Faculty of Engineering and Physics Sciences

PHYM021 Physics Masters Dissertation 2018-2019

Geant4 Simulations of the Detection of Underground Water Sources

Leonardo Subramani Diana-Kuramapu

Supervisor: Dr Christopher Steer

Date of submission: 17th September 2019

Abstract

CRESTA a radiation transport code which uses Geant4 code libraries was used to run simulations investigating the use of cosmic ray neutrons above ground to detect underground sources of water. The research was divided into two parts. First a set of simulations were run to determine the attenuation length of cosmic ray neutrons in a SiO_2 soil of porosities between $0.4 - 0.7$, and saturation levels between 0% - 50% water by volume; allowing us to consider their limitations in the depths to which they could be used to detect underground water sources. Second, another set of simulations were run, simulating a roving neutron detector which stepped across the surface of soil containing a shallow a body of water at varying depths under its surface. The results showed for a small body of water of dimensions 20m x 5m x 0.5m, 1m below the surface of a dry 0.5 porosity SiO_2 soil that a significant (2σ confidence limit) depreciation in the flux of cosmic ray neutrons with an energy less than 0.1 MeV – 1 MeV is noticeable above the ground when using a roving neutron detector to count neutrons for periods of approximately 333s per detector position. This effect was shown to be very dependant on the depth to the body of water, as when the simulations were repeated using the body of water at 2m depth, there was almost no discernible drop in the number of neutrons over the water body. This is encouraging evidence that supports the ability of cosmic ray neutrons to be used in the detection of underground sources.

Contents

1 Introduction & Motivation **3**

2 Physical Background/Theory **4**
2.1 The Cosmic Ray Neutron Sensing Method 4
2.2 The Physics of Neutrons 5
2.2.1 Neutrons 5
2.2.2 Nuclear interactions and cross sections 6
2.2.3 The role of Hydrogen in CRNS 8
2.3 Cosmic Rays & The Generation Of Neutrons 9
2.4 Codes/Software used & The Monte Carlo Method 11
2.4.1 The Monte Carlo Method 11
2.4.2 Geant4 11
2.4.3 Cosmic Ray Shower Library (CRY) 12
2.4.4 CRESTA 12
2.4.5 ROOT 12
2.4.6 Python and shell scripting 12

3 Method **13**
3.1 Finding the attenuation length of neutrons in soil 13
3.2 Finding ground water using cosmic rays 17

4 Results and Discussion **19**
4.1 Attenuation Lengths 19
4.2 Finding water 23

5 Conclusion **29**

A Atomic composition code **34**

B Script used to automate process **36**

C Data from Simulations **40**

D Flux size data **42**

1 Introduction & Motivation

Along with access to food and shelter, having access to water is one of life's necessities. While in cities water is freely available through the engineering of modern day plumbing, granting us access to water in most places, this is not the case everywhere on Earth. Natural sources of water such as rivers, lakes and springs can only provide adequate access to so many and their utility decreases the further you are situated from them. Wells take advantage of the fact that vast quantities of water are stored underground in forms such as underground rivers, underground lakes, aquifers and other saturated zones. When deciding where to dig a well, there are many unknown factors. Much research must be done about the local hydrology, which typically involves looking to official government resources for information on things such as the depth to the local water table, and the presence of aquifers. Complications may arise if the information available is incomplete, incorrect, or lacks precision. Trial and error can cost a prohibitive amount time and money, especially in impoverished or remote areas. The ability to determine the location or depth of underground bodies of water before drilling, could be game changing for well-construction projects world wide, changing the lives of millions of people for the better.

The CRNS (Cosmic Ray Neutron Sensing) method, is a method that utilises the constant ambient flux of neutrons generated by cosmic rays near and at ground level to infer how saturated soil is with water[1]. CRNS has uses in a wide variety of sciences and may be a crucial tool in being able to locate underground sources of water.

The ability to detect underground sources of water, may not only be useful for niche industries such as well-drilling. Evidence has been gathered that water exists on other bodies in our solar system, such as Mars where high energy cosmic ray neutrons have already been used to provide evidence of water[2]. The process of searching for water and life on Mars is hindered by the fact that we do not want to introduce any earthly microbial life to potential ecosystems for the fear that we introduce a source of false positives in our search for extraterrestrial life, or disrupt any extant ecosystems[3]. The CRNS method could be instrumental in the search for water or life on Mars, as it provides us with a method of non intrusively finding bodies of water.

Lastly the CRNS as a field of research is still quite young, arguably having only been around since 2007/2008[4]. Its limitations are not yet completely understood and this project opens up the door to a unique, new direction that CRNS can be taken.

2 Physical Background/Theory

2.1 The Cosmic Ray Neutron Sensing Method

CRNS has many applications over a wide range of sciences including but not limited to hydrogeophysical, meteorological, astrobiological/planetary and agricultural sciences[5][2]. In 1966, L. D. Hendrick and R. D. Edge found that the flux of neutrons near the surface of the earth was strongly related to the soil water-content[6]. About 15 years later research by Kodama found that the water equivalent of snow could be monitored using a neutron detector placed under snow packs[7] and later on, a further study, by Kodama et al. discussed the potential for using neutrons generated near the ground by cosmic rays to monitor the moisture content of soil[8]. After NASA's Mars Odyssey Orbiter was able to find evidence for water in the southern highlands and the northern lowlands of Mars using high energy cosmic ray neutrons[2], it was suggested that albedo cosmic ray neutrons could be used to determine water saturation levels in soil[9]. Finally, in 2007, there was a joint presentation by Zreda and Desilets where first measurements and simulations were shown at two international conferences and water content was related to neutron observations above the ground[4].

The CRNS method uses a neutron detector placed above the ground to measure the ambient flux of neutrons generated by cosmic rays in and near the surface. In doing this it is possible to infer how saturated with water the soil is in the surrounding area. As will be discussed in §2.2.3, CRNS hinges upon the ability of hydrogen to efficiently moderate[I] the neutrons generated by cosmic rays that are incident upon the Earth's atmosphere and surface. Water, due to the fact that it contains 2 hydrogen atoms, is a very efficient moderator of neutrons. Neutrons generated near the surface of the earth by cosmic rays travel in all directions. Those travelling along a path that would take them into the ground, or that are generated in the ground themselves are moderated significantly more by the soil when ground water is present, even in small percentages[10] and are less likely to leave the ground to contribute back to the rest of the ambient flux of neutrons.

[I] To moderate a particle is to reduce its energy by causing it to pass through some material, called a moderator[11]. In this case, the particles in question are the neutrons generated by cosmic rays and the material is soil/water.

This technique has some attractive advantages over other conventional soil moisture measuring methods.

1. Detected neutrons can come from as deep down as 76cm in dry soils, much deeper than other non invasive methods such as satellite based methods[10]

2. Neutrons can diffuse very quickly in air and can travel across 100s of meters, allowing this technique to have a very large footprint at the hectometre scale. This is a nice intermediate between point measurements and satellite ones and is useful for measuring areas comparable with crop fields[10][5]

3. It is non invasive and non destructive[10]

4. it's reasonably sensitive to water levels and insensitive to variations in soil chemistry[10]

This method of research is still early in its development and there is still much to learn about it, but that is not the focus of this report.

2.2 The Physics of Neutrons

2.2.1 Neutrons

The neutron is a hadron of a neutral charge and a mass of approximately 940 MeV/c^2, almost identical to that of a proton which has a mass of approximately 938 MeV/c^2. Both neutrons and protons, together referred to as nucleons, are the two main particles that make up atomic nuclei. While free from a nucleus, the proton is theoretically stable against radioactive decay, with a speculative lifetime of $\tau > 10^{32}$ years[12]. Isolated neutrons have a life time of $\tau \simeq 15$ minutes[13] before decaying via β^--emission, into a proton, an electron and an anti-neutrino:

$$n \longrightarrow p + e^- + \bar{\nu_e} \tag{1}$$

Neutrons are often grouped into energy bands depending on their kinetic energy:

Energy Band	**Energy**
Thermal	$E \simeq 25$ meV
Epithermal	$E \simeq 1$ eV
Slow	$E \simeq$ keV
Fast	$E \simeq 100$ keV $\rightarrow 10$ MeV
High energy	$E > 10$ MeV

Table 1: A table listing the names and energies of the neutron energy bands.

2.2.2 Nuclear interactions and cross sections

Neutrons interact differently with matter depending on their energy. There there are two main types of interaction to consider: scattering and absorption.

Scattering is the process by which the neutron scatters off another particle, changing its momentum. There are two types of scattering: elastic and inelastic scattering. Elastic scattering is a special case of scattering where the kinetic energy of the system is conserved and inelastic scattering is when energy from the interaction is lost to processes such as exciting one of the interacting bodies.

Absorption is a process where the incoming neutron is absorbed by a nucleus. There are several main channels that a nucleus can take after an absorption event with a neutron:

1. The nucleus is stable against radioactive decay, changing the binding energy of the nucleus. The mass number of the nucleus increases by 1.
2. The nucleus is unstable against radioactive decay and subsequently decays.
3. The nucleus is radioactively stable but is excited and emits gamma rays to de-excite.

In order to better understand nuclear interactions and by extension neutron interaction, it helps to define the term cross section σ; which can be thought as a measure of the probability for a reaction to occur. Consider a particle in a particle beam, moving towards a target particle. The cross section can be interpreted as the area, measured in barns b ($1\text{b} = 10^{-28}\text{m}^2$), that the target particle takes up as "seen" by the in the beam[12]. In practice we find that σ values can vary wildly with energy and particle. For instance, the relationship between the absorption cross section of a neutron by a nucleus and its kinetic energy is generally given by $\sigma_{abs} \propto E_n^{-0.5}$. As a result, this type of interaction tends to dominate at lower energies and is relatively unimportant at higher energies, where scattering processes tend to dominate.

Cross sections for individual interactions are called microscopic cross sections; the microscopic cross sections for absorption and scattering are given the symbols σ_{abs} and σ_s respectively. During any interaction these microscopic cross sections "compete", and the eventual outcome of the interaction, if any, is statistical/non-deterministic[II]. It may be that cross sections are similar in size and therefore similar in probability of outcome, or it may be that one completely dominates; Fig. 1 shows a plot of cross section vs incident energy for two different interactions of neutrons with a ^1_1H nucleus, you can see that at some points they differ by a factor of more than 10^5. Microscopic cross sections are summed together linearly to form total cross sections, which is generally given the symbol σ.

[II]It is the statistical nature of nuclear interactions, that is why Monte Carlo methods so well suited

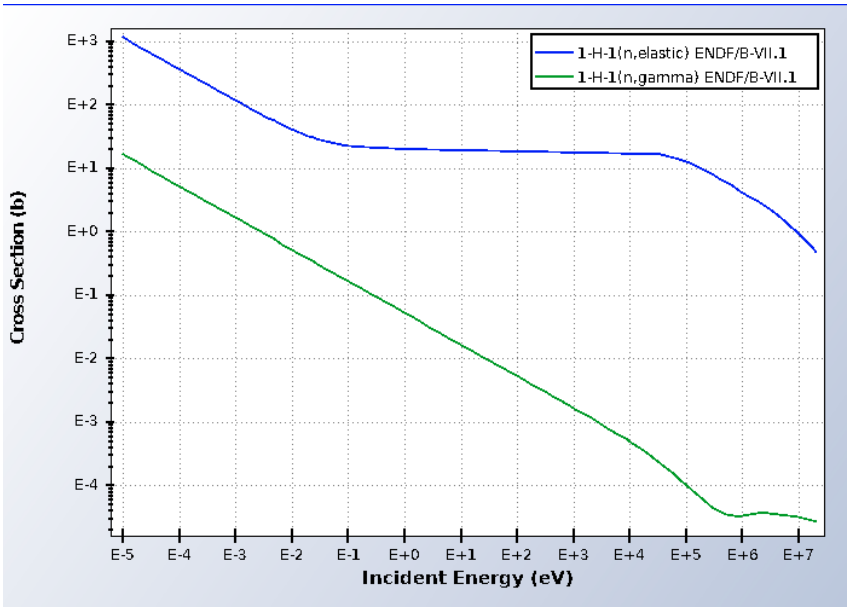

Fig. 1: A figure showing the σ_{abs} and σ_s for neutrons interacting with Hydrogen. Note how the cross sections vary by factors of 10^5 in some energy regimes. Taken from the National Nuclear Data Centre's website `https://www.nndc.bnl.gov/`[14]

The intensity of a narrow mono-energetic beam of neutrons that has travelled some distance x, through a material is given by[12]:

$$I(x) = I_0 e^{-x\Sigma} = I_0 e^{-n\sigma x} \tag{2}$$

Where Σ is the macroscopic cross section, n is the number density of the material and σ is the total cross section. Σ has units of m^{-1}, and describes the probability of interaction for a neutron travelling through a material, per unit length. This helps us define the attenuation length, λ, of the beam in the material which is the distance the beam travels through the material before it reaches e^{-1} of its original intensity.

$$\lambda = \Sigma^{-1} \tag{3}$$

While cosmic ray neutrons are far from similar to a mono-energetic narrow beam, the idea of attenuation length can be applied to them in a similar way when considering that their intensity with distance penetrated into the earth's surface. The number able to penetrate down to a certain depth will decrease as shown by the results in §3.2, in an exponential manner to which you can apply the idea of an attenuation length.

for use in radiation transport codes, as will be slightly further explained in §2.4.1.

