

Declaration of Academic Integrity

*to be used throughout the University from 2012-13
for submitted work at undergraduate and taught postgraduate levels:*


I promise that in the attached submission I have not presented or attempted to present anyone else's work as my own except where I have explicitly so indicated. I understand that to do so would mean that I had committed plagiarism, the University's definition of which is published. I know that if I commit plagiarism I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance.

I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes.

I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the School's policy on mitigation and procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Full Name.....Matthew Briggs.....

Student Number.....200275847.....

Signed..........

Date.....May 7th 2013.....

Plagiarism is defined as presenting someone else's work as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance, or any combination of these.

DEPARTMENT OF PHYSICS AND ASTRONOMY
UNIVERSITY OF LEEDS

MPHYS PROJECT THESIS

Geoneutrinos at SNO+

Author:

Matthew BRIGGS

ID: 200275847

E-mail: py09mb@leeds.ac.uk

Supervisor:

Dr. Joachim ROSE

Assesor:

Dr. Almut BEIGE

May 7, 2013

Abstract

The SNO+ detector is in the process of being reconfigured to a liquid scintillator material that will enable it to detect lower energy events than were previously available. A small number of geoneutrino events will for the first time be captured at the site in addition to other experiments. Work needs to be done to model the geoneutrinos that are theoretically detectable which includes modelling the spectra and flux of incoming geoneutrinos and simulate how they would appear in the detector using custom tools. The project had two main goals: to write a standalone C++ program to generate antineutrino decay spectra for isotopes that will produce detectable geoneutrinos at the SNO+ site and to simulate how these geoneutrinos would appear inside the detector. This article briefly outlines the SNO+ detector, the methods of generating spectra from theory and how they are implemented through code including a description of the complex gamma function and lanczos spline interpolation. It also covers how the SNO+ reconstruction and analysis tools are used to simulate events. The tools were used alongside ROOT to generate and plot the antineutrino energy cross section in inverse betat decay.

Contents

1	Introduction	1
1.1	Geoneutrinos	1
1.2	Solar Neutrino Problem	1
1.3	SNO+ Detector	2
1.4	Goals	3
2	Antineutrino Energy Spectra	3
2.1	Decay Chain Tables	4
3	C++ Implementation of Decay Spectra	5
3.1	Complex Gamma Function	6
3.2	Spline Function	7
3.3	Results	7
4	Detector Simulation	8
4.1	Generating Geoneutrinos with RAT	9
4.2	C++ To Analyse ROOT Output	10
4.3	Simulation Results	11
5	Conclusions	11

1 Introduction

1.1 Geoneutrinos

Geoneutrinos are electron antineutrinos produced through the inverse beta decay of radioactive elements from within the surface of the Earth^[1]. The detection of geoneutrinos is important as the spectra and fluxes can give insights into the geophysical structure of the Earth such as the abundances and distributions of various radiogenic elements in the crust, mantle, and core — which due to their depth are very difficult to measure with other methods because of their relatively small cross section and low interaction probability; therefore, a good way to measure this is with geoneutrinos. Since the geoneutrinos come from radioactive sources in the Earth they can also be used to measure the radiogenic heat output of the Earth. Previous methods to measure the composition have come from estimations made by studying chondritic meteorites that are believed to be representative, by having similar bulk compositions, of the early Earth^[2]. Geoneutrinos were first proposed to be studied in this way by Marx^[3] and Eder^[4] in the late 1960s but were not successfully detected until the KamLAND detector published its first results in 2005. A large increase in detector sensitivity was first needed.

1.2 Solar Neutrino Problem

The Homestake experiment began operation in 1965; designed by Davis it was the first of its kind to detect solar neutrinos in an attempt to verify the standard solar model predicted by Bahcall. A one-hundred thousand gallon tank of tetrachloroethylene was placed in a mine in South Dakota, upon a neutrino striking the liquid it would cause an inverse electron capture and cause the production of Argon and an electron^[5]. The Argon was collected and from the volume of gas a prediction of the solar neutrino flux could be calculated. Upon analysis of several years of data it was found that the measured capture rate was more than a third lower than the model made by Bahcall, some were missing. This discrepancy was coined as the Solar Neutrino Problem. Improvements in detection in later experiments such as Super Kamiokande and GALLEX were able to measure neutrinos produced from more prevalent decay chains in the sun and also give the direction

of their origin, indicating these were indeed solar neutrinos, but it wasn't until the SNO experiment which started in 1999 that the problem was finally put to rest. The SNO detector was designed with a capability to be sensitive to all three types of neutrino through the Charged Current, Elastic Scattering, and Neutral Current channels - whereas previous experiments only utilised the CC channel which can detect only electron neutrinos. SNO conclusively showed a flux in agreement with the solar model and therefore gave evidence of electron neutrinos produced in the sun changing flavour between the time they were created and their arrival at the detector^[6].

