

Christiaan PHD Thesis on NUSTAR, FAIR

Christiaan Alwin Douma

May 9, 2018

Contents

1	Introduction and Motivation	1
1.1	help for the introductory essay	1
2	Theory	2
2.1	Nuclear Shell model	2
2.1.1	Basics of the Nuclear Shell Model: Mean field approximation	2
2.1.2	Including the nucleon spins into the shell-model	6
2.1.3	Including the Pauli-principle in the shell model	8
2.1.4	Inclusion of the Spin-orbit coupling in the shell model	9
2.1.5	Features of the Nuclear Shell model	10
2.2	Syuomo PHD	14
2.3	Najafi PHD: Quasi-Free Proton and Neutron Knockout Reactions in ^{20}O	14
2.3.1	Basic concepts	14
2.3.2	Fragmentation effects on the spectra	16
2.3.3	Experimental effects and detector principles	18
2.3.4	Analysis and results of the experiment	21
2.4	Alex PHD	23
2.5	Zahra PHD	23
2.6	Marcel PHD	23
2.7	Solmaz PHD	23
2.8	Gita Postdoc	23
3	Construction of Super-FRS	24
3.1	Practical information	24
3.2	NX-9 tutorials	25
3.3	X-slit system	25
3.3.1	Matlab simulation of Densimet	25
3.3.2	Implementation of Radiation cooling by using the Debye Model	26
3.3.3	Implementation of the beam	28
3.3.4	Thermal conductivity	29
3.3.5	Time integration	30
3.4	Target system	30
4	Materials and Method of Analysis	31
4.1	Principles of operation for NeuLAND	31
4.1.1	Basic principles of scintillators	31
4.1.2	Hardware triggers on the signals	32
4.1.3	ADC, QDC and TDC	34
4.1.4	Pulse time and self-triggering	34
4.1.5	Event reconstruction and software triggers	35
4.1.6	Detecting neutrons with NeuLAND and the VETO-detector	35
4.1.7	Statistical principles of detector efficiency	35
4.2	Documentation on R ³ Broot	40
4.2.1	Activating the digitizer	41
4.3	Full infrastructure of the NeuLAND detector	41
4.3.1	Electronic infrastructure of NeuLAND	41
4.3.2	Operation principles of the TDC and QDC on TACQUILA	44
4.3.3	Software structure of NeuLAND	45

4.3.4	Synchronization of the photomultipliers of NeuLAND	46
4.3.5	Synchronization of the different bars	47
4.3.6	Correcting QCD for the light attenuation in the scintillator bar	48
4.3.7	Operation principles of TAMEX	48
4.3.8	Monitoring of NeuLAND during IMPACT experiment	49
4.3.9	Use of the upgraded algorithm for monitoring histograms	50
4.3.10	Using the SAMURAI online monitor	52
4.3.11	Restarting a crashed DAQ	52
4.3.12	How to get access to the real data of NeuLAND in coincidence with the beam	53
4.4	SAMURAI Analysis procedure	55
4.4.1	Performing analysis on SAMURAI data	55
4.4.2	Performing particle ID on the SAMURAI NeuLAND setup	56
4.4.3	SAMURAI setup	57
4.4.4	MINOS	58
4.4.5	NEBULA	59
5	Experiments at the RCNP facility	61
5.0.1	Overview of RCNP	61
5.0.2	Practical information on the experiments	62
5.0.3	Operation principles of the Grand Raiden Spectrometer	69
5.0.4	Meeting report 25 April	71
5.0.5	Tasks and operation principles of the Analyser	76
5.0.6	Analysis tricks you can do yourself (NOT done by the analyzer)	77
5.0.7	How to do Online Monitoring	77
5.0.8	Planning of my PhD Thesis	81
5.0.9	Michael his analyzer	81
5.0.10	Binary Code from the wiki page	82
5.0.11	Accelerator	82
5.0.12	Summary of the Experimental Log	82
5.0.13	Identifying states	85
5.1	Running an experiment at RCNP	85
5.2	Experimental setup of FAIR and NUSTAR	94
5.3	Own Analysis steps	94
6	Results	96
7	Conclusion	97
8	Operation of the Fold package	98
8.1	oxbash	98
8.2	wsaw	99
8.3	normod	100
8.4	fold	101
8.5	dwhi	103
8.6	MDA	105
8.7	Data fit	106
8.8	Final notes	107
A	Dead Ends	116

Abstract

I still need to fill this in...

Chapter 1

Introduction and Motivation

1.1 help for the introductory essay

Going for the story of Alex, it has an introduction, which is a very broad overview of the physics and experimental setup, PANDA in his case. No actual work needs to be done for this, it is just a summary of what is known already. Chapter 2 is an explanation of the setup itself. Again, no actual work. Chapter 3 is an overview of the physics goals for PANDA and why PANDA is so good and special. This does again not require real work. In Chapter 4 the analysis framework of PANDA is explained. It starts with some general remarks on PANDAroot and then some real work comes on simulations that are carried out. Conclusions about these simulations and the selected decay channel are then made and some predictions on which work has to be done is presented. So the actual work is only on this chapter! Then chapter 5 is just the conclusions and chapter 6 are references. That sounds pretty easy, but not in the sense that no work has to be done! See [1].

Chapter 2

Theory

2.1 Nuclear Shell model

2.1.1 Basics of the Nuclear Shell Model: Mean field approximation

The nucleus of an atom is composed of N neutrons and Z protons. The mass number A is defined as $A = N + Z$. The motion of these A nucleons is (when nuclear reactions are absent) non-relativistic [2]. Therefore the behavior of the nucleons in the nucleus is fully described if one knows the Quantum Mechanical wave function describing all these nucleons together [3]. Finding this wave functions means solving the Schrodinger equation [3]. Notice that for this model the protons and neutrons are considered to be elementary particles.

An atom has a radius of approximately 10^{-10} m [4]. A nucleon has a radius of about $r = r_0 \cdot A^{1/3}$ with $r_0 = 1.3 \text{ fm} = 1.3 \cdot 10^{-15} \text{ m}$. Energy scales inside the proton and neutron are of the order of 1 GeV (their mass, which is generated dynamically). This is because most of their mass is generated dynamically. Nuclear effects have however an energy scale of several MeV: gamma rays. Hence the energy scales of nucleons versus nuclear effects are very different. Therefore the approximation of considering protons and neutrons as elementary particles is justified for crude models like the pure shell model that we discuss here. More sophisticated models must however treat protons and neutrons as non-elementary.

Notice also that one searches for a single wave function that describes *all* nucleons at the same time:

$$\Psi = \Psi(t, \vec{x}_1, \vec{x}_2, \dots, \vec{x}_Z, \vec{y}_1, \vec{y}_2, \dots, \vec{y}_N) \quad (2.1)$$

In equation (2.1) the spatial coordinates of protons are labeled with x and the spatial coordinates of the neutrons are labeled with y . Only a single time coordinate is required, since we are in the non-relativistic region. The full Schrodinger equation that has to be solved now reads:

$$\left(\sum_{i=1}^Z \frac{\hbar^2}{2M_P} \vec{\nabla}_{\vec{x}_i}^2 + \sum_{j=1}^N \frac{\hbar^2}{2M_N} \vec{\nabla}_{\vec{y}_j}^2 \right) \cdot \Psi + \mathbf{V} \cdot \Psi = i\hbar \frac{\partial \Psi}{\partial t} \quad (2.2)$$

Meaning that the Hamiltonian reads:

$$\mathbf{H}_0 = \sum_{i=1}^Z \frac{\hbar^2}{2M_P} \vec{\nabla}_{\vec{x}_i}^2 + \sum_{j=1}^N \frac{\hbar^2}{2M_N} \vec{\nabla}_{\vec{y}_j}^2 + \mathbf{V} \quad (2.3)$$

The potential energy \mathbf{V} in equation (2.2) is a sum of the potentials of all forces working on the nuclei (like the standard model):

$$\mathbf{V} = V_{EM} + V_{Weak} + V_{Strong} \quad (2.4)$$

Since velocities of the nuclei are non-relativistic, the electromagnetic potential is approximately a coulomb potential:

$$V_{EM} \approx + \sum_{i,j=1}^Z \frac{1}{4\pi\epsilon_0} \cdot \frac{e^2}{\|\vec{x}_i - \vec{x}_j\|} \quad (2.5)$$

The weak force is (due to its limited range) of very little importance in the overall potential, so it can be neglected:

$$V_{Weak} \approx 0 \quad (2.6)$$

The strong potential is quite difficult to model. A crude modeling however would be to use the pion exchange model in combination with the Yukawa potential. We know that

$$\alpha_{EM} = \frac{e^2}{4\pi\epsilon_0\hbar c}$$

so we could write a single Coulomb potential like:

$$\frac{1}{4\pi\epsilon_0} \cdot \frac{e^2}{r} = \alpha_{EM} \cdot \frac{\hbar c}{r}$$

Meaning that a single Strong Yukawa potential based on the pion exchange model would be:

$$-\hbar c \alpha_{Strong} \cdot \frac{e^{-k \cdot m_\pi \cdot r}}{r}$$

The minus sign reflects that we have an attraction instead of repulsion. Hence the full strong potential will be:

$$V_{Strong} \approx - \sum_{i,j=1}^Z \hbar c \alpha_{Strong} \cdot \frac{e^{-k \cdot m_\pi \cdot \|\vec{x}_i - \vec{x}_j\|}}{\|\vec{x}_i - \vec{x}_j\|} - \sum_{i,j=1}^N \hbar c \alpha_{Strong} \cdot \frac{e^{-k \cdot m_\pi \cdot \|\vec{y}_i - \vec{y}_j\|}}{\|\vec{y}_i - \vec{y}_j\|} - \sum_{i=1}^Z \sum_{j=1}^N \hbar c \alpha_{Strong} \cdot \frac{e^{-k \cdot m_\pi \cdot \|\vec{x}_i - \vec{y}_j\|}}{\|\vec{x}_i - \vec{y}_j\|} \quad (2.7)$$

With $k = 2.82 \cdot 10^{42} \text{ kg}^{-1} \text{ m}^{-1}$ [5]. Better approximations for the strong potential are for example the Paris potential, the Argonne AV18 potential, the CD-Bonn potential and the Nijmegen potentials. Notice that such models might have an explicit time-dependence in the potential.

Solving the full Schrodinger equation (including the time dependence) would mean that we have to solve a $3A + 1$ dimensional PDE. This is beyond our computational power for most nuclei, so we use the so-called Mean Field approximation: we solve the Schrodinger equation for a single nucleon moving in a time-averaged potential that is generated by all the other nucleons. This potential generated by all the other nucleons is called the Mean Field potential. This potential is somehow obtained from averaging the potentials defined above.

It is assumed that all the nucleons feel the same Mean Field potential, which is treated as a external time-independent potential. It is assumed that there are no interactions between the nuclei, all they feel is the external Mean Field potential. Since the Mean Field potential is time independent, the time dependence of the wave-funcntion factorizes out. Since the interactions are also switched off now, equation (2.1) can then be written as:

$$\Psi = \psi(t) \cdot \prod_{i=1}^Z \psi_{x,i}(\vec{x}_i) \cdot \prod_{j=1}^N \psi_{y,j}(\vec{y}_j) \quad (2.8)$$

The full potential \mathbf{V} is a function of all the spatial vectors and time (since the nucleons move around in the nucleus). This time dependence was NOT taken along in the crude models for the potential that we wrote down, but could be taken along in more sophisticated models for the potential. In general we have:

$$\mathbf{V} = \mathbf{V}(t, \vec{x}_1, \vec{x}_2, \dots, \vec{x}_Z, \vec{y}_1, \vec{y}_2, \dots, \vec{y}_N) \quad (2.9)$$

Then since $|\psi_{x,i}(\vec{x})|^2$ is the probability density that proton i is at position \vec{x} , the time averaged mean field potential for proton p is given by:

$$\bar{V}(\vec{x}_p) = \frac{1}{T} \int_0^T dt \prod_{i=1, i \neq p}^Z \left(\int_{\mathbb{R}^3} d^3 x_i \right) \prod_{j=1}^N \left(\int_{\mathbb{R}^3} d^3 y_j \right) \mathbf{V}(t, \vec{x}_1, \vec{x}_2, \dots, \vec{x}_Z, \vec{y}_1, \vec{y}_2, \dots, \vec{y}_N) \prod_{i=1, i \neq p}^Z |\psi_{x,i}(\vec{x}_i)|^2 \prod_{j=1}^N |\psi_{y,j}(\vec{y}_j)|^2 \quad (2.10)$$

For the neutrons a similar expression holds. Obviously the mean field potential is identical for different protons and also for different neutrons. There is however a difference between neutrons and protons, since for the neutrons the coulomb potential is integrated out completely while a Coulomb term remains for the protons (as it should be).