2.2.3 The role of Hydrogen in CRNS

The purpose of CRNS is to be able to measure the amount of soil moisture using cosmic ray neutrons. This method is possible due to the remarkable neutron scattering properties of hydrogen. Hydrogen ($^1_1$H) is an incredibly efficient nucleus for moderating neutrons, due to the similarity of their masses. Energy losses to the neutrons during elastic collisions are roughly correlated with the mass of the nuclei they collide with. The ratio between a neutron's energy after an elastic collusion, E', to before E an elastic collision is given by[12]:

$$\frac{E'}{E} = \frac{A^2 + 1 + 2A\cos(\theta)}{(A+1)^2} \tag{4}$$

Where θ is the scattering angle and A is the mass number of the nuclei the neutron has collided with. We can see that for a head on collision ($\theta = 180$) we get maximum energy transferred and that for no scattering ($\theta = 0$) that we get no energy transferred. As $A = 1$ for Hydrogen, we can see that for a head on collision, the entire kinetic energy of the neutron is transferred. The energies of neutrons generated by cosmic rays in the soil and near the ground tend to be around 1 MeV (further explanation in §2.3), this is important as the elastic scattering cross section of hydrogen, shown in Fig. 2, starts to dominate around 1 MeV. This shows that cosmic ray neutrons generated near or in the ground are the most likely to collide with hydrogen if it is present, which coincidentally is also the most efficient nucleus for moderating them. Hence hydrogen's great moderation of neutrons.

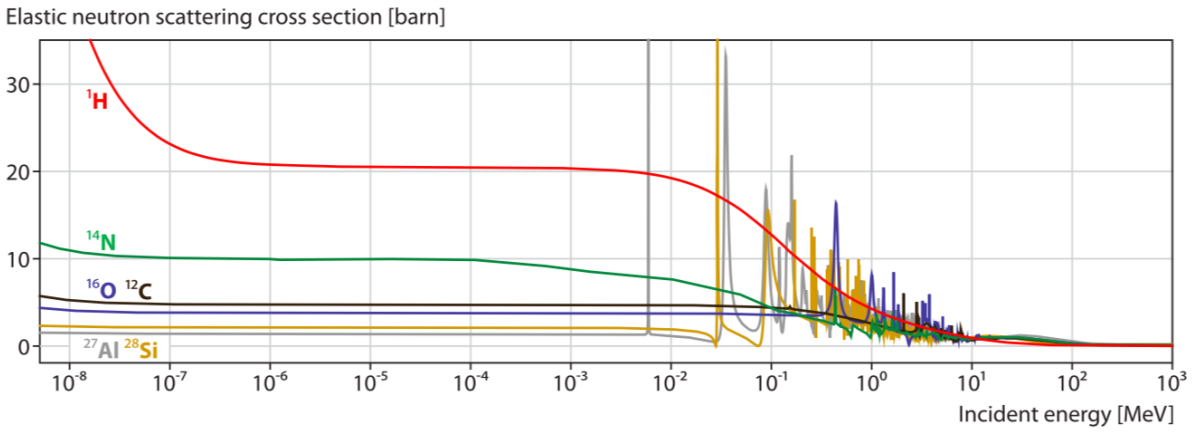

Fig. 2: Data taken from JENDL/HE-2007 (Shibata et al. 2011)[15], showing the elastic neutron scattering cross sections of various elements. Credit for the figure goes to Köhli et al. (2015)[5].

2.3 Cosmic Rays & The Generation Of Neutrons

Cosmic rays are energetic particles, that are constantly bombarding the earth at a rate of around 1000 $m^{-2}s^{-1}$[16]. Taking the form of charged particles, the majority being comprised of protons, with a smaller quantity being electrons and other charged nuclei[16], they typically originate from within the galactic disk, although there is a proportion of them which are solar as well as extra-galactic in origin[17][18]. It is widely agreed that supernovae remnants are the source of the acceleration of galactic cosmic rays, with evidence being found to support this[17]. Their diffusion through space is thought to be isotropic[19] and they are omnipresent at the surface of the earth, with some variation depending on latitude, the sun's solar cycle, and the Earth's and the Sun's magnetic fields[4].

Cosmic rays crashing into the upper atmosphere are classified as Primary Cosmic Rays (PCRs), which typically have energies in the order of around 1 GeV[16]. Upon entering the Earth's atmosphere, the PCRs interact with atmospheric nuclei producing a multitude of hadrons, collectively referred to as Secondary Cosmic Rays (SCRs). SCRs can either go on to generate further showers of particles, or decay into photons[4]. High energy neutrons are one type of SCR, created through spallation events with incoming proton PCRs and atmospheric nuclei. Spallation is a nuclear reaction in which a target nucleus is bombarded by a particle, and broken up in the process[11]. A sketch depicting spallation is shown in Fig. 3. These high energy neutrons generated in such events have a range of energies centred around an energy of ~ 0.1 GeV[20][21].

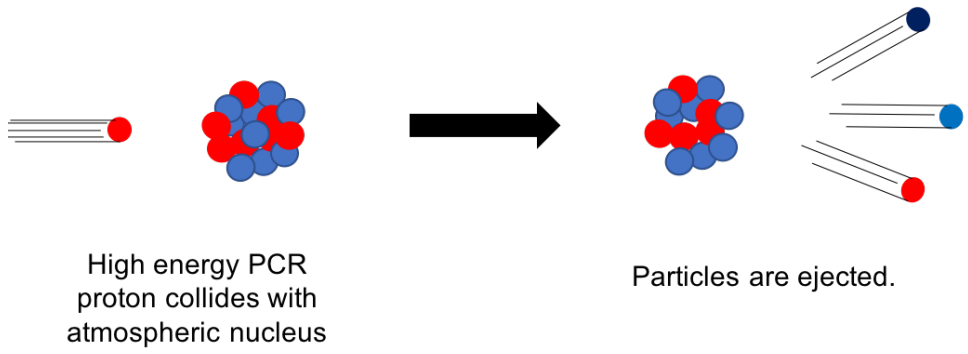

Fig. 3: A sketch depicting a spallation event where a PCR proton collides with an atmospheric nucleus, breaking it up and scattering particles.

The way in which these high energy neutrons interact with further nuclei after being generated changes as they progress along their journey to the ground, as said before, interaction cross sections can be very energy dependent. Most important for the CRNS method is a process called evaporation where the high energy neutrons go on to generate neutrons of a lower energy by becoming captured by an atmospheric nuclei, and causing

it to emit neutrons in order to reach a more stable energy configuration. This process is depicted in Fig. 4. Evaporation neutrons have a spectrum of energies, centred around 1 MeV[20][21], which come under the fast neutron energy band.

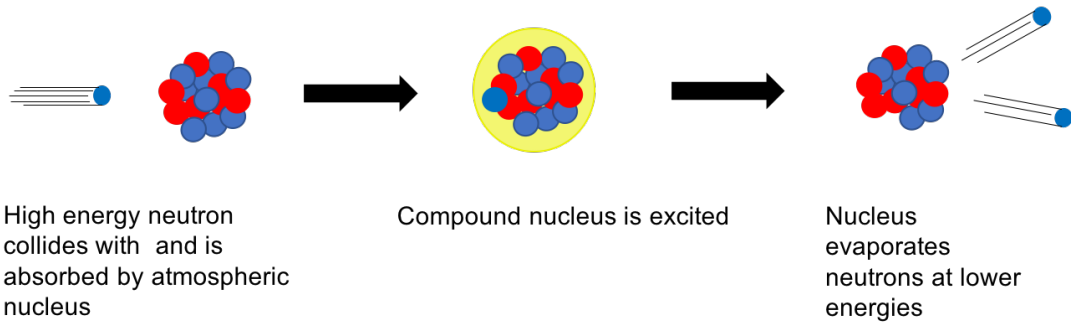

Fig. 4: A sketch depicting an evaporation event. Reading the sketch from left to right: a high energy neutron generated by a spallation event between a PCR proton and an atmospheric nucleus is captured by a further nucleus. This nucleus becomes excited and emits some neutrons in order to reach a more stable energy configuration.

Interactions in the sub MeV regime are dominated by elastic collisions and their flux near the ground is highly dependent on the soil moisture content. The reason for this, as described in §2.2.3, is that fast neutrons are extremely well moderated by hydrogen and by extension, water. The more water in the soil, the fewer fast neutrons that are generated in or scattered into the soil, can make it back out to contribute back to the ambient flux.

A useful plot for visualising the complete spectrum of cosmic ray neutrons as seen at the Earth's surface is shown in Fig. 5. Neutron spectra from several sources are plotted on the same axes. High energy neutrons created through spallation are shown in red; neutrons created by evaporation are shown in green; evaporation neutrons which have reached an energy sensitive to elastic scattering are shown in blue, and completely thermalised neutrons are shown in grey.

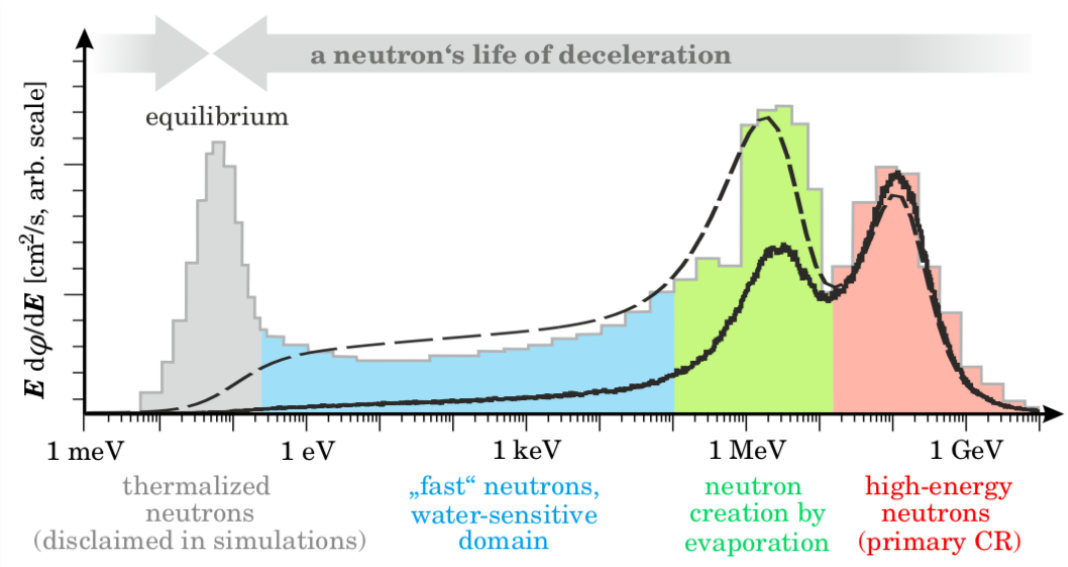

Fig. 5: A graph displaying neutron energy spectra at the Earth's surface found by several studies. This graph contains data from several sources: measurements by Goldhagen et al. (2002)[20] (grey line); Sato and Niita (2006)[21] (dashed line) and the isolated incident component by Köhli et al. (2015)[5] (black line). Credit for the figure goes to Köhli et al. (2015).

2.4 Codes/Software used & The Monte Carlo Method

2.4.1 The Monte Carlo Method

The Monte Carlo method is a class of computational algorithm used in a wide array of disciplines. The name allegedly refers to the city of Monte Carlo in Monaco, famous for its gambling. The method uses huge quantities of random numbers to evaluate possible outcomes in situations where the final events are either entirely, or partly undetermined. This method is very well suited to nuclear physics where outcomes of events tend to be non-deterministic and large numbers of results are sometimes needed to find a result.

2.4.2 Geant4

Geant4 is a radiation transport code written in C++, developed by the Geant4 collaboration, which utilises Monte Carlo methods to allow users to simulate the effects of particles and radiation interacting with matter[22]. Thanks to its incredible versatility, Geant4 is used in a wide variety of areas such as medical science, nuclear science and space science[23]. Geant4 has been used by a wide community of scientists for years now and is quite well validated, as can be seen on their website[24].

2.4.3 Cosmic Ray Shower Library (CRY)

This is small code library used to generate the cosmic ray particle showers at a range of energies and elevations above sea level. Its main purpose is for use along side radiation transport codes. This has also been validated[25].

2.4.4 CRESTA

If Geant4 and CRY were instruments on a scientific probe, then CRESTA (Cosmic Ray Environmental Science & Technology Application) would be the probe itself. Written in C++, CRESTA uses Geant4 in conjunction with CRY to simulate cosmic rays and how they interact with matter.

