed understanding will require modelling of at least the level of detail used in preparation of this thesis. The exposure of operatives will be primarily influenced by the production of short lived γ emitters, where as for ingestion long lived β emitters pose a greater threat.

Neutron interrogation of cargo has significant potential for improving the detection of illicit goods at border crossings. In order to deploy a neutron interroga-

tion system a suitable neutron source must be used. This thesis has shown that a variety of reactions have the potential to provide the necessary neutron beams, both with compound nucleus and deuteron dissociation reactions. The energy of the neutrons has an effect on both the threat detection but also the production of radioisotopes. By considering a range of possible neutron producing reactions it may be possible to maximise the detection of contraband whilst minimising radiotoxicology in both the neutron producing target and the irradiated goods.

8.2 Future Work

8.2.1 Neutron Source Energy Spectrum

There can be little doubt that neutron interrogation has significant potential for improving the prevention of illicit goods being smuggled across national borders. The majority of research into the use of these techniques has focused on the use of 14 MeV $T(d,n)$ sealed tube fusion sources, with a small number at lower energies with $D(d,n)$ sources of ^{252}Cf sources. The energy of a γ produced by inelastic neutron scattering is unrelated to the energy of the scattered neutron; however, the cross-section has a strong energy dependence and reaches 0 when $E_n < E_\gamma$.

If the energy dependence of γ emission is sufficient then it may be possible to irradiate a container at two different energies in a similar manner to dual energy X-ray interrogation. Using two spectra together may enable better background reduction and element identification.

Along with the inelastic scattering cross-section the total interaction cross-section for neutrons is also energy dependent. In general the total neutron in-

teraction cross section decreases with energy. For cargo interrogation this would result in greater penetration potentially improving material recognition further from the source.

8.2.2 Neutron Production Experiments

The work presented in this thesis has been performed entirely using simulation. Effective simulations are dependent upon accurate numerical models of the system being simulated, which is not always possible. When simulating nuclear interactions that are difficult to model numerically it is possible to compile experimental data into data libraries that can be used in place of models.

Low energy compound nucleus reactions do not have effective numerical models available, there are also large numbers of reactions with no data available. Deuteron induced reactions in particular have very little data available but are some of the most promising for production of high energy neutrons. In order to develop new novel neutron sources it is essential that more experimental work is carried out. Experimental work is necessary both to provide the data to populate data libraries but also to enable a better understanding of the reactions for model development.

8.2.3 Neutron Activation Experiments

The activation of materials is likely to be directly proportional to the neutron fluence used in interrogation. When used in a cargo interrogation setting it would be very beneficial to know the maximum fluence a material can safely receive. The maximum safe fluence will depend on multiple factors, the sum of the activation

cross-sections for a given material, the storage time before it is given to consumers, and how it is used. Use and storage time will be the most significant, anything ingested or inhaled must have a lower safety threshold than goods which are not, and those items which are in close proximity to a person must have lower thresholds. In addition the longer something is stored the higher the initial activation may be as the increased storage time will allow a greater reduction in the activity.

The total number of radioisotopes produced by a given neutron fluence should be independent of the flux, however the flux will affect the activity after irradiation. A lower flux will result in a longer scan time allowing short lived isotopes to decay during the scan. A longer scan will cause a lower activity immediately after irradiation, the reverse being true for high fluxes. After irradiation it is possible that goods will need to be stored in a hot cell until the activity has dropped below a safe threshold. Since people cannot approach a container during irradiation, or whilst it is in the hot cell the time between the start of irradiation to radiation being below the safe threshold is what is most important, and this may be flux dependent.

The results presented in this thesis were entirely dependent upon the activation cross-sections for neutron reactions being known, which in some cases they are not. An increased level of detail in the neutron activation cross-sections is essential for cargo interrogation research, but also other areas such as next generation nuclear reactors.

Bibliography

[1] A. Bergantino, E. Musso, and F. Porcelli. Port management performance and contectual variables: Which relationship? methodological and empirical issues. *Research in Transportation Business and Management*, 8:39–49, 2013.

[2] G. Boghen, A. Donzella, V. Filippini, A. Fontana, M. Lunardon, S. Moretto, S. Pesente, and A. Zenoni. MCNP calculations for container inspection with tagged neutrons. *Nuclear Instruments and Methods in Physics Research B*, 241:831–834, 2005.

[3] J. E. Eberhardt, S. Rainey, R. J. Stevens, B. D. Sowerby, and J. R. Tickner. Fast neutron radiography scanner for the detection of contraband in air cargo containers. *Applied Radiation and Isotopes*, 63:179–188, 2005.

[4] J.Obhodas et al. Analysis of containerized cargo in the ship container terminal. *Nuclear Instruments and Methods in Physics Research A*, 619:460–466, 2010.

Appendix A

Derivations

This appendix contains the derivations used in the preparation of this thesis.

A.1 Compound Nucleus Source Neutron Energy Approximation

If the target nucleus is considered at rest until the neutron is emitted and we ignore relativistic effects the kinetic energy of a neutron produced in an $A(x,n)Y$ reaction can be approximated by

$$E_n \approx \frac{E_p + Q}{1 + \frac{M_n}{M_{DN}}} \tag{A.1}$$

E_n is the neutron kinetic energy, E_p is the proton kinetic energy, Q is the Q of the reaction, M_n is the neutron rest mass and M_{DN} is the decay nucleus rest mass. As is shown in the relevant chapters this approximation is sufficient in the

low energy ($E_p \leq 10\ MeV$) cases considered. A proton with kinetic energy of 10 MeV has velocity $\beta \approx 0.144c$.

To derive equation A.1 we assume all of the proton kinetic energy has been transferred to the target nucleus and it remains stationary in an excited state with energy

$$E_t = Q + E_p + M_n + M_{DN} \tag{A.2}$$

E_t is the total energy of the system, which is the masses of the decay nucleus and neutron, the Q of the reaction and the proton kinetic energy. After the excited state decays we have two ejectiles together with energy

$$E_t = M_n + E_n + M_{DN} + E_{DN} \tag{A.3}$$

with energy being conserved between the two states, therefore

$$Q + E_p + M_n + M_{DN} = M_n + E_n + M_{DN} + E_{DN} \tag{A.4}$$

subtracting the masses from each side gives

$$Q + E_p = E_n + E_{DN} \tag{A.5}$$

leaving us with Q and E_p as knowns and E_n and E_{DN} as unknowns. Whilst the kinetic energies of the ejectiles are not known the magnitudes of their momenta must be equal therefore

$$M_n \times v_n = M_{DN} \times v_{DN} \tag{A.6}$$

or

$$\frac{M_n \times v_n}{M_{DN} \times v_{DN}} = 1 \tag{A.7}$$

from the momentum we can calculate the ratio of the kinetic energies as

$$\frac{E_n}{E_{DN}} = \frac{\frac{1}{2} \times M_n \times v_n \times v_n}{\frac{1}{2} \times M_{DN} \times v_{DN} \times v_{DN}} \tag{A.8}$$

if we ignore relativistic effects. If we consider equation A.6 we can rewrite A.8 as

$$\frac{E_n}{E_{DN}} = \frac{\frac{1}{2} \times M_{DN} \times v_{DN} \times v_n}{\frac{1}{2} \times M_n \times v_n \times v_{DN}} \tag{A.9}$$

which simplifies to

$$\frac{E_n}{E_{DN}} = \frac{M_{DN}}{M_n} \tag{A.10}$$

telling us that the ratio of the kinetic energies is equal to the inverse ratio of the masses. Returning to equation A.5 we can rearrange as

$$\frac{Q + E_p}{E_n} = 1 + \frac{E_{DN}}{E_n} \tag{A.11}$$

into which we can substitute A.9 giving

$$\frac{Q + E_p}{E_n} = 1 + \frac{M_n}{M_{DN}} \tag{A.12}$$

and rearranging the terms gives

$$E_n = \frac{Q + E_p}{1 + \frac{M_n}{M_{DN}}} \tag{A.13}$$

however as momentum conservation has been neglected the equals sign is correctly written as an approximation as in equation A.1.

A.2 Deuteron Dissociation Equations

A number of the equations used in the simulation of deuteron dissociation had to be derived, those derivations are presented here.

A.2.1 Hulthen Function Integration

This section gives the integration of the Hulthen function, used in the deuteron dissociation models. The Hulthen function was used in the form

$$N(P) \propto P^2 \left(\frac{1}{\alpha^2 + P^2} - \frac{1}{\beta^2 + P^2} \right)^2, \tag{A.14}$$

To solve this numerically it was necessary to use Newton's method, which required the integration of this function, given by

$$\begin{aligned}\int N(P)dP =& \frac{4\alpha\beta(\alpha+\beta)}{\pi(\alpha-\beta)^2} \times \Bigg[\\ & \left(\frac{1}{2\alpha} + \frac{2\alpha}{\beta^2-\alpha^2}\right) tan^{-1}\frac{P}{\alpha} + \\ & \left(\frac{1}{2\beta} - \frac{2\beta}{\beta^2-\alpha^2}\right) tan^{-1}\frac{P}{\beta} - \\ & \frac{1}{2}\left(\frac{P^2}{P^2+\alpha^2} + \frac{P^2}{P^2+\beta^2}\right)\Bigg].\end{aligned} \tag{A.15}$$

To integrate equation A.14 it is necessary to include a constant of proportionality (A) and define the probability (P) of a given momentum p as

$$P(p) = AP^2\left(\frac{1}{\alpha^2+p^2} - \frac{1}{\beta^2+p^2}\right)^2. \tag{A.16}$$

From equation A.16 the integral can be defined as

$$\int P(p)dp = A\int P^2\left(\frac{1}{\alpha^2+p^2} - \frac{1}{\beta^2+p^2}\right)^2 dp, \tag{A.17}$$

Expansion and rearranging of the terms enables the declaration of three separate integrals such that

$$\begin{aligned}\frac{1}{A}\int P(p)dp = &\int \frac{p^2}{(p^2+\alpha^2)^2}dp \\ &- \int \frac{2p^2}{(p^2+\alpha^2)(p^2+\beta^2)}dp \\ &+ \int \frac{p^2}{(p^2+\beta^2)^2}dp.\end{aligned} \tag{A.18}$$

The first and last integrals can be solved using the fact that

$$\int \frac{2x^2}{(x^2+a^2)^2}dx = \frac{1}{a}tan^{-1}\left(\frac{x}{a}\right) + \frac{x}{x^2+a^2}, \tag{A.19}$$

Therefore it can be seen that

$$\int \frac{p^2}{(p^2+\alpha^2)^2}dp = \frac{1}{2\alpha}tan^{-1}\left(\frac{p}{\alpha}\right) + \frac{p}{2(p^2+\alpha^2)}, \tag{A.20}$$

and

$$\int \frac{p^2}{(p^2+\beta^2)^2}dp = \frac{1}{2\beta}tan^{-1}\left(\frac{p}{\beta}\right) + \frac{p}{2(p^2+\beta^2)}. \tag{A.21}$$

The center integral of equation A.18 can be further separated such that

$$\int \frac{2p^2}{(p^2+\alpha^2)(p^2+\beta^2)}dp = \frac{1}{\beta^2-\alpha^2}\left(\int \frac{2p^2}{p^2+\alpha^2}dp - \int \frac{2p^2}{p^2+\beta^2}dp\right), \tag{A.22}$$

Which gives

$$\int \frac{2p^2}{(p^2+\alpha^2)(p^2+\beta^2)}dp = \frac{2}{\beta^2-\alpha^2}\left(\beta tan^{-1}\left(\frac{p}{\beta}\right) - \alpha tan^{-1}\left(\frac{p}{\alpha}\right)\right). \quad \text{(A.23)}$$