Now if the potential \mathbf{V} in equation (2.2) is then replaced by the mean field potential for a single nucleon \bar{V} , Then all factors in equation (2.9) are divided out, except the factor for which the mean field potential was designed. Since the Mean Field potential is no longer time dependent, the time dependence also factorizes out and the time-independent Schrodinger equation we are left with reads:

$$\frac{\hbar^2}{2M_P} \vec{\nabla}_{\vec{x}}^2 \psi_{proton}(\vec{x}) + \bar{V}_{proton}(\vec{x}) \cdot \psi_{proton}(\vec{x}) = E \cdot \psi_{proton}(\vec{x}) \quad (2.11)$$

For the neutrons a similar expression holds. There is however one major problem with this approach. In order to solve for $\psi_{proton}(\vec{x})$ or $\psi_{neutron}(\vec{x})$, \bar{V} has to be known first, but that can only be computed once these wave functions are known already (see equation (2.10)). So the real challenge of performing a mean field approximation is to find a solution to this problem, usually again by resorting to approximations. If no approximations would be used in evaluating equation (2.10), the answer we get would still be incorrect, because the Mean Field potential is an expectation value of the real world, not the real world itself. This is what causes that there are no interactions between the nucleons. Approximations are made again because the evaluation of equation (2.10) is far from trivial.

Typical shapes of the radial dependence of the Mean Field potentials are given by figure 2.1.

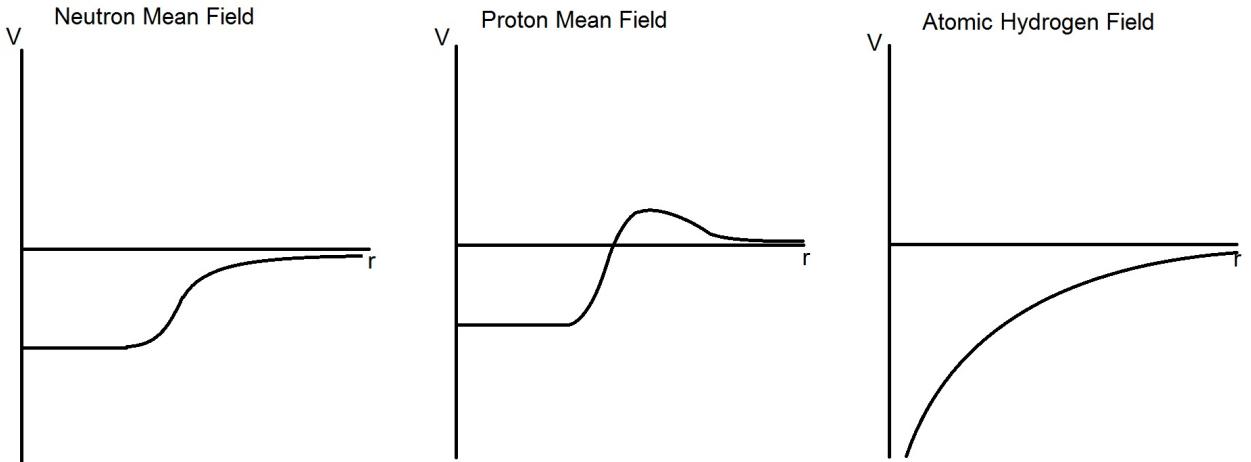


Figure 2.1: Typical Mean Field Potentials

There is usually no angular dependence for the Mean Field potentials. Now solving the Schrodinger equation is done in precisely the same way as for the hydrogen atom. The potential shape is however different. Due to the specific shape of the Atomic Hydrogen potential, one obtains a shell quantum number $n = 1, 2, \dots$, an orbital momentum number $l = 0, 1, 2, \dots, n - 1$ and a magnetic quantum number $m = -l, -(l - 1), \dots, -1, 0, 1, \dots, l - 1, l$.

However, the nuclear Mean Field potentials are different: they do not have a singularity at $r = 0$. Therefore the same quantum numbers are obtained, but the orbital quantum number is now no longer limited to $n - 1$ but can take any value. The energy E is again dependent only on n and l , but not on m , meaning that (including spin) the multiplicity of each energy level equals $2(2l + 1)$. For the actual solution procedure the Mean Field potentials are sometimes even approximated by a Harmonic oscillator potential or a box-potential.

The accuracy of the mean field approximation depends now on several factors:

1. The accuracy of the models used for V_{EM} , V_{weak} and V_{strong} .
2. The accuracy of the solution to the chicken-egg problem of obtaining the Mean Field potential. This means how accurate the actual mean field approximation is.
3. The Mean Field approximation is an expectation value of the real world, not the real world itself. This means that direct interactions between nuclei like collisions cannot be described by the Mean Field approximation. These interactions are also called correlations. We will get back to this later.

The model used in this section for V_{Strong} is very crude and finding a correct expression is far from trivial. The models for V_{EM} and V_{weak} are however not far from the truth. There is however one more aspect that should be included in V_{EM} : the spin-spin interactions. Due to the non-relativistic motion of the nuclei, all EM -phenomena are negligible except the Coulomb interaction (which we modeled) and the spin-spin interactions. The latter were not included, because spin itself was not included in the Schrodinger equation we used. This has a very specific reason which we will get back to in the next section.

For now, let us just give a picture of the energy levels obtained from a pure solution of equation (2.11). This will then give a pure shell-model for the nucleus and one can simply fill the levels one-by-one to obtain the Nuclear shell model in figure 2.2.

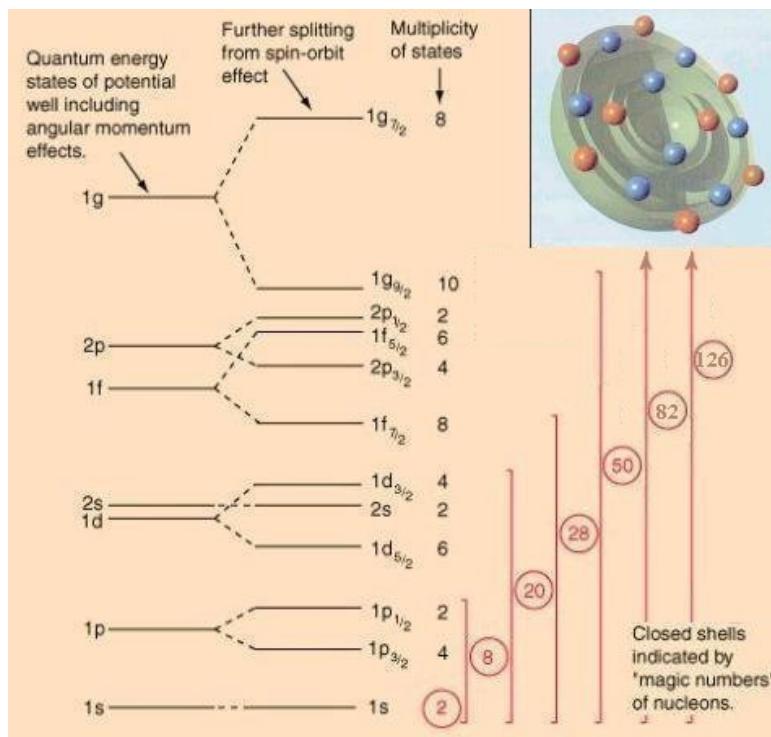


Figure 2.2: Energy levels from the Nuclear shell model

The most left series of levels is what is obtained from solving equation (2.11). The spin-orbit coupling and the magic numbers will be explained in subsequent sections. Notice that we get two shell models placed parallel to each other: one for the protons and one for the neutrons. The proton energy levels are all a bit higher than the neutron levels, due to figure 2.1, which is caused by the Coulomb interaction.

This is basically the full principle of the Shell model. The next subsections are dedicated to refining the shell model by taking spin, Pauli, spin-orbit coupling, correlations, etc. along in the model.

2.1.2 Including the nucleon spins into the shell-model

The Pauli exclusion principle and spin are NOT included in equation (2.11) [6], because they are not effects of classical Quantum Mechanics [3]. It is an effect of Relativistic Quantum Mechanics. Therefore they have to be inserted by hand. Inserting the spin will be discussed first.

By inserting the spin, the wave function that describes all nucleons at once will be a product of a spatial part and a spin part [7], [3]. The spatial part is a scalar function with time and several spatial vectors as arguments. The spin-part is an abstract time-dependent quantum state:

$$\Psi = \Psi_{spatial}(t, \vec{x}_1, \vec{x}_2, \dots, \vec{x}_Z, \vec{y}_1, \vec{y}_2, \dots, \vec{y}_N) \cdot \Psi_{spin}(t) \quad (2.12)$$

Each nucleon can have spin-up or spin-down. These states are represented by the abstract quantum states $|s, m_s\rangle$. To represent the spin-state of the entire nucleus, products of these states can be used. These product states form a complete basis, meaning that the most general spin-state of the nucleus is a linear combination of these product states:

$$\Psi_{spin}(t) = \sum_k c_k |x_{1,k}(t)\rangle \dots |x_{Z,k}(t)\rangle \cdot |y_{1,k}(t)\rangle \dots |y_{N,k}(t)\rangle \quad \sum_k |c_k|^2 = 1 \quad (2.13)$$

In equation (2.13), each $|x_{i,k}(t)\rangle$ is a spin-state $|s, m_s\rangle$. For each nucleon we know that $s = 1/2$. The magnetic quantum numbers might however be different ($m_s = \pm 1/2$), depending on which proton i is chosen, which term k is used and depending on the time t .

A spin-sensitive operator now consists of a mathematical combination of single-nucleon spin operators. A single-nucleon spin-operator only operates on the spin state of the nucleon it is designed for and leaves the other parts alone. One can simply specify the actions such an operator does on the state and specify the commutator relations of the operator. This is the usual way in Quantum Mechanics. For discrete quantum states like spin, one could however also choose a specific basis and write the single-nucleon quantum states and single-nucleon operators in matrix-vector form with respect to this basis. Usually the z -axis is chosen to project the spin on.

For a single-nucleon, the spin-operator is a vector operator: $\vec{S} = (\mathbf{S}_x \mathbf{S}_y \mathbf{S}_z)$. In the basis with respect to the z -axis, the spin operators \mathbf{S}_x , \mathbf{S}_y and \mathbf{S}_z are proportional to the pauli-matrices:

$$\mathbf{S}_x = \frac{\hbar}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \mathbf{S}_y = \frac{\hbar}{2} \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} \quad \mathbf{S}_z = \frac{\hbar}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2.14)$$

In the same basis, a single-nucleon quantum state $|s, m_s\rangle$ is represented by a 2-element column matrix:

$$|s, m_s\rangle = \begin{pmatrix} \psi_1(t) \\ \psi_2(t) \end{pmatrix} \quad |\psi_1(t)|^2 + |\psi_2(t)|^2 = 1 \quad \forall t \quad (2.15)$$

The elements of this matrix are pure complex numbers that might be time-dependent. This is how one can explicitly model the time-dependence of a spin-state. If the spin is projected out along a specific axis, the spin-state is described by an eigenvector of the Pauli-matrices ($s = \frac{1}{2}$ in all cases):

$$\begin{aligned} |s, +\frac{1}{2}\rangle_{x-axis} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} & |s, -\frac{1}{2}\rangle_{x-axis} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ |s, +\frac{1}{2}\rangle_{y-axis} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix} & |s, -\frac{1}{2}\rangle_{y-axis} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix} \\ |s, +\frac{1}{2}\rangle_{z-axis} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} & |s, -\frac{1}{2}\rangle_{z-axis} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned} \quad (2.16)$$

If the spin of single nucleons is added, one could obtain a quantum state of larger s that is described by column matrices of more than 2 elements. The spin-operators for such states are higher dimensional versions of the

Pauli-matrices. This is however something completely different than the product states of equation (2.13). The product of spin states remains an abstract mathematical product and one can only visualize a single nucleon spin-state at a time by using matrix notation. The product between spin states is NOT a matrix product.

The Hamiltonian of equation (2.3) now consists of a kinetic term and a potential term. The kinetic term is a purely spatial operator and leaves the spin-part of the wavefunction unchanged. The potential \mathbf{V} consists now of pure spatial terms and pure spin-terms:

$$\mathbf{H}_0 = \sum_{i=1}^Z \frac{\hbar^2}{2M_P} \vec{\nabla}_{\vec{x}_i}^2 + \sum_{j=1}^N \frac{\hbar^2}{2M_N} \vec{\nabla}_{\vec{y}_j}^2 + \mathbf{V} \quad (2.17)$$

$$\mathbf{V} = V_{EM} + V_{weak} + V_{Strong} \approx V_{coulomb} + V_{Strong} + V_{spin-spin}$$

$V_{coulomb}$ and V_{Strong} are pure spatial operators. They act by multiplying the wave-function with the potential function (just as usual). $V_{spin-spin}$ is a pure spin-operator and acts only on the spin-part of the wave-function.