The term to describe the spatial relationship between the bodies being simulated and the nature of the incident radiation upon them is the geometry. The geometry of a simulation is described in a file, called a geometry file. The output of CRESTA is in the form of a ROOT file (more on this in §2.4.5). CRESTA simulations are initiated via a command entered into the command line making them a good candidate for automation using bash shell scripts. For example:

```
$ cosmicraysim -g geometry_file.geo -n 100 -o outputfilename
```

This would run a simulation, using the geometry file `geometry_file.geo`, run the simulation for 100 particles, and output the results to a default ROOT file named `outputfilename.0.0.root`.

2.4.5 ROOT

ROOT is a data processing framework created by CERN[26]. CRESTA automatically outputs its data in the form of a ROOT file. In this project ROOT was used to process the output data from CRESTA and was also used to fit lines of best fit to the data in order to determine mean free paths/attenuation lengths.

2.4.6 Python and shell scripting

Python scripts were used on their own and in tandem with bash shell scripts several times in this project. Scripts written in python were used to either automate tasks, or for data analysis and graphing.

A large part of the project was simulation based. As shall be explained further in §3.1, this took large amounts of time due to the sheer number of individual simulations that needed to be run and required a lot of human interaction. Using a combination of bash scripts to initiate simulations and change permission files, and python to edit parameters in both the bash scripts and the geometry file; the simulation process became

effectively fully automated, saving vast amounts of time, and human errors. The first python script written to automate a simulation process is shown inAppendix B; there were others that were written for other simulations, but they have not been included as they function fairly similarly. If the reader is particularly interested though, all scripts used, data relevant to this project can be found online at `https://github.com/leonardodk/DissertationCode/`.

Another script written was used to find the elemental mass fractions and average density of an SiO_2 soil. It allowed the user to enter a desired porosity and saturation level and it came in extremely useful when completing parts of the method. This script is shown in Appendix A.

3 Method

The method for this project was divided in two parts. The first part was to use CRESTA/ Geant4 (hereon just referred to as CRESTA) to investigate the attenuation lengths of cosmic ray neutrons in soils of various porosities and water saturation levels. This verified that cosmic ray neutrons could penetrate down to the depths required to detect water sources. The second part used CRESTA to simulate a scenario where an underground source of water could be detected, by moving a neutron detector across the surface of a plane of soil containing a body of water under the surface, and monitoring the change in the flux of neutrons as it went.

3.1 Finding the attenuation length of neutrons in soil

To find out whether cosmic ray neutrons can penetrate down through soil to a hypothetical underground water source, the relationship of the effect of soil porosity[III] and saturation on the attenuation length of cosmic ray neutrons needed to be understood.

While the ability to create volumes and bodies in codes like CRESTA is fairly versatile, it may not be easy, or even sensible to try and exactly replicate the dimensions of the scenario being tested. In general, it is acceptable to use approximate simple 3D volumes like cuboids and spheres, to imitate what you are trying to simulate. The entire simulation takes place inside the "world" - which is the name of the simulated euclidean space within which we describe our simulation geometry. Using a Cartesian coordinate system, the world is set to be the shape of a cuboid with sides of length 20m x 20m x 15m in the x, y and z axes respectively, centred around the origin as shown in Fig. 6.

[III] porosity is a measure of how much of the soil is free space, available to be filled by air, water or some other fluid. Porosity is done by fraction, ie. a porosity of 0.4 corresponds to 40% free space by volume.

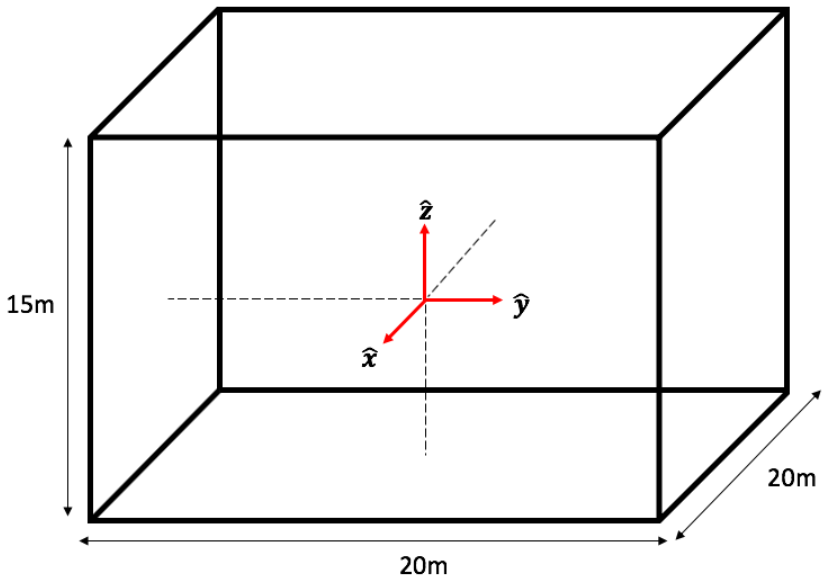

Fig. 6: Sketch depicting the cuboid dimensions of the world volume used in the simulations for §3.1 and §3.2. The world volume goes from -10m to +10m in both the x and y axes, and -7.5m to +7.5m in the z axis.

To find the attenuation length of the incident flux of cosmic ray neutrons in soil, a layer of soil was simulated above a layer of water, imitating a water table or water supported by an aquifer. An incident flux of cosmic ray neutrons was then set incident upon them, and the number of neutrons which were able to penetrate through the soil into the water were counted. The next step was to repeat the process, incrementally increasing the thickness of the soil, and counting the number of neutrons able make it through each thickness. Making a plot of the number of neutrons able to penetrate soil of a certain thickness, by symmetry allows us to plot the relationship between the intensity of cosmic ray neutrons flux and depth into the Earth. Assuming that the intensity of the cosmic ray neutron flux in soil follows a similar decay-like trend to that described by (2), we can use ROOT's in-built line fitting software to fit some function of the form $f(x) = a \cdot e^{-x/b}$ to our plots, where a and b are constants to be fitted. Inspection of (2) and (3), shows that b directly gives us the attenuation length. The parameter a is not of importance to this study but has been included for completeness, in the complete table of values for attenuation lengths shown in Appendix C.

Both the soil and the water were modelled as cuboids spanning the entire $x-y$ plane. The thickness of the water was kept constant at 4.5m, at this thickness, it is enough to capture 99.5% of incoming cosmic ray neutrons even if the thickness of the soil is set to 0. The depth of the soil was varied throughout the simulations, from 10cm, to 600cm. The flux of neutrons, while generally travelling in the -$\hat{z}$ direction, had a distribution of velocities and energies, as was dictated by the CRY code library (§2.4.3) and came from

a planar source 1mm thick which spanned the $x - y$ plane and was placed at the top of the world volume. A visualisation of the geometry is shown in Fig. 7.

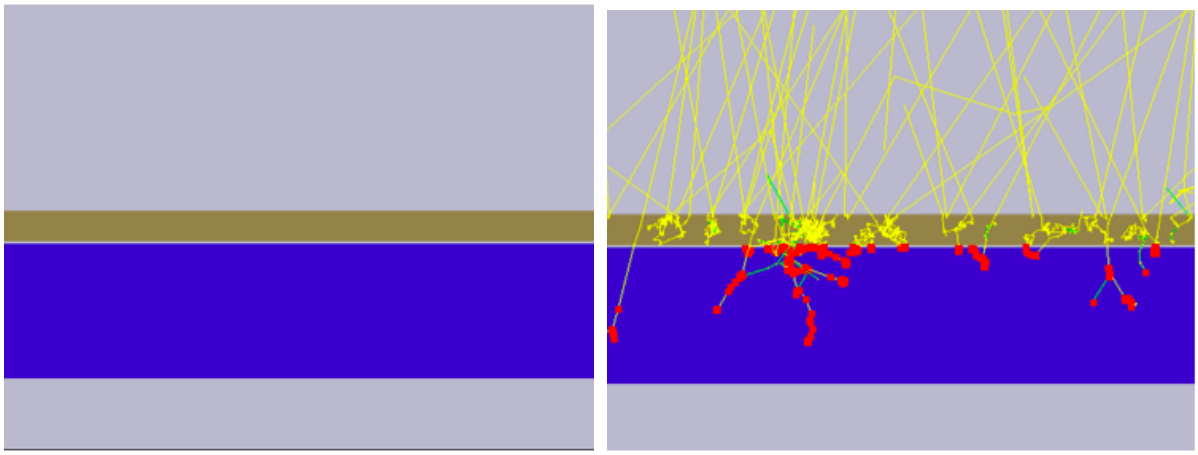

Fig. 7a: Fig. 7b:

Fig. 7: Screen shots of the visualisation of one of the geometries used to determine the attenuation lengths of cosmic ray neutrons in water. Water is shown in blue, soil in dark yellow and air (dry) in grey. Fig. 7b, shows the geometry of the simulation with paths of 30 neutrons that have been generated also shown. The plane the geometries are beign viewed from is the $x - z$ plane.

Elementally, all soils were comprised entirely of SiO_2, this was deemed an acceptable approximation as oxygen and silicon are the first and second most abundant elements in the Earth's crust found primarily in the form of SiO_2[27]. This soil was initially simulated at three levels of porosity: 0.5, 0.6 and 0.75, each porosity was also simulated at a range of saturations from 0% to 50% saturation by volume. Increasing the thickness of the soil gradually in increments: starting at 10cm, 25cm, 50cm, 60cm & 75cm before increasing in steps of 25cm until a final thickness of 600cm. The irregularity in increment at small thicknesses was due to a check implemented in the designing stages of the geometry, which ended up remaining. The total number of simulations was initially 702 simulations but later a soil porosity of 0.4 was added, making the final total number of simulations needed to be run was 936. The elemental abundance and density specifications required by CRESTA for the 0.4 porosity soil were calculated using the python script found in Appendix A.

Each simulation was run for $3{\times}10^5$ particles and took on average 4.3 minutes to complete. Initially, each simulation was set off "by hand". As shown in §2.4.4, the relevant command to start CRESTA would be entered into the computer terminal and allowed to complete. Following this, the geometry file would be opened to update the parameters of the simulation. The process would then be started again, but with the terminal input changed, so that CRESTA would output its results to a new ROOT file.

It quickly became apparent that if all the necessary simulations were to be done on time, all human interaction would need to be circumvented. To achieve this, a combination of Python and Bash scripts were written, which could run all the required simulations back to back, cycling through each soil porosity and saturation level, and output the data from each run into its own ROOT file.

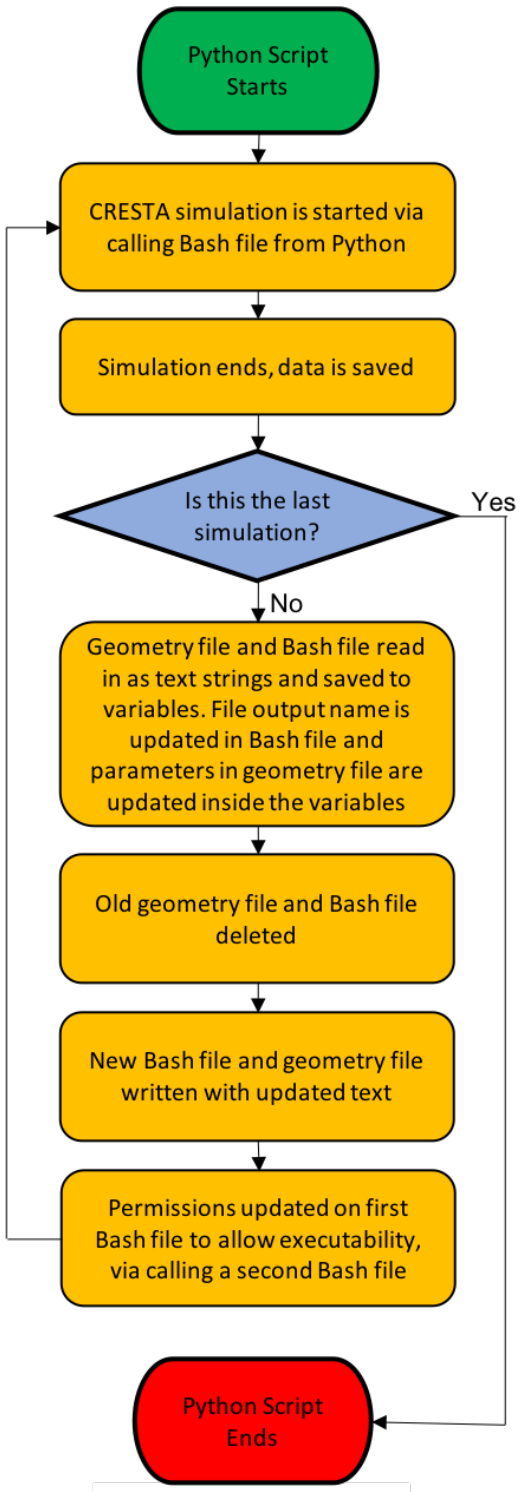

Fig. 8: A flow diagram depicting how a combination of Python and two Bash scripts, allowed almost full automation of the simulation process

The terminal command to start CRESTA was written to a bash file that would then be called by the Python script to start the first simulation. After finishing, Python would then read both the Bash script and the geometry file in as plain text, edit the necessary parts, before overwriting them and saving new versions. Python would then call the new Bash script and the process would repeat until all soil types had been done. A much simplified flow chart, explaining how the script works is also shown in Fig. 8, the script itself is shown in full in Appendix B.