Bringing together equations A.20, A.21 and A.23, and substituting into equation A.18 gives

$$\begin{aligned}\frac{1}{A}\int P(p)dp =& \frac{1}{2\alpha}tan^{-1}\left(\frac{p}{\alpha}\right) + \frac{p}{2(p^2+\alpha^2)} \\ &- \frac{2}{\beta^2-\alpha^2}\left(\beta tan^{-1}\left(\frac{p}{\beta}\right) - \alpha tan^{-1}\left(\frac{p}{\alpha}\right)\right) \\ &+ \frac{1}{2\beta}tan^{-1}\left(\frac{p}{\beta}\right) + \frac{p}{2(p^2+\beta^2)}.\end{aligned} \quad \text{(A.24)}$$

Rearranging equation A.24 gives

$$\begin{aligned}\frac{1}{A}\int P(p)dp =& \left(\frac{1}{2\alpha} + \frac{2\alpha}{\beta^2-\alpha^2}\right)tan^{-1}\left(\frac{p}{\alpha}\right) \\ &+ \left(\frac{1}{2\beta} + \frac{2\beta}{\beta^2-\alpha^2}\right)tan^{-1}\left(\frac{p}{\beta}\right) \\ &+ \frac{1}{2}\left(\frac{p}{p^2+\alpha^2} + \frac{p}{p^2+\beta^2}\right).\end{aligned} \quad \text{(A.25)}$$

The final requirement is to identify the value of A, this is done by taking the limits of equation A.25 as $p \to \infty$, giving

$$\int P(p)dp \to 1 \tag{A.26}$$

$$tan^{-1}\left(\frac{p}{x}\right) \to \frac{\pi}{2} \tag{A.27}$$

$$\frac{p}{p^2+x} \to 0, \tag{A.28}$$

From this it can be seen that

$$\frac{1}{A} = \left(\frac{1}{2\alpha} + \frac{2\alpha}{\beta^2-\alpha^2}\right)\frac{\pi}{2} + \left(\frac{1}{2\beta} + \frac{2\beta}{\beta^2-\alpha^2}\right)\frac{\pi}{2}, \tag{A.29}$$

Which rearranges to

$$A = \frac{4\alpha\beta(\alpha+\beta)}{\pi(\alpha-\beta)^2}. \tag{A.30}$$

Substituting equations A.30 and A.25 into A.17 then recovers the full integration of the Hulthen Function as

$$\begin{aligned}\int P(p)dP =& \frac{4\alpha\beta(\alpha+\beta)}{\pi(\alpha-\beta)^2} \times \Bigg[\\ & \left(\frac{1}{2\alpha} + \frac{2\alpha}{\beta^2-\alpha^2}\right) tan^{-1}\frac{p}{\alpha} + \\ & \left(\frac{1}{2\beta} - \frac{2\beta}{\beta^2-\alpha^2}\right) tan^{-1}\frac{p}{\beta} - \\ & \frac{1}{2}\left(\frac{p^2}{p^2+\alpha^2} + \frac{p^2}{p^2+\beta^2}\right)\Bigg].\end{aligned} \tag{A.31}$$

A.2.2 Nucleon Momentum After Dissociation

The equation used is

$$|P| = \left[\left(\frac{E_T^2 - M_p^2 + M_n^2}{2E} \right)^2 - M_n^2 \right]^{\frac{1}{2}}, \tag{A.32}$$

where P is the momentum of the nucleons, E_T is the total energy of the system and M_p and M_n are the proton and neutron rest mass.

Before break up the total energy is given by

$$E_T = E_p + E_n + \nu, \tag{A.33}$$

where E_p is the proton energy, E_n is the neutron energy and ν is the deuteron potential energy, which is given by the energy of the negative mass quasi-particle. After dissociation the energy is then given by

$$E'_T = E'_p + E'_n, \tag{A.34}$$

however since energy must be conserved E_T and E'_T must be equivalent. Squaring equation A.34 gives

$$E_T^2 = E_p'^2 + 2E'_pE'_n + E_n'^2, \tag{A.35}$$

using the relativistic energy equation $E^2 = m^2 + p^2$ it can be seen that

$$E_T^2 = P'^2 + M_p^2 + 2E'_pE'_n + P'^2 + M_n^2, \tag{A.36}$$

where P' is the magnitude of the proton and neutron momenta, which are equal and opposite in direction in this frame of reference. Rearranging equation A.34 for E'_n, substituting into A.36 and rearranging terms gives

$$E_T^2 = 2P'^2 + M_p^2 + M_n^2 + 2E'_p(E_T - E'_p). \tag{A.37}$$

From equation A.37 we can then take the steps

$$E_T^2 = 2P^2 + M_p^2 + M_n^2 + 2E_pE_T - 2E_p^2 \tag{A.38}$$

$$= 2P^2 + M_p^2 + M_n^2 + 2E_pE_t - 2(P + M_p^2) \tag{A.39}$$

$$= M_n^2 - M_p^2 + 2E_pE_T \tag{A.40}$$

$$= M_n^2 - M_p^2 + 2(P^2 + M_p^2)^{\frac{1}{2}}E_T, \tag{A.41}$$

A.2.3 ΔP_{min} Minimum Change in Proton Momentum

Calculating the minimum change in the proton momentum that will enable the deuteron to be dissociated is required for calculation of scattering angle.

The dissociation is assumed to be due to a change in the energy of the proton after it receives a momentum kick in the Coulomb field of a nucleus. In order that the dissociation can happen the change in proton energy must satisfy the condition

$$\Delta E_p \geq |\varepsilon_B| \tag{A.42}$$

ΔE_p is the change in the proton energy and ε_B is the binding energy of the deuteron. ΔE_p can be calculated by

$$\Delta E_p = \left[\left(P_p + \Delta P_p^2\right) + M_p^2\right]^{\frac{1}{2}} - E_i \tag{A.43}$$

P_p is initial proton momentum, ΔP_p is the change in the proton momentum, M_p is the proton mass and E_i is the initial energy of the proton. Combining equations A.42 and A.43 gives

$$|\varepsilon_B| \leq \left[\left(P_p + \Delta P_p^2\right) + M_p^2\right]^{\frac{1}{2}} - E_i \tag{A.44}$$

and rearranging terms makes it possible to calculate ΔP_p as

$$\Delta P_p \geq \left[(|\varepsilon_B| + E_i)^2 - M_p^2\right]^{\frac{1}{2}} - P_p \tag{A.45}$$

From equation A.45 it is clear that the minimum value of ΔP_p (ΔP_{min}) that will allow dissociation is given by

$$\Delta P_{min} = \left[(|\varepsilon_B| + E_i)^2 - M_p^2\right]^{\frac{1}{2}} - P_p \tag{A.46}$$

A.2.4 θ_{min} Minimum Scattering Angle

The minimum scattering angle is used as a limit to determine the scattering angle, and therefore momentum kick, which will be received during deuteron dissociation.

After scattering from a target nucleus a deuteron will have momentum components relative to the initial momentum direction of

$$P_{\perp} = \mu u \times [cos(\theta) - 1] \tag{A.47}$$

$$P_{\parallel} = \mu u \times sin(\theta) \tag{A.48}$$

where $P_{\perp}$ is the perpendicular component, $P_{\parallel}$ is the parallel component, μ is the reduced mass of the deuteron/target system and θ is the scattering angle. To calculate the magnitude of the momentum kick (P_k) we use the components from equation A.47

$$P_k^2 = P_{\perp}^2 + P_{\parallel}^2 \tag{A.49}$$

$$= \mu^2 u^2 \times [cos(\theta) - 1]^2 + \mu^2 u^2 \times sin^2(\theta) \tag{A.50}$$

$$= \mu^2 u^2 \left[cos^2(\theta) - 2cos(\theta) + 1 + sin^2(\theta)\right] \tag{A.51}$$

$$= \mu^2 u^2 [2 - 2 \times cos(\theta)] \tag{A.52}$$

from which it can be seen that

$$P_k = \mu u [2 - 2 \times cos(\theta)]^{\frac{1}{2}} \tag{A.53}$$

To calculate the minimum scattering angle it is necessary to introduce the condition for minimum scattering angle, θ_{min}, such that minimum kick ΔP_{min} is achieved as

$$\Delta P_{min} = \mu u [2 - 2 \times cos(\theta_{min})]^{\frac{1}{2}} \tag{A.54}$$

and solving for θ_{min} gives

$$\mu u[2 - 2 \times cos(\theta_{min})]^{\frac{1}{2}} = \Delta P_{min} \tag{A.55}$$

$$2 - 2 \times cos(\theta_{min}) = \frac{\Delta P_{min}^2}{\mu^2 u^2} \tag{A.56}$$

$$cos(\theta_{min}) = 1 - \frac{\Delta P_{min}^2}{2\mu^2 u^2} \tag{A.57}$$

and so θ_{min} can be calculated as

$$\theta_{min} = Acos\left(1 - \frac{\Delta P_{min}^2}{2\mu^2 u^2}\right) \tag{A.58}$$

Appendix B

C++ Deuteron Dissociation Code

B.1 Inclusion of Model in Geant4

To include external code in a Geant4 simulation requires two additional methods to be included, which can then be called by the main program.

```
1
2  G4bool NumericalModel::IsApplicable (const G4HadProjectile &theTrack,
3      G4Nucleus &theTarget)
4          {
5          if (/*ConditionalTest*/) {return true;}
6          else {return false;}
7          }
8
9  G4HadFinalState* NumericalModel::ApplyYourself(const G4HadProjectile&
10     theTrack, G4Nucleus& theTarget)
11         {
12         /*Compute and add secondaries to theParticleChange*/
13
14       return &theParticleChange;
15       }
16
```

The IsApplicable method is a test, returning true or false, if the model is can be used by a given combination of projectile (G4HadProjectile), and target nucleus (G4Nucleus). The ApplyYourself method is used to access the various routines of a model and is used to pass primary particles in, and secondary particles back out.