1.3 SNO+ Detector

The SNO experiment is located at INCO's Creighton mine in Sudbury, Ontario. The detector was originally designed to study neutrino flavour oscillations, as discussed above, and effectively put to rest the Solar Neutrino Problem. The detector shown in Figure 1 is 2039m underground and has significant granite overburden equivalent to 6000m of water leading to a small muon background rate of only 70 passing through the PMT array per day. After the success of the previous SNO experiments the detector is in the process of being reconfigured to start a new batch of experiments under the moniker SNO+. The change mainly involves swapping the heavy water inside the detector vessel to a liquid scintillator material. ^[7] and adding extra support ropes to hold the detector in place since the scintillator is lighter than water and the vessel could float upward.

The benefits of switching to the liquid scintillator over the previous Cherenkov-based water detection method is a vast increase in detection efficiency. The light yield is boosted by a factor of fifty allowing a much lower detection threshold energy. This allows SNO+ to explore new areas of physics that the previous experiment could not. Aside from geoneutrinos there is the possibility to study electron antineutrinos from commercial nuclear reactor power plants which can provide a confirmation of KamLAND's LMA neutrino oscillation experiments. The flux and spectrum of reactor antineutrinos are well studied so reactor oscillation experiments provide a good method of measuring oscillation parameters. There is also the possibility of detecting much lower energy solar neutrinos such as those in the PEP, CNO, and ⁷Be decay chains. The most exciting investigation the upgrade brings is the future possibility of a neutrino less double beta decay experiment (a staged approach, available after more development and the addition of a doping agent into the scintillator) which could answer whether neutrinos are distinct from their anti-particles (Dirac particle) or if they are identical to their own anti-particle except for helicities (Majorana particle). Considering a double beta decay process: if neutrinos are Majorana particles one of the two antineutrinos would be absorbed by the other as a neutrino and become neutrino less double beta decay. If this reaction is observed this would provide confirmation and, since the rate of decay is related to the neutrino mass, other promising insights in neutrino physics.

SNO+ shows a promising possibility for geoneutrino detection. Using the inverse beta decay reaction as shown in Equation 1, antineutrinos react with protons in the liquid scintillator and produce positrons and neutrons. The positrons have a fixed kinetic energy of 1.804 MeV and it annihilates an electron inside the detector to immediately produce two photons which deposit 0.8 mEv energy into the detector. This is considered to be the prompt event. A second, delayed, event occurs after a mean time of 200 micro seconds when the neutron is captured by a proton which produces another photon with a discrete energy of 2.2 MeV. The delayed coincidence events provide a distinct signal that separates the geoneutrino detection from background events reliably.

$$\bar{\nu}_e + p \rightarrow n + e^+ \quad (1)$$

In comparison with other collaborations attempting to detect geoneutrinos the SNO+ detector should potentially be able to detect 64 events per year which is more than double that of KamLAND's 29 events and six times greater than Borexino's 10 events per year. This is due to the site at Sudbury being surrounded by the

Canadian Shield, a thicker section of continental crust which has higher concentrations of U and Th than the oceanic crust found in Japan at the KamLAND detector^[8].

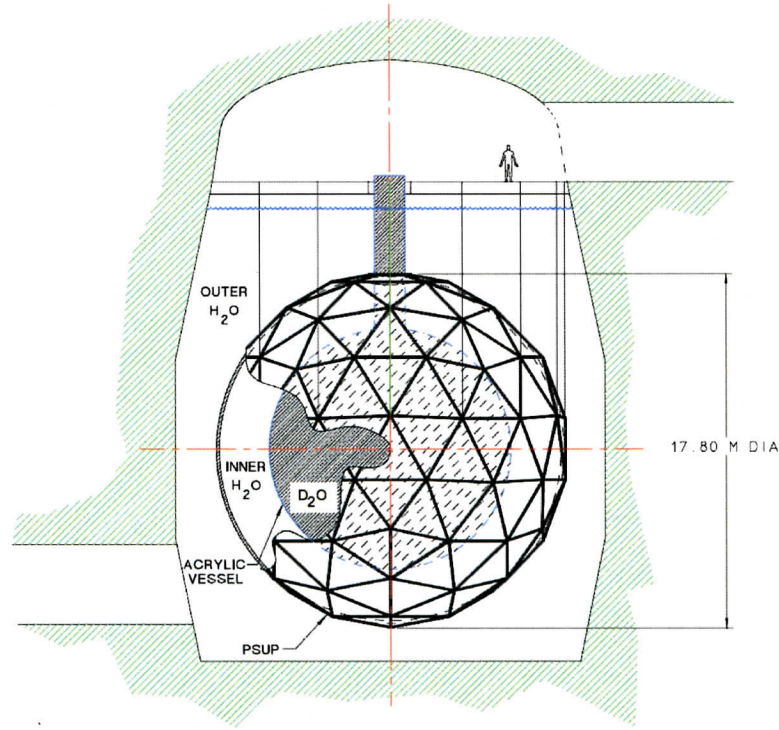


Figure 1: Layout of the original SNO detector cavity system showing the acrylic vessel and outer PMT support structures (PSUP)^[9]