If \vec{r} is the distance vector between two classical magnetic dipoles, the classical interaction energy between them is given by:

$$E = \frac{-\mu_0}{4\pi||\vec{r}||^3} \cdot (3(\vec{\mu}_1 \cdot \vec{r})(\vec{\mu}_2 \cdot \vec{r}) - \vec{\mu}_1 \cdot \vec{\mu}_2)$$

For nuclei, the magnetic moment is given by $\vec{\mu} = \frac{g_r}{\hbar} \mu_N \vec{S}$. Here g_r is a real number, the gyromagnetic ratio, which is known for protons and neutrons. $\mu_N = \frac{e\hbar}{2M_P}$ for both neutrons and protons. The difference is between them is then absorbed in different g_r . \vec{S} is the spin operator. Hence we obtain for $V_{spin-spin}$ the following expression:

$$\begin{aligned} V_{spin-spin} = & \sum_{i,j=1, i < j}^Z \frac{-\mu_0 \mu_N^2 g_p^2}{4\pi \hbar^2 ||\vec{r}_{ij}||^3} \cdot (3(\vec{S}_{i,p} \cdot \vec{r}_{ij})(\vec{S}_{j,p} \cdot \vec{r}_{ij}) - \vec{S}_{i,p} \cdot \vec{S}_{j,p}) + \\ & \sum_{i,j=1, i < j}^N \frac{-\mu_0 \mu_N^2 g_n^2}{4\pi \hbar^2 ||\vec{r}_{ij}||^3} \cdot (3(\vec{S}_{i,n} \cdot \vec{r}_{ij})(\vec{S}_{j,n} \cdot \vec{r}_{ij}) - \vec{S}_{i,n} \cdot \vec{S}_{j,n}) + \\ & \sum_{i=1}^Z \sum_{j=1}^N \frac{-\mu_0 \mu_N^2 g_p g_n}{4\pi \hbar^2 ||\vec{r}_{ij}||^3} \cdot (3(\vec{S}_{i,p} \cdot \vec{r}_{ij})(\vec{S}_{j,n} \cdot \vec{r}_{ij}) - \vec{S}_{i,p} \cdot \vec{S}_{j,n}) \end{aligned} \quad (2.18)$$

The spin-operators in equation (2.18) are properly defined now, so the full Hamiltonian is now fully defined in terms of equation (2.17). Mathematically the spin state of the full wavefunction is also fully defined, making that the Schrodinger equation that has to be solved is now given by:

$$i\hbar \cdot \frac{\partial \Psi}{\partial t} = \mathbf{H}_0 \Psi \quad (2.19)$$

One could now apply the same Mean Field approximation discussed in the previous section. The only difference is that the potential is more complicated, because it depends on spin now. Still, because the Mean Field potential is a time-averaged potential, the time dependence will factor out of equation (2.19) nevertheless. The same time-independent Schrodinger equation results in the end (for both protons and neutrons):

$$\frac{\hbar^2}{2M_P} \vec{\nabla}_{\vec{x}}^2 \psi_{proton}(\vec{x}) + \bar{V}_{proton} \psi_{proton}(\vec{x}) = E \cdot \psi_{proton}(\vec{x}) \quad (2.20)$$

However, this time the wave function $\psi_{proton}(\vec{x})$ has a spatial part and a spin part in it. The Mean Field potential might also still have a spin-operator in it. The Mean Field potential for protons is now computed by:

$$\bar{V}(\vec{x}_p) = \frac{1}{T} \int_0^T dt \prod_{i=1, i \neq p}^Z \left(\int_{\mathbb{R}^3} d^3 x_i \right) \prod_{j=1}^N \left(\int_{\mathbb{R}^3} d^3 y_j \right) \sum_{m_s=\pm 1/2 \text{ } \forall \text{ protons but } p} \sum_{m_s=\pm 1/2 \text{ } \forall \text{ neutrons}} \quad (2.21)$$

$$\prod_{i=1, i \neq p}^Z \psi_{x,i}^*(\vec{x}_i) \prod_{j=1}^N \psi_{y,j}^*(\vec{y}_j) \mathbf{V}(t, \vec{x}_1, \vec{x}_2, \dots, \vec{x}_Z, \vec{y}_1, \vec{y}_2, \dots, \vec{y}_N) \prod_{i=1, i \neq p}^Z \psi_{x,i}(\vec{x}_i) \prod_{j=1}^N \psi_{y,j}(\vec{y}_j)$$

And for neutrons a similar expression holds. The spin-part of the expression can be evaluated in an abstract manner, or with respect to a particular basis.

An even better approximation of V_{EM} would be to not only include $V_{coulomb}$ and $V_{spin-spin}$, but to include the full magnetic interaction between the nucleons: $V_{EM} \approx V_{coulomb} + V_{\mu-\mu}$. $V_{\mu-\mu}$ has exactly the same expression as equation (2.18), but now all spin operators \vec{S} are replaced by full angular momentum operators $\vec{J} = \vec{L} + \vec{S}$. Then \vec{S} operates on the spin-part of the wave function and \vec{L} operates on the spatial part of the wave function. So this potential is then no longer just a multiplication for the spatial part. The gyromagnetic ratios are replaced by their full ratios: $g_p \rightarrow g_p^J$ and $g_n \rightarrow g_n^J$. For the rest the equation remains the same.

The relation between full gyromagnetic ratios and spin gyromagnetic ratios is given by

$$g_p^J = \frac{j(j+1) - s(s+1) + l(l+1)}{2j(j+1)} + g_p \cdot \frac{j(j+1) + s(s+1) - l(l+1)}{2j(j+1)} \quad (2.22)$$

Equation (2.22) is justified because orbital gyromagnetic ratios are equal to unity. Obviously these numbers can only be computed after the j and l of each nucleon is known. But in order to solve the Schrodinger equation, the operator $V_{\mu-\mu}$ must be fully defined. It depends however on the nucleons state through these numbers. Another loop in the definitions that must be solved by the Mean Field approximation.

Notice however that $V_{\mu-\mu}$ also introduces spin-orbit couplings in the potential: products of the operators \mathbf{L} and \mathbf{S} . Since $V_{\mu-\mu}$ would be the full magnetic interaction between all nucleons, this would be a way to include the spin-orbit coupling of section 2.1.4. It is however not the most elegant way (but a correct way). The best way is to keep using $V_{EM} \approx V_{Coulomb} + V_{spin-spin}$ and include the spin-orbit coupling as a first order correction to special relativity, see section 2.1.4. The \mathbf{L}^2 contributions in $V_{\mu-\mu}$ are second order contributions and can therefore be neglected.

2.1.3 Including the Pauli-principle in the shell model

Including the Pauli principle in the shell model is conceptually not very difficult. One first solves the Schrodinger equation (2.19) to obtain the wave function (2.12) without being concerned about the Pauli principle. Then one antisymmetrizes this wave function over all neutrons and over all protons adjusting the normalization. Note that this has to be done for both the spin and the spatial part in such a way that the full wave function is antisymmetric under the exchange of labels. Then the Pauli principle is satisfied.

If the wave function is a pure product state of single particle states, this antisymmetrization process is done by means of a determinant. In other cases, this antisymmetrization process is more complicated. In this case the antisymmetrizer operator for protons is defined as:

$$A_{protons} = \frac{1}{Z!} \sum_{\hat{P} \in S_Z} (-1)^{\pi_P} \hat{P} \quad (2.23)$$

In equation (2.23), S_Z is the group of all permutations on Z and \hat{P} is one of those permutations of the labels on the particles. π_P is the number of 2-cycles needed to compose P . Remember your group theory.

Hence one has to apply an antisymmetrizer operator for the protons and one for the neutrons to put in the Pauli principle.

For the Mean Field approximation the story is then exactly the same. One could solve equation (2.20) for a single proton and for a single neutron and then build a product state according to equation (2.9) and then antisymmetrize this equation. This will give a pure Mean Field Model with the Pauli principle taken along properly.

For the potential V_{Strong} we used a pure Yukawa potential that is purely attractive. More realistic nucleon-nucleon potentials have however long-range attraction and short-range repulsion. This repulsion is a consequence of heavier meson exchange (Σ -mesons, composed of two pion, etc.) that becomes possible on short ranges. Long-range is only pion exchange. Another possible view would be to attribute this short-range repulsion to the Pauli principle, but this is the wrong picture. Short-range repulsion between nucleons is a pure physical effect that originates from the particles that can be exchanged on that range. The Pauli principle should then only be taken along by antisymmetrizing the wave function and in no other way. Obviously it is the full wavefunction of spin and spatial part together that should be antisymmetrized with the antisymmetrizer.

With the spin and the Pauli principle inserted, the picture of figure 2.2 is still accurate. From solving the Schrodinger equation we obtain the left-most energy levels with corresponding multiplicities. Due to the spin, those multiplicities are doubled, giving the numbers in the figure. With the Pauli-principle one than obtains that the levels should be filled one-by-one, giving the nuclear Aufbauprinzip.

2.1.4 Inclusion of the Spin-orbit coupling in the shell model

One important effect should however still be included in the shell model: The spin-orbit coupling. The real reason for this phenomenon is that one should actually solve the Dirac equation for the protons and neutrons, not the Schrodinger equation. One would have a system of two coupled Dirac equations: one for the proton field and one for the neutron field. Remember that a single Dirac field describes all particles and antiparticles of one species at the same time. Now expand this system in terms of v/c , where v is the speed of a specific nucleon (Multi-dimensional Taylor expansion). The zero-th order term is now the Schrodinger equation (2.19). The first order terms are the spin-orbit couplings.

One can take these first order terms along in the Schrodinger equation by adding a term to the Hamiltonian in equation (2.19). One obtains:

$$i\hbar \cdot \frac{\partial \Psi}{\partial t} = \mathbf{H}\Psi \quad \mathbf{H} = \mathbf{H}_0 + \Delta H \quad (2.24)$$

In equation (2.24), \mathbf{H}_0 is the Hamiltonian defined in equation (2.17) (with spin-contributions) and ΔH is the contribution of the spin-orbit coupling: The first order Relativistic correction to the Schrodinger equation. This term turns out to be far from negligible [8]. Hence it must be incorporated into the model.

A more phenomenological explanation (but not precisely correct) would be that the spin-orbit coupling arises from interactions between the spin magnetic moment of a nucleon and the magnetic field generated by the motion of the other nuclei. The magnetic field seen by the specific nucleon is proportional to its orbital angular momentum and the magnetic moment of the nucleon is proportional to its spin. Hence the interaction energy would be $\Delta H = -\vec{\mu} \cdot \vec{B} \propto \vec{S} \cdot \vec{L}$. Summing over all nucleons then gives the total spin-orbit coupling of the full Hamiltonian. It will however be off by a scalar factor due to the phenomenological approach. A pure quantum mechanical approach does give the right answer.

For The Quantum Mechanical model of the Hydrogen atom one knows the spin-orbit coupling:

$$\Delta H = \frac{\mu_B}{\hbar m_e c^2} \frac{1}{r} \frac{\partial V}{\partial r} \cdot \mathbf{L} \cdot \mathbf{S} \quad V = -\frac{1}{4\pi\epsilon_0} \frac{Ze^2}{r}$$

This equation is based on the fact that the nucleus is at rest and the electron is moving, meaning that the electron will only feel an electrostatic central potential.

The nuclei are not at rest with respect to each other, so the potential a single nucleon sees, it not completely a central static potential. The velocities of the nuclei are however quite low. The spin-orbit coupling is already a first order effect in the velocity of the specific nucleon. Then incorporating the velocity of the other nucleons in the potential V would mean dealing with secondary order effects. For a first crude model of the spin-orbit coupling, it can be safely ignored. This gives the full spin-orbit coupling to be:

$$\Delta H = \sum_{i=1}^Z \sum_{j=1, j \neq i}^Z \frac{\mu_N}{\hbar M_p c^2} \cdot \vec{\nabla}_{\vec{x}_i} V_{EM} \cdot \vec{L}_i \cdot \vec{S}_j + \sum_{i=1}^Z \sum_{j=1}^N \frac{\mu_N \frac{M_p}{M_n}}{\hbar M_n c^2} \cdot \vec{\nabla}_{\vec{x}_i} V_{EM} \cdot \vec{L}_i \cdot \vec{S}_j \quad (2.25)$$

Notice that for nuclear spin-orbit coupling and electron spin-orbit coupling there is a sign difference, due to the different electrical charges (watch the formulas closely!)

This gives that \mathbf{H}_0 is defined by equation (2.17) with $V \approx V_{Coulomb} + V_{spin-spin} + V_{Strong}$. $V_{Coulomb}$ is given by equation (2.5), $V_{spin-spin}$ is given by equation (2.18) and V_{Strong} is a model of choice, for example equation (2.7). Together with equations (2.24) and (2.25) this gives us the full Schrodinger equation of the nucleus to solve.

If now mean field approximation is applied with the help of equation (2.21), a spin-orbit coupling term will remain in the final potential. The final mean-field potential will look like:

$$\bar{V} = V(\vec{x}) + \alpha \vec{S}^2 + \beta \vec{L} \cdot \vec{S} \quad (2.26)$$

The term proportional to α is the Mean Field result of $V_{spin-spin}$, a pure spin term in the potential. It contains only \vec{S}^2 because that is the only scalar one can make out of a vector operator \vec{S} that is invariant under all coordinate transformations. The term proportional to β is the Mean Field result of the spin-orbit coupling. The term $V(\vec{x})$ is the Mean Field result of the terms $V_{Coulomb}$ and V_{Strong} , which are both pure central spatial potentials. It is a pure multiplicative term that does not contain any operators. This is because one assumes a spherically symmetric distribution of nuclei over the nucleus. As such, the Mean Field potential will be spherically symmetric and there will be no term that contains a pure orbital angular momentum operator. Hence $V(\vec{x})$ will follow the shapes of figure 2.1. The size of α and β are given by the specific mathematical calculation used to perform the Mean Field approximation.