B.2 Compound Nucleus Reaction

B.2.1 idealisedCNReaction.hh

B.2.2 idealisedCNReaction.cc

```
1
2  /*
3          Author: Simon Albright
4          Date: 21-May-2014
5          Affiliation: University Of Huddersfield,
6                         International Institute for Accelerator
7                             Applications
8          Version: 1
9
10         Highly idealised simulation of compound nucleus neutron
11            production.
12
13         Uses the kinetic energy of the target and projectile in the CoM
14         frame to calculate the energy available for the produced neutron
15         and decay nucleus.
16
17         Assumes 100% of energy goes to kinetic and only includes single
18         neutron final state.
19 */
20
21
22 #include "idealisedCNReaction.hh"
23
24 //************************************
25 //Default empty constructor
```

```
26 //*********************************
27 idealisedCNReaction::idealisedCNReaction() :
28     G4HadronicInteraction("idealisedCNReaction")
29         {
30         }
31
32 //*********************************
33 //Constructor used by IsApplicable
34 //method
35 //*********************************
36 idealisedCNReaction::idealisedCNReaction(const G4HadProjectile&
37     projectile, G4Nucleus& theTarget)
38    {
39    const G4ParticleDefinition *definitionP = theTrack.GetDefinition();
40    const G4String ParticleName = definitionP -> GetParticleName();
41
42    G4double projectileMass  = projectile.GetDefinition() ->
43        GetPDGMass();
44    G4double targetMass   = theTarget.AtomicMass(theTarget.GetA_asInt(),
45        theTarget.GetZ_asInt());
46
47    G4double decayMass;
48
49         if(ParticleName == "proton")  decayMass =
50             theTarget.AtomicMass(theTarget.GetA_asInt(),
51             theTarget.GetZ_asInt()+1);
52    else if(ParticleName == "deuteron")  decayMass =
53        theTarget.AtomicMass(theTarget.GetA_asInt()+1,
54        theTarget.GetZ_asInt()+1);
55    else if(ParticleName == "alpha")  decayMass =
56        theTarget.AtomicMass(theTarget.GetA_asInt()+3,
57        theTarget.GetZ_asInt()+2);
58
59    G4double projectileKineticEnergy = projectile.Get4Momentum().e() -
60        projectileMass;
61
62    G4LorentzVector* CoMMom  = new
63        G4LorentzVector(projectile.Get4Momentum() +
64        G4LorentzVector(G4ThreeVector(), targetMass));
65    G4ThreeVector* betaCoM   = new G4ThreeVector(calculateBeta(CoMMom));
66    G4ThreeVector* betaLab   = new G4ThreeVector(*betaCoM*-1);
67
```

```
68    G4LorentzVector* projLab = new
69        G4LorentzVector(projectile.Get4Momentum());
70    G4LorentzVector* targetLab = new G4LorentzVector(G4ThreeVector(),
71        targetMass);
72    G4LorentzVector* targetCoM = new
73        G4LorentzVector(lorentzBoost(targetLab, betaCoM));
74    G4LorentzVector* projCoM = new G4LorentzVector(lorentzBoost(projLab,
75        betaCoM));
76
77    G4double totalInitialEnergy = projCoM->e() + targetCoM->e();
78    G4double finalMassEnergy = NRestMass + decayMass;
79    G4double availableEnergy = totalInitialEnergy - finalMassEnergy;
80
81    G4double neutronEnergyCoM = availableEnergy/(1+NRestMass/decayMass);
82    neutronMomCoM      =
83        sqrt((neutronEnergyCoM+NRestMass)*(neutronEnergyCoM+NRestMass) -
84        NRestMass*NRestMass);
85
86    emissionAngleThetaPhi();
87    ParPerpComponentsCoM();
88    XYZComponentsCoM();
89
90    G4LorentzVector* neut4MomCoM = new
91        G4LorentzVector(*neutThreeMomentum, neutronEnergyCoM+NRestMass);
92
93    *neut4MomLab       = lorentzBoost(neut4MomCoM, betaLab);
94    }
95
96 //************************************
97 //Destructor
98 //************************************
99 idealisedCNReaction::~idealisedCNReaction()
100   {
101   }
102
103 //************************************
104 //IsApplicable method used to test
105 //if code is applicable to passed
106 //particle
107 //************************************
108 G4bool idealisedCNReaction::IsApplicable (const G4HadProjectile
109     &theTrack, G4Nucleus &theTarget)
110       {
```

```
111         const G4ParticleDefinition *definitionP =
112             theTrack.GetDefinition();
113         const G4String ParticleName = definitionP -> GetParticleName();
114
115    delete definitionP;
116
117         if (ParticleName == "deuteron" || particleName == "proton" ||
118             particleName == "alpha") {return true;}
119         else {return false;}
120         }
121
122 //**********************************
123 //ApplyYourself method used to interface
124 //with Geant4
125 //**********************************
126 G4HadFinalState* idealisedCNReaction::ApplyYourself(const
127     G4HadProjectile& theTrack, G4Nucleus& theTarget)
128         {
129         theParticleChange.Clear();
130         theParticleChange.SetStatusChange(stopAndKill);
131
132    G4LorentzVector* Deuteron_LF_4Momentum = new
133        G4LorentzVector(theTrack.Get4Momentum());
134
135    idealisedCNReaction* CNReaction = new idealisedCNReaction(theTrack,
136        theTarget);
137
138    G4DynamicParticle* recoilNucleus = new G4DynamicParticle();
139         recoilNucleus -> SetDefinition(G4GenericIon::Definition());
140         recoilNucleus ->
141             SetMass(theTarget.AtomicMass(theTarget.GetA_asInt(),
142             theTarget.GetZ_asInt()));
143    recoilNucleus -> SetMomentum(G4ThreeVector());
144
145    G4DynamicParticle* returnNucleus = new G4DynamicParticle();
146    returnNucleus -> SetDefinition(G4GenericIon::Definition());
147    returnNucleus ->
148        SetMass(theTarget.AtomicMass(theTarget.GetA_asInt()+1,
149        theTarget.GetZ_asInt()+1));
150    returnNucleus -> SetMomentum(-1*(CNReaction->neut4MomLab->vect()));
151
152
153    G4DynamicParticle* returnNeutron = new G4DynamicParticle();
```

```
154         returnNeutron -> SetDefinition(G4Neutron::Definition());
155    returnNeutron -> SetMomentum(CNReaction->neut4MomLab->vect());
156 
157    theParticleChange.AddSecondary(returnNeutron);
158    theParticleChange.AddSecondary(returnNucleus);
159 
160    return &theParticleChange;
161    }
162 
163 //**********************************
164 //Method to calculate neutron emission
165 //angle
166 //**********************************
167 void idealisedCNReaction::emissionAngleThetaPhi()
168        {
169        G4double v1=2*Randq()-1;
170        neutTheta = acos(v1);
171        neutPhi = Randq()*2*M_PI;
172        }
173 
174 //**********************************
175 //Method to calulate components of
176 //neutron momentum relative to source
177 //particle direction
178 //**********************************
179 void idealisedCNReaction::ParPerpComponentsCoM()
180        {
181        G4double totP = neutronMomCoM;
182        G4double angle = neutTheta;
183        neutParP=totP*cos(angle);
184        neutPerpP=abs(totP*sin(angle));
185        }
186 
187 //**********************************
188 //Method to caluclate and assign
189 //X, Y and Z components of momentum
190 //**********************************
191 void idealisedCNReaction::XYZComponentsCoM()
192        {
193        G4double PerpP = neutPerpP;
194        G4double angle = neutPhi;
195        G4double neutPY=PerpP*sin(angle);
196        G4double neutPX=PerpP*cos(angle);
```

```
197
198    neutThreeMomentum -> setZ(neutParP);
199         neutThreeMomentum -> setY(neutPY);
200         neutThreeMomentum -> setX(neutPX);
201         }
202
203
204
205 /*
206    "Utility" functions:
207    Lorentz boosts, minor calculations, etc
208 */
209
210 //***********************************
211 //Lorentz boost method
212 //***********************************
213 G4LorentzVector idealisedCNReaction::lorentzBoost(G4LorentzVector*
214     sourceParticle, G4ThreeVector* betaComponents)
215         {
216         G4double beta = betaComponents -> mag();
217    G4LorentzVector* BoostedVector = new G4LorentzVector();
218
219    if(beta>0)
220       {
221            G4double gamma = 1/sqrt(1-beta*beta);
222
223          G4double betaX = betaComponents -> getX();
224            G4double betaY = betaComponents -> getY();
225          G4double betaZ = betaComponents -> getZ();
226
227            G4double beta2 = beta*beta;
228
229          G4double gammaBetaX = gamma * betaX;
230            G4double gammaBetaY = gamma * betaY;
231          G4double gammaBetaZ = gamma * betaZ;
232
233            G4double gamma1 = gamma - 1;
234
235          G4double sourcePx = sourceParticle->px();
236            G4double sourcePy = sourceParticle->py();
237          G4double sourcePz = sourceParticle->pz();
238            G4double sourceP = sqrt(sourcePx*sourcePx +
239                sourcePy*sourcePy + sourcePz*sourcePz);
```

```
240         G4double sourceEnergy = sourceParticle->e();
241
242           G4double boostedEnergy = gamma*sourceEnergy -
243              gammaBetaX*sourcePx - gammaBetaY*sourcePy -
244              gammaBetaZ*sourcePz;
245
246         G4double boostedPx = -1*gammaBetaX*sourceEnergy +
247            (1+gamma1*(betaX*betaX/beta2))*sourcePx +
248            gamma1*betaX*betaY/beta2*sourcePy +
249            gamma1*betaX*betaZ/beta2*sourcePz;
250           G4double boostedPy = -1*gammaBetaY*sourceEnergy +
251              gamma1*betaY*betaX/beta2*sourcePx +
252              (1+gamma1*(betaY*betaY/beta2))*sourcePy +
253              gamma1*betaY*betaZ/beta2*sourcePz;
254         G4double boostedPz = -1*gammaBetaZ*sourceEnergy +
255            gamma1*betaZ*betaX/beta2*sourcePx +
256            gamma1*betaZ*betaY/beta2*sourcePy +
257            (1+gamma1*(betaZ*betaZ/beta2))*sourcePz;
258
259       *BoostedVector = G4LorentzVector(boostedPx, boostedPy, boostedPz,
260           boostedEnergy);
261       }
262     else
263       {
264       *BoostedVector = *sourceParticle;
265       }
266     G4LorentzVector returnVector;
267         returnVector.setPx(BoostedVector->px());
268         returnVector.setPy(BoostedVector->py());
269         returnVector.setPz(BoostedVector->pz());
270         returnVector.setE(BoostedVector->e());
271
272         delete BoostedVector;
273         return(returnVector);
274         }
275
276 //***********************************
277 //Method to caluclate beta from 4-momentum
278 //***********************************
279 G4ThreeVector idealisedCNReaction::calculateBeta(G4LorentzVector*
280     particle)
281     {
282     G4double betaX = particle->px()/particle->e();
```

```
283    G4double betaY = particle->py()/particle->e();
284    G4double betaZ = particle->pz()/particle->e();
285
286    return G4ThreeVector(betaX, betaY, betaZ);
287    }
288
289 //*********************************
290 //Method to select random number
291 //from uniform distribution [-1 1]
292 //*********************************
293 G4double idealisedCNReaction::Randq()
294        {
295        G4double v1 = (G4double)rand()/(G4double)RAND_MAX;
296        return v1;
297        }
298
```