1.4 Goals

- Generate and test initial standalone code for geoneutrino emission spectra
- Compare predicted emission spectra against measured beta decay spectra with literature
- Generate simulated geoneutrino events including detector reconstruction
- Plot the reconstructed (measured) geoneutrino energy spectrum in ROOT

2 Antineutrino Energy Spectra

Antineutrino spectra of the inverse beta decay of Uranium and Thorium are required to model the signal that can be achieved in the SNO+ detector. The theoretical count rates can then be utilised as an input into detector simulation software to obtain a representation of what the detector will see when activated and collecting signal. The detector at SNO+ has a certain resolution to the energies which it can detect (1.8 MeV minimum) so only decays which produce antineutrinos above this threshold are investigated. There are only three significant radioactive elements that produce geoneutrinos with high enough contribution to the total signal to be considered significant; these are ^{238}U , ^{232}Th , and ^{40}K but only the first two produce antineutrinos with sufficient energy to meet the detector requirements.

Each element when it undergoes radioactive decay has a number of decay modes and possibilities of the daughter particles and subsequent energy ranges of the antineutrinos it can produce so this has to be considered when calculating the spectra and each decay mode and branch has to be summed to give the total spectrum for the isotope. The beta decay $i \rightarrow j$ describes transitions to different nuclear states which develop multiple chains with different end points. $I_{i,j;k}$ is the percentage intensity of the k -th beta transition (number of decay modes) and $f_{i,j;k}(E)$ is the corresponding antineutrino energy spectrum for that decay. Each branch in the chain has a probability of occurrence as $R_{i,j}$. Therefore the total antineutrino spectrum for an element is described by the following expression^[10]:

$$f(E) = \sum_{ij} R_{i,j} \sum_k I_{i,j;k} f_{i,j;k}(E) \quad (2)$$

This describes the total spectrum but one must also consider how to arrive at the spectrum of each individual decay. This is done by considering energy conservation in the beta decay process. Since there is an electron generated alongside an antineutrino and the electron energy spectrum has a well defined and studied shape it can be worked from backwards to arrive at the antineutrino energy spectrum. In the beta decay of a nucleus i to a nucleus j in the state k the antineutrino energy spectrum $f_{i,j;k}(E)$ is related to the electron energy spectrum $\phi_{i,j;k}(W)$ as:

$$f_{i,j;k}(E) = \phi_{i,j;k}(W)|_{W=W_{max}-E} \quad (3)$$

And W is the total electron energy with W_{max} the maximum electron energy in that transition where $W_{max} = m_e c^2 + E_{max}$ and E_{max} being the maximum antineutrino energy and m_e is the standard electron rest mass energy. The electron energy spectrum is considered through an application of perturbations in quantum mechanics and the use of Fermi's Golden Rule. The Fermi function used here is relativistic and only considers the Coulombic attraction between the emitted and final state nucleus without any electron screening or nuclear size events. If these were to be used a small correction would be applied but only a few percent which, in comparison to other uncertainties, is not significant enough to be considered.

$$\phi_{i,j;k}(W) = \frac{1}{N} W (W_{max} - W)^2 (W^2 - m_e^2 c^4)^{\gamma-1/2} e^{\pi y} |\Gamma(\gamma + iy)|^2 \quad (4)$$

Where γ and y are simply truncated expressions for clarity and given in full as:

$$\gamma = \sqrt{1 - (\alpha Z)^2} \quad (5) \quad y = \alpha Z \frac{W}{\sqrt{W^2 - m_e^2 c^4}} \quad (6)$$

Z specifies the atomic number of the daughter nucleus (important to stress this is the daughter nucleus and not the parent), α is the fine structure constant, γ is the complex gamma function and N is a normalisation constant such that the sum of the electron energy spectrum is equal to one. Something noteworthy for when discussing the code later on can be seen from the integration limits in the normalisation, and from energy conservation, the energy of the electron is always greater than $m_e c^2$.

$$\int_{m_e c^2}^{W_{max}} dW \phi_{i,j;k}(W) = 1 \quad (7)$$

So, the above set of equations describe the theoretical outline of obtaining the antineutrino energy spectrum for a radioactive isotope undergoing beta decay. The two main isotopes that are the most significant sources of geoneutrinos, and can be detected by the SNO+ project, are described in the following section.

2.1 Decay Chain Tables

There are eight α and six β decays in the process of ^{238}U decaying to ^{206}Pb . Of these only three nuclides provide antineutrinos with sufficient energies to be detected (^{234}Pa , ^{214}Bi , and ^{210}Ti) but the contribution from ^{210}Ti is insignificant compared to its low probability of occurrence. The main contribution from ^{238}U

to the antineutrino energy spectrum is from the other two isotopes ^{234}Pa and ^{214}Bi which account for 98% of its contribution. Of these two isotopes there are a total of five decays which are listed in Table 1

Similarly, there is a chain of six α and four β decays comprising the total ^{232}Th decay into ^{208}Pb . Of these only two nuclides (^{232}Ac and ^{212}Bi) provide antineutrinos with sufficient energy to be detected. The effective thorium contribution comes from three β decays as listed below in Table 2.