This combination of scripts allowed for work to be done on other tasks while simulations were running and also allowed for simulations to be run when it was not possible to be at the desk, like overnight. Lastly but not least importantly, as it did all the file naming and saving automatically, it reduced the potential for introduction of human error, and made managing the results from the simulation much easier.

3.2 Finding ground water using cosmic rays

The idea behind using cosmic ray neutrons to search for ground water is that the flux of neutrons above ground, over the source of water, will be noticeably lower, allowing it to be detected. The neutrons that were "missing" were neutrons that had been able to penetrate the soil some depth and which would have scattered back up to the surface to be released into the atmosphere, had the source of water not been there to stop them. To test this hypothesis using CRESTA, a suitable simulation needed to be set up.

If one wants to properly answer the question of whether cosmic ray neutrons can indeed be used to search for underground bodies of water, a large number of variables need to be considered. A summary the variables considered in this project and a short summary of how they may affect the data is shown in Table 2.

Table 2

Variable	**Effect of variable**
Shape of water source	Bodies of underground water do not conform to neat geometric volumes, nor do they often have hard, self containing boundaries. The type of water body being modelled needs to be well approximated by a volume in CRESTA.
Soil porosity	An increase in soil porosity means that more of the soil is empty space. This alone allows neutrons to penetrate further into the soil, and also allows for more saturation with water
Soil saturation	An increase in saturation leads to a reduction in the attenuation length of the neutrons in the soil
Angle/pitch of the surface	Changes the amount of material in between the detector and the source of water. This will change the energy of the neutrons being counted/not counted, and possibly the number of neutrons counted
Soil elemental composition	Different elements and densities will affect the attenuation length of neutrons
Shape of the underground water	Will change the spatial position the neutron detector detects a change in the flux of the neutrons
Detector type	Different detectors pick up different energies of neutrons, changing the detector type could have significant effect on the end results

Continued on next page

Table 2 – *Continued from previous page*

Variable	**Effect of variable**
Path taken by the detector	Will change the spatial position the neutron detector detects a change in the flux of the neutrons

Table 2: A table of variables to be considered in §3.2

Writing a geometry file for an underground water sources is difficult as they can take many forms. The water table, the region at which soil becomes completely saturated with water, is not really a self contained body with hard boundaries but more of a region which gradually becomes increasingly saturated until it is fully saturated. It also will undulate with the surrounding environment, something which might be tricky to emulate in CRESTA. Aquifers, bodies of porous rock which contain water either in their pores or in between their cracks and fissures[28], face similar problems in that there is perhaps no single geometric representation for them which is "the most" appropriate. Underground rivers and lakes, still, have the same problems.

The final model most appropriate and simplest to simulate was to model a small, isolated underground lake, covered by dry 0.5 porosity SiO_2 soil. For this scenario the dimensions can be approximated adequately using the basics shapes available for the CRESTA geometry. A neutron detector could then be stepped over the surface, counting neutrons at each position for some period of time. Hypothetically, once at a point on the surface where the detector is also near or above the body of water, we will see a drop in the count of neutrons.

The geometry was set up as follows: the dimension of the world volume were a cuboid of dimensions 40m x 40m x 15m in the x, y and z planes respectively, centred around the origin. This is similar to the world as shown in Fig. 6, but with a larger x-y plane to allow for the roving detector to move across. The soil was modelled as a cuboid, spanning the entire $x - y$ plane of the world volume, and had dimensions 40m x 40m x 5.5m. This cuboid was positioned so that it shared its bottom surface, with the bottom of the world volume. The body of water was set to be a cuboid of dimensions of 20m x 5m x 0.5m, centred about the origin in the $x - y$ plane, positioned such that it was 1m deep below the surface of the soil. Finally, in place of a detector, a $1m^3$ cube of benzene was simulated that moved along the surface of the soil. Benzene was chosen as the neutron detector material as chemically it is similar in structure and density to other liquid scintillators used in neutron detection. While the active volumes of portable neutron detectors are typically much smaller than this, a volume this size ensures more neutrons will be counted, helping to improve the final data. A visualisation of this set up is shown in Figs. 9a and 9b. The flux of neutrons came from a source of of dimensions

30m x 30m x 0.1mm cuboid placed at the top of the world volume. The energies and velocities of the neutrons were all controlled by the CRY code library. The reason the source did not cover the entire $x - y$ plane of the world is because at this size, more neutrons would hopefully contribute to the area of interest which is where the roving detector crosses the underground lake, at the origin in the $x - y$ plane. This should reduce the number of neutrons needed to be simulated and in doing so, computational time. One effect from this may be that we see fewer counts of neutrons towards the edges.

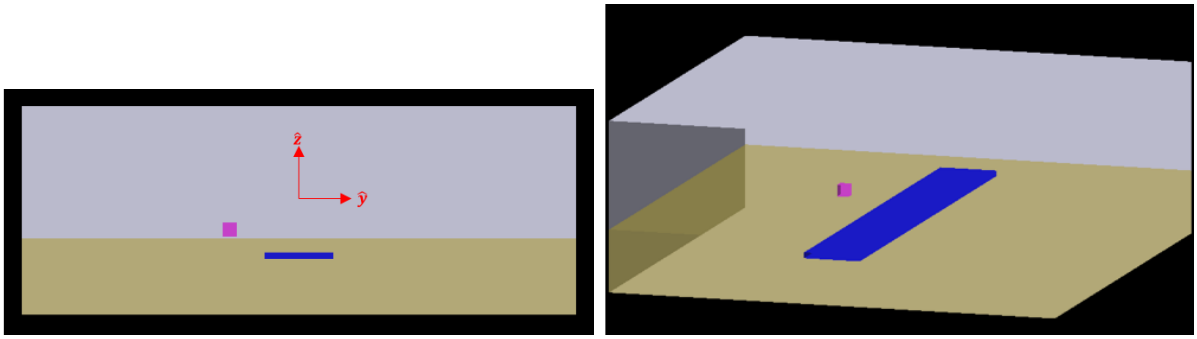

Fig. 9a: Fig. 9b:

Fig. 9: Screen shots of the visualisation of the geometry as described in §3.2. Soil is in yellow, water body in blue, and the detector in pink. in Fig. 9a, axis are shown in red, approximately where the origin is, to show where the directions are.

Using Cartesian coordinates, the detector was then stepped over the surface of the soil, along the y-axis from (0, -10m, -3m) to (0m, +10m, -3m) and CRESTA was ran for $3{\times}10^6$ neutrons at each detector position. In doing this we hoped to see a significant drop in the number of neutrons at points which were the detector was over the water table. A set simulations were also run, where the body of water was not present as a control. Plots were made of counts of neutrons counts vs position of the detector's position along the y-axis.

4 Results and Discussion

All data output by CRESTA was in the form of ROOT files; after retrieving the relevant data from these files using ROOT, the data was tabulated in spreadsheets and visualised using Python and its code library Matplotlib.

4.1 Attenuation Lengths

The results for the investigation into the effects of soil saturation and porosity on the attenuation length of cosmic ray neutrons is shown in Fig. 10. Each point was derived by fitting a line of the form $f(x) = a \cdot e^{-x/b}$ to a plot of neutrons counted passing through a

soil of varying thickness, and recording the parameter b which is equal to the attenuation length. An example of one such plot, with the line of best fit determined by ROOT is shown in Fig. 11. Table 3 in Appendix C shows all attenuation lengths determined in this manner but was not included in this section as the focus of this report is on the phenomenological effects observed.

The results suggest that cosmic ray neutrons in a SiO_2 soil may have a wide range of attenuation lengths, depending on the soil porosity and saturation with water. In general, the attenuation length increases with soil porosity and decreases with water saturation which is a logical and agreeable finding based on the physics discussed in §2.2.1 and §2.2.3. If the porosity increases or the saturation decreases, it means there is less matter along the path of the neutrons to be scattered off or be absorbed by, allowing it to further penetrate the soil. This data is encouraging as it shows that given the right circumstances, that the attenuation length of cosmic ray neutrons is not a prohibitively short distance and may be long enough for use in the detection of underground sources of water.

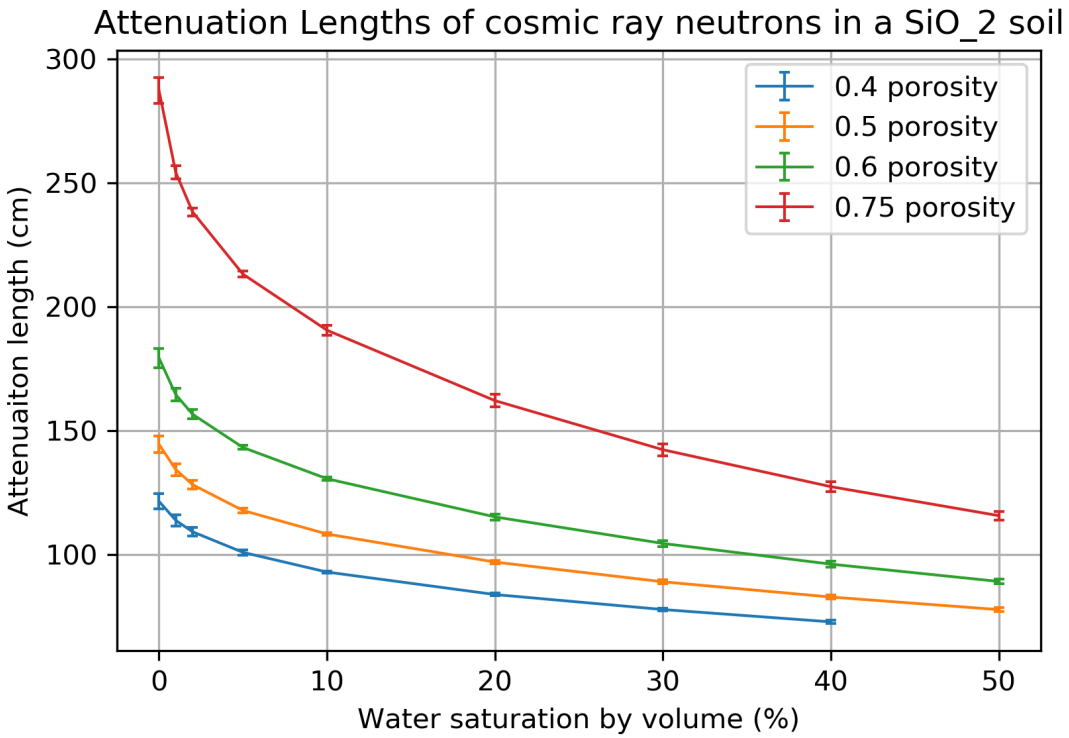

Fig. 10: A graph showing effect of soil porosity and water saturation on cosmic ray neutron attenuation lengths in a pure SiO_2 soil. Porosities ranged from 0.4 to 0.75, and water saturation levels from 0% to 50% water by volume where possible. Uncertainties were determined by ROOT, and are presented in Table 3.

Several steps were taken to ensure that the data was optimal and correct. While designing the simulations, a small investigation into the effect of source size was done. This was done to address the issue that any neutrons travelling along a path which would take them outside the world volume would cease to be counted. The concern was that if the size of the source was too large, a significant number of neutrons generated close to, or scattered near the edge of the world volume, might be lost. While insignificant in small enough quantities, too many neutrons being lost could change the results dramatically. Running a small set of simulations, keeping the $x-y$ shape of the source square, the side length of the source was varied between 2m and 20m; one simulation of 10^6 neutrons was done for each source size. The neutrons were incident on a 5m thick cuboid of water, which spanned the entire world $x-y$ plane and the total number of neutrons which entered the water were counted. The data showed that while the most neutrons were counted for a source size of 10m in side length, it was only 1% larger than the minimum which was when the source size was 20m, making it effectively constant across the range of source sizes. This data is shown in Appendix D. Because of this, I decided to use a source size of 10m.