B.3 Low Precision

B.3.1 brokenDeuteron.hh

```
1
2  /*
3          Author: Simon Albright
4          Date: 05-Sept-2013
5          Affiliation: University Of Huddersfield,
6                          International Institute for Accelerator
7                              Applications
8          Version: 2.2
9
10             Header file
11
12             Methods, constants and variables declared
13
14         Deuteron Breakup class to simulate the neutrons produced by
15             deuteron breakup.
16
17         The probability of a nucleon having given momentum (p) within a
18             deuteron is
19         given by the Hulthen function:
20
21                         p^2(1/(a^2+p^2)-1/(b^2+p^2))
22
```

```
23          The integral of the Hulthen function provides the momentum for
24              a random number
25          using Newton's method in the Deuteron Rest Frame. The random
26              number is provided
27          by a uniform distribution between 0 and 1.
28
29          The angle of emission (in the CMS) is random with the polar
30              co-ordinate uniformly
31          distributed by cos(theta) varying from -1 to 1. The azimuthal
32              direction is
33          uniform between 0 and 2*pi.
34
35          The neutron momentum in the CMS is lorentz boosted into the lab
36              frame based
37          on the momentum of the initial deuteron.
38  */
39
40
41
42
43  #ifndef BROKEN_DEUTERON_HH
44  #define BROKEN_DEUTERON_HH
45
46  #include "G4ThreeVector.hh"
47  #include "G4HadronicInteraction.hh"
48  #include "G4HadProjectile.hh"
49  #include "G4Deuteron.hh"
50  #include "G4Track.hh"
51  #include "G4Nucleus.hh"
52  #include "G4HadFinalState.hh"
53  #include "G4LorentzRotation.hh"
54  #include "G4LorentzVector.hh"
55  #include "G4Neutron.hh"
56  #include "G4ParticleTable.hh"
57  #include "G4IonTable.hh"
58
59  #include <iostream>
60  #include <cmath>
61  #include <cstdlib>
62  #include <stdio.h>
63  #include <stdlib.h>
64  #include <fstream>
65
```

```
66
67  class BrokenDeuteron : public G4HadronicInteraction
68      {
69
70  //*************************************************************
71  //Private member methods and variables required only within class
72  //*************************************************************
73      private:
74
75
76  //*************************************************************
77  //Constants used in Hulthen function and Newton's method
78  //value of "NEWT_SMALL" can be reduced to increase accuracy
79  //but computation time will greatly increase
80  //*************************************************************
81          const G4double HULT_ALPHA = 42.7;
82          const G4double HULT_BETA = 320;
83          const G4double HULT_CONS =
84              (4*HULT_ALPHA*HULT_BETA*(HULT_ALPHA+HULT_BETA))/(M_PI*pow((HULT_ALPHA-HULT_B
85          const G4double HULT_B2A2 =
86              HULT_BETA*HULT_BETA-HULT_ALPHA*HULT_ALPHA;
87          const G4double NEWT_SMALL = 0.00001;
88
89
90  //*************************************************************
91  //Rest mass of Deuteron and Neutron in MeV
92  //*************************************************************
93          const G4double DRestMass = G4Deuteron::Definition() ->
94              GetPDGMass();
95          const G4double NRestMass = G4Neutron::Definition() ->
96              GetPDGMass();
97          const G4double PRestMass = G4Proton::Definition() ->
98              GetPDGMass();
99
100
101 //*************************************************************
102 //Variables used in calculations and assigned to produced particles
103 //Shorthand used to relate variables to particles and
104 //frames of reference:
105 //DRF: Deuteron Rest Frame
106 //LF:  Lab Frame
107 //N:   Neutron
108 //D:   Deuteron
```

```
109 //P:   Proton
110 //e.g. DRF_N_Momentum = Neutron momentum in deuteron rest frame
111 //*********************************************************
112         G4double DRF_N_Theta;
113         G4double DRF_N_Momentum;
114         G4double DRF_N_ParP;
115         G4double DRF_N_PerpP;
116         G4double DRF_N_Phi;
117         G4double DRF_N_PY;
118         G4double DRF_N_PZ;
119 
120         G4double DRF_P_Momentum;
121 
122         G4ThreeVector* DRF_N_ThreeMomentum = new G4ThreeVector();
123         G4ThreeVector* LF_D_ThreeMomentum = new G4ThreeVector();
124         G4ThreeVector* LF_N_ThreeMomentum = new G4ThreeVector();
125         G4ThreeVector* LF_P_ThreeMomentum = new G4ThreeVector();
126         G4ThreeVector* DRF_P_ThreeMomentum = new G4ThreeVector();
127 
128 //*********************************************************
129 //Methods used to calculate the momentum of the neutron and proton in
130    the DRF
131 //*********************************************************
132         void MomentumDRF();
133         G4double integratedHulthen(G4double mom);
134         G4double Hulthen(G4double mom);
135         G4double Randq();
136 
137         void CalculateProtonMomentumDRF();
138 
139 //*********************************************************
140 //Method to calculate emission angle in the DRF
141 //*********************************************************
142         void emissionAngleThetaPhi();
143 
144 
145 //*********************************************************
146 //Methods to transform neutron momentum from DRF to LF
147 //*********************************************************
148         void ParPerpComponentsDRF();
149         void YZComponentsDRF();
150         void setMomenta();
151 
```

```
152         G4LorentzVector* projectile4Vector = new G4LorentzVector();
153         void BoostNToLF(G4LorentzVector* projectile4Vector);
154 
155         G4LorentzVector lorentzBoost(G4LorentzVector* sourceParticle,
156             G4ThreeVector* betaComponents);
157 
158 
159 //*************************************************************
160 //*************************************************************
161 //Public member variables and functions required to operate
162 //class externally
163 //*************************************************************
164 //*************************************************************
165     public:
166 
167 
168         void SetNucleonMomenta(G4ThreeVector* neutron, G4ThreeVector*
169             proton);
170 
171 //*************************************************************
172 //Overwrite virtual IsApplicable and ApplyYourself functions
173 //from G4HadronicInteraction
174 //*************************************************************
175         G4HadFinalState* ApplyYourself(const G4HadProjectile &theTrack,
176             G4Nucleus &theTarget);
177         G4bool IsApplicable(const G4HadProjectile &theTrack, G4Nucleus
178             &theTarget);
179 
180 //*************************************************************
181 //Methods to access component momenta in LF and DR
182 //*************************************************************
183         G4ThreeVector* GetNMomentumThreeVectorDRF();
184         G4ThreeVector* GetNMomentumThreeVectorLF();
185         G4ThreeVector* GetPMomentumThreeVectorLF();
186         G4ThreeVector* GetPMomentumThreeVectorDRF();
187 
188         G4double GetNMomentumDRF();
189         G4double GetNMomentumLF();
190         G4double GetPMomentumDRF();
191 
192 //*************************************************************
193 //Constructor to produce a neutron from a given deuteron
194 //Constructor to initialise but not produce a neutron
```

```
195 //Destructor
196 //*************************************************************
197         G4ThreeVector* Deuteron3Momentum = new G4ThreeVector();
198         BrokenDeuteron(G4ThreeVector* Deuteron3Momentum);
199         BrokenDeuteron();
200         ~BrokenDeuteron();
201
202
203 //*************************************************************
204 //Constructor to produce a neutron from a given deuteron
205 //Constructor to initialise but not produce a neutron
206 //*************************************************************
207         void RandomNeutron(G4ThreeVector* Deuteron3Momentum);
208         void RandomNeutron();
209     };
210 #endif
211
```