The Mean Field Schrodinger equation that now has to be solved is given by:

$$\frac{\hbar^2}{2M_p} \vec{\nabla}_{\vec{x}}^2 \psi_{proton}(\vec{x}) + \bar{V}_{proton} \psi_{proton}(\vec{x}) = E \cdot \psi_{proton}(\vec{x}) \quad (2.27)$$

And a similar equation holds for the neutron. Remember that $V(\vec{x})$, α and β differ slightly for the neutron and the proton (see figure 2.1).

Including the spin-orbit coupling in the Hamiltonian will cause a splitting in the energy levels. Without the spin-orbit coupling, the energy E depends on n and l , but not on m_l . Hence the multiplicity of each level is given by $2(2l+1)$. After adding the spin-orbit coupling, each level splits into $j = l + \frac{1}{2}$ and $j = l - \frac{1}{2}$. The level $j = l + \frac{1}{2}$ always has the lower energy. Both levels have a multiplicity of $2j+1$.

Using a harmonic oscillator Mean Field potential

$$\bar{V} = \frac{k}{2} \vec{x}^2 + \beta \vec{L} \cdot \vec{S}$$

The energy levels obtained are given by an update of figure ??: figure 2.3.

2.1.5 Features of the Nuclear Shell model

Energy levels from obtaining the solutions of equation (2.27) are labeled E_{nlj}^p for protons or E_{nlj}^n for neutrons. Here $n = 1, 2, 3, \dots$, $l = s, p, d, f, g, h, \dots$ and $j = 1/2, 3/2, \dots$. The corresponding quantum states are labeled $|nl_j, m_l\rangle_p |s, m_s\rangle$ or $|nl_j, m_l\rangle_n |s, m_s\rangle$. These states have a spatial part that can be represented with a wave function $|nl_j, m_l\rangle = \psi(\vec{x})$ and a spin-part $|s, m_s\rangle$ discussed in section 2.1.2. Each energy state can hold $2j+1$ nucleons. A full quantum state of the nucleon is now build from a properly antisymmetrized product state of these single-particle states.

The energy levels are now filled from below with nucleons. One has a separate energy scheme for protons and for neutrons. The proton energy levels are a bit higher due to their Coulomb repulsion (see figure 2.1). The highest filled energy level is the Fermi-energy from Statistical Mechanics. One has a separate Fermi Energy for protons and neutrons. This is now the *ground* state of the nucleus. Excited states are discussed later.

Big gaps in the energy level structure indicate the so-called magic numbers. If a nucleus is filled exactly up to this gap, it is particularly stable, while adding another nucleon would cost a lot more energy and hence

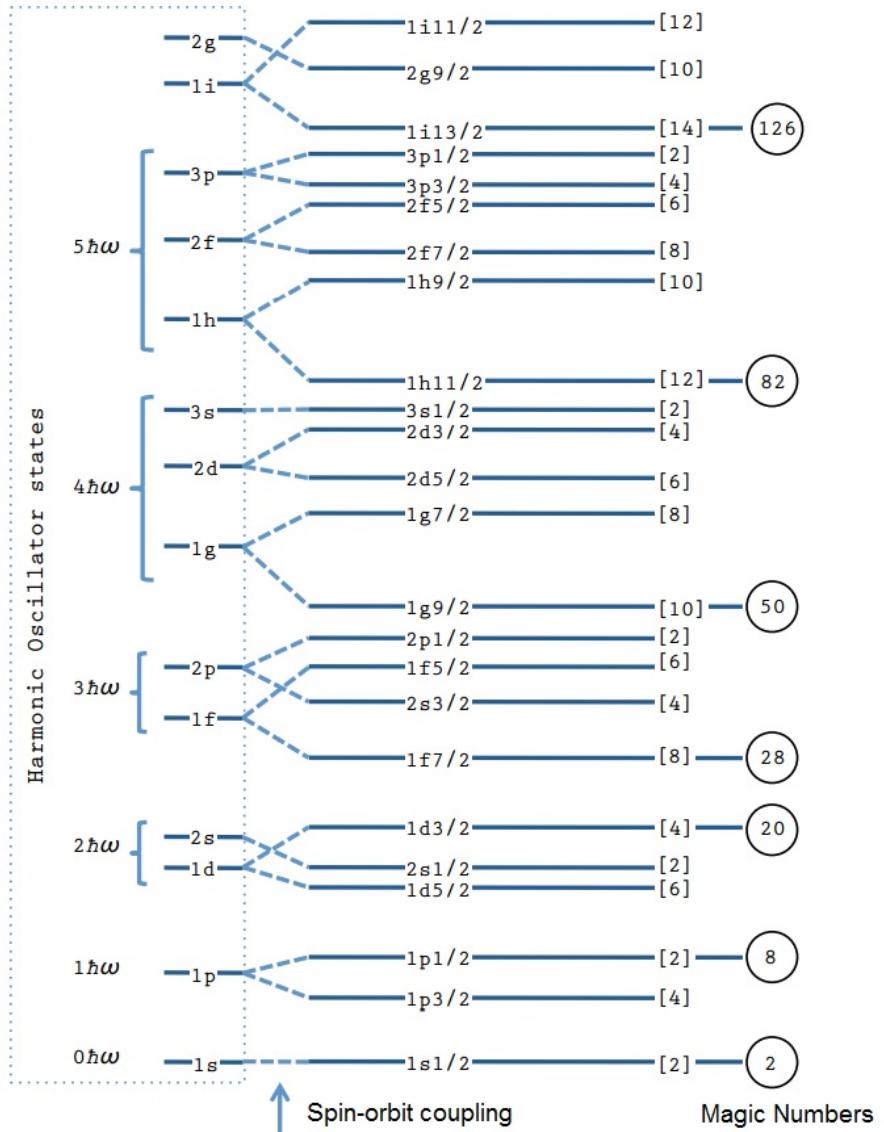


Figure 2.3: Energy levels from the Harmonic Oscillator Nuclear shell model with spin-orbit coupling.

gives instability. These magic numbers indicate the nuclear *shells*. Energy levels with multiplicity now give the nuclear *orbitals*, just like for atoms. The orbitals are however difficult to visualize. An example of a worked out nuclear shell model is for ^{16}O :

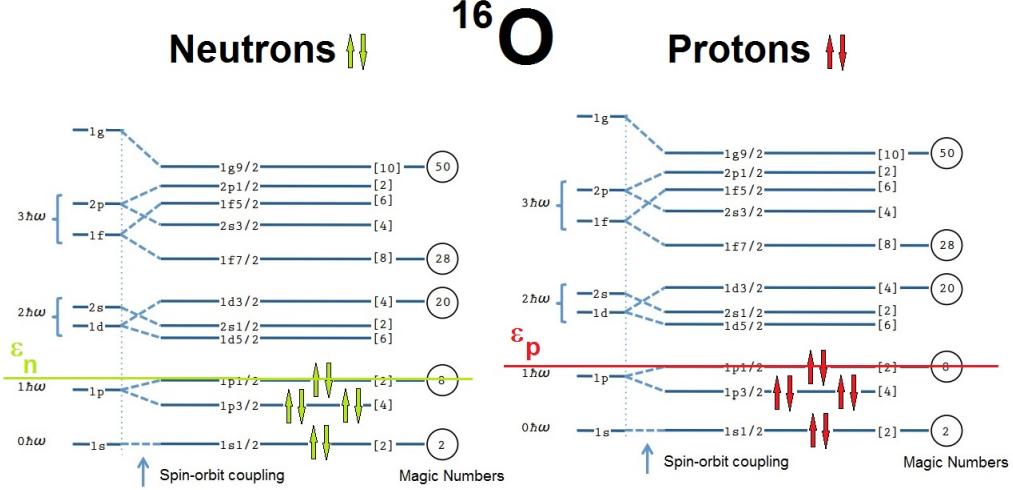


Figure 2.4: Harmonic Oscillator Shell Model for ^{16}O .

^{16}O is a so-called doubly-magic nucleus, because both Fermi-energies are at a magic number. A single magic nucleus only has one of its Fermi-energies at a magic number. Magic numbers favour stability of a nucleus but do not give a guarantee for it, nuclear stability is governed by more than only magic numbers.

The spectroscopic factor of a single-particle proton state labeled $E_{nl_j}^p$ of a nucleus with Z protons and N neutrons is defined as [2]:

$$S_{Z-1}^Z(nl_j) = |\langle N, Z-1 | a_{nl_j}^p | N, Z \rangle|^2 \quad (2.28)$$

In equation (2.28), $|N, Z\rangle$ is the full nuclear wave function (2.12) of a nucleus with N neutrons and Z protons. A similar statement is true for $|N, Z-1\rangle$. The operator $a_{nl_j}^p$ annihilates a proton in the energy state $E_{nl_j}^p$. This operator cannot distinguish between multiple protons that might populate this level, since these protons are indistinguishable; it simply annihilates one of them. This annihilation operator is defined in a normalized way, such that $a_{nl_j}^p |N, Z\rangle$ is again a valid wave function.

Then to compute the bracket, one first operates with $a_{nl_j}^p$ on $|N, Z\rangle$, then computes the inner product with $\langle N, Z-1|$ and finally integrates over all \Re^3 space and sums over all spin states of the nucleus. Then we compute the absolute square. Otherwise interference between different states of the nucleus is neglected.

If the specific Mean Field energy level $E_{nl_j}^p$ is populated with at least one proton, obviously one obtains that $a_{nl_j}^p |N, Z\rangle = |N, Z-1\rangle$, giving that

$$S_{Z-1}^Z(nl_j) = |\langle N, Z-1 | a_{nl_j}^p | N, Z \rangle|^2 = |\langle N, Z-1 | N, Z-1 \rangle|^2 = 1$$

Which is just the normalization of a quantum state. If there are however no protons available in the specific energy state, then $a_{nl_j}^p |N, Z\rangle = 0 \cdot |N, Z-1\rangle$, giving that

$$S_{Z-1}^Z(nl_j) = |\langle N, Z-1 | a_{nl_j}^p | N, Z \rangle|^2 = |\langle N, Z-1 | 0 | N, Z-1 \rangle|^2 = 0$$

Which is also trivial.

If a pure shell model is assumed for the nucleus according to figure 2.4, then each energy level $E_{nl_j}^p$ either contains one or more protons, or it does not. Hence according to a pure shell model, each spectroscopic factor is either zero or unity, depending whether there is at least one proton present in the energy level, or not. In the ground state of a nucleus, all spectroscopic factors below the Fermi energy are unity and above they are zero.

For a real nucleus, the shell model is however not exact. Still, the quantum states of the shell model provide a mathematically complete basis. Hence the protons are in superpositions of different energy states. In this case,

the spectroscopic factors become more interesting: the mathematical definition now directly translates into that the spectroscopic factor is the probability that a given energy level is populated.

This interpretation of a spectroscopic factor being the probability of a state that is occupied is very important to remember.

Other Features of the Shell model also need to be discussed. The Shell model assumes no interactions between the nucleons, which is obviously an incorrect assumption. In reality there are collision and interactions between the nucleons. This results in a number of deviations between the real world and the shell model. We will briefly discuss them.

At first the number of collisions and interactions becomes stronger when one goes deeper into the nucleus [2]. This causes the Shell model to be quite accurate for nucleons in the higher energy levels, but inaccurate for the nucleons in the lower energy levels: the nucleons in the lower energy levels do not occupy a single state, but are distributed over multiple states. One says that their wave function is fragmented over different energy states. The spectroscopic factors might therefore deviate from unity significantly. By their definition however, they can never exceed unity.

Second, the Shell model is quite accurate for situations like figure 2.4, where the nucleus contains equal numbers of protons and neutrons. If there is however a large difference between the numbers N and Z , take for example ^{26}O , the Shell model would look like figure 2.5.

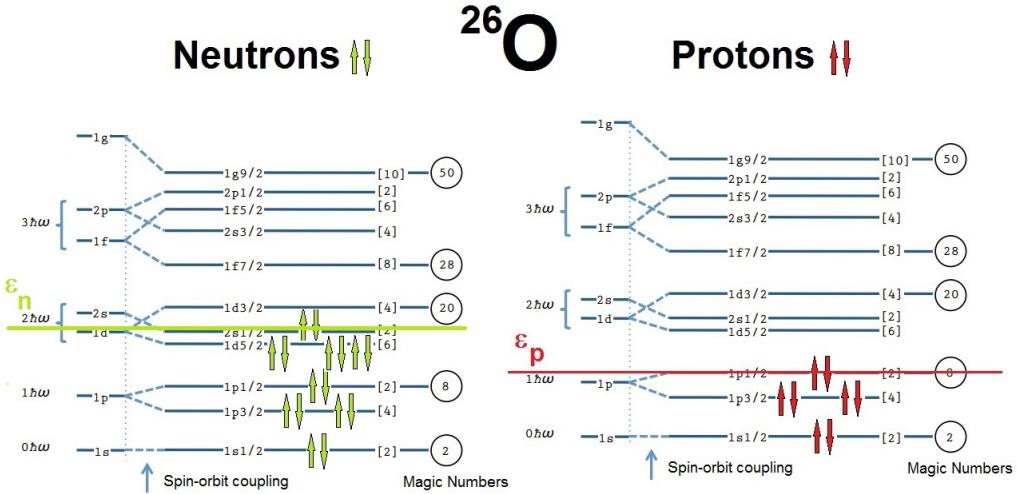


Figure 2.5: Harmonic Oscillator Shell Model for ^{26}O .