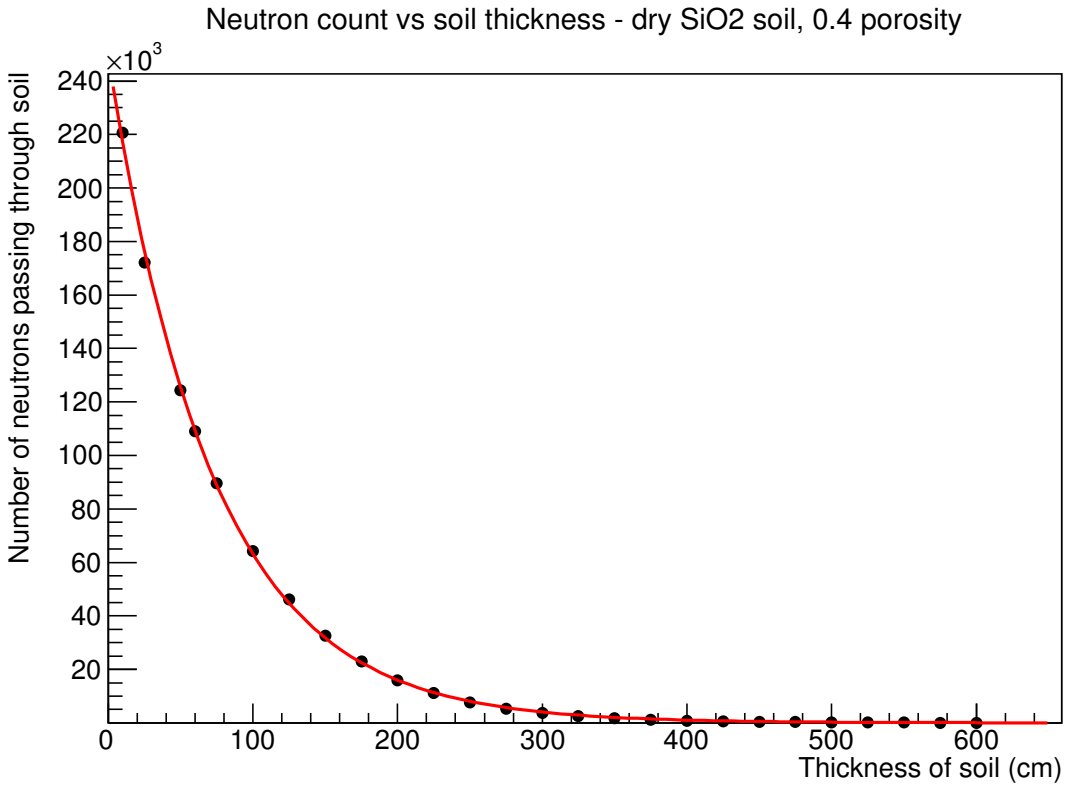

Fig. 11: A plot of the number of neutrons making it through a 0.4 porosity, fully saturated soil, vs. soil thickness generated. A clear exponential decay-like curve can be seen, to which ROOT's internal fitting algorithms fit a function of the form $f(x) = a \cdot e^{-x/b}$, granting us the attenuation length (b) of the neutrons in this particular soil.

To help validate this data a set of simulations was done using a "soil" comprised of pure water, and the attenuation length found both by using the method described in §3.1 and by hand. To make calculations simpler, it was assumed that every neutron possessed the same kinetic energy, equal to that of the mean energy generated by CRESTA. This came to be approximately 170 MeV. Values for the elastic scattering cross sections of 200MeV incident upon H-1 and O-16, the two elements present in water, were taken to be 50 mb and 150 mb respectively[29] to work out the attenuation length; the attenuation length of cosmic ray neutrons in water came to be approximately 124cm. Comparing this to the value derived by the method described in §3.1 which came to 120cm ± 2cm, it was deemed a satisfactory approximation.

The most significant source of error in finding the attenuation lengths, was the practical limitations of running simulations of a manageable size. Each simulation yielded a result which would have had a poissonian spread associated with it. This means that we know if the final number of neutrons counted after a simulation is N, that the associated spread of this point has a standard deviation. (std. dev.) of $\pm\sqrt{N}$. This is important as it implies that as N decreases, the larger its associated error is with respect to itself. Depending on the thickness and saturation, the number of neutrons counted could be quite low, leading to relatively large uncertainties about those points. A simple way in which errors could have been reduced would be to have been to increase the number of neutrons per simulation. As it stands, the simulations were already taking a very long time and increasing it any further may not have been a sensible or practicable. The other source of error was the line fitting algorithm used by ROOT: `Migrad Algorithm ("Minuit", "Migrad")`. The accuracy of this fitting algorithm could have been improved by increasing the number of data points passed to it, but that again would have required more simulations, for which there was not enough time. With the equipment available, the best was done.

There are many ways in which this research could be expanded upon. Perhaps the most natural extension would be to investigate the attenuation lengths of cosmic ray neutrons in lower porosity levels of the same soil. Porosities 0.5-0.75 were an adequate place to start but may not represent the vast majority of soils out there[30]–[33], most of which are much denser. Other ways that research could have been improved and expanded upon include: using a wider range of saturations for the 0.6 and 0.75 pore soils; use soils of different elemental compositions; run simulations where the surface of the soil isn't level, or has an incline. Research could also be carried out in the field using real soils and using the the actual flux of cosmic rays as found in nature.

4.2 Finding water

The results for the roving detector can across the sub-figures of Figures 12 and 13. In Fig. 12, the water body was set to a depth of 1m below the surface and in Fig. 13, the water body was set to a depth of 2m below the surface. Each sub-figure displays two plots on the same axes, one where a water body is present and one where it is not, showing the number of neutrons counted below a certain energy vs. the position of the detector along the y-axis. Each set of plots come from the same set of simulations.

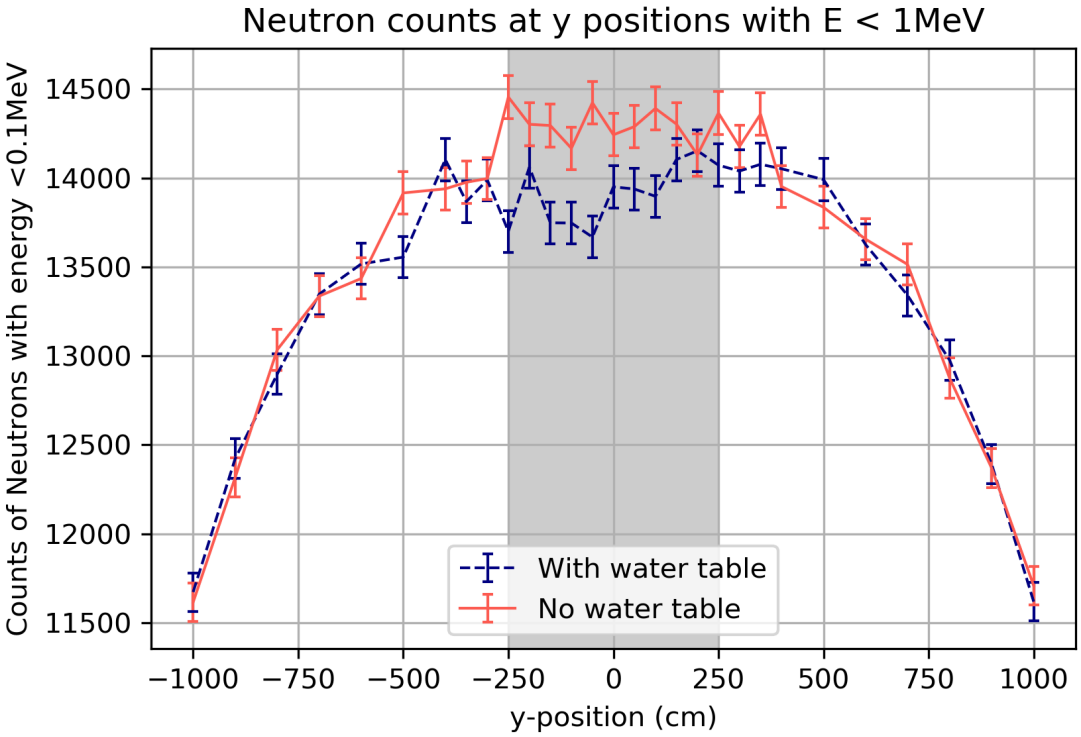

Fig. 12a: A plot of neutrons with $E_n < 1$ MeV vs. y-position of the roving detector, error bars set to one std. dev.. The water body was at a depth of 1m below the surface of the 0.5 porosity SiO_2 soil. Each point is the result of one simulation run for 3×10^6 neutrons. Grey shading indicates where the detector was directly over the water body.

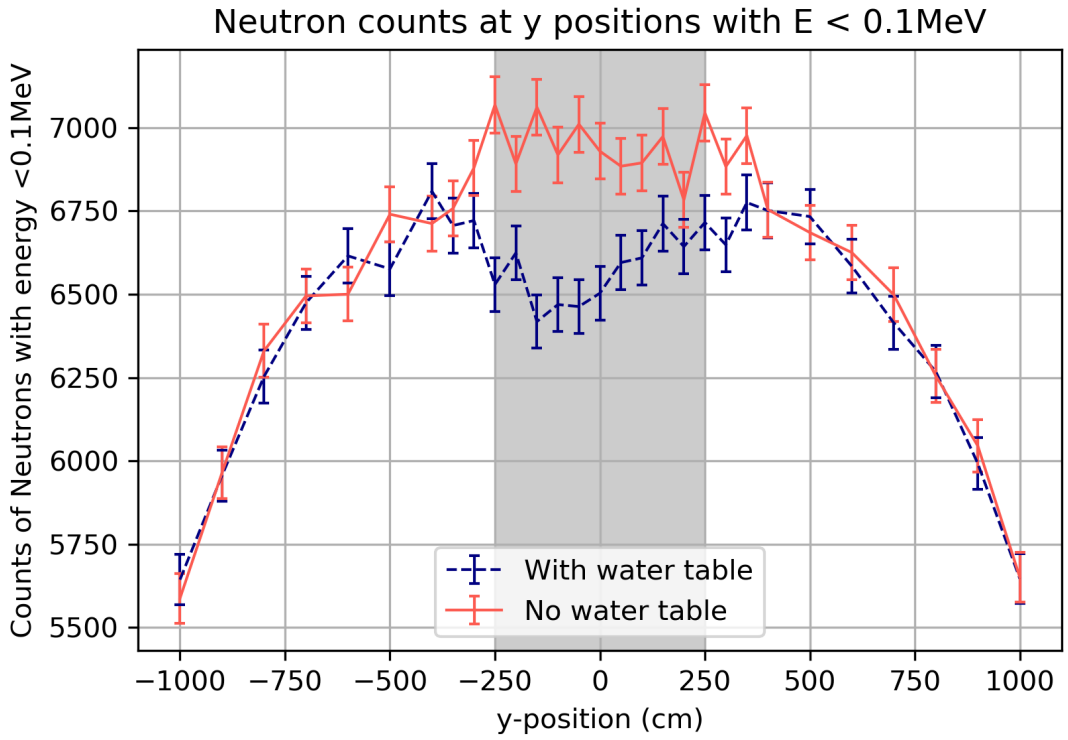

Fig. 12b: A plot of neutrons with $E_n < 0.1$ MeV vs. y-position of the roving detector, error bars set to one standard deviation. The water body was at a depth of 1m below the surface of the 0.5 porosity SiO_2 soil. Each point is the result of one simulation run for 3×10^6 neutrons. Grey shading indicates where the detector was directly over the water body.

Fig. 12

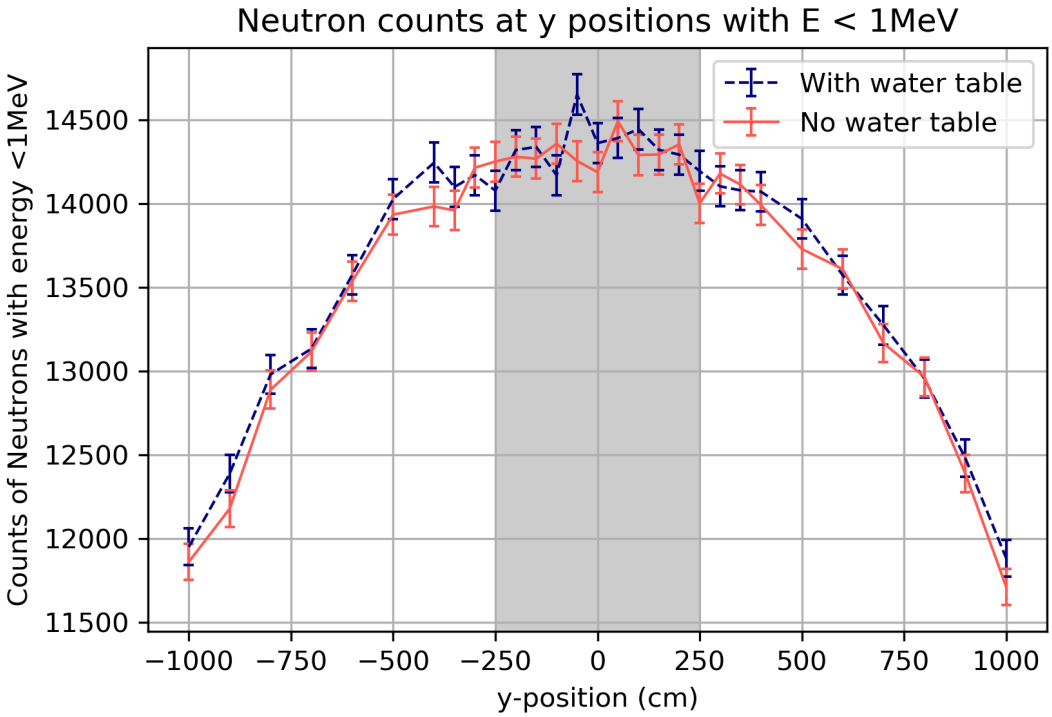

Fig. 13a:

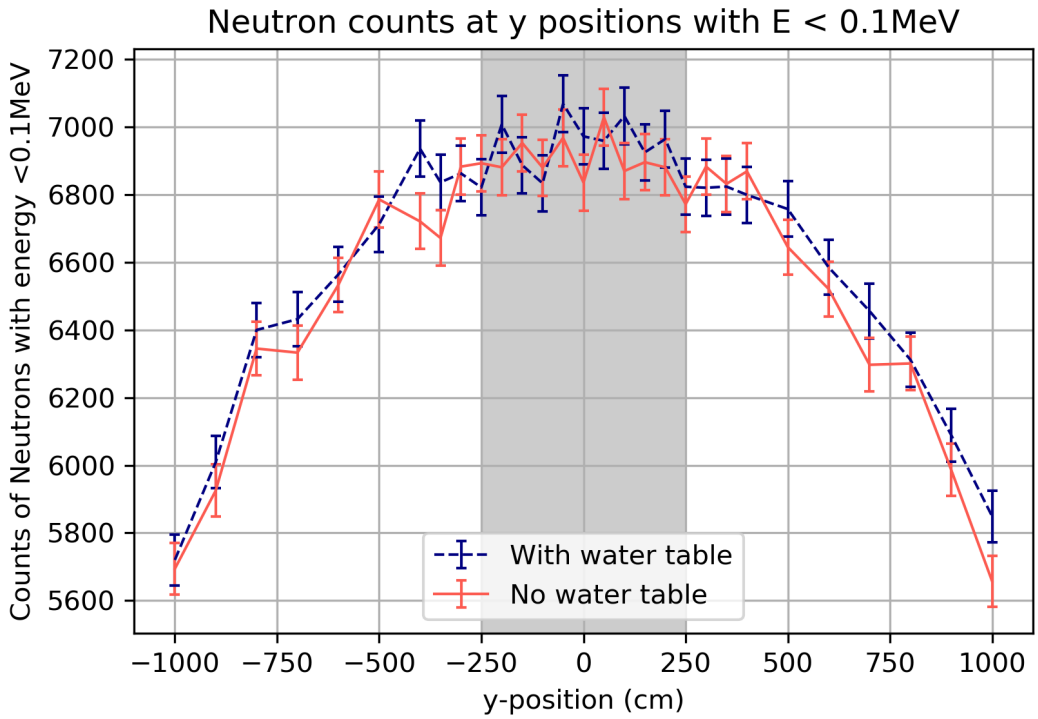

Fig. 13b:

Fig. 13: Plots of neutrons counted with $E_n < 1$ and $E_n < 0.1$ MeV vs. y-position of the roving detector shown in Fig. 13a and Fig. 13b respectively, error bars set to one standard deviation. The water body was at a depth of 1m below the surface of the 0.5 porosity SiO_2 soil. Each point is the result of one simulation run for 3×10^6 neutrons. Grey shading indicates where the detector was directly over the water body.

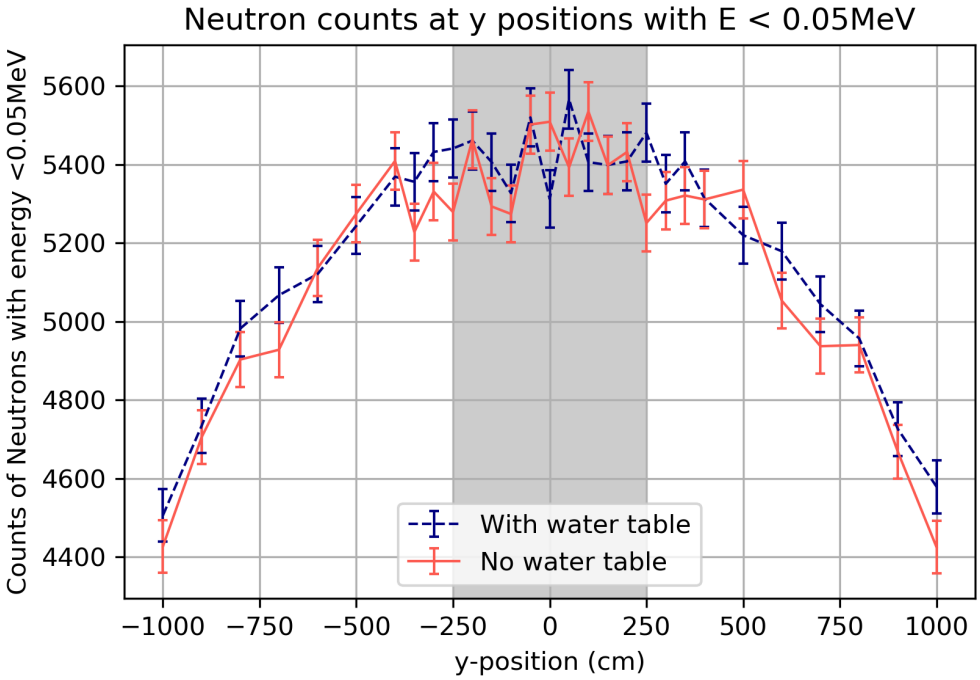

Fig. 13c: A plot of neutrons with $E_n < 0.05$ MeV vs. y-position of the roving detector, error bars set to one standard deviation. The water body was at a depth of 1m below the surface of the 0.5 porosity SiO_2 soil. Each point is the result of one simulation run for 3×10^6 neutrons. Grey shading indicates where the detector was directly over the water body.

Initially the simulations were run with the water body 1m below the surface of the soil (Fig. 12) and after moderately successful results they were repeated with the water body at a depth of 2m (Fig. 13). The final data output by CRESTA was cut so that only neutrons below a certain energy thresholds were plotted. This was because while CRESTA was set to count every neutron that entered the detector region and log its energy, the neutrons of interest were only the neutrons which had been able to penetrate down to some depth and scatter back up to the surface. These neutrons after completing such a journey most likely would have lost a significant amount of energy, in accordance with (4). The quantity of such neutrons able to do this in comparison to the total number of neutrons which would pass through the detector would likely be very small as the probability that a particle would scatter along such a path is also quite small.

In the case where the water body was at a depth of 1m, a significant drop in neutrons was observed when the detector was over the water. Two energy cuts were made: Fig. 12a shows the counts of neutrons with $E_n < 1$ MeV and Fig. 12b shows the counts of neutrons with an $E_n < 0.1$MeV. The drop in neutrons is much clearer in Fig. 12b, likely due to the further filtering out of higher energy, unscattered, cosmic ray neutrons. For the purpose of clarity, the error bars in these plots are all set to 1 std. dev. which gives a 68% confidence limit around each point. Higher accuracy error bars were not used in order to

keep the figures clear but it should be noted that even when the error bars are increased to 2 std. dev. that there is still no overlap between the error bars between the plot lines in Fig. 12b at points directly over the body of water.

The results from the simulations where the water was placed at 2m depth show no clear drop in the number of neutrons counted above the water source and require further discussion. Figs. 13a, 13b and 13c show the results for the number of neutrons counted with energies below 1MeV, 0.1MeV and 0.05 MeV respectively. Neutrons able to reach a depth of 2m and scatter back up to the surface were expected to be of even lower energy than neutrons which had made the same journey to a depth of only 1m, hence a third lower energy cut was made. As with Figures 12a and 12b, error bars are set to 1 std. dev. confidence limit to make the figures clearer. This being said, higher accuracy error bars would not be needed as none of the plots exhibit a significant difference in the number of counted neutrons above the body of water with the 1 std. dev. error bars

While these results indicates that using the CRNS method to search for underground sources of water may be limited quite considerably to areas with shallow bodies of water, it is worth noting, that each simulation was only run for 9×10^6 neutrons, which CRESTA calculates as approximately 333s of exposure time in the real world. In practice, out in the field, this would be a very short amount of time to leave a detector to take a reading[10] and it is more likely that detectors would be set in place for periods on the order of hours. If it were possible, these simulations would have been run for longer which may have yield different result but time constraints made this unfeasible. Something that could have helped yield different results would have been to look at lower energy cuts, but the number of neutrons detected very quickly drops off and past 0.05MeV, the number of neutrons being counted becomes too low to plot anything meaningful.

A feature all figures display is a clear drop off in the number of neutrons counted at all energy cut offs, at detector positions greater than approximately 3m away from the origin. This drop off is observed both when the water body is and is not present and is likely manifest due to the size and location of the source. The detector path ranged from -10m to +10m along the y-axis; by comparison, the source with a 30m side length ranged from -15m to +15m, so as the detector started and ended its journey it would have been fairly close to the edge of the source, picking up fewer neutrons because it was further away from where the bulk of neutrons were being generated. An effect similar to this was predicted by the results shown in Appendix D, but shows a much smaller effect than what was observed.

There are two ways the accuracy and quality of the data plotted in Fig. 12 and Fig. 13 could have been improved. The step size of the roving detector could have been decreased, giving a less granular plot, smoothing out some of the noise. The uncertainty

associated with each individual point could have been improved by running more neutrons per simulation. This is because, similar to §4.1, the spread associated with each is poissonian, meaning an increase in counts improves the associated error. The reason these improvements were not implemented at the time was due to the time constraints. The data in Figs. 12 and 13 took approximately 8 days of continuous run time to generate[IV], which is already large proportion of the allocated time for the project. A strong recommendation for further work would be to allocate more time for gathering data, use a computer with better specifications than the one used for this project or, if possible, to book time on a super computer.

The fact that the results show that it may be feasible to use a CRNS set up to detect sources of water underground shows that is encouraging but there is a lot of room for further investigation with this project. The simulations shown in this section explore one possible setup out of numerous possible arrangements of variables which could have affected the results in different ways. Understanding how these variables, which are outlined in Table 2, could affect the data, would be key to understanding the limits and effectiveness of this method. Comparing Figs. 12 and 13, we can see that the ability to detect water sources depends strongly on the depth to the source of water. The exact depth this method is limited to remains to be seen; had there been more time, it would have been interesting to run more simulations with the water body at a range of depths in between 1m and 2m, to see where the limits of depth are, for being able to detect a water source using a roving detector, scanning for short periods of time.

[IV] Two sets of simulations, one with a water table and one without, each one took 4 days to complete.

5 Conclusion

The purpose of this project was to use the radiation transport code CRESTA, which used Geant4 and CRY code libraries, to determine whether it is possible to use cosmic ray neutrons to detect sources of underground water. Simulations were run and results were analysed using ROOT and Python.

The results from §4.1, showed that the attenuation length of the flux of cosmic ray neutrons in soils increases with increasing soil porosity and decreases with increasing water saturation. While the soil porosities used were quite high and potentially not representative of most soils found around the world, they show that the attenuation length of cosmic ray neutrons varies around 1m in SiO_2 based soils of porosity 0.4 - 0.75, even when fully or partially saturated with water. The results from §4.2 then showed promising evidence that in the right circumstances, a roving neutron detector can be used to look for a reduction in the flux of cosmic ray neutrons above the ground, signalling the potential presence of an underground water source, or more generally, any hydrogen rich material. Results for a water body representing an underground lake 1m below the ground's surface showed a clear drop in the flux of fast neutrons, around 0.1 MeV - 1 MeV, above the water body. However when the depth to the water body was increased to 2m, the counts of neutrons no longer showed any discernible or significant drop when the detector was above the water body. Whether this was due to not running enough neutrons per simulation or other factors, remains to be seen and could be an excellent candidate for further research and an extension of this project. Other factors that could be explored for future research are listed in Table 2. The biggest limiting factor throughout the project was the time. Each simulation was run for 9×10^6 neutrons, which was equivalent to 333s of exposure in the real world, this is much a shorter time than one would typically leave a neutron detector out in the field. It would be interesting to see how the results shown in Fig. 13 would have differed had the simulations been run for a more appropriate length of time.

Shown in Fig. 14, is a global map from a study by Y. Fan, H. Li, and G. Miguez-Macho[34], which shows the depths to water tables around the world. It was produced by combining physical data taken at locations around the world and data from simulations. Using this map, and the results from §3.1, we can see that there are potentially a wealth of places where the water table is < 2.5m from the surface, and might be able to benefit from this technique. This includes remote regions like central Africa, South America, and central Australia. These regions and the people in them might really benefit from this technology if were able to be developed further, into something reliable.