B.3.2 brokenDeuteron.cc

```
1
2  /*
3     Author: Simon Albright
4     Date: 05-Sept-2013
5     Affiliation: University Of Huddersfield,
6          International Institute for Accelerator Applications
7     Version: 2.2
8
9     Deuteron Breakup class to simulate the neutrons produced by deuteron
10       breakup.
11
12    The probability of a nucleon having given momentum (p) within a
13       deuteron is given by the Hulthen function:
14
15         p^2(1/(a^2+p^2)-1/(b^2+p^2))
16
17    The integral of the Hulthen function provides the momentum for a
18       random number using Newton's method in the Deuteron Rest Frame.
19       The random number is provided  by a uniform distribution between
20       0 and 1.
21
22    The angle of emission (in the CMS) is random with the polar
23       co-ordinate uniformly distributed by cos(theta) varying from -1
24       to 1. The azimuthal direction is uniform between 0 and 2*pi.
```

```
25
26    The neutron momentum in the CMS is lorentz boosted into the lab
27        frame based on the momentum of the initial deuteron.
28 */
29
30 #include "brokenDeuteron.hh"
31
32 BrokenDeuteron::BrokenDeuteron(G4ThreeVector* Deuteron3Momentum) :
33     G4HadronicInteraction("brokenDeuteron")
34     {
35     LF_D_ThreeMomentum -> set(Deuteron3Momentum -> getX(),
36         Deuteron3Momentum -> getY(), Deuteron3Momentum -> getZ());
37     RandomNeutron();
38     }
39
40 BrokenDeuteron::BrokenDeuteron()
41     {
42     }
43
44 BrokenDeuteron::~BrokenDeuteron()
45     {
46     delete DRF_N_ThreeMomentum;
47     delete LF_D_ThreeMomentum;
48     delete LF_N_ThreeMomentum;
49     delete DRF_P_ThreeMomentum;
50     delete projectile4Vector;
51     delete Deuteron3Momentum;
52     }
53
54 //************************************************************
55 //Produce a new neutron using current deuteron momentum
56 //************************************************************
57 void BrokenDeuteron::RandomNeutron()
58     {
59     MomentumDRF();
60     emissionAngleThetaPhi();
61     ParPerpComponentsDRF();
62     YZComponentsDRF();
63     CalculateProtonMomentumDRF();
64     }
65
66 //************************************************************
67 //Produce a new neutron using current given deuteron momentum
```

```
68 //*************************************************************
69 void BrokenDeuteron::RandomNeutron(G4ThreeVector* Deuteron3Momentum)
70     {
71     LF_D_ThreeMomentum -> set(Deuteron3Momentum -> getX(),
72         Deuteron3Momentum -> getY(), Deuteron3Momentum -> getZ());
73     MomentumDRF();
74     emissionAngleThetaPhi();
75     ParPerpComponentsDRF();
76     YZComponentsDRF();
77     CalculateProtonMomentumDRF();
78     }
79
80 //*************************************************************
81 //Produce a neutron in the Deuteron Rest Frame
82 //A randon number is produced
83 //Newton's method is used to converge on the assosciated momentum
84 //A loop counter is used to prevent an infinite loop
85 //An error is printed and the trial restarts with a new random number
86 //A check is used to prevent divergence and +/-inf results
87 //*************************************************************
88 void BrokenDeuteron::MomentumDRF()
89     {
90     G4bool ViableAnswer = false;
91     while(ViableAnswer==false)
92         {
93         G4double q = Randq();
94         G4double trialMomentum;
95         if(q<0.9)
96             {
97             trialMomentum = 70;
98             }
99         else
100            {
101            trialMomentum = 150;
102            }
103        G4double newFp = integratedHulthen(trialMomentum);
104        G4int i = 0;
105        G4int tmp = 50;
106        while ((q-newFp>NEWT_SMALL || q-newFp<-1*NEWT_SMALL) && i<50)
107            {
108            if (abs(newFp) == 1)
109                {
```

```
110                 G4cout << "\n\n***DIVERGENCE DETECTED, FORCING REDUCTION
111                     OF VARIABLE***\n\n";
112                 trialMomentum = sqrt(abs(trialMomentum));
113                 newFp = newFp/2;
114                 }
115             G4double hulthenResult = Hulthen(trialMomentum);
116             trialMomentum = trialMomentum + (q-newFp)/hulthenResult;
117             newFp=integratedHulthen(trialMomentum);
118             i++;
119             }
120         if(i<50&&trialMomentum>0)
121             {
122             ViableAnswer=true;
123             DRF_N_Momentum = trialMomentum;
124             DRF_P_Momentum = trialMomentum;
125             }
126         else
127             {
128             G4cout <<"\n\n";
129             G4cout <<"****************************************\n";
130             G4cout <<"ERROR: POTENTIAL INFINITE LOOP DETECTED\n";
131             G4cout <<"  RESTARTING WITH NEW RANDOM NUMBER \n";
132             G4cout <<"****************************************\n";
133             G4cout << "i = " << i << ", trialMom= " << trialMomentum <<
134                 ", q = " << q << G4endl;
135             }
136         }
137     }
138
139 //************************************************************
140 //The integrated form of the Hulthen function
141 //Returns a number between 0 and 1 for a given momentum
142 //Used by Newton's method to identify "correct" momentum for a given
143     random number
144 //************************************************************
145 G4double BrokenDeuteron::integratedHulthen(G4double mom)
146     {
147     G4double intHul = HULT_CONS*((1/(2*HULT_ALPHA) +
148         (2*HULT_ALPHA)/HULT_B2A2)*atan(mom/HULT_ALPHA) +
149         (1/(2*HULT_BETA)-(2*HULT_BETA)/HULT_B2A2)*atan(mom/HULT_BETA) -
150         0.5*(mom/(mom*mom + HULT_ALPHA*HULT_ALPHA) + mom/(mom*mom +
151         HULT_BETA*HULT_BETA)));
152     return intHul;
```

```
153     }
154
155 //************************************************************
156 //The Hulthen function
157 //Returns the probability of a given momentum
158 //Used by Newton's method to identify "correct" momentum for a given
159     random number
160 //************************************************************
161 G4double BrokenDeuteron::Hulthen(G4double mom)
162     {
163     G4double Hul = HULT_CONS*pow(mom, 2)*pow( (1/(mom*mom +
164         HULT_ALPHA*HULT_ALPHA)-1/(mom*mom + HULT_BETA*HULT_BETA)) , 2);
165     return Hul;
166     }
167
168 //************************************************************
169 //Produce a random number between 0 and 1
170 //************************************************************
171 G4double BrokenDeuteron::Randq()
172     {
173     G4double v1 = (G4double)rand()/(G4double)RAND_MAX;
174     return v1;
175     }
176
177 //************************************************************
178 //Provide the polar and azimuthal emission angle of the neutron
179 //DRF_N_Theta varies with cos(theta) between -1 and 1
180 //Polar angle is NOT frame independent
181 //LF_DRF_Phi varies between 0 and 2*pi
182 //Azimuthal angle is frame independent
183 //************************************************************
184 void BrokenDeuteron::emissionAngleThetaPhi()
185     {
186     G4double v1=2*Randq()-1;
187     DRF_N_Theta = acos(v1);
188     DRF_N_Phi = Randq()*2*M_PI;
189     }
190
191 //************************************************************
192 //Polar emission angle (DRF_N_Theta) used to convert total momentum
193 //into parrallel and perpendicular components in deuteron
194 //propagation direction
195 //************************************************************
```

```
196 void BrokenDeuteron::ParPerpComponentsDRF()
197     {
198     G4double totP = DRF_N_Momentum;
199     G4double angle = DRF_N_Theta;
200     DRF_N_ParP=totP*cos(angle);
201     DRF_N_PerpP=abs(totP*sin(angle));
202     DRF_N_ThreeMomentum -> setX(DRF_N_ParP);
203     }
204 
205 //*************************************************************
206 //Azimuthal emission angle used to split perpendicular emission
207 //angle into Y and Z components
208 //*************************************************************
209 void BrokenDeuteron::YZComponentsDRF()
210     {
211     G4double PerpP = DRF_N_PerpP;
212     G4double angle = DRF_N_Phi;
213     DRF_N_PY=PerpP*sin(angle);
214     DRF_N_PZ=PerpP*cos(angle);
215     DRF_N_ThreeMomentum -> setY(DRF_N_PY);
216     DRF_N_ThreeMomentum -> setZ(DRF_N_PZ);
217     }
218 
219 //*************************************************************
220 //Neutron momentum boosted from Deuteron Rest Frame into CoM
221 //frame of deuteron and nucleus passed into ApplyYourself()
222 //*************************************************************
223 void BrokenDeuteron::BoostNToLF(G4LorentzVector* projectile4Vector)
224     {
225     G4ThreeVector* projectileMomentum = new
226         G4ThreeVector(projectile4Vector -> px(), projectile4Vector ->
227         py(), projectile4Vector -> pz());
228     G4ThreeVector* projectileDirection = new
229         G4ThreeVector(projectile4Vector -> vect()/sqrt(
230                 (projectile4Vector -> px())*(projectile4Vector -> px()) +
231                 (projectile4Vector -> py())*(projectile4Vector -> py()) +
232                 (projectile4Vector -> pz())*(projectile4Vector -> pz())
233             ));
234 
235     G4double betaX = projectileDirection -> getX()*(projectileMomentum
236         -> mag()/(projectile4Vector -> e()));
237     G4double betaY = projectileDirection -> getY()*(projectileMomentum
238         -> mag()/(projectile4Vector -> e()));
```

```
239     G4double betaZ = projectileDirection -> getZ()*(projectileMomentum
240         -> mag()/(projectile4Vector -> e()));
241 
242     G4ThreeVector* betaComponents = new G4ThreeVector(betaX, betaY,
243         betaZ);
244 
245     G4LorentzVector* DRF_N_4Momentum = new
246         G4LorentzVector(*DRF_N_ThreeMomentum, sqrt(pow(DRF_N_Momentum,
247         2) + pow(NRestMass, 2)));
248     G4LorentzVector * LF_N_4Momentum = new
249         G4LorentzVector(DRF_N_4Momentum -> boost(*betaComponents));
250 
251     G4LorentzVector* DRF_P_4Momentum = new
252         G4LorentzVector(*DRF_P_ThreeMomentum, sqrt(pow(DRF_P_Momentum,
253         2) + pow(PRestMass, 2)));
254     G4LorentzVector * LF_P_4Momentum = new
255         G4LorentzVector(DRF_P_4Momentum -> boost(*betaComponents));
256 
257     LF_N_ThreeMomentum -> set(LF_N_4Momentum->px(),
258         LF_N_4Momentum->py(), LF_N_4Momentum->pz());
259     LF_P_ThreeMomentum -> set(LF_P_4Momentum->px(),
260         LF_P_4Momentum->py(), LF_P_4Momentum->pz());
261 
262     delete projectileMomentum, delete projectileDirection, delete
263         betaComponents, delete DRF_N_4Momentum, delete LF_N_4Momentum;
264     }
265 
266 //************************************************************
267 //Method to calculate proton momentum in DRF using momentum
268 //conservation as opposite of neutron momentum
269 //************************************************************
270 void BrokenDeuteron::CalculateProtonMomentumDRF()
271     {
272     DRF_P_ThreeMomentum -> set(-1*DRF_N_ThreeMomentum->getX(),
273         -1*DRF_N_ThreeMomentum->getY(), -1*DRF_N_ThreeMomentum->getZ());
274     }
275 
276 //************************************************************
277 //Called to check model is available for target projectile
278 //Returns TRUE if projectile is a deuteron
279 //************************************************************
280 G4bool BrokenDeuteron::IsApplicable (const G4HadProjectile &theTrack,
281     G4Nucleus &theTarget)
```

```
282     {
283     const G4ParticleDefinition *definitionP = theTrack.GetDefinition();
284     const G4String ParticleName = definitionP -> GetParticleName();
285     if (ParticleName == "deuteron")
286         {
287         return true;
288         }
289     else
290         {
291         return false;
292         }
293     }
294
295 //************************************************************
296 //Calculate result of deuteron break up and return:
297 //Neutron with appropriate momentum
298 //Proton with opposite momentum boosted into CoM
299 //G4GenericIon with A, Z and P equal to nucleus passed into code
300 //
301 //Proton momentum calculated based on momentum conservation
302 //as projectile deuteron momentum - produced neutron momentum
303 //************************************************************
304 G4HadFinalState *BrokenDeuteron::ApplyYourself(const G4HadProjectile
305     &theTrack, G4Nucleus &theTarget)
306     {
307     theParticleChange.Clear();
308     theParticleChange.SetStatusChange(stopAndKill);
309
310     G4ThreeVector* DeuteronMomentum = new G4ThreeVector();
311     DeuteronMomentum -> set(theTrack.Get4Momentum().getX(),
312         theTrack.Get4Momentum().getY(), theTrack.Get4Momentum().getZ());
313
314     BrokenDeuteron* newNeut = new BrokenDeuteron(DeuteronMomentum);
315
316     G4LorentzVector* track4Mom = new
317         G4LorentzVector(theTrack.Get4Momentum());
318
319     newNeut -> BoostNToLF(track4Mom);
320
321     G4ThreeVector* neutronMomentum = new G4ThreeVector(*newNeut ->
322         LF_N_ThreeMomentum);
323
```

```
324     G4DynamicParticle* DynParticle = new
325         G4DynamicParticle(G4Neutron::Definition(), *newNeut ->
326         LF_N_ThreeMomentum);
327     G4DynamicParticle* RecoilProton = new
328         G4DynamicParticle(G4Proton::Definition(), *(new
329         G4ThreeVector(theTrack.Get4Momentum().getX()-neutronMomentum ->
330         getX(),
331             theTrack.Get4Momentum().getY()-neutronMomentum -> getY(),
332             theTrack.Get4Momentum().getZ()-neutronMomentum -> getZ())));
333
334     G4DynamicParticle* RecoilNucleus = new G4DynamicParticle();
335     RecoilNucleus -> SetDefinition(G4GenericIon::Definition());
336     RecoilNucleus ->
337         SetMass(theTarget.AtomicMass(theTarget.GetA_asInt(),
338         theTarget.GetZ_asInt()));
339     RecoilNucleus -> SetMomentum(theTarget.GetFermiMomentum());
340
341     theParticleChange.AddSecondary(DynParticle);
342     theParticleChange.AddSecondary(RecoilProton);
343     theParticleChange.AddSecondary(RecoilNucleus);
344
345     delete newNeut, delete track4Mom, delete neutronMomentum, delete
346         DynParticle, delete RecoilProton, delete RecoilNucleus;
347
348     return &theParticleChange;
349     }
350
351 //*************************************************************
352 //Assigns data members based on passed values
353 //*************************************************************
354 void BrokenDeuteron::SetNucleonMomenta(G4ThreeVector* neutron,
355     G4ThreeVector* proton)
356     {
357     *DRF_N_ThreeMomentum = *neutron;
358     *DRF_P_ThreeMomentum = *proton;
359
360     DRF_P_Momentum = proton->mag();
361     }
362
363 //*************************************************************
364 //Lorentz boost function to calculate boost 4-vectors between
365 //arbitrary frames of reference by cross produce V' = A X V
366 //*************************************************************
```

```
367 G4LorentzVector brokenDeuteron::lorentzBoost(G4LorentzVector*
368     sourceParticle, G4ThreeVector* betaComponents)
369     {
370     G4double beta = betaComponents -> mag();
371 
372     G4LorentzVector* BoostedVector = new G4LorentzVector();
373 
374     if(beta>0)
375         {
376         G4double gamma = 1/sqrt(1-beta*beta);
377 
378         G4double betaX = betaComponents -> getX();
379         G4double betaY = betaComponents -> getY();
380         G4double betaZ = betaComponents -> getZ();
381 
382         G4double beta2 = beta*beta;
383 
384         G4double gammaBetaX = gamma * betaX;
385         G4double gammaBetaY = gamma * betaY;
386         G4double gammaBetaZ = gamma * betaZ;
387 
388         G4double gamma1 = gamma - 1;
389 
390         G4double sourcePx = sourceParticle->px();
391         G4double sourcePy = sourceParticle->py();
392         G4double sourcePz = sourceParticle->pz();
393         G4double sourceP = sqrt(sourcePx*sourcePx + sourcePy*sourcePy +
394             sourcePz*sourcePz);
395         G4double sourceEnergy = sourceParticle->e();
396 
397         G4double boostedEnergy = gamma*sourceEnergy -
398             gammaBetaX*sourcePx - gammaBetaY*sourcePy -
399             gammaBetaZ*sourcePz;
400 
401         G4double boostedPx = -1*gammaBetaX*sourceEnergy +
402             (1+gamma1*(betaX*betaX/beta2))*sourcePx +
403             gamma1*betaX*betaY/beta2*sourcePy +
404             gamma1*betaX*betaZ/beta2*sourcePz;
405         G4double boostedPy = -1*gammaBetaY*sourceEnergy +
406             gamma1*betaY*betaX/beta2*sourcePx +
407             (1+gamma1*(betaY*betaY/beta2))*sourcePy +
408             gamma1*betaY*betaZ/beta2*sourcePz;
```

```
409         G4double boostedPz = -1*gammaBetaZ*sourceEnergy +
410             gamma1*betaZ*betaX/beta2*sourcePx +
411             gamma1*betaZ*betaY/beta2*sourcePy +
412             (1+gamma1*(betaZ*betaZ/beta2))*sourcePz;
413
414
415         *BoostedVector = G4LorentzVector(boostedPx, boostedPy,
416             boostedPz, boostedEnergy);
417         }
418     else
419         {
420         *BoostedVector = *sourceParticle;
421         }
422     return *BoostedVector;
423     }
424
425
426 //************************************************************
427 //************************************************************
428 //Methods for accessing data follow, no calculation is performed
429 //after this section
430 //************************************************************
431 //************************************************************
432
433 G4ThreeVector* BrokenDeuteron::GetNMomentumThreeVectorDRF()
434     {
435     return DRF_N_ThreeMomentum;
436     }
437
438 G4ThreeVector* BrokenDeuteron::GetNMomentumThreeVectorLF()
439     {
440     return LF_N_ThreeMomentum;
441     }
442
443 G4ThreeVector* BrokenDeuteron::GetPMomentumThreeVectorDRF()
444     {
445     return DRF_P_ThreeMomentum;
446     }
447
448 G4ThreeVector* BrokenDeuteron::GetPMomentumThreeVectorLF()
449     {
450     return LF_P_ThreeMomentum;
451     }
```

```
452
453 G4double BrokenDeuteron::GetPMomentumDRF()
454     {
455     return DRF_P_Momentum;
456     }
457
458 G4double BrokenDeuteron::GetNMomentumDRF()
459     {
460     return DRF_N_Momentum;
461     }
462
463 G4double BrokenDeuteron::GetNMomentumLF()
464     {
465     return LF_N_ThreeMomentum -> mag();
466     }
467
```