This large difference in Fermi-energies between the protons and neutrons causes the real world to deviate from the shell model and alters the structure of energy levels. Neutron-rich nuclei have a so-called 'halo' of neutrons, an outer shell composed mostly of neutrons. Neutron-poor nuclei on the other hand have their neutrons usually bound deeply into the nucleus to minimize the Coulomb-repulsion between the protons. Neutron-rich nuclei can be explained better if the Mean Field approximation assumes neutron-clusters instead of individual neutrons. This asymmetric behavior is also observed in the semi-empirical mass formula:

$$M_{nucleus} = N \cdot M_n + Z \cdot M_p - \frac{E_{bind}}{c^2} \quad (2.29)$$

$$E_{bind} = a_V \cdot A - a_S \cdot A^{2/3} - a_C \cdot \frac{Z^2}{A^{1/3}} - a_A \cdot \frac{N-Z}{A} + \delta(A, Z)$$

The volume term is due to the Strong force. Not each nucleon interacts with each other, due to the limited range of the Strong force. They only interact with their nearest neighbors. Hence, it is proportional to A , not A^2 . The binding energy between individual nucleons is of the order of 40 MeV. But due to the Pauli principle, not all nucleons can be in the lowest energy state. Given the nucleus is a dense Fermi-gas, basic statistical mechanics prescribe a kinetic energy of approximately $\frac{3}{5} A \epsilon_F$. Estimates give $\epsilon_F \approx 28$ MeV, giving that $a_V \approx 17$ MeV.

The surface term is a correction to the volume term, because nucleons on the surface cannot interact with as many neighbors as nucleons inside the nucleus. $A^{2/3}$ gives that this term is proportional to the surface. The coulomb term is obviously proportional to Z^2/r since Z is the number of protons. $r \propto A^{1/3}$, which explains this

term. The asymmetry term is caused by the effect just described. The effect can be estimated by describing the nucleus again as a Fermi gas. Finally, the pairing term $\delta(A, Z)$ is given by

$$\delta(Z, A) = \begin{cases} +\delta_0 & N, Z \text{ both even} \\ 0 & A = \text{odd} \\ -\delta_0 & N, Z \text{ both odd} \end{cases}$$

With $\delta_0 = a_P/A^{1/2}$. It describes the effect of spin-pairings in energy levels. The $A^{-1/2}$ dependence follows from liquid drop models and is no easy task.

Hence the Shell model is very good in describing nucleons in the outer part of the nucleus (high energy levels), usually for non-exotic nuclei. For nucleons deeper into the nucleus or for exotic nuclei, other models are more successful, like dense-gas models (with many interactions, precisely what is important for deeply-bound nucleons) or the liquid-drop model. In order to make the Shell model more successful, 2-body interactions (NN -forces) and sometimes 3-body interactions ($3N$ -forces) are added to the model. These interactions are also called correlations. This will be discussed later on.

One can plot Z on the y -axis and N on the x -axis to obtain the cart of existing nuclei. The regions where the binding energy goes from positive to negative are known as the drip lines. The proton drip line is where there are too few neutrons to create a nucleus and the neutron drip line is where there are too many neutrons to create a nucleus. Different models predict different drip-lines, making the true neutron drip-lines a bit uncertain. A picture is given in figure 2.6.

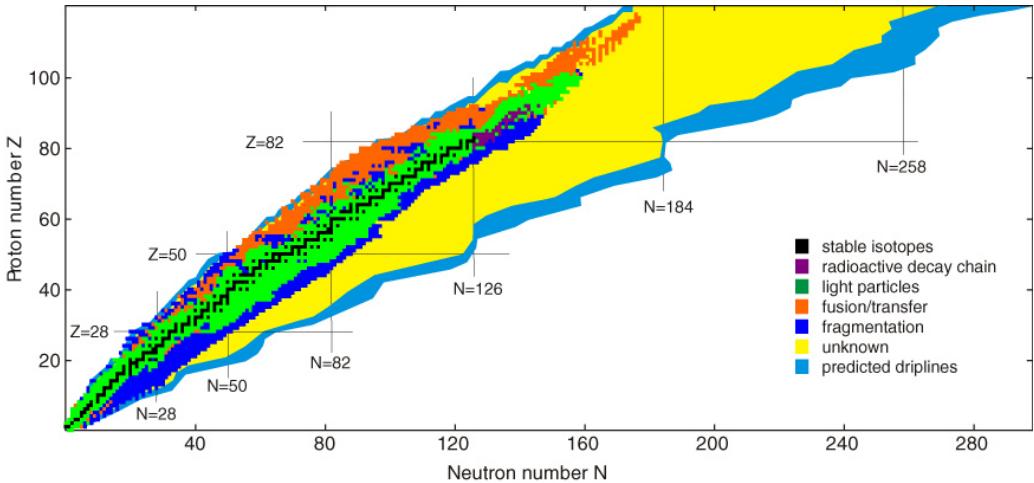


Figure 2.6: Nuclear cart with driplines.

The various ways a nucleus can decay are α -decay, β -decay, Fission, vaporisation, multifragmentation and spallation. γ -decay is only a transition between energy states of a nucleus. Gamow-Teller decay is when the beta-particle and the neutrino have parallel spins, meaning that the decaying nucleon must undergo a spin-flip. Fermi decay is the opposite case.

2.2 Syuomo PHD

2.3 Najafi PHD: Quasi-Free Proton and Neutron Knockout Reactions in ^{20}O

2.3.1 Basic concepts

In this PHD nuclei are shot with 3 different projectiles: a proton, a neutron or an electron. With this projectile, a nucleon (either a proton or a neutron) is knocked out of the nucleus. At first we will discuss some terminology: an exclusive reaction is when the full kinematics of the reactions can be reconstructed. An inclusive reaction is when one has to integrate or sum over some states due to the lack of kinematic information. Hence exclusive

reactions are most favourable. A peripheral reaction is when a nucleon is knocked out from the outer shell of the nucleus.

Electron projectiles can only be used to remove protons from the nucleus [2], since neutrons are uncharged. They can however be used to knock a proton out of any place of the nucleus, because inside the nucleus, their mean-free path is much larger than the nucleus. Hence there will be (approximately) one collision at a time. Hadron projectiles must however be used only for peripheral reactions, because their mean free path is much smaller than the nucleus. Multiple interactions between nucleus and projectile are extremely difficult to analyze, so a peripheral reaction is the only way to prevent multiple scattering.

The cross-section of such a knockout reaction (for knocking out a proton) is given by [2]:

$$\sigma_{nlj} = \left(\frac{A-1}{A} \right)^2 \cdot S_{Z-1}^Z(nl_j) \cdot \sigma_{\text{single particle}} \quad (2.30)$$

It reads that the nuclear cross section of removing a proton from the nl_j -state is a form factor times the spectroscopic factor of that state (the probability that it is occupied) times the single-particle cross-section. This single-particle cross section is the pure cross-section of a collision between the proton that should be knocked out and the projectile. This cross-section does depend on the energy-state of the proton, hence a non-interacting proton in the Mean Field potential (with the nl_j -energy) should be used. For a neutron a similar expression holds. **Note that there is a Clebs Gordan isospin coupling constant somewhere too, but I do not understand it....**

For such a knockout reaction, one denotes the 4-momentum of the projectile before the collision by $p_{in} = (E_{in}, \vec{k}_{in})$. The 4-momentum of the projectile after the collision is denoted by $p_{out} = (E_{out}, \vec{k}_{out})$. The energy of the knocked out nucleon and the nucleus after the collision (the energy transfer) is denoted by $p = p_{in} - p_{out} = (\omega, \vec{q})$. This momentum is distributed between the knocked out nucleon and the remaining daughter nucleus. The 4-momentum of the knocked out nucleon is denoted by $p_{nuc} = (T_p, \vec{p})$. The remaining 4-momentum of the daughter nucleus is then given by $p_{rec} = (E_{exc}, \vec{q}_{rec})$. One can now plot either E_{exc} or $\|\vec{q}_{rec}\|$ versus counts (which represents cross-section). After correcting for experimental aspects like background or acceptance and systematic errors, the pure shell model predicts the following pictures for a proton knockout reaction on ^{16}O (similar for neutron knockout reactions):

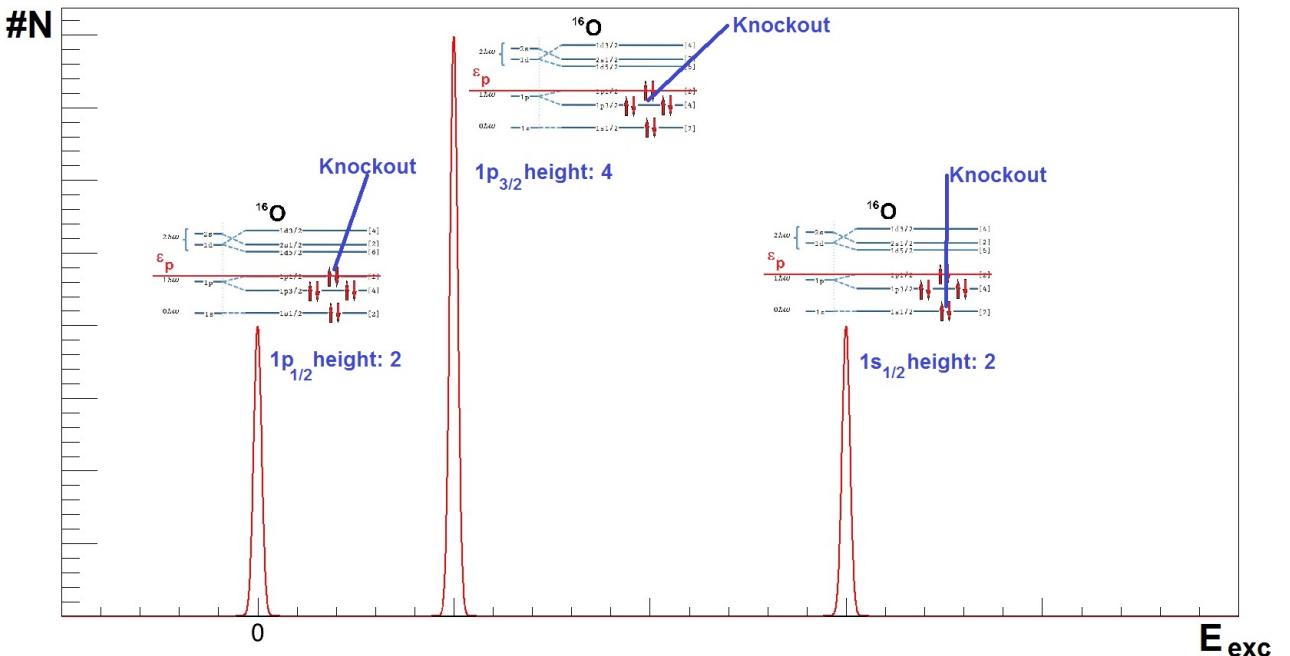


Figure 2.7: Excitation energy of daughter nucleus from ^{16}O after proton knockout reaction.