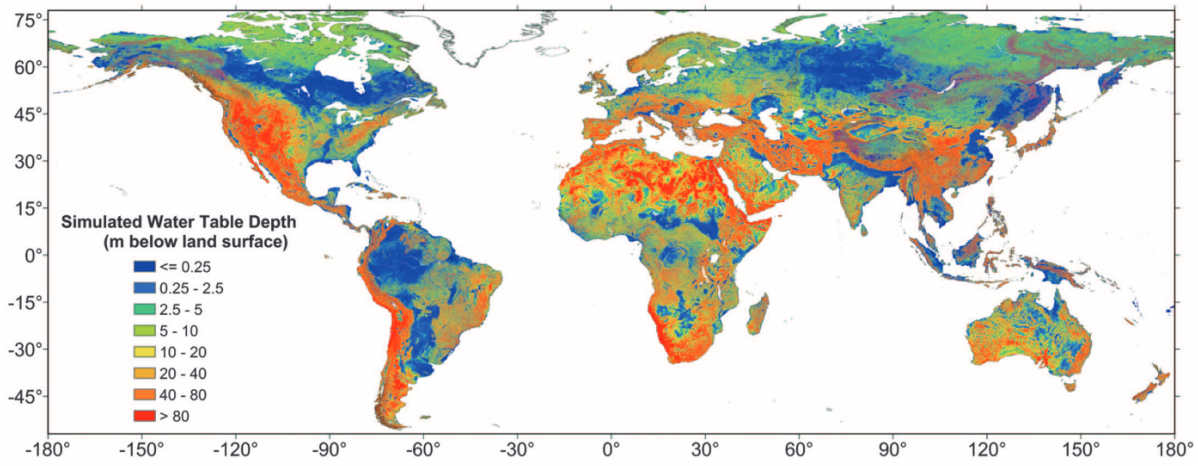

Fig. 14: Map of the world displaying the simulated water table depth. Taken from *Global Patterns of Groundwater Table Depth* by Y. Fan, H. Li, G. Miguez-Macho[34]

If further research were done into this area and the method honed it would be my hope that this research could help people around the world, in areas where water is shallow enough to be accessible if only its presence could be discovered.

References

[1] M. Zreda, W. Shuttleworth, X. Zeng, C. Zweck, D. Desilets, T. Franz, and R. Rosolem, "Cosmos: The cosmic-ray soil moisture observing system," *Hydrology and Earth System Sciences*, vol. 16, no. 11, pp. 4079–4099, 2012.

[2] I. Mitrofanov, D. Anfimov, A. Kozyrev, M. Litvak, A. Sanin, V. Tret'yakov, A. Krylov, V. Shvetsov, W. v. Boynton, C. Shinohara, *et al.*, "Maps of subsurface hydrogen from the high energy neutron detector, mars odyssey," *Science*, vol. 297, no. 5578, pp. 78–81, 2002.

[3] S. S. Board, N. R. Council, *et al.*, *An Astrobiology Strategy for the Exploration of Mars*. National Academies Press, 2007, ch. 7. DOI: 10.17226/11937.

[4] M. Schrön, "Cosmic-ray neutron sensing and its applications to soil and land surface hydrology," 2017.

[5] M. Köhli, M. Schrön, M. Zreda, U. Schmidt, P. Dietrich, and S. Zacharias, "Footprint characteristics revised for field-scale soil moisture monitoring with cosmic-ray neutrons," *Water Resources Research*, vol. 51, no. 7, pp. 5772–5790, 2015.

[6] L. Hendrick and R. Edge, "Cosmic-ray neutrons near the earth," *Physical Review*, vol. 145, no. 4, p. 1023, 1966.

[7] M. Kodama, "Continuous monitoring of snow water equivalent using cosmic ray neutrons," *Cold Regions Science and Technology*, vol. 3, no. 4, pp. 295–303, 1980.

[8] M. Kodama, S. Kudo, and T. Kosuge, "Application of atmospheric neutrons to soil moisture measurement," *Soil science*, vol. 140, no. 4, pp. 237–242, 1985.

[9] L. Dorman, *Cosmic rays in the Earth's atmosphere and underground*. Springer Science & Business Media, 2004, vol. 303. DOI: 10.1007/978-1-4020-2113-8.

[10] M. Zreda, D. Desilets, T. Ferré, and R. L. Scott, "Measuring soil moisture content non-invasively at intermediate spatial scale using cosmic-ray neutrons," *Geophysical research letters*, vol. 35, no. 21, 2008. DOI: 10.1029/2008GL035655.

[11] C. J. Cleveland and C. G. Morris, *Dictionary of energy*. Elsevier, 2015.

[12] K. S. Krane, D. Halliday, *et al.*, *Introductory nuclear physics*. Wiley New York, 1988, vol. 465.

[13] C. J. Christensen, A. Nielsen, A. Bahnsen, W. Brown, and B. Rustad, "Free-neutron beta-decay half-life," *Physical Review D*, vol. 5, no. 7, p. 1628, 1972.

[14] National Nuclear Data Centre. (). Plot of hydrogen cross section from the National Nuclear Data Centre, [Online]. Available: https://www.nndc.bnl.gov/sigma/plotCart.jsp?plot=Neutron%20reactions:14960.3.102&submit=Plot%20selected%20files (visited on 07/2019).

[15] K. Shibata, O. Iwamoto, T. Nakagawa, N. Iwamoto, A. Ichihara, S. Kunieda, S. Chiba, K. Furutaka, N. Otuka, T. Ohsawa, *et al.*, "Jendl-4.0: A new library for nuclear science and engineering," *Journal of Nuclear Science and Technology*, vol. 48, no. 1, pp. 1–30, 2011.

[16] T. K. Gaisser, R. Engel, and E. Resconi, *Cosmic rays and particle physics*. Cambridge University Press, 2016.

[17] M. Ackermann, M. Ajello, A. Allafort, L. Baldini, J. Ballet, G. Barbiellini, M. Baring, D. Bastieri, K. Bechtol, R. Bellazzini, *et al.*, "Detection of the characteristic pion-decay signature in supernova remnants," *Science*, vol. 339, no. 6121, pp. 807–811, 2013.

[18] J. Vink, J. Bleeker, K. van der Heyden, A. Bykov, A. Bamba, and R. Yamazaki, "The x-ray synchrotron emission of rcw 86 and the implications for its age," *The Astrophysical Journal Letters*, vol. 648, no. 1, p. L33, 2006.

[19] A. W. Strong, I. V. Moskalenko, and V. S. Ptuskin, "Cosmic-ray propagation and interactions in the galaxy," *Annu. Rev. Nucl. Part. Sci.*, vol. 57, pp. 285–327, 2007. DOI: 10.1146/annurev.nucl.57.090506.123011.

[20] P. Goldhagen, M. Reginatto, T. Kniss, J. Wilson, R. Singleterry, I. Jones, and W. Van Steveninck, "Measurement of the energy spectrum of cosmic-ray induced neutrons aboard an er-2 high-altitude airplane," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 476, no. 1-2, pp. 42–51, 2002. DOI: 10.1016/S0168-9002(01)01386-9.

[21] T. Sato and K. Niita, "Analytical functions to predict cosmic-ray neutron spectra in the atmosphere," *Radiation research*, vol. 166, no. 3, pp. 544–555, 2006. DOI: 10.1667/RR0610.1.

[22] S. Agostinelli, J. Allison, K. a. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. 2. Barrand, *et al.*, "Geant4—a simulation toolkit," *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, pp. 250–303, 2003.

[23] (Jun. 2019). CERN homepage, [Online]. Available: https://geant4.web.cern.ch/ (visited on 08/16/2019).

[24] CERN. (Jun. 2019). Geant4 testing and validation, [Online]. Available: https://geant4.web.cern.ch/publications_validations/testing_and_validation (visited on 08/16/2019).

[25] C. Hagmann, D. Lange, and D. Wright, "Monte carlo simulation of proton-induced cosimc ray cascades in the atmosphere," Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), Tech. Rep., 2007.

[26] CERN. (2018). About ROOT: ROOT a data analysis framework, [Online]. Available: `https://root.cern.ch/about-root` (visited on 08/16/2019).

[27] B. S. Tubana, T. Babu, and L. E. Datnoff, "A review of silicon in soils and plants and its role in us agriculture: History and future perspectives," *Soil Science*, vol. 181, no. 9/10, pp. 393–411, 2016.

[28] *Aquifers types, impacts, and conservation*, eng, ser. Environmental science, engineering and technology. New York: Nova Science Publisher's, Inc., 2012, ISBN: 1-61942-096-1.

[29] N. Soppera, E. Dupont, M. Bossant, and M. Fleming, "JANIS book of neutron-induced cross-sections," *OECD NEA Data Bank,Comparison of evaluated and experimental data from EAF-2010, ENDF/B-VIII.0, JEFF-3.3, JENDL-4.0u, JENDL/HE-2007, IRDFF-1.05, TENDL-2017 and EXFOR*, May 2018. [Online]. Available: `https://www.oecd-nea.org/janis/book/book-neutron-2018-05.pdf` (visited on 09/10/2019).

[30] S. Swiss Standard, "670 010b," *Characteristic Coefficients of soils, Association of Swiss Road and Traffic Engineers*, 1999.

[31] B. M. Das, *Advanced soil mechanics*. Crc Press, 2013.

[32] B. K. Hough, *Basic soils engineering*. Ronald Press Co., 1969.

[33] K. Terzaghi, R. B. Peck, and G. Mesri, *Soil mechanics in engineering practice*. John Wiley & Sons, 1996.

[34] Y. Fan, H. Li, and G. Miguez-Macho, "Global patterns of groundwater table depth," *Science*, vol. 339, no. 6122, pp. 940–943, 2013.

A Atomic composition code

This code was used to determine the atomic composition of the 0.4 porosity soil simulated. In its current form it only works for SiO_2 based soils but has the capacity to be expanded fairly easily using python's dictionary data types. It allows a user to enter a desired soil porosity and water saturation level.

```
#!/usr/bin/env python3
"""
This script takes a user entered soil porosity and saturation level and returns the atomic
masses per cm^3 and mass fractions for each element present. Density is done in cm^3
and all volumes are 1cm^3 which allows for easy calculations of density as they are
effectively the same.

Only one soil composition was needed for this research but the use of dictionaries allows for
easy expansion to more soil types.

"""
Av = 6.02e23
# avogadros constant

Mr = {'H': 1, 'C': 12, 'N': 14, 'O': 16, 'Si': 28, 'Ar': 40}
# Mrs of relevant elements

rho = {'H2O': 1, 'SiO2': 2.65}
# densities of relavent materials in g/cm^3

ppcm3Air = {'N': 4.1925e19, 'O': 1.1314e19, 'Ar': 2.5531e17, 'C': 1.3438e16}
# particles per cm^3 in air, calculated by hand assuming 1 mole of any gas occupies 22.4dm^3
# at rtp, and that the atmosphere is 78% Nitrogen gas (N2), 21% oxygen gas (O2), 0.95% Argon
# gas, 0.05% carbon dioxide CO2 gas


# this determines how much of the volume is empty space
porosity = float(input('Please enter a porosity in the form.\n'
                       '0.XXX...: '))

print(porosity)
# how much of the empty space is to be filled by water
saturation = float(input('How saturated is this soil (%): \n'
                         'Please enter a number percentage between 0 and 100: ')) / 100

```

```
36 print(saturation)
37
38 # work out how much volume the air, water and soil each has
39 total_vol = 1  # cm3
40 vol_space = (total_vol * porosity)
41 vol_solid = total_vol - vol_space
42 vol_H2O = (vol_space * saturation)
43 vol_air = vol_space - vol_H2O
44
45 # prints out volumes of air, water and soil (solid)
46 print('\n\nthis gives us:\n'
47       '\t :-\t {0}cm3 air/cm3\n'  #
48       '\t :-\t {1}cm3 water/cm3\n '
49       '\t :-\t {2}cm3 SiO2\n\n'.format(vol_air, vol_H2O, vol_solid))
50
51 # masses of each elemeent in air
52 mass_N = (vol_air * ppcm3Air['N']) / Av * Mr['N']
53 mass_O_air = (vol_air * ppcm3Air['O']) / Av * Mr['O']
54 mass_Ar = (vol_air * ppcm3Air['Ar']) / Av * Mr['Ar']
55 mass_C = (vol_air * ppcm3Air['C']) / Av * Mr['C']
56
57
58 # masses of elements in water
59 massH = vol_H2O * (2 / 18)  # water is 2/18ths H by mass
60 massO_H2O = vol_H2O * (16 / 18)  # water is 16/18ths by mass
61
62 # masses of elements in soil
63 moles_solid = (vol_solid * rho['SiO2']) / (Mr['Si'] + (Mr['O'] * 2))
64 mass_Si = moles_solid * Mr['Si']
65 mass_O_solid = moles_solid * 2 * Mr['O']
66
67 total_mass_O = mass_O_air + massO_H2O + mass_O_solid
68
69 totalM = massH + total_mass_O + mass_N + mass_Si + mass_Ar + mass_C
70
71 masses = [massH, total_mass_O, mass_Si, mass_N, mass_Ar, mass_C]
72 elements = ['H', 'O', 'Si', 'N', 'Ar', 'C']
73
74 for i in range(len(masses)):
75     if i == 0:
76         print('This makes the soil:')
77         print('\t {0:.3}g {1} atoms; mass fraction {2:.5}'.format(masses[i],
           ↪  elements[i], masses[i] / totalM))
78     else:
79         print('\t {0:.3}g {1} atoms; mass fraction {2:.5}'.format(masses[i],
           ↪  elements[i], masses[i] / totalM))
80
```

```
print(totalM) # prints total mass which is also the density as everything is done by cm3
```

B Script used to automate process

This script was used to almost fully automate the part of the simulation process described in §3.1. While the script itself is written in Python, it uses two Bash scripts to circumnavigate commands I could not manage to do directly from the Python script. Scripts like this were used throughout the project but only one was included in the appendix, as more would be unnecessary due to their similarity.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
This script begins by calling CRsim.sh which sets off a simulation for 3e5 particles.
After completion,  CRsim.sh is then opened and read in as a string to a variable which
is then edited to change the name of the output ROOT file CRESTA outputs to, in
preparation for the next simulation. The old CRsim is then deleted and a new bash
script of the exact same name is created and written using the altered text.
This new CRsim.sh is then given the correct permissions using the file chmod.sh.