B.4 High Precision

B.4.1 brokenDeuteronKick.hh

```
1
2 /*
3         Author: Simon Albright
4         Date: 18-Feb-2015
5         Affiliation: University Of Huddersfield,
6                         International Institute for Accelerator
7                             Applications
8         Version: 1
9
10    Higher accuracy Deuteron Breakup class to incorporate the effects of
11    the deuteron potential energy and coulomb scattering from the target
12    nucleus.
13
14    The potential energy is incorporated as the 4-vector of a -ve mass
15    quasiparticle, which is transferred between frames of reference
16    along with the proton and neutron, preserving energy and momentum.
17
18    The Coulomb scattering is calculated using the standard Rutherford
19    scattering formula with a maximal value of impact parameter used
20    to select from a random number distribution what value the impact
21    parameter will take.
22
23    The Coulomb scatter is applied to the proton increasing its energy
```

```
24    in the CoM frame and overcoming the potential energy. The energy
25    and momentum of the quasi particle are removed from the system and
26    the proton and neutron are returned to the lab frame before being
27    returned to Geant4 for further tracking.
28 */
29
30 #ifndef BROKEN_DEUTERON_KICK_HH
31 #define BROKEN_DEUTERON_KICK_HH
32
33
34 #include "brokenDeuteron.hh"
35 #include "G4HadronicInteraction.hh"
36 #include "globals.hh"
37 #include "G4Neutron.hh"
38 #include "G4Deuteron.hh"
39 #include "G4Proton.hh"
40 #include <iostream>
41
42 class brokenDeuteronKick : public G4HadronicInteraction
43 {
44
45 private:
46
47
48    //*************************************************************
49    //Constants used in code, masses, natural constants etc
50    //*************************************************************
51    const G4double fractionalErrorAllowed = 0.025;
52    const G4double DRestMass   = G4Deuteron::Definition() ->
53        GetPDGMass();
54    const G4double NRestMass   = G4Neutron::Definition() -> GetPDGMass();
55    const G4double PRestMass   = G4Proton::Definition() -> GetPDGMass();
56    const G4double DBindingEnergy  = DRestMass-NRestMass-PRestMass;
57    const G4double epsilon_0   = 8.854187817*pow(10, -12);
58    const G4double absElectronCharge = 1.60217657*pow(10,-19);
59    const G4double COULOMB_CONSTANT  =
60        (4*M_PI*epsilon_0)/(pow(absElectronCharge,2));
61
62
63    //*************************************************************
64    //Variables and methods related to potential energies and
65    //non-constant masses
66    //*************************************************************
```

```
67    G4double reducedMass;
68    G4Nucleus* TargetNucleus = new G4Nucleus();
69    G4double potentialEnergyDRF;
70    G4double potentialEnergyNKP;
71    G4double NTotEnergyDRF;
72    G4double PTotEnergyDRFPostKick;
73    void CalcInterNucleonPotentialEnergyDRF();
74    void CalcInterNucleonPotentialEnergyNKP();
75    void calcReducedMass();
76
77
78    //*********************************************************
79    //Variables and methods for scattering angle and momentum kick
80    //*********************************************************
81    G4double deltaPpMin;
82    inline G4bool isPpEnough();
83    void calcDeltaPpMin();
84    void calcThetaMin();
85    G4double thetaMin;
86
87
88    //*********************************************************
89    //4-vectors of proton (P), neutron (N) and quasi-particle (Q) in
90        various frames
91    //Frames of reference are:
92    //DRF: Deuteron Rest Frame
93    //LF: Lab Frame
94    //NKP: Neutron and Kicked Proton center of momentum frame
95    //
96    //Suffix PK refers to Post Kick and UB refers to Un Bound
97    //*********************************************************
98    G4LorentzVector* DRF_P_4Momentum = new G4LorentzVector();
99    G4LorentzVector* DRF_N_4Momentum = new G4LorentzVector();
100   G4LorentzVector* DRF_Q_4Momentum = new G4LorentzVector();
101
102   G4LorentzVector* LF_Target_4Momentum = new G4LorentzVector();
103
104    G4LorentzVector* NKP_N_4Momentum = new G4LorentzVector();
105    G4LorentzVector* NKP_P_4Momentum = new G4LorentzVector();
106   G4LorentzVector* NKP_P_4Momentum_PK = new G4LorentzVector();
107   G4LorentzVector* NKP_Q_4Momentum = new G4LorentzVector();
108
109   G4LorentzVector* LF_P_4Momentum  = new G4LorentzVector();
```

```
110    G4LorentzVector* LF_N_4Momentum  = new G4LorentzVector();
111    G4LorentzVector* LF_P_4Momentum_PK = new G4LorentzVector();
112    G4LorentzVector* LF_Q_4Momentum  = new G4LorentzVector();
113
114    G4LorentzVector* NKP_P_4Momentum_UB = new G4LorentzVector();
115    G4LorentzVector* NKP_N_4Momentum_UB = new G4LorentzVector();
116
117     G4LorentzVector* LF_P_4Momentum_UB = new G4LorentzVector();
118     G4LorentzVector* LF_N_4Momentum_UB = new G4LorentzVector();
119
120
121
122    //*********************************************************
123    //Variables and methods for transferring between frames of reference
124    //*********************************************************
125    void NucleonsToLF(BrokenDeuteron* deuteron);
126    void boostToNKPFrame();
127    void ReturnToDRF();
128    void BoostToLF(G4LorentzVector* track4Mom);
129    void FragmentsToLF();
130    void CalculateNKP4Momenta();
131    void CalculateBetaNKP();
132    G4LorentzVector lorentzBoost(G4LorentzVector* sourceParticle,
133       G4ThreeVector* betaComponents);
134    G4ThreeVector calculateBeta(G4LorentzVector* particle);
135
136    G4ThreeVector* PBetaComponents     = new G4ThreeVector();
137    G4ThreeVector* DRFBetaComponents   = new G4ThreeVector();
138    G4ThreeVector* NKP_Beta          = new G4ThreeVector();
139
140
141
142    //*********************************************************
143    //Constants, variables and methods for calculating angles
144    //and values of momentum kick
145    //*********************************************************
146    const G4double THETA_MIN = M_PI/480;
147    G4double ThetaKick;
148    G4double PhiKick;
149    G4double ScatterMin;
150    G4double KickPerp;
151    G4double KickParr;
152    G4ThreeVector* kick = new G4ThreeVector();
```

```
153     void CalculateKickAngle();
154     void CalculateKickComponents();
155     void KickProton();
156     G4ThreeVector Rotate(G4ThreeVector* inputVector);
157 
158 
159     //*************************************************************
160     //Variables and methods for breaking deuteron
161     //*************************************************************
162     G4bool IsBreakable();
163     G4bool IsBroken;
164     inline G4bool KickedEnough();
165     void BreakDeuteron();
166     G4double CalculateNKPMomentumMag();
167 
168 
169     //*************************************************************
170     //Declaration of initial deuteron to be broken
171     //*************************************************************
172     BrokenDeuteron* deuteron;
173 
174 
175 public:
176 
177 
178     //*************************************************************
179     //IsApplicable and ApplyYourself methods used by Geant4
180     //*************************************************************
181     G4bool IsApplicable(const G4HadProjectile &theTrack, G4Nucleus
182         &theTarget);
183     G4HadFinalState* ApplyYourself(const G4HadProjectile& theTrack,
184         G4Nucleus& theTarget);
185 
186 
187     //*************************************************************
188     //Constructors and destructor
189     //*************************************************************
190     brokenDeuteronKick();
191     brokenDeuteronKick(G4LorentzVector LF_D_4Momentum, G4Nucleus&
192         theTarget);
193     ~brokenDeuteronKick();
194 };
195 
```

```
196 #endif
197
```