Notice that one must be careful if the knocked out nucleon and the projectile are the same particle. For this analysis one must know which is the projectile and which is the knocked out particle. Also notice that uncharged

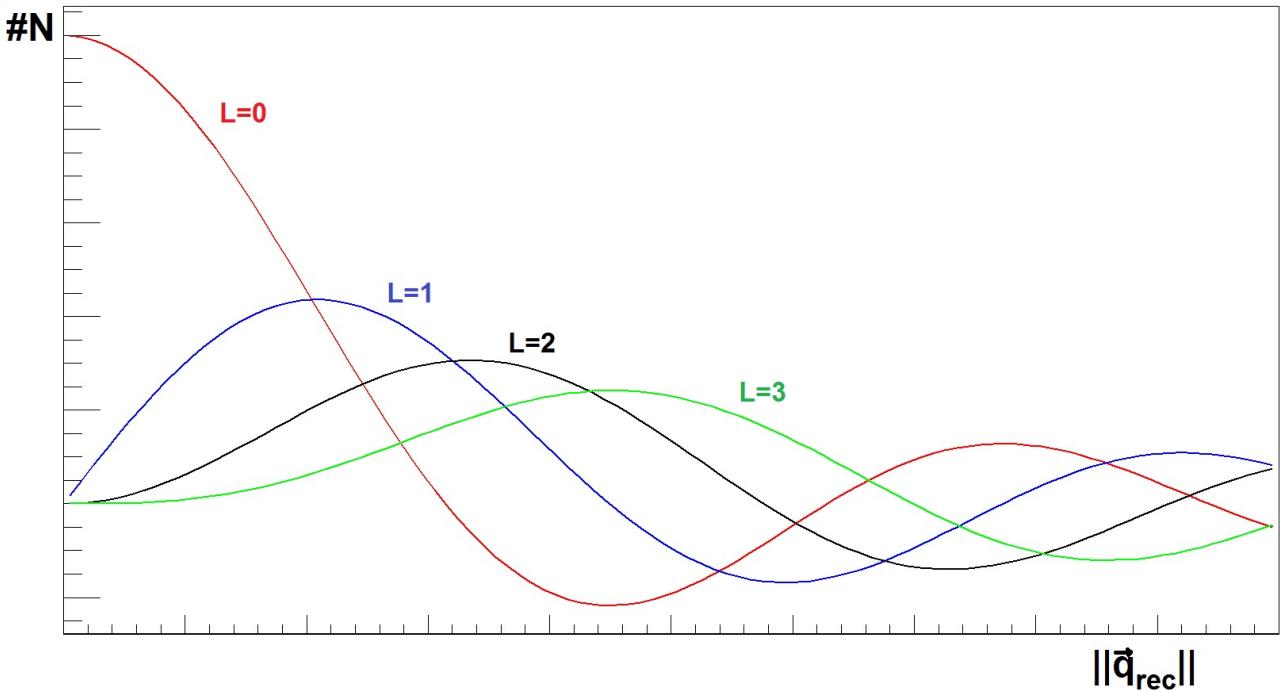


Figure 2.8: Angular momentum profile of daughter nucleus from ^{16}O after proton knockout reaction.

particles are difficult to reconstruct experimentally. In figure 2.7, we talk about the excitation energy of the *daughter* nucleus. Hence if we knock out one of the upper most nucleons, the daughter remains in the ground state, so the upper shell gives a peak at zero. More deeper bound nucleons give then a peak at higher values of E_{exc} .

Notice also that for light nuclei the Shell structure is clear, so one knows exactly which peak corresponds to which state. For heavier unexplored nuclei the shell structure is however not always clear, meaning that for example the $1p_{1/2}$ and $1p_{3/2}$ might be reversed. Due to fragmentation (explained in a minute), one cannot use the amplitudes to distinguish these peaks (there will still be two peaks in the spectrum). In such cases figure 2.8 can still unambiguously determine l . From the energy spectrum n is usually also clear. The spin-orbit coupling will then be visible in figure 2.7, but one will *not* know which peak corresponds to which j (read: which spin). In such cases one needs additional information to uniquely determine j . This can be done by working with polarized nuclei or by directly measuring the spin of the knocked out nucleon. Polarization will however never be 100 % and there are transfer coefficients between the spin of the nucleon inside the nucleus and the spin of the nucleon after the knockout reaction. Hence such methods will never give a 100 % answer.

2.3.2 Fragmentation effects on the spectra

In the real world the picture of figures 2.7 and 2.8 are quite different from the ideal case of the Shell model. Figure 2.8 can still unambiguously determine l , but a certain event selection will most likely always give a superposition of the functions, since these event selections can correspond to different energy levels. The task is then to figure out which events belong to which momentum function.

For the figure 2.7 the story is however quite different. The wave functions are fragmented over different energy states. This effect is most pronounced for the lowest energy levels. This causes that the peaks in figure ??2.7 are reduced in height or *quenched*. This fragmentation is shown in figure 2.9.

If now a proton is knocked out from, for example, the $1s_{1/2}$ shell (read: the large fragment in the $1s_{1/2}$ -shell is hit), two things can happen. Either the proton simply leaves and we get the same peak as before, or the small fragments of the wave function provide a coupling to other protons in the nucleus that are then *also* excited (together with knocking out the proton). After all, fragmentation is caused by correlations (nucleon-nucleon interactions). This then gives a highly excited state of the nucleus. Hence, fragmentation provides additional peaks at high excitation energies in the spectrum. Since there are many highly excited states possible, a continuum of small peaks at high E_{exc} is obtained. See figure 2.10.

16 O Proton fragmentation

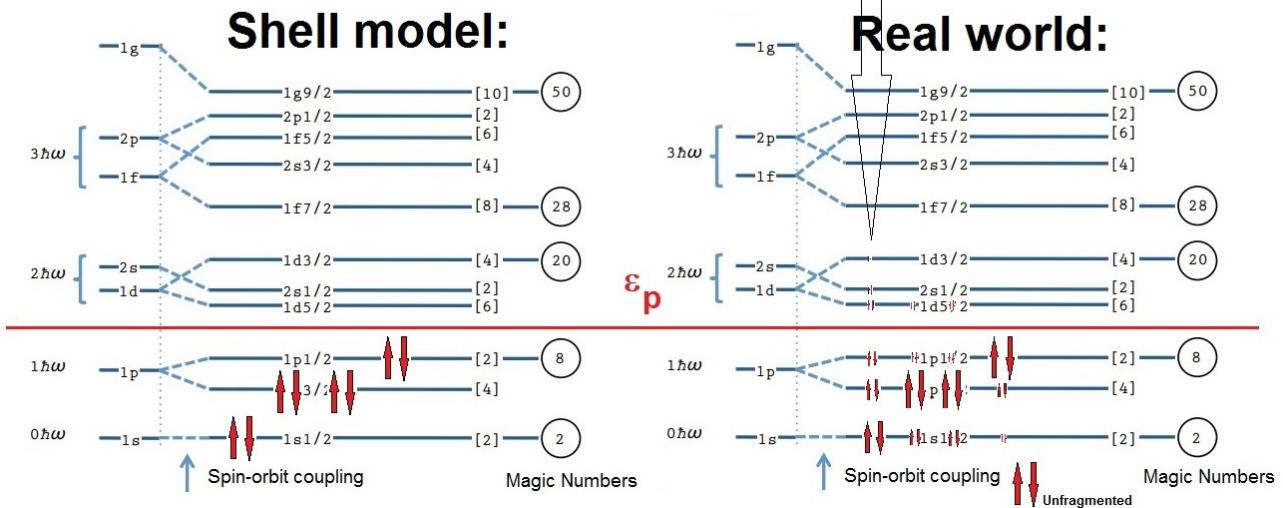


Figure 2.9: Proton wave function fragmentation of ^{16}O .

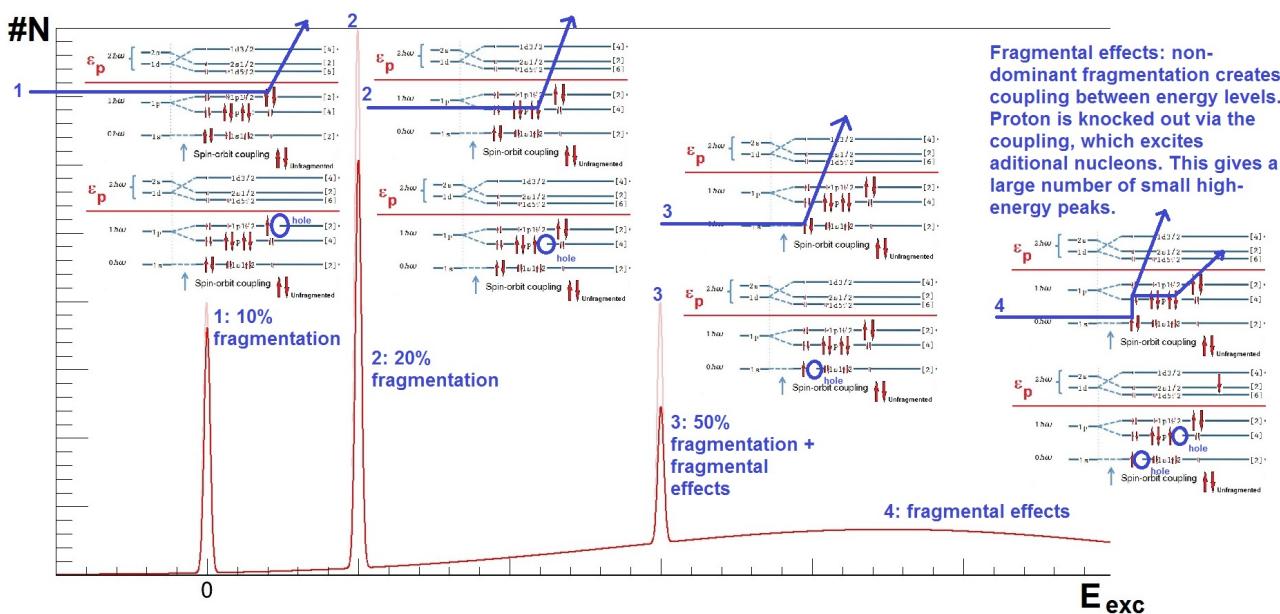


Figure 2.10: Excitation energy of daughter nucleus from ^{16}O after proton knockout reaction.

If one places an event selection criterion on the energy to specifically look at the fragmentation part of figure 2.10 and then makes a plot like figure 2.8, one can determine again l . Hence one can determine the Shell model energy level from which the fragmentation originates. If the plot gives $l = 0$, then one knows that the fragmentation originates from an s -state. The same for $l = 1$, etc. One might however also find a superposition. Most of the time the $1s_{1/2}$ -state will be fragmented the most. This is because fragmentation is the result of correlations, that are stronger when one goes into the nucleus deeper.

2.3.3 Experimental effects and detector principles

Different detector components are tested in-beam at KVI in chapter 3. These components are DSSD (Double-sided Silicon Strip Detectors, see your Master Thesis), Si(Li) (lithium-drifted silicon detectors) and CsI Scintillators. These components were arranged into detector arrays that are tested in-beam and compared to Monte-Carlo data. Extensive analysis and calibration was performed. Keep the Bethe-Bloch formula from your proton therapy research in mind, since it also applies to particles traveling through a detector!

Chapter 4 is about the S393-experiment carried out with the FRS at GSI. To explain the principle of the FRS, we will discuss the Super-FRS which is more relevant to us. The principles are the same, but the Super-FRS is more versatile and bigger, see figure 2.11.

The Super-FRS at FAIR

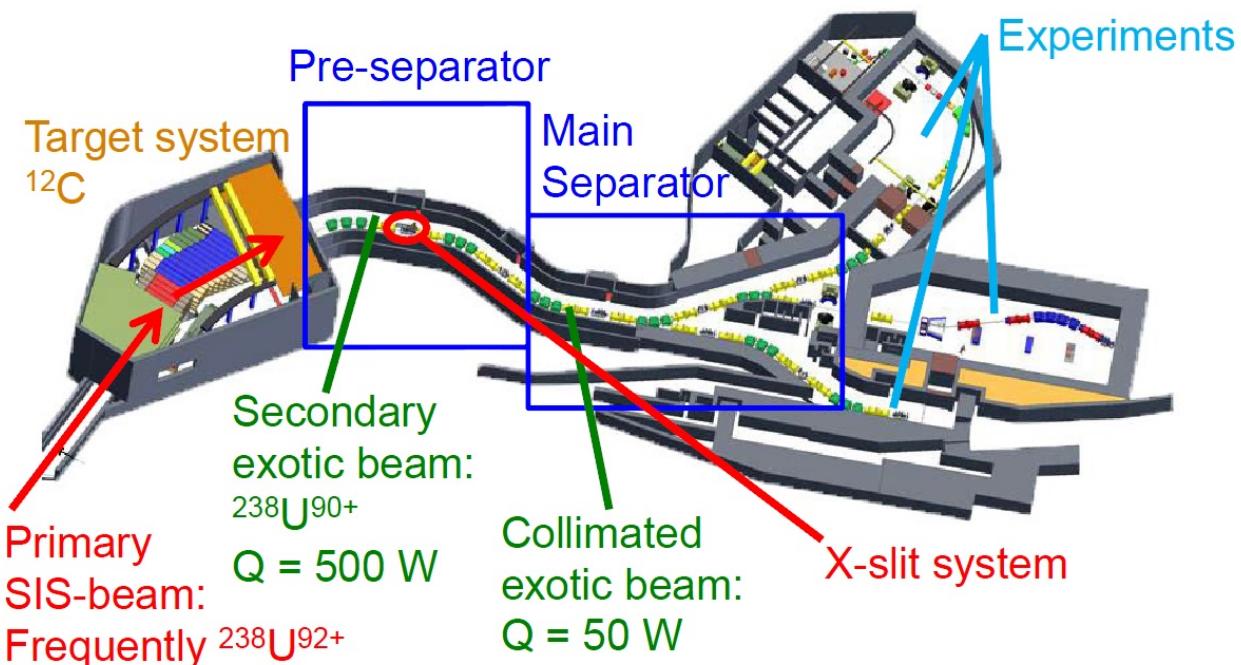
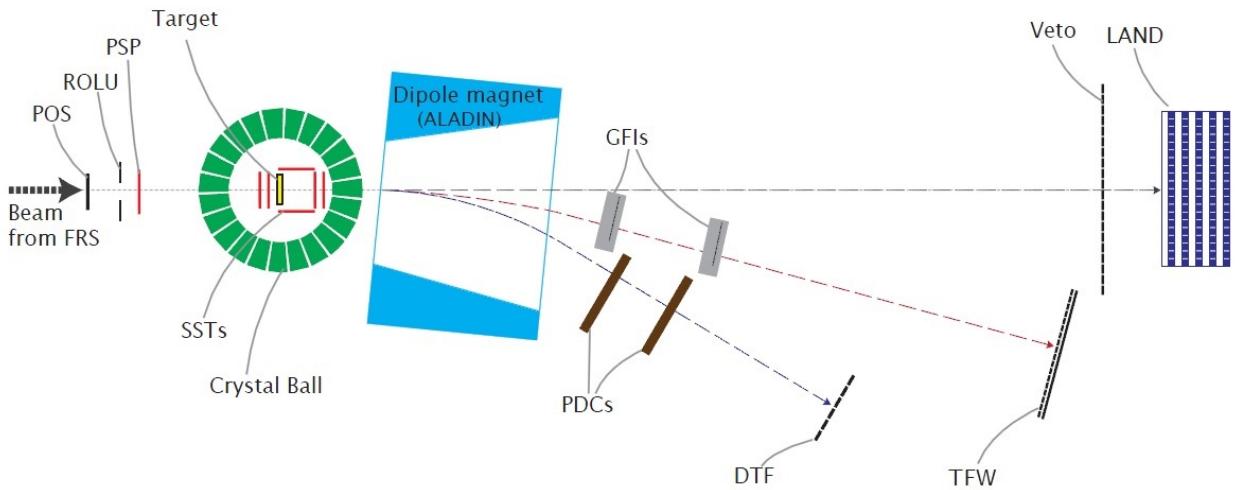


Figure 2.11: Overview of the Super-FRS at FAIR [2].