Table 1: The five effective ^{238}U transitions with their associated decay probabilities, maximum antineutrino energy, the percentage intensity I_k with associated error, type and also the percentage contribution to the Uranium and total geoneutrino signals. Assuming a chondritic ratio of $\text{Th}/\text{U} = 3.9$

$i \rightarrow j$	R_{ij}	E_{max} (keV)	I_k	ΔI_k	Type	S_u	S_{tot}
$^{234}\text{Pa}_m \rightarrow ^{234}\text{U}$	0.9984	2268.92	0.9836	0.002	1st Forbidden $(0^-) \rightarrow 0^+$	39.62	31.21
$^{214}\text{Bi} \rightarrow ^{214}\text{Po}$	0.9998	3272.00	0.182	0.006	1st Forbidden $1^- \rightarrow 0^+$	58.21	45.84
		2662.68	0.017	0.006	1st Forbidden $1^- \rightarrow 2^+$	1.98	1.55
		1894.32	0.0743	0.0011	1st Forbidden $1^- \rightarrow 2^+$	0.18	0.01
		1856.51	0.0081	0.0007	1st Forbidden $1^- \rightarrow 0^+$	0.01	0.01

Table 2: The three effective ^{238}Th transitions with their associated decay probabilities, maximum antineutrino energy, the percentage intensity I_k with associated error, type and also the percentage contribution to the Uranium and total geoneutrino signals. Assuming a chondritic ratio of $\text{Th}/\text{U} = 3.9$

$i \rightarrow j$	R_{ij}	E_{max} (keV)	I_k	ΔI_k	Type	S_u	S_{tot}
$^{212}\text{Bi}_m \rightarrow ^{212}\text{Po}$	0.6406	2254	0.8658	0.0016	1st Forbidden $1^- \rightarrow 0^+$	94.15	20.00
$^{228}\text{Ac} \rightarrow ^{228}\text{Th}$	1.000	2069.24	0.08	0.06	Allowed $3^+ \rightarrow 2^+$	5.66	1.21
		1940.18	0.008	0.006	Allowed $3^- \rightarrow 4^+$	0.19	0.04

3 C++ Implementation of Decay Spectra

Isotopes are created in their own class as a vector object. This class also contains functions that are able to resize the vector to allow more entries, and constructor, and destructors and other functions to pass variables outside the class. As of right now the entries are fixed for each decay chain. Meaning there is a function called SetVect() that contains the E_{max} and I of the two decay chains of one Isotope hardcoded into the text. For optimal robustness it would be preferable to have a data file which is read into the program that can automatically fill these functions, that way many different isotopes could have their decay spectra generated easily. Unfortunately the code is not dynamic enough to be able to handle such a case but it did not seem very important since only a small number of decay chains and isotopes needed to be calculated. More planning toward scalability and robustness should have been considered initially. Moving back to the code. In the main function there are the usual declaration of variables with the isotopes being created as vectors and a minimum and maximum energy range entered (variables A , and B) with the step size being C . This is useful later for the loop that generates the spectrum. There is also a calculation of the total number of data points, P , or equivocally the number of iterations for the loop to run over. More variables are declared such as the arrays and pointers to arrays that hold energy values etc and a datafile is created for output. There is also a declaration of the constants in this section, m_e , α , and c . These are not in S.I. units but rather in terms of energy which should not cause any problems as everything cancels out in the end and an energy in keV is

given and saves time not having to convert back from S.I.

There is then a short while loop that fills in information about the chains. Here is where the z , R , and s values are sent to create a new isotope. Functions are then called from the isotope class to resize its vector from these declarations. This is a currently a little redundant given the method of inputting isotope information as stated previously but gives a good basis for any further modifications that enable more dynamic methods — the size of the vectors are automatically changed depending on the input. A list is then used. Isotopes created are added to the end of a list containing the isotope classes. This is used in the next section of the code that runs a loop from the start to end of this list and can again be utilised for more dynamic methods; i.e. having a variable number of isotopes, letting one add as many as required without having to change indexes in loops etc.

Now, arriving at the main calculation portion of the code. A for loop runs from zero to the index P , calculated earlier from the min/max energies and the increment size. At each step an energy is calculated from the minimum energy plus the current index multiplied by the increment — essentially the current increment gets added each time the loop completes. We run another loop inside this that iterates through the vector list items. Grabbing the R and s values from functions in the vector class and then another loop inside this that cycles through the number of branches in the chain of the current isotope in the list. Expressions inside utilise the formula described in Section 2 to first calculate W_{max} from the E_{max} of the current list item and then W from this. A condition such that $W > m_e c^2$ has to be met for further calculations in the loop otherwise nothing else is added to the spectra. If the condition is met further calculations are called to calculate Φ and then depending on the number of states available (For Uranium if $S = 1$ or $S = 4$) the results for ϕ are syphoned into separate arrays so individual decays can be graphed later on and not just the total. Outside this second nested loop there are then the standard expressions to clear variables etc and statements to turn the f values of the spectra to intensities by a multiplication of the current energy of the iteration. Once all the values for the spectra at each energy have been calculated they can then be passed to a spline function as described in Section 3.2 or graphed directly using some of the ROOT internal plotting commands. ROOT's own graphing routines are quite a task to work with and the documentation is quite convoluted but being constrained by C++ it is difficult to find a better solution without using an external program. It is much more practical to have everything contained in one program that graphs its output during runtime without having to keep importing data to separate programs each run.