B.4.2 brokenDeuteronKick.cc

```
1
2  /*
3          Author: Simon Albright
4          Date: 18-Feb-2015
5          Affiliation: University Of Huddersfield,
6                       International Institute for Accelerator
7                           Applications
8          Version: 1
9
10   Higher accuracy Deuteron Breakup class to incorporate the effects of
11   the deuteron potential energy and coulomb scattering from the target
12   nucleus.
13
14   The potential energy is incorporated as the 4-vector of a -ve mass
15   quasiparticle, which is transferred between frames of reference
16   along with the proton and neutron, preserving energy and momentum.
17
18   The Coulomb scattering is calculated using the standard Rutherford
19   scattering formula with a maximal value of impact parameter used
20   to select from a random number distribution what value the impact
21   parameter will take.
22
23   The Coulomb scatter is applied to the proton increasing its energy
24   in the CoM frame and overcoming the potential energy. The energy
25   and momentum of the quasi particle are removed from the system and
26   the proton and neutron are returned to the lab frame before being
27   returned to Geant4 for further tracking.
28 */
29
30
31 #include "brokenDeuteronKick.hh"
32
33 //************************************
34 //Default empty constructor
35 //************************************
36 brokenDeuteronKick::brokenDeuteronKick() :
37     G4HadronicInteraction("brokenDeuteronKick")
38         {
39         }
```

```
40
41
42 //***********************************
43 //Constructor used by ApplyYourself method
44 //***********************************
45 brokenDeuteronKick::brokenDeuteronKick(G4LorentzVector LF_D_4Momentum,
46     G4Nucleus& theTarget)
47    {
48    *TargetNucleus = theTarget;
49
50    *DRFBetaComponents = calculateBeta(&LF_D_4Momentum);
51
52    deuteron = new BrokenDeuteron();
53    deuteron -> RandomNeutron();
54
55    NucleonsToLF(deuteron);
56
57    G4int loopCounter = 1;
58    G4int bigCounter = 0;
59    G4int nOfReSamples = 0;
60    G4int thetaJumps = 0;
61    G4bool breakSuccesful = false;
62    G4bool reSample = false;
63    G4bool firstItt = true;
64
65    G4double accuracyLimit =
66        fractionalErrorAllowed*(LF_D_4Momentum.e()-DRestMass);
67
68    //***********************************
69    //If the proton is travelling away from the target
70    //at point of interaction the deuteron will not
71    //break. This is a crude way of implimenting
72    //the flux factor.
73    //***********************************
74    while(LF_P_4Momentum->vect().getZ()<0)
75       {
76       deuteron -> RandomNeutron();
77       NucleonsToLF(deuteron);
78       }
79
80    calcDeltaPpMin();
81    calcThetaMin();
82    calcReducedMass();
```

```
83    CalculateKickAngle();
84    CalculateKickComponents();
85    KickProton();
86    CalculateNKP4Momenta();
87    BreakDeuteron();
88    FragmentsToLF();
89
90    G4bool kicked;
91    G4double targetMass = theTarget.AtomicMass(TargetNucleus ->
92        GetA_asInt(), TargetNucleus -> GetZ_asInt());
93    G4double Ecm = sqrt(pow(DRestMass,2) + targetMass*(targetMass +
94        2*LF_D_4Momentum.e())) - targetMass - DRestMass;
95
96    while(breakSuccesful == false)
97       {
98       loopCounter = 1;
99                kicked = KickedEnough();
100      while(reSample == true)
101         {
102         reSample = false;
103
104         deuteron -> RandomNeutron();
105         NucleonsToLF(deuteron);
106
107         if(LF_P_4Momentum->vect().getZ()<0){reSample=true;}
108         else
109            {
110            calcReducedMass();
111                  calcDeltaPpMin();
112                  calcThetaMin();
113            CalculateKickAngle();
114                 CalculateKickComponents();
115                 KickProton();
116                 CalculateNKP4Momenta();
117            BreakDeuteron();
118               FragmentsToLF();
119            kicked = KickedEnough();
120            }
121         loopCounter+=1;
122         }
123      while((kicked==0 && loopCounter <= 500))
124         {
125         if(firstItt==true) firstItt = false;
```

```
126         reSample = false;
127
128         calcThetaMin();
129
130         if(nOfReSamples>100)
131            {
132            nOfReSamples=0;
133            thetaJumps += 1;
134            thetaMin = thetaMin+(M_PI/2000)*thetaJumps;
135            if(thetaMin>M_PI){thetaMin=M_PI;}
136            }
137         if(bigCounter>500)thetaMin=M_PI;
138
139         calcReducedMass();
140         calcDeltaPpMin();
141                      calcThetaMin();
142         CalculateKickAngle();
143             CalculateKickComponents();
144         KickProton();
145         CalculateNKP4Momenta();
146         BreakDeuteron();
147                       FragmentsToLF();
148         kicked = KickedEnough();
149
150         loopCounter += 1;
151         }
152      if(kicked==1 && magEnergyChange(LF_D_4Momentum)<accuracyLimit)
153         {
154         breakSuccesful=true;
155         thetaJumps = 0;
156         BreakDeuteron();
157         FragmentsToLF();
158         }
159      else if(kicked==1) reSample=true;
160      if(loopCounter>=500)
161         {
162         loopCounter = 0;
163         nOfReSamples += 1;
164         reSample = true;
165         }
166      }
167   }
168
```

```
169 //************************************
170 //Constructor used by ApplyYourself method
171 //************************************
172 brokenDeuteronKick::~brokenDeuteronKick()
173    {
174     delete NKP_N_4Momentum;
175     delete NKP_P_4Momentum;
176    delete NKP_P_4Momentum_PK;
177    delete NKP_Q_4Momentum;
178    delete NKP_P_4Momentum_UB;
179    delete NKP_N_4Momentum_UB;
180    delete LF_N_4Momentum;
181    delete LF_P_4Momentum;
182    delete LF_P_4Momentum_PK;
183    delete LF_Q_4Momentum;
184    delete LF_N_4Momentum_UB;
185    delete LF_P_4Momentum_UB;
186    delete LF_Target_4Momentum;
187    delete kick;
188    delete TargetNucleus;
189    delete DRFBetaComponents;
190    delete PBetaComponents;
191    delete NKP_Beta;
192    delete deuteron;
193    }
194
195
196 //************************************
197 //Test to see if class is suitable to a
198 //a passed combination of particle and
199 //target nucleus
200 //************************************
201 G4bool brokenDeuteronKick::IsApplicable (const G4HadProjectile
202     &theTrack, G4Nucleus &theTarget)
203    {
204    const G4ParticleDefinition *definitionP = theTrack.GetDefinition();
205    const G4String ParticleName = definitionP -> GetParticleName();
206
207    delete definitionP;
208
209    if (ParticleName == "deuteron" &&
210        theTrack.GetKineticEnergy()>(-1*DBindingEnergy)) {return true;}
211    else {return false;}
```

```
212    }
213
214  //************************************
215  //Method used to run class by Geant4.
216  //Particle and target are passed from outside
217  //used within the class and secondaries
218  //are returned to be tracked.
219  //************************************
220  G4HadFinalState* brokenDeuteronKick::ApplyYourself(const
221      G4HadProjectile& theTrack, G4Nucleus& theTarget)
222    {
223     theParticleChange.Clear();
224     theParticleChange.SetStatusChange(stopAndKill);
225
226    GLorentzVector* Deuteron_LF_4Momentum = new
227        G4LorentzVector(theTrack.Get4Momentum());
228
229    brokenDeuteronKick* brokenDeuteron = new
230        brokenDeuteronKick(*Deuteron_LF_4Momentum, theTarget);
231
232    G4DynamicParticle* recoilNucleus = new G4DynamicParticle();
233     recoilNucleus -> SetDefinition(G4GenericIon::Definition());
234     recoilNucleus ->
235         SetMass(theTarget.AtomicMass(theTarget.GetA_asInt(),
236         theTarget.GetZ_asInt()));
237    recoilNucleus ->
238        SetMomentum(brokenDeuteron->LF_Target_4Momentum->vect());
239
240    G4DynamicParticle* returnProton = new G4DynamicParticle();
241    returnProton -> SetDefinition(G4Proton::Definition());
242    returnProton ->
243        SetMomentum(brokenDeuteron->LF_P_4Momentum_UB->vect());
244
245    G4DynamicParticle* returnNeutron = new G4DynamicParticle();
246     returnNeutron -> SetDefinition(G4Neutron::Definition());
247     returnNeutron ->
248         SetMomentum(brokenDeuteron->LF_N_4Momentum_UB->vect());
249
250    theParticleChange.AddSecondary(recoilNucleus);
251    theParticleChange.AddSecondary(returnProton);
252    theParticleChange.AddSecondary(returnNeutron);
253
```

```
254    delete Deuteron_LF_4Momentum, delete brokenDeuteron, delete
255        recoilNucleus, delete returnProton, delete returnNeutron;
256
257    return &theParticleChange;
258    }
259
260 //**********************************
261 //Method to calculate lab components
262 //of Coulomb kick from scattering angles.
263 //**********************************
264 void brokenDeuteronKick::CalculateKickComponents()
265    {
266    KickPerp = reducedMass * PBetaComponents -> mag() * sin(ThetaKick);
267    KickParr = reducedMass * PBetaComponents -> mag() *
268        (cos(ThetaKick)-1);
269
270    G4ThreeVector* pKick = new G4ThreeVector(KickPerp*cos(PhiKick),
271        KickPerp*sin(PhiKick), KickParr);
272
273    *kick = Rotate(pKick);
274
275    delete pKick;
276    }
277
278 //**********************************
279 //Constructor used by ApplyYourself method
280 //**********************************
281 void brokenDeuteronKick::CalculateKickAngle()
282    {
283     G4double RNum = ((G4double)rand()/(G4double)RAND_MAX);
284    RNum = sqrt(RNum*RNum);
285    G4int targetZ = TargetNucleus -> GetZ_asInt();
286
287    *PBetaComponents = calculateBeta(LF_P_4Momentum);
288
289    G4double targetMass = TargetNucleus -> AtomicMass(TargetNucleus ->
290        GetA_asInt(), TargetNucleus -> GetZ_asInt());
291    G4double ScatteringTerm = (reducedMass*pow(PBetaComponents->mag(),
292        2)/targetZ)*COULOMB_CONSTANT;
293    G4double B_MAX = (1/tan(thetaMin/2)) * (1/COULOMB_CONSTANT) *
294        (targetZ/(reducedMass * pow(PBetaComponents->mag(), 2)));
295    G4double ImpParameter = B_MAX*sqrt(RNum);
296
```

```
297     ThetaKick = 2*atan(1/(ScatteringTerm*ImpParameter));
298
299    //*************************************************
300    //Essential to resample RNum before generating PhiKick to prevent
301        correlation between Theta and Phi
302    //*************************************************
303    RNum = ((G4double)rand()/(G4double)RAND_MAX);
304     RNum = sqrt(RNum*RNum);
305    PhiKick = RNum*2*M_PI;
306    }
307
308 //**********************************
309 //Method to calculate the minimum change
310 //in momentum that would be required to
311 //break the deuteron under optimum
312 //conditions
313 //**********************************
314 void brokenDeuteronKick::calcDeltaPpMin()
315        {
316    G4double protMom = LF_P_4Momentum->vect().mag();
317    G4double protEn = LF_P_4Momentum -> e();
318
319        deltaPpMin = - protMom+sqrt((protEn-DBindingEnergy) *
320            (protEn-DBindingEnergy) - PRestMass*PRestMass);
321        }
322
323 //**********************************
324 //Method to apply the momentum kick
325 //to the proton
326 //**********************************
327 void brokenDeuteronKick::KickProton()
328    {
329        G4ThreeVector* LF_P_3Momentum_PK = new G4ThreeVector(
330                                    LF_P_4Momentum->vect().getX() +
331                                        kick->getX(),
332                                             LF_P_4Momentum->vect().getY()
333                                                 + kick->getY(),
334                                             LF_P_4Momentum->vect().getZ()
335                                                 + kick->getZ());