With the FRS a secondary beam of oxygen isotopes was created and directed to the experimental chambers. There it collides with a target and the events are recorded with the crystal ball detector (and several other detectors), see figure 2.12.

Before the scattering experiment occurs, there are some detectors placed in the secondary beam for beam diagnostics: POS, ROLU and PSP. POS generates a signal when an ion passes through. In this experiment, it is used for triggering. ROLU is a slit system that detects how narrow the beam is. Ions that do not pass through the slit are anti-triggered. PSP measures the energy from the Bethe-Bloch energy deposition in the system.



The schematic view of the detection setup in cave C.

Figure 2.12: Overview of the experimental setup used by Najafi [2].

PSP and POS are also used for particle identification. Then the beam comes to the scattering chamber, where some Silicon strip detectors are placed inside the crystal ball, consistent with figure 2.13. These detectors are calibrated only for energy and position, not time. By correcting for several detector artifacts, including diffusion of electric charge, the position resolution can become better than the distance between succeeding strips. This is the story about the inter-strip parameter η .

Some of the silicon strip detectors are passed BEFORE the scattering and some are passed AFTER the scattering. Next is the crystal ball detector, which is a 4π array of scintillators. Notice that crystal ball therefore ONLY detects photons! By giving each detector another readout channel, also protons (and neutrons) can be detected with crystal ball. Hence the Silicon strip detectors detect the initial and resulting NUCLEUS of the knockout reaction and the crystal ball detects the photons from the excitations and the knocked out nucleons! Some of the nucleons might have sufficient energy to pass through the crystal ball, causing that not their full energy can be reconstructed. Obviously everything has to be calibrated, for which different ions and cosmic muons are used. Cluster algorithms are used to reconstruct energy (See Marcel his work, although the precise algorithms are different). Notice that one should correct for the Doppler shifts of the photons too.

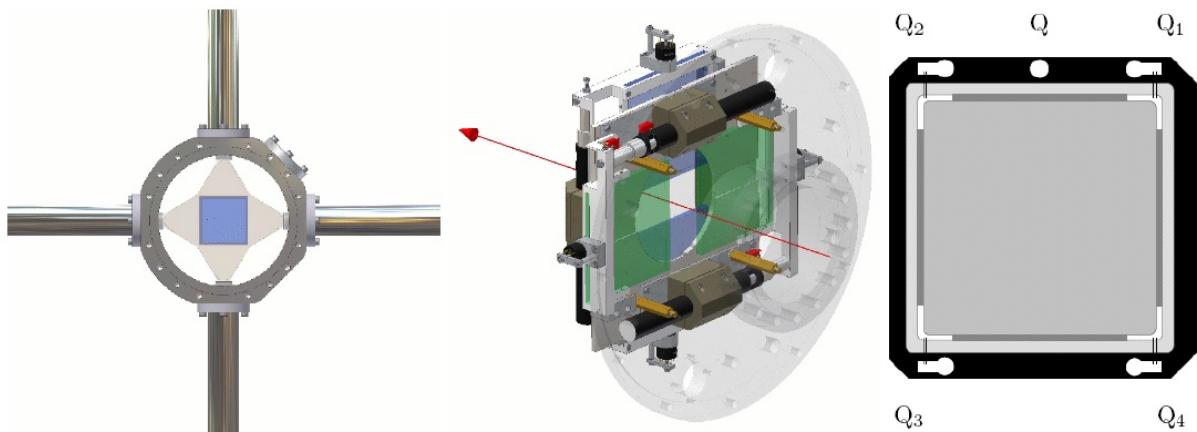
After the fragments and the knocked out nucleons passed through the crystal ball, they go through the magnet and then through some other detectors. LAND is one of them, which has a background detector in front of it that anti-triggers the background signals.

Detailed descriptions of the construction of each detector is given, its achievable resolution and principles of operation and an overview of its calibration.

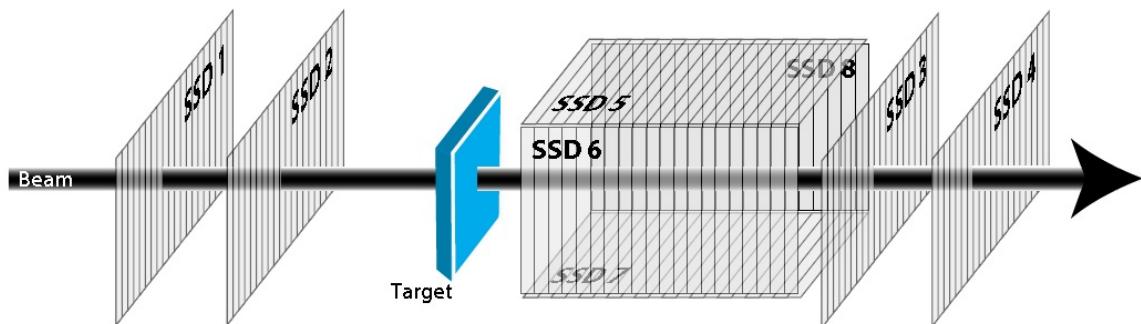
How does this calibration work? I can see that you use different ions to do a 'blind' calibration, but how do you know the properties of the calibration ions so that you can tune your detectors? **How does calibration of a scintillator paddle work?** What is QDC and TDC? This procedure is used for all relevant detectors after the crystal ball.

Special attention is required for the PDC, the proton drift chamber. It is composed of a large number of wires with gas in between and electric fields. When the protons pass through, the gas is ionized and this is felt by the wires since the ionized particles move in the electric fields toward the wires. From the wires that fired a signal one can reconstruct the proton trajectory. This is very simply the principle. By taking the time of detection of several ionizations into account and performing an advanced statistical procedure, an accurate reconstruction of the trajectory is possible.

The rest of chapter 4 is mainly descriptions of calibration procedures and trigger conditions. **What are scaling factors of trigger conditions?** I know what a trigger is: a condition to YES/NO record an event. It can be a hardware limitation, or a software condition. But how does the scaling work?



The drawings of POS with the four photomultiplier tubes (left), the active slit with its moving system, ROLU (middle), and the pin-diode, PSP (right).



The schematic geometry of the eight silicon strip detectors with respect to the target. The arrow shows the beam direction.

Figure 2.13: The pre-scattering detectors from the experiment of Najafi [2].

Particle identification is a combination of almost all beam diagnostic detectors and also several other detectors. The main procedure is to measure energy directly and time of flight to get the velocity. This gives information about the mass. Furthermore, the deflection magnet is used to extract the charge to mass ratio from the knowledge of the magnetic field and the reconstructed path through the magnet from the position measurements on the detector. Another possibility of particle identification is Cerenkov-light (not used in this setup). Usually particle identification is a combination of all such pieces of information. Iterative fitting procedures are used to combine this information in the best possible way, even when the magnetic field is not known very precisely.

The setup from figure 2.12 will be almost equal to the setup of R^3B (Reactions with Radioactive Relativistic Beams) in FAIR with the Super-FRS. Some detectors and additional elements will be added to greatly improve the already successful setup. A mass-spectrometer will be added, so that one can choose between large-acceptance detectors that can detect almost all produced particles in most directions or between the high-resolution mode in which the products are guided to the mass spectrometer, see figure 2.14.

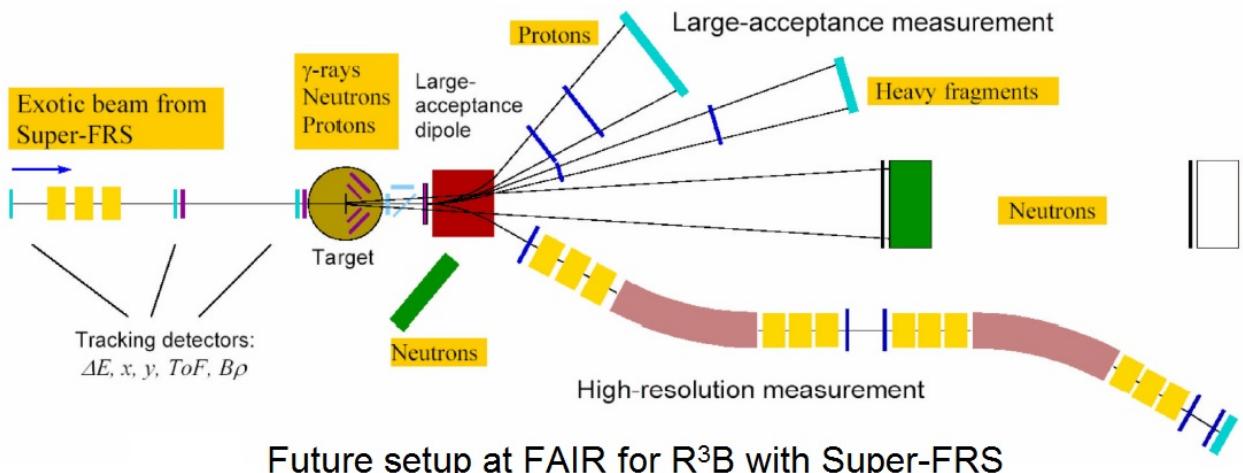
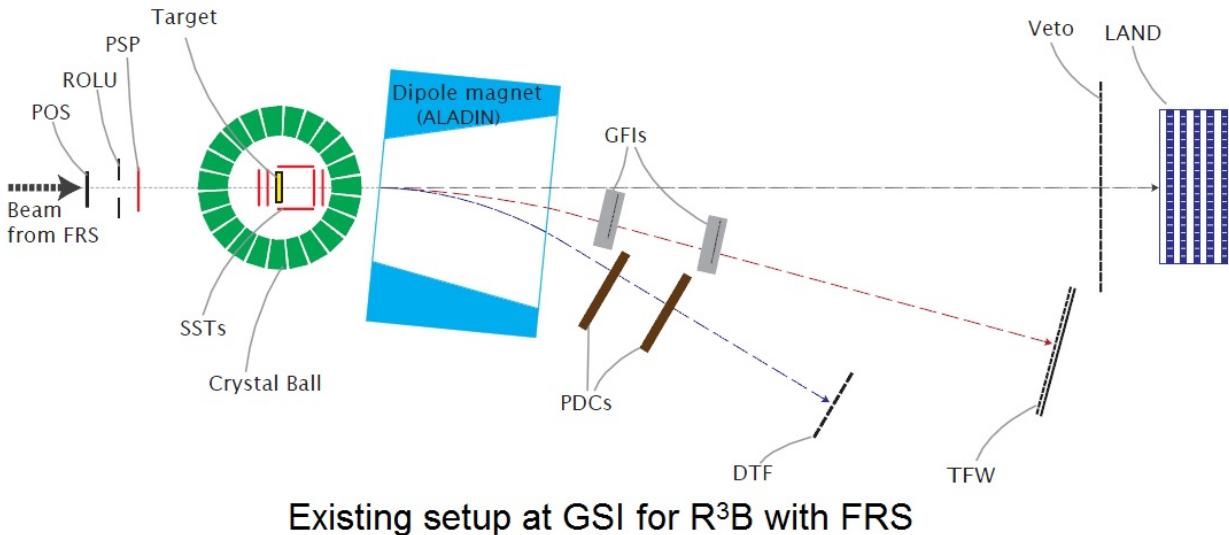
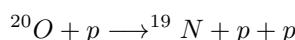
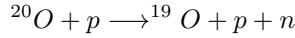


Figure 2.14: Comparison between the existing and future setup of R^3B [2].