3.1 Complex Gamma Function

From Equation 4 one can notice the use of the complex Gamma function in the calculation of the electron energy spectrum, leading to a question regarding what the best method of application is when generating $\phi_{i,j;k}(W)$ in the code. The Gamma function in basic terms is an extension of the factorial and is used to calculate the factorial for all real and complex values. Many different approaches can be taken toward the computation of the Gamma function — although the Sterling Approximation is the most popular there exists a more efficient method for calculating the gamma of large polynomials developed by Lanczos. An implementation of the Lanczos approximation was used with the addition of pre-calculated coefficients to speed up computation. A comprehensive work by Pugh^[11] has been published which states the Lanczos method is the most efficient compared to the others with the lowest relative error in calculation. The implementation is listed in Chapter 6.1 of the Numerical Recipes book^[12]:

$$\Gamma(z + 1) = \sqrt{2\pi} \left(z + g + \frac{1}{2} \right)^{z + \frac{1}{2}} e^{-(z + g + \frac{1}{2})} A_g(z) \quad (8)$$

Where $A_g(z)$ is a convergent series that gives a certain precision based on the number of terms to sum, it can be made sufficiently easy to calculate with a fixed value of g and coefficients calculated and stored beforehand so no extra calculations are needed at runtime and thus increasing the performance of the program. Generally

a g value of 9 is the most common as stated in the Numerical Recipes text which is where the coefficients were taken from to make the numerical sum $A_g(z)$ have the following simplified form:

$$A_g(z) = c_0 + \sum_{k=1}^N \frac{c_k}{z+k} \quad (9)$$

Thus, the problem of calculating the complex gamma function reduces to implementing the previous equations with predetermined coefficients (each coefficient has a precision of at least 10 decimal places) and in cases where $z < 0.5$ (right-hand) it can be extended to the entire complex plane with Euler's reflection formula:

$$\Gamma(z) = \frac{\pi}{\sin(\pi z)} \frac{1}{\Gamma(1-z)} \quad (10)$$

3.2 Spline Function

The calculation of antineutrino energy spectra is handled by way of taking an minimum energy input and calculating the intensity at that one point and then running in a loop over a set interval of energy increases until the maximum energy is reached. To make the results more easily available for future use a function needed to be implemented that could return an intensity for any energy that was given to it and not just the discrete set of specific energies given as the input. To achieve this a standard spline function from the GSL code libraries was used. A spline function is a function that can, essentially, connect a series of points using polynomials to provide a smooth curve. There are many different methods of implementing the spline function such as a basis or interpolation spline. The basis spline is more often used to connect a series of noisy data where the curve does not have to pass through every data point^[12]. For the problem at hand it is required to use the interpolation spline which does pass through every datapoint since the calculated data points do indeed have to be adhered to since the data points are not arrived at through experiment. The GSL library has six available interpolation routines; some of which can be discounted such as the polynomial which introduces oscillations when used on large numbers of data points and the linear interpolation which cannot handle more complex functions. The method chosen for this work was a cubic interpolation spline that uses piecewise polynomials with continuous first and second derivatives at the data points. Little testing was done with regards to the accuracy of interpolated output except basic comparisons on a few data points which did agree with what was expected. A modified version of this function called the Akima spline is also available to GSL. The Akima method is once again a local interpolation but is more stable at outliers. In addition it is purported to be more efficient with regards to memory usage and may offer faster runtime execution^[13]. Changing to this method should only require a simple alteration of a line of code: *gsl_interp_cspline* to *gsl_interp_akima* and could prove worthwhile to investigate.

3.3 Results

In Figures 2 and 3 the result of the decay chain spectra program are shown. The total spectrum for each isotope labelled in the bold line with each individual contribution coloured according to the legend. It can be seen that the drop-off energy point corresponds exactly to the E_{max} values as listed in Tables 1 and 2 which shows correspondence with expected values on this axis. In comparison with the results from Fiorentini^[10] it can also be seen that the general shape is mostly the same (more so for Thorium than Uranium). There is a slight problem with normalisation however and with a direct comparison the y-values for the neutrino intensity do not match up to those in Fiorentini. This is a perplexing problem. All spectra were normalised such that the area under the total was equal to the number of decays in the chain as stated in literature. This was double checked with a test in the code that would give an warning if the normalisation was not correct. The spectra have the correct shape and end-points so the problem must lie with normalisation.