336
337        LF_P_4Momentum_PK -> set(*LF_P_3Momentum_PK,
338            sqrt(LF_P_3Momentum_PK->mag()*LF_P_3Momentum_PK->mag() +
339            PRestMass*PRestMass));
```

```
340
341    G4double targetMass = TargetNucleus -> AtomicMass(TargetNucleus ->
342        GetA_asInt(), TargetNucleus -> GetZ_asInt());
343
344    LF_Target_4Momentum -> set(*kick*-1, sqrt(kick->mag()*kick->mag() +
345        targetMass*targetMass));
346
347    delete LF_P_3Momentum_PK;
348    }
349
350 //*********************************
351 //Method to recalculate particle momenta
352 //in the CoM frame of the neutron and
353 //proton after the kick has been applied
354 //*********************************
355 void brokenDeuteronKick::CalculateNKP4Momenta()
356    {
357    CalculateBetaNKP();
358
359    *NKP_P_4Momentum  = lorentzBoost(LF_P_4Momentum, NKP_Beta);
360    *NKP_N_4Momentum  = lorentzBoost(LF_N_4Momentum, NKP_Beta);
361    *NKP_P_4Momentum_PK = lorentzBoost(LF_P_4Momentum_PK, NKP_Beta);
362    *NKP_Q_4Momentum  = lorentzBoost(LF_Q_4Momentum, NKP_Beta);
363    }
364
365 //*********************************
366 //Method to calculate beta of the CoM
367 //frame of the neutron and kicked proton
368 //*********************************
369 void brokenDeuteronKick::CalculateBetaNKP()
370    {
371    G4LorentzVector* NKP_4Momentum = new
372        G4LorentzVector(*LF_P_4Momentum_PK + *LF_N_4Momentum);
373
374    *NKP_Beta = calculateBeta(NKP_4Momentum);
375    delete NKP_4Momentum;
376    }
377
378 //*********************************
379 //Method to calculate the total momentum
380 //of the proton and neutron after breaking
381 //*********************************
382 G4double brokenDeuteronKick::CalculateNKPMomentumMag()
```

```
383    {
384    G4double totalEnergy = NKP_N_4Momentum->e() +
385        NKP_P_4Momentum_PK->e() + NKP_Q_4Momentum->e();
386    G4double momMag = sqrt(pow((totalEnergy*totalEnergy -
387        PRestMass*PRestMass + NRestMass*NRestMass)/(2*totalEnergy),2) -
388        NRestMass*NRestMass);
389
390    return(momMag);
391    }
392
393 //***********************************
394 //Method to calculate if the momentum
395 //kick is sufficient to overcome the
396 //deuteron potential energy
397 //***********************************
398 inline G4bool brokenDeuteronKick::KickedEnough()
399    {
400    G4double E1 = NKP_P_4Momentum_PK->e() + NKP_N_4Momentum->e() +
401        NKP_Q_4Momentum->e();
402    G4double E2 = NRestMass + PRestMass;
403
404    if(E1>E2){return true;}
405    else{return false;}
406    }
407
408 //***********************************
409 //Method to create 4-vectors with proton
410 //and neutron 4-momenta after removing
411 //potential energy
412 //***********************************
413 void brokenDeuteronKick::BreakDeuteron()
414    {
415    G4double momentum = CalculateNKPMomentumMag();
416    G4ThreeVector* P_Dir = new G4ThreeVector(NKP_P_4Momentum_PK->
417        vect()/NKP_P_4Momentum_PK->vect().mag());
418     G4ThreeVector* N_Dir = new G4ThreeVector(NKP_N_4Momentum->
419         vect()/NKP_N_4Momentum->vect().mag());
420
421    G4ThreeVector* P_3Momentum = new G4ThreeVector(*P_Dir*momentum);
422    G4ThreeVector* N_3Momentum = new G4ThreeVector(*N_Dir*momentum);
423
```

```
424    NKP_P_4Momentum_UB -> set(*P_3Momentum,
425        sqrt(P_3Momentum->mag()*P_3Momentum->mag() +
426        PRestMass*PRestMass));
427    NKP_N_4Momentum_UB -> set(*N_3Momentum,
428        sqrt(N_3Momentum->mag()*N_3Momentum->mag() +
429        NRestMass*NRestMass));
430
431    delete P_Dir, delete N_Dir, delete P_3Momentum, delete N_3Momentum;
432    }
433
434 //***********************************
435 //Method to boost the seperated proton
436 //and neutron back to the lab frame
437 //***********************************
438 void brokenDeuteronKick::FragmentsToLF()
439    {
440    G4ThreeVector* LFBeta = new G4ThreeVector(*NKP_Beta*-1);
441
442    *LF_P_4Momentum_UB = lorentzBoost(NKP_P_4Momentum_UB, LFBeta);
443    *LF_N_4Momentum_UB = lorentzBoost(NKP_N_4Momentum_UB, LFBeta);
444
445    delete LFBeta;
446    }
447
448 //***********************************
449 //Method to boost the proton, neutron
450 //and quasi-particle from the deuteron
451 //rest frame to the lab frame
452 //***********************************
453 void brokenDeuteronKick::NucleonsToLF(BrokenDeuteron* deuteron)
454    {
455    DRF_N_4Momentum -> set(*deuteron->GetNMomentumThreeVectorDRF(),
456        sqrt(deuteron->GetNMomentumThreeVectorDRF()->mag() *
457        deuteron->GetNMomentumThreeVectorDRF()->mag() +
458        NRestMass*NRestMass));
459    DRF_P_4Momentum -> set(*deuteron->GetPMomentumThreeVectorDRF(),
460        sqrt(deuteron->GetPMomentumThreeVectorDRF()->mag() *
461        deuteron->GetPMomentumThreeVectorDRF()->mag() +
462        PRestMass*PRestMass));
463
464    DRF_Q_4Momentum -> set(DRestMass - DRF_N_4Momentum->e() -
465        DRF_P_4Momentum->e(), G4ThreeVector());
466
```

```
467    G4ThreeVector* BetaComps = new G4ThreeVector(*DRFBetaComponents*-1);
468
469    *LF_N_4Momentum = lorentzBoost(DRF_N_4Momentum, BetaComps);
470    *LF_P_4Momentum = lorentzBoost(DRF_P_4Momentum, BetaComps);
471    *LF_Q_4Momentum = lorentzBoost(DRF_Q_4Momentum, BetaComps);
472
473    delete BetaComps;
474    }
475
476
477 /*
478    "Utility" functions:
479    Lorentz boosts, minor calculations, etc
480 */
481
482 //**********************************
483 //An alternative Lorentz boost method
484 //**********************************
485 G4LorentzVector brokenDeuteronKick::lorentzBoost(G4LorentzVector*
486     sourceParticle, G4ThreeVector* betaComponents)
487    {
488    G4double beta = betaComponents -> mag();
489    G4LorentzVector* BoostedVector = new G4LorentzVector();
490
491    if(beta>0)
492       {
493       G4double gamma = 1/sqrt(1-beta*beta);
494
495       G4double betaX = betaComponents -> getX();
496       G4double betaY = betaComponents -> getY();
497       G4double betaZ = betaComponents -> getZ();
498
499       G4double beta2 = beta*beta;
500
501       G4double gammaBetaX = gamma * betaX;
502       G4double gammaBetaY = gamma * betaY;
503       G4double gammaBetaZ = gamma * betaZ;
504
505       G4double gamma1 = gamma - 1;
506
507       G4double sourcePx = sourceParticle->px();
508       G4double sourcePy = sourceParticle->py();
509       G4double sourcePz = sourceParticle->pz();
```

```
510       G4double sourceP = sqrt(sourcePx*sourcePx + sourcePy*sourcePy +
511           sourcePz*sourcePz);
512       G4double sourceEnergy = sourceParticle->e();
513
514       G4double boostedEnergy = gamma*sourceEnergy - gammaBetaX*sourcePx
515           - gammaBetaY*sourcePy - gammaBetaZ*sourcePz;
516
517       G4double boostedPx = -1*gammaBetaX*sourceEnergy +
518           (1+gamma1*(betaX*betaX/beta2))*sourcePx +
519           gamma1*betaX*betaY/beta2*sourcePy +
520           gamma1*betaX*betaZ/beta2*sourcePz;
521       G4double boostedPy = -1*gammaBetaY*sourceEnergy +
522           gamma1*betaY*betaX/beta2*sourcePx +
523           (1+gamma1*(betaY*betaY/beta2))*sourcePy +
524           gamma1*betaY*betaZ/beta2*sourcePz;
525       G4double boostedPz = -1*gammaBetaZ*sourceEnergy +
526           gamma1*betaZ*betaX/beta2*sourcePx +
527           gamma1*betaZ*betaY/beta2*sourcePy +
528           (1+gamma1*(betaZ*betaZ/beta2))*sourcePz;
529
530       *BoostedVector = G4LorentzVector(boostedPx, boostedPy, boostedPz,
531           boostedEnergy);
532       }
533     else
534       {
535       *BoostedVector = *sourceParticle;
536       }
537
538     G4LorentzVector returnVector;
539     returnVector.setPx(BoostedVector->px());
540     returnVector.setPy(BoostedVector->py());
541     returnVector.setPz(BoostedVector->pz());
542     returnVector.setE(BoostedVector->e());
543
544     delete BoostedVector;
545     return(returnVector);
546     }
547
548 //***********************************
549 //Method to calculate beta for a given
550 //4-momentum
551 //***********************************
```

```
552 G4ThreeVector brokenDeuteronKick::calculateBeta(G4LorentzVector*
553     particle)
554    {
555    G4double betaX = particle->px()/particle->e();
556    G4double betaY = particle->py()/particle->e();
557    G4double betaZ = particle->pz()/particle->e();
558 
559    return G4ThreeVector(betaX, betaY, betaZ);
560    }
561 
562 //***********************************
563 //Method to calculate the reduced mass
564 //of two particles
565 //***********************************
566 void brokenDeuteronKick::calcReducedMass()
567    {
568    G4double targetMass = TargetNucleus -> AtomicMass(TargetNucleus ->
569        GetA_asInt(), TargetNucleus -> GetZ_asInt());
570    G4double gamma =
571        sqrt(1+(pow(LF_P_4Momentum->vect().mag()/PRestMass,2)));
572 
573    reducedMass =
574        (targetMass*gamma*PRestMass)/(targetMass+gamma*PRestMass);
575    }
576 
577 //***********************************
578 //Method to check if the proton momentum
579 //will allow the deuteron to be broken
580 //***********************************
581 inline G4bool brokenDeuteronKick::isPpEnough()
582    {
583    G4double upperBound = sqrt(pow(LF_P_4Momentum->e() +
584        sqrt(DBindingEnergy*DBindingEnergy),2)-PRestMass*PRestMass);
585    G4double lowerBound = sqrt(pow(LF_P_4Momentum->e() +
586        sqrt(DBindingEnergy*DBindingEnergy),2)-PRestMass*PRestMass) -
587        2*reducedMass*DRFBetaComponents->mag();
588 
589    return (LF_P_4Momentum->vect().mag()<=upperBound &&
590        LF_P_4Momentum->vect().mag()>=lowerBound);
591    }
592 
593 //***********************************
594 //Method to calculate the minimum scattering
```

```
595 //angle required to provide the minimum kick
596 //**********************************
597 void brokenDeuteronKick::calcThetaMin()
598    {
599    thetaMin = acos(1-(0.5*deltaPpMin*deltaPpMin)/
600        (2*reducedMass*reducedMass*DRFBetaComponents->mag() *
601        DRFBetaComponents->mag()));
602    }
603 
604 //**********************************
605 //Method to realign the kick to the lab
606 //axis after calculating it in axis
607 //with proton momentum aligned along Z
608 //**********************************
609 G4ThreeVector brokenDeuteronKick::Rotate(G4ThreeVector* inputVector)
610    {
611    G4ThreeVector* LabPP = new G4ThreeVector(LF_P_4Momentum->vect());
612    G4ThreeVector* LinearPP = new G4ThreeVector(0, 0,
613        LF_P_4Momentum->vect().mag());
614 
615    G4double rotTheta = acos((LinearPP->getX()*LabPP->getX() +
616        LinearPP->getY()*LabPP->getY() +
617        LinearPP->getZ()*LabPP->getZ())/ (sqrt(pow(LabPP->getX(),2) +
618        pow(LabPP->getY(),2)+pow(LabPP->getZ(),2)) *
619        sqrt(pow(LinearPP->getX(),2) +
620        pow(LinearPP->getY(),2)+pow(LinearPP->getZ(),2))));
621 
622    G4double rotPhi = atan(LabPP->getY()/LabPP->getX());
623 
624    if((LabPP->getY()<0 || LabPP->getX()<0) || LabPP->getZ()<0){rotTheta
625        = -rotTheta;}
626    if(LabPP->getZ()<0&&LabPP->getX()>0){rotPhi = M_PI+rotPhi;}
627    else if(LabPP->getY()<0&&LabPP->getZ()>0){rotPhi = rotPhi+M_PI;}
628 
629    G4ThreeVector* trueKick = new G4ThreeVector(
630        (inputVector->getX()*cos(rotTheta) +
631        inputVector->getZ()*sin(rotTheta))*cos(rotPhi) -
632        inputVector->getY()*sin(rotPhi),
633    (inputVector->getX()*cos(rotTheta) +
634        inputVector->getZ()*sin(rotTheta))*sin(rotPhi) +
635        inputVector->getY()*cos(rotPhi),
636    -1*inputVector->getY()*sin(rotTheta) +
637        inputVector->getZ()*cos(rotTheta));
```

```
638    delete LabPP, delete LinearPP;
639
640    G4ThreeVector returnKick;
641
642    returnKick = *trueKick;
643
644    delete trueKick;
645    return returnKick;
646    }
647
```