Next the appropriate changes need to be made to the geometry file for the next simulation.
This is done in a similar way to CRsim.sh; the text is read in, edited, the old file is
deleted and a new one is written. No permission change is necessary here.

This is repeated until all required simulations have been run
"""

import subprocess  # allows python to interact with the terminal.



# Here I define some functions which will be used several times in the main body
# of my code.

# this function deletes the original CRsim.sh script,then it writes a
# new updated script also called CRsim.sh using the text it's been passed,
# and then grants it the neccesary permissions.
def delwrtchmodCRsim(text_CRsim):
    subprocess.call(["rm", "CRsim.sh"], shell=True)  # delete old file

    with open("CRsim.sh", "w") as file:  # creates a new file with same name
```

```
32          file.write(text_CRsim)  # write in updated text using text passed to function
33
34      subprocess.call(["./changemod.sh"], shell=True)  # grant neccessary permissions
35
36
37  # this function deletes the old geometry file, and then writes a new updated one
38  # using text passed to the function permission changes are not required here
39  def delwrtgeo(geo_lines):
40      subprocess.call(["rm", "cosmos_simple.geo"])  # delete old file
41
42      with open("cosmos_simple.geo", "w") as file:  # create new file with same name
43          for line_geo in geo_lines:  # write new doc using list of lines passed to
            ↪  function
44              file.write(line_geo)
45
46
47  #
48  # main body of code
49  #
50
51  # Helow I have a list of soil thicknesses, moisture levels, and porosities used in
    ↪  the
52  # simulations. A simple substitution in the right place in CRsim.sh and the geometry
    ↪  file
53  # allows me to edit them in preparation to run them again with updated parameters.
54  thickness = ['10', '25', '50', '60', '75', '100', '125',
55               '150', '175', '200', '225', '250', '275', '300',
56               '325', '350', '375', '400', '425', '450', '475', '500',
57               '525', '550', '575', '600']
58
59
60  mst_lvl = ["H20_0p0", "H20_0p01", "H20_0p02", "H20_0p05", "H20_0p1",
61             "H20_0p2", "H20_0p3", "H20_0p4", "H20_0p5"]
62
63
64  porosities = ["Pore_0p5", "Pore_0p6", "Pore_0p75"]
65
66
67  # initial replacement variables set to the string of characters to first be
68  # replaced in the files
69  currnt_thcknss = "10cm"  # initial thickness for substituion in  CRsim.sh
70  currnt_thcknss2 = "10*cm"  # initial thickness for substituion in the geo file
71  currnt_mst = "H20_0p0"  # initial moisture level for substituion in CRsim.sh and the
    ↪  geo file
72  currnt_por = "Pore_0p5"  #i nitial porosity  for substituion in CRsim.sh and the geo
    ↪  file
73
74  ############################################### start of porosity loop
```

```
75  for i in range(len(porosities)):
76      ########################################## start of saturation loop
77      for j in range(len(mst_lvl)):
78          ######################################## start of thicknesses loop
79          for k in range(len(thickness)):
80              subprocess.call(["./CRsim.sh"], shell=True)  # run simulation
81
82              if k == (len(thickness) - 1):  # at the last one, break, as the final
                ↪  simulations have already run
83                  break
84
85              with open("CRsim.sh", "r") as file:  # open CRsim in read mode
86                  text_k = file.read() # read file in as text
87                  nxt_thcknss = thickness[k + 1] + "cm"  # set replacement thickness to
                    ↪  next one in the list
88
89                  text_k = text_k.replace(currnt_thcknss, nxt_thcknss)  # replace
                    ↪  current thickness with next one
90                  currnt_thcknss = nxt_thcknss # prepare for next loop
91
92              delwrtchmodCRsim(text_k)  # delete current CRsim.sh, write a new one and
                ↪  changes its permissions
93
94              with open("cosmos_simple.geo", "r") as file: # open geometry file in read
                ↪  mode
95                  lines_k = file.readlines()  # read geometry file in as list of lines
96                  nxt_thcknss2 = thickness[k + 1] + '*cm'
97                  lines_k[8] = lines_k[8].replace(currnt_thcknss2, nxt_thcknss2)  #
                    ↪  replace correct bit with next bit
98
99                  currnt_thcknss2 = nxt_thcknss2
100
101             delwrtgeo(lines_k)  # deletes current geo and writes new one
102
103         ######################################## end of thicknesses loop
104
105         if j == (len(mst_lvl) - 1):
106             break
107
108         with open("CRsim.sh", "r") as file:
109             text_j = file.read()
110             nxt_thcknss = "10cm"
111             text_j = text_j.replace(currnt_thcknss, nxt_thcknss)  # reset thickness in
                ↪  file name to 10cm
112             currnt_thcknss = nxt_thcknss
113
114
115             nxt_mst = mst_lvl[j + 1]
```

```
116             text_j = text_j.replace(currnt_mst, nxt_mst)
117
118         delwrtchmodCRsim(text_j)
119
120         with open("cosmos_simple.geo", "r") as file:
121             lines_j = file.readlines()
122
123             nxt_thcknss2 = "10*cm"
124             lines_j[8] = lines_j[8].replace(currnt_thcknss2, nxt_thcknss2) # reset
                ↪  thickness in file name to 10cm
125
126             currnt_thcknss2 = nxt_thcknss2
127
128             lines_j[39] = lines_j[39].replace(currnt_mst, nxt_mst)
129             currnt_mst = nxt_mst
130
131         delwrtgeo(lines_j)
132     ################################################ end of saturation loop
133
134     if i == (len(porosities) - 1):
135         break
136
137     with open("CRsim.sh", "r") as file:
138         text_i = file.read()
139         nxt_thcknss = "10cm"
140         nxt_mst = "H20_0p0"
141
142         text_i = text_i.replace(currnt_thcknss, nxt_thcknss)
143         currnt_thcknss = nxt_thcknss
144
145         text_i = text_i.replace(currnt_mst, nxt_mst)
146
147         nxt_por = porosities[i + 1]
148         text_i = text_i.replace(currnt_por, nxt_por)  # change the porosity to the
            ↪  next one along
149
150     delwrtchmodCRsim(text_i)
151
152     with open('cosmos_simple.geo', 'r') as file:
153         lines_i = file.readlines()
154         nxt_thcknss2 = "10*cm"
155
156         lines_i[8] = lines_i[8].replace(currnt_thcknss2, nxt_thcknss2)
157
158         currnt_thcknss2 = nxt_thcknss2  # reprimes curret2
159
160         lines_i[39] = lines_i[39].replace(currnt_mst, nxt_mst)
161         currnt_mst = nxt_mst  # moves current along one
```

```
162
163         lines_i[39] = lines_i[39].replace(currnt_por, nxt_por)
164         currnt_por = nxt_por  # moves current along one
165
166     delwrtgeo(lines_i)
167
168     ############################################## end of porosity loop
```

C Data from Simulations

This table shows all of the attenuation lengths of cosmic ray neutrons for a SiO_2 soil of various porosities and water saturation levels, done in percentage by volume. A line of the form $f(x) = ae^{-x/b}$ where the variables a and b are parameters to be fit by ROOT's inbuilt line fitting algorithm: `Migrad Algorithm ("Minuit", "Migrad")`. While parameter b is the only one of relevance to this project, the parameter a was included for completeness.

Table 3: A table showing all attenuation lengths found by the method describes in §3.1

Soil Type and water saturation (%)	Parameter a	Parameter b
SiO_2, Porosity 0.4		
0.00%	272700 ± 4413	122 ± 2
1.00%	272436 ± 3583	114 ± 2
2.00%	271353 ± 2974	109 ± 2
5.00%	268955 ± 1822	101 ± 1
10.00%	264953 ± 864.	92.9 ± 0.4
20.00%	258185 ± 996.	83.9 ± 0
30.00%	252955 ± 1295	77.8 ± 1
40.00%	248997 ± 1366	72.9 ± 1
SiO_2, Porosity 0.5		
0.00%	272311 ± 4098	144 ± 3
1.00%	272083 ± 3202	134 ± 2
2.00%	271356 ± 2561	128 ± 2
5.00%	268475 ± 1331	118 ± 1
10.00%	263768 ± 795.	108 ± 1
20.00%	256280 ± 1246	96.7 ± 1
30.00%	250937 ± 1558	89.0 ± 1
40.00%	246996 ± 1597	82.9 ± 1
50.00%	243625 ± 1523	77.8 ± 1

Continued on next page

Table 3 – *Continued from previous page*

Soil Type and water saturation (%)	Parameter a	Parameter b
SiO_2, Porosity 0.6		
0.00%	271534 ± 3776	179 ± 4
1.00%	271555 ± 2783	165 ± 3
2.00%	270420 ± 2132	157 ± 2
5.00%	266706 ± 983	143 ± 1
10.00%	261522 ± 868	131 ± 1
20.00%	254274 ± 1717	115 ± 1
30.00%	249039 ± 1796	105 ± 1
40.00%	245164 ± 1880	96.2 ± 1
50.00%	241443 ± 1842	89.2 ± 1
SiO_2, Porosity 0.75		
0.00%	268186 ± 2847	287 ± 5
1.00%	268462 ± 1743	254. ± 2
2.00%	267294 ± 1086	238 ± 2
5.00%	263297 ± 885.	213 ± 1
10.00%	257108 ± 1725	191 ± 2
20.00%	249008 ± 2423	162 ± 3
30.00%	244137 ± 2577	142 ± 2
40.00%	240409 ± 2480	127 ± 2
50.00%	237382 ± 2221	116 ± 2

D Flux size data

Fig. 15 shows a plot showing how flux box sized affected total number of particles getting through the x-y plane of the simulation. The larger the source size, the more particles were generated near the edges and subsequently lost, from the simulation entirely.

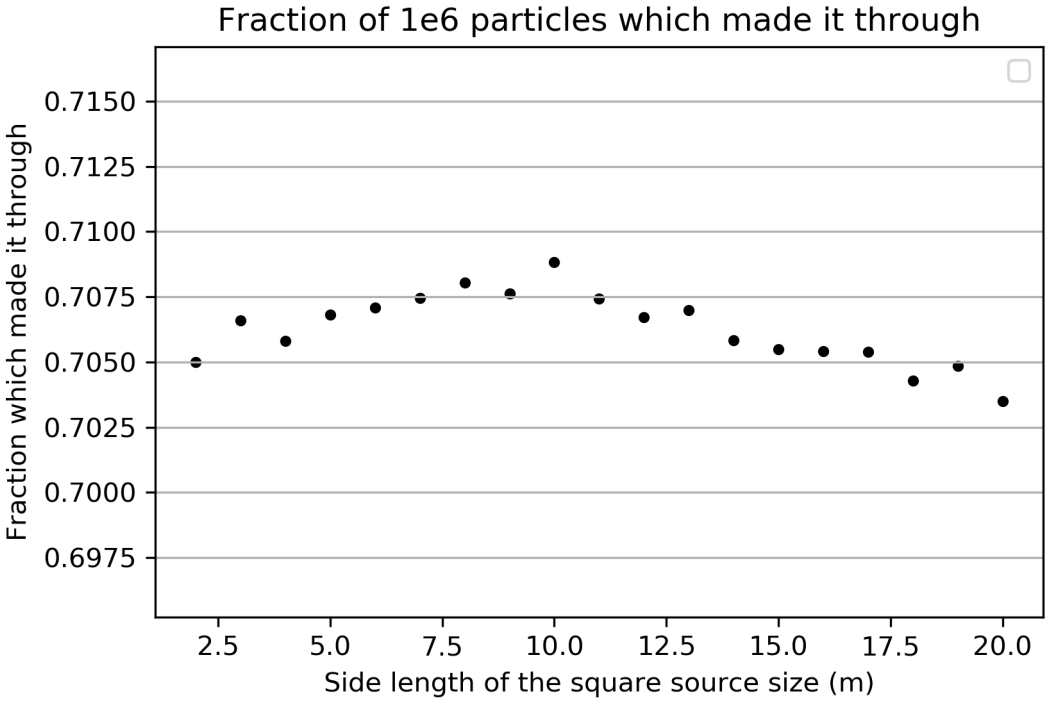

Fig. 15: A plot of the change in fraction of particles able to be counted when flux source size was changed between 2m and 20m side length.