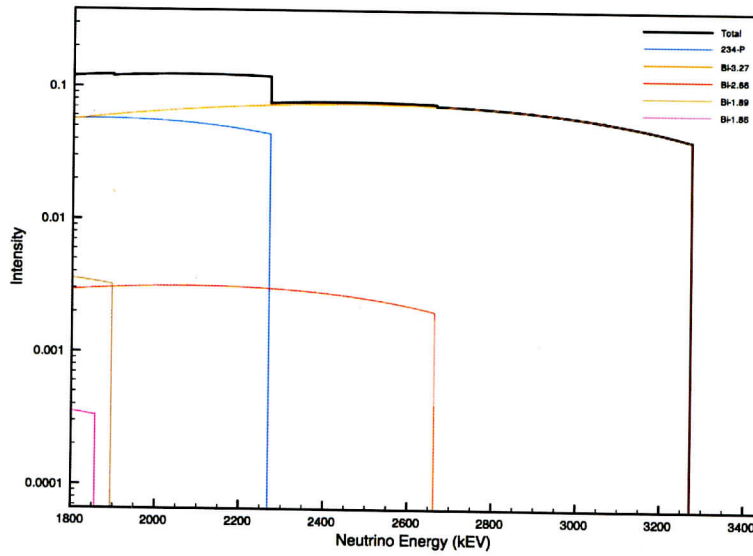


Figure 2: Output of the normalised Uranium beta decay spectrum calculated from theory. The bold line shows the total spectrum whilst the individual lines represent each contribution to the total. For isotopes with multiple entries the maximum energy in MeV is shown to the right of the entry in the legend. The total spectrum was normalised so the area under the curve equals six and each individual contribution normalised with the condition specified in Equation 7

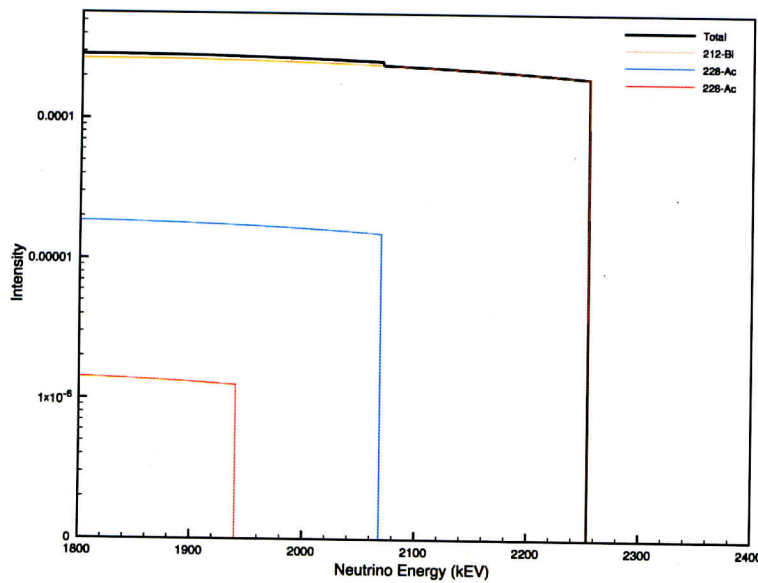


Figure 3: Output of the normalised Thorium beta decay spectrum calculated from theory. The bold line shows the total spectrum whilst the individual lines represent each contribution to the total. The total spectrum was normalised so the area under the curve equals four and each individual contribution normalised with the condition specified in Equation 7

4 Detector Simulation

Detector simulation was carried out using a custom tool called RAT which is a simulation and analysis package built with GEANT4^[14], ROOT, and C++, originally developed by S. Seibert for the Braidwood

Collaboration. It is now being used (and developed) by the miniCLEAN, DEAP-3600, and SNO+ collaborations. Using RAT one can generate particles from a multitude of sources: solar neutrinos, reactor neutrinos, inverse beta decay, and many more. The SNO+ team have modelled the detector inside this framework so any simulated events can be sent a very accurate representation of the detector to test all possibilities before construction. RAT is a large, complex, and powerful tool and can be used for so much more than studied here. The main focus of this aspect of the project was to start investigations toward programming some basic simulations and writing code to analyse their output.

4.1 Generating Geoneutrinos with RAT

Before starting to simulate antineutrinos passing through the detector the decision was made to tackle a smaller task and instead focus on getting the inverse beta decay generator set-up and starting to generate some basic events. The simulations are controlled by macro files that are loaded into RAT, these are analogous to batch files which send a set of commands in succession. Hadronic and muonic processes are omitted since the inverse beta decay is only needed for analysis of the antineutrinos and electrons/positrons from the resulting decay, other particles such as neutrons do not need to be tracked or even simulated. A statement is inserted at the top of the macro to turn off these processes and speed up simulation:

```
/glg4debug/glg4param omit_muon_processes 1.0
/rlg4debug/glg4param omit_hadronic_processes 1.0
```

The simulation software can also use different detector geometry profiles, setting up custom shapes and configurations including the ropes holding the detector, belly plates, etc. This was turned off again since the simulation was not using the detector to process any particles. They were only being generated. Nothing was passed through a detector. Simple detector geometry was initialised through the following line in the macro.

```
/rat/db/set DETECTOR geo_file "geo/snoplus_Simple.geo"
```

Next are the standard initialisation commands that set up various processor elements. Here basic commands were used to set a counter — all this does is give an output to the terminal window displaying which event it is currently processing and the update command updates the output every one event. There is also a command to set the output as a ROOT file and the corresponding file name. Much more processes can be added at this stage such as the event builder, PMT calibration, front end DAC electronics for a very in-depth simulation.

```
/run/initialize
/rat/proc count
/rat/procset update 1
/rat/proc outroot
/rat/procset file "output.root"
```

Finally here are the commands used to create an inverse beta decay using the combo generator the vertex set command refers to the isotropic generation of neutrinos and takes spectrum information from a database file. The spectrum was set in the database file to be a flat spectrum with constant flux between a minimum and maximum specified energy range. The number of events to generate is also set here.

```
/generator/add combo ibd:fill:poisson
/generator/pos/set 0 0 0
/generator/vtx/set 0 0 0 IBD:testspect:nue
/rat/run/start 20000
```

The RAT macro can of course be extended to run full detector simulations. The inverse beta decay simulation here was the first part of a larger intended project to simulate geoneutrinos and pass them through a simulation of the detector to give an idea of the signal that could be expected at SNO+. This builds upon the work done for the energy spectra which were intended to be passed to a flux model calculation. This flux could then be set up in the database file that the macro calls for a full simulation of the geoneutrino signal.

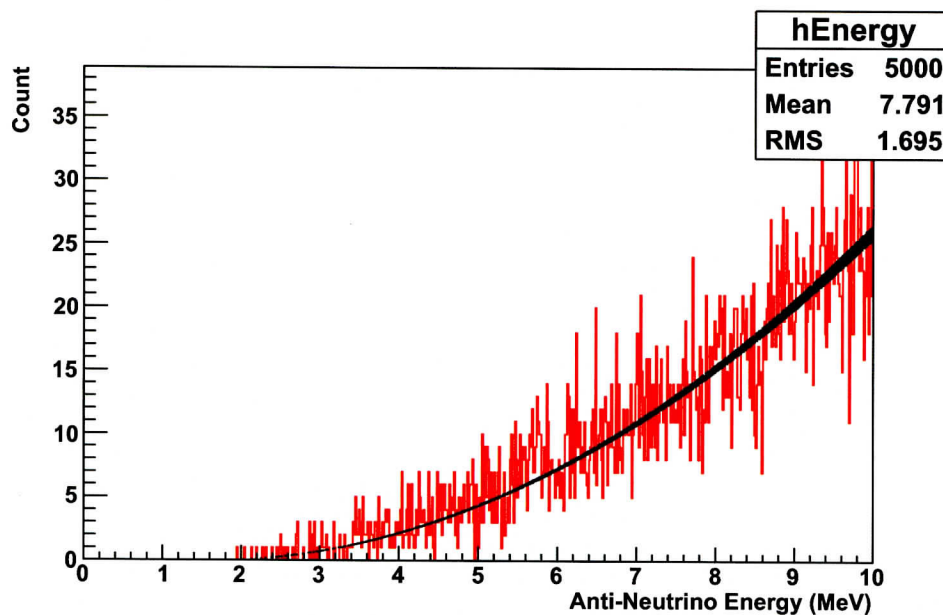


Figure 4: The simulation of expected geoneutrino IBD cross section from MC simulations using the RAT software is shown in red. The black line overlaid is the expected IBD cross section calculated from theory.

4.2 C++ To Analyse ROOT Output

The RAT simulation outputs to a standardised ROOT file which can then be read and analysed using the ROOT C++ framework. The data are stored in a tree-like data structure with independent branches that can be accessed depending on what information needs to be extracted. The main tree structure contains a data structure branch with the following sub-branches: MC, EV, Calibration, and Header Information. The MC (Monte Carlo) branch is what is required here to access the simulated data. EV represents a class of detected events as defined by the trigger simulation and not particle information. So, the ROOT file is loaded into the C++ program, the main Tree is read and the datastructure is accessed. Once this is accessed a simple command can be used to check the number of events generated in the run. This is used as an index inside a loop to process over all particles effectively. Now, a loop is run over every particle inside the DS and the MC branch for each particle is accessed. This allows an extraction of all the information about each particle including their energies, momentum, particle data group codes, etc. The way RAT was coded initially only stored the daughter particles of the decay in the IBD generator. A patch was coded so that parent information could be extracted which allowed the geoneutrino energies to be accessed. Inside the main loop a secondary loop is applied under the condition that the daughter particle accessed from the DS has a parent. If this condition is true a loop will run over the number of parent particles and store the energy inside a histogram. Another nested loop also keeps track of each parent particle's associated positron to store the kinetic energy and momentum used for further calculations. Finally, a standard ROOT histogram plotting command is applied to store the final output of geoneutrino energies. Using this Tree style data structure is very efficient on large data sets since only the required branches need to be loaded into memory and effectively partitions off unwanted information. For a five thousand event simulation reading through the data structure and putting the energies into a histogram only takes a number of seconds.

4.3 Simulation Results

The output of the inverse beta decay simulation is shown in Figure 4. A brief check by eye of the expected geoneutrino spectrum was carried out with an overlay of the calculation of the inverse beta decay with the following equation:

$$\sigma = \frac{2\pi^2 m_e^5}{f_{p,s}^R \tau_n} E_e p_e \quad (11)$$

This is an empirical formula with the Fermi-Coulomb corrected phase space factor $f_{p,s}^R = 1.7152$ and a precise measurement of the neutron lifetime from ultra cold neutrons $\tau_n = 885.7 \text{ sec}^{[15]}$. This form of the cross section provided an straightforward method of implementation due to the fact that the energy and momentum of the electron was readily available from the monte-carlo data structure. Thus, the calculation could be run inside an already existing loop multiplying a constant by the energy and the square root of the sum of the squared individual x, y, and z momentum components. Taking a look at Figure 4 it can be seen the overlaid plot of the cross section indeed takes the general shape of the modelled geoneutrino signal and therefore it can be confidently stated the simulation was indeed correct. A more thorough method of verification would be through an log-likelihood fit using the fit style options in ROOT but this problem was a little more tricky and requires a more competent programmer. Again, only 5000 interactions were simulated leading to quite a noisy histogram and so further simulations could also be run using the RAT macro if so desired but the tradeoff between usefulness and computer time did not seem worthwhile as each event takes around half a minute to simulate.

5 Conclusions

Programs have thus been coded to carry out the specified goals as outlined previously in Section 1.4. The generation of the geoneutrino spectrum for ^{238}U and ^{232}Th , was mostly successful except from the stated normalisation problem which unfortunately may leave the generated spectra unusable in flux calculations at the current time. The second half of the project had better results with a good foundation built for further work in simulation through RAT. Code has successfully been written to generate IBD events and access the ROOT data structures to plot a histogram of geoneutrino energies with the RAT source code updated to include the collection of parent particle information. Nevertheless, if the spectra can be correctly normalised this should allow a geoneutrino signal flux to be calculated at the SNO+ site which can then be passed to the simulator and give a prediction of the expected geoneutrino signal arriving at the detector.

References

- [1] S.T. Dye. Geo-neutrinos and silicate earth enrichment of U and Th. *Earth and Planetary Science Letters*, 297(1-2):1–9, August 2010.
- [2] Christoph Clauser. Radiogenic Heat Production of Rocks. In *Encyclopedia of Solid Earth Geophysics*. Springer, Dordrecht, 2nd edition, 2011.
- [3] G. Marx. Geophysics by neutrinos. *Czechoslovak Journal of Physics*, 19(12):1471–1479, December 1969.
- [4] Gernot Eder. Terrestrial Neutrinos. *Nuclear Physics*, 78(3):657–662, 1966.
- [5] BT Cleveland and Timothy Daily. Measurement of the solar electron neutrino flux with the Homestake chlorine detector. *The Astrophysical Journal*, 1998.

- [6] QR Ahmad, RC Allen, and TC Andersen. Direct Evidence For Neutrino Flavor Transformation From Neutral-Current Interactions in the Sudbury Neutrino Observatory. *Physical Review Letters*, 05:6–11, 2002.
- [7] M Boulay, M Chen, M Di Marco, et al. A Letter Expressing Interest in Staging an Experiment at SNOLAB Involving Filling SNO with Liquid Scintillator Plus Double Beta Decay Candidate Isotopes, 2004.
- [8] Chunlin Lan. SNO + and Geoneutrino Physics. Master's thesis, Queen's University, 2007.
- [9] J Boger, R.L Hahn, J.K Rowley, et al. The Sudbury Neutrino Observatory. *Nuclear Instruments and Methods in Physics Research A*, 449(1-2):172–207, July 2000.
- [10] Gianni Fiorentini, Marcello Lissia, and Fabio Mantovani. Geo-neutrinos and Earth's interior. *Physics Reports*, 2007.
- [11] Glendon Ralph Pugh. *An Analysis of the Lanczos Gamma Approximation*. PhD thesis, The University of British Columbia, 2004.
- [12] William H. Press. *Numerical Recipes in C*. Cambridge University Press, 2002.
- [13] Christoper W. Ueberhuber. *Numerical Computation: Methods, Software, and Analysis*, chapter 9. Springer, 1997.
- [14] S. Agostinelli, J. Allison, K. Amako, et al. Geant4: A simulation toolkit. *Nuclear Instruments and Methods in Physics Research A*, 506(3):250–303, July 2003.
- [15] Herwig Schopper. *Landolt Bornstein Series Volume 21: Elementary Particles*. Springer, 2008.