2.3.4 Analysis and results of the experiment

In figure 2.12, the beam from the FRS consists of ^{20}O ions that are then collided onto a CH_2 target. A collision with the protons from the hydrogen will then induce the knockout reactions of interest:





In order to select the correct reactions, conditions on the final products are implied: triggers. This is however not enough to ensure a *quasi-free* knockout reaction. In order to have this, we must have that the beam energy per nucleon is much bigger than the binding energy per nucleon, so that the collision is almost like a collision between free nucleons: quasi-free. A clear signature of this behaviour is the angular correlation between the two free nucleons in the final state. in the LAB-frame (z -axis along beam line and origin at target) the difference between the θ -angles of the two nucleons should be about 81° and the difference between the ϕ -angles should be about 180° . This is what comes out of this 3-body process when one assumes isotropic distributions in the CM-frame. Therefore a cut on these angles ensures the selection of quasi-free nucleons.

Measuring an inclusive cross section (meaning: integrated over all kinematic variables: energies and angles) is conceptually very easy:

$$\sigma = \frac{R}{I \cdot n} \quad (2.31)$$

I is simply the beam intensity: the number of ions in the beam within a certain time interval. R is the reaction rate: the total number of observed reactions within the same time interval as was used to measure I . n is the number density from the target:

$$n = \frac{t \cdot N_A}{M}$$

Here t is the areal density of the target: in gcm^{-2} . N_A is the Avogadro number and M is the molar mass of the target in gmol^{-1} . This way, n is the number of particles per cm^2 ; the number of particles that can be hit by the beam. This way, both R and I have units s^{-1} . n has units cm^{-2} . This gives σ units of cm^2 as it should be. However, usually one has to apply lots of corrections to the data to get R out in the right way.

Measuring the angular distribution (or energy distribution) of a cross section is therefore also very easy: one simply applies the above procedure but now one imposes a condition that only reactions are counted for R for which the angle θ of the outgoing particle is in a certain interval $\Delta\theta$. Then in the end, one divided by $\Delta\theta$:

$$\frac{\partial\sigma}{\partial\theta}|_{\theta=\theta^*} \approx \frac{R(\theta \in [\theta^* - \Delta\theta/2, \theta^* + \Delta\theta/2])}{I \cdot n \cdot \Delta\theta}$$

For spatial angles:

$$\frac{\partial\sigma}{\partial\Omega}|_{\theta=\theta^*, \phi=\phi^*} \approx \frac{R(\theta \in [\theta^* - \Delta\theta/2, \theta^* + \Delta\theta/2], \phi \in [\phi^* - \Delta\phi/2, \phi^* + \Delta\phi/2])}{I \cdot n \cdot \Delta\theta \cdot \sin\theta^* \cdot \Delta\phi}$$

and for energy a similar trick can be applied.

Invariant mass always has to be conserved. Invariant mass is given by

$$M_{inv} = \left\| \sum_i p_{i,out} \right\| = \left\| \sum_i p_{i,in} \right\| \quad (2.32)$$

In equation (2.32), $\|\cdot\|$ is the usual Space-time metric from special relativity and $p_{i,out}$ are the 4-momentum vectors from all outgoing particles. $p_{i,in}$ are the 4-momentum vectors from all ingoing particles before the reaction. Hence equation (2.32) is a conservation law. It comes from conservation of momentum. In the case of a decaying nucleus, the invariant mass before the decay (in its rest frame) is equal to its rest mass + its excitation energy. Hence the excitation energy can be recovered.

The luminosity of a beam is equal to $L = n_{target} \cdot N_{beam}$ where n_{target} is the areal number density discussed earlier and N_{beam} is the absolute number of ions in the beam in a certain time interval. Hence the units of L are $\text{cm}^{-2}\text{s}^{-1}$. In order to subtract the background of reactions with the Carbon in the CH_2 -target, one also

spents some beamtime to observe these reactions and then corrects them for different luminosities. Then these measured reactions can be used to identify the background, so it can be subtracted.

Obviously when measuring R , the detector geometrical acceptance (where there is no detector, there is no hit) and the efficiency of the detector (when the detector produces a signal, given there is a particle) must be corrected for. Together, these two effects are the full detector acceptance. If however the same acceptance is used to deduce I as well, no correction needs to be applied, because it divides out. This is true for most setups, but not for Crystal Ball.

In FairROOT and R3BROOT geometrical packages of the detectors are available. Hence purely geometrical acceptances can easily be evaluated. Efficiencies etc are however NOT available; one has to implement that by hand.

2.4 Alex PHD

2.5 Zahra PHD

Works on the energy levels and resonance states of charmonium. She analyzes specific decay channels of this.

2.6 Marcel PHD

Works on a computer algorithim (C++) to reconstruct gamma photons from the scintillator arrays of PANDA. The algorithm will be used in PANDA when it is completed.

2.7 Solmaz PHD

2.8 Gita Postdoc

Chapter 3

Construction of Super-FRS

3.1 Practical information

The thermal conductivity coefficient of Densimet as a function of temperature in Kelvin is given by:

$$K(T) = 64.2935 + 156.0611 \cdot e^{-0.00372332 \cdot T} + 60,0588 \cdot e^{-0.0037213 \cdot T} + 47,6263 \cdot e^{-0.000480648 \cdot T} \text{ W m}^{-1} \text{ K}^{-1} \quad (3.1)$$

a picture is given by:

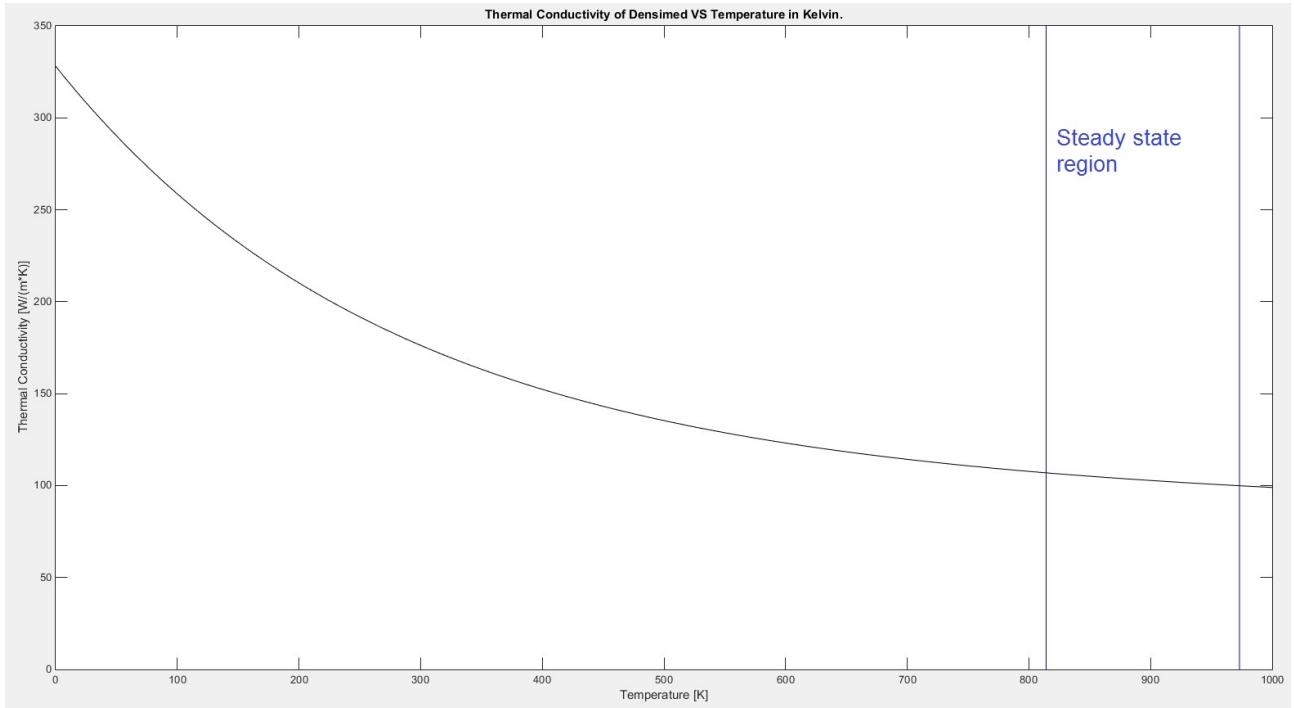


Figure 3.1: Thermal conductivity for Densimet.

3.2 NX-9 tutorials

3.3 X-slit system

3.3.1 Matlab simulation of Densimet

In this simulation we solve the diffusion equation on a rectangular box with uniform spatial meshes. The meshes in x , y and z might be different from each other. A rectangular grid is imposed over the Densimet block. The grid points are labelled $T_{i,j,k}$ where $i = 1, \dots, n_x$, $j = 1, \dots, n_y$ and $k = 1, \dots, n_z$. These grid points might change as a function of time. To discretize the Laplacian a second order discretization is applied:

$$\frac{\partial^2 T_{i,j,k}}{\partial x^2} = \frac{T_{i+1,j,k} - 2T_{i,j,k} + T_{i-1,j,k}}{h_x^2} \quad (3.2)$$

Discretization in the direction of y and z is performed analogously. The Laplacian is then given by:

$$\vec{\nabla}^2 T_{i,j,k} = \frac{\partial^2 T_{i,j,k}}{\partial x^2} + \frac{\partial^2 T_{i,j,k}}{\partial y^2} + \frac{\partial^2 T_{i,j,k}}{\partial z^2} \quad (3.3)$$

On interior points there is no problem, but on the boundaries there is. Neumann boundary conditions are applied there to allow for computation of the Laplacian. This is done by resorting to ghost points: a single layer of points is added outside the Densimet block to which the Neumann condition is applied. This allows then for direct computation of the Laplacian on the boundary points. The Neumann conditions read:

$$\left. \frac{\partial T_{i,j,k}}{\partial x} \right|_{i=1,n_x} = 0 \quad \left. \frac{\partial T_{i,j,k}}{\partial y} \right|_{j=1,n_x} = 0 \quad \left. \frac{\partial T_{i,j,k}}{\partial z} \right|_{j=1,n_z} = 0 \quad (3.4)$$

The practical implementation is then given by:

$$\left. \frac{\partial^2 T_{i,j,k}}{\partial x^2} \right|_{i=1} = \frac{T_{i+1,j,k} - 2T_{i,j,k} + T_{ghost}}{h_x^2} \quad T_{ghost} = T_{i,j,k} \quad (3.5)$$

Notice than in equation (3.5), T_{ghost} plays the role of $T_{i-1,j,k}$. By equating it to $T_{i,j,k}$ the Neumann condition is then applied. On $i = n_x$ a similar condition is applied. For the derivatives to y and z the procedure is similar.

Now given a grid with values of $T_{i,j,k}$, the above procedure applies for a computation of the full Laplacian. Then the diffusion equation

$$\frac{\partial T_{i,j,k}}{\partial t} = D \cdot \vec{\nabla}^2 T_{i,j,k} \quad (3.6)$$

gives us the time derivatives of the grid $T_{i,j,k}$. Hence this gives a system of ordinary differential equations. The time integration is then performed with forward Euler. Notice that the absolute stability criterion for forward Euler on the Diffusion equation (with second order spatial discretization) is then given by

$$\Delta t \leq \frac{1}{2^3 \cdot D} \cdot \min\{h_x^2, h_y^2, h_z^2\} \quad (3.7)$$

The Neumann boundary condition makes sense, since the boundaries of the Densimet block are 'loose ends' in vacuum (see the analogue with waves in a rope). When cooling by radiation or heating by a beam is however applied, the endpoints must be able to heat up and cool down as well. This is accomplished by adding a term to the time derivative of equation (3.8):

$$\frac{\partial T_{i,j,k}}{\partial t} = D \cdot \vec{\nabla}^2 T_{i,j,k} - \langle \text{Radiation} \rangle + \langle \text{Beam} \rangle \quad (3.8)$$

The Neumann boundary condition is however applied between the endpoints and the ghost points. This means the endpoints are still free to cool down and heat up. The boundary condition only limits the value of the ghost points, which we are not interested in. Moreover, note that the Neumann condition is on the spatial derivative, not on the time derivative. A hand-wave argument is to state that the Neumann boundary condition and the ghost points are only used for computing the Laplacian. Then the other terms can still provide cooling or heating at the endpoints.

3.3.2 Implementation of Radiation cooling by using the Debye Model

This leaves us with modeling the radiation cooling and beam heating. Since these are given in terms of energy gains and energy loss and not in terms of temperature, one needs a way to relate the temperature of the Densimet to the internal energy of the Densimet. For this we will use the Debye model. This model gives the absolute internal energy of a solid state substance in terms of the temperature of this substance:

$$U = 3Nk_B T \cdot D_3\left(\frac{T_D}{T}\right) \quad D_n(x) = \frac{n}{x^n} \cdot \int_0^x \frac{t^n}{e^t - 1} dt \quad (3.9)$$

$$T_D = \frac{h \cdot v_{eff}}{2k_B} \cdot \sqrt[3]{\frac{6N}{\pi V}} \quad \frac{1}{v_{eff}^3} = \frac{1/3}{v_{long}^3} + \frac{2/3}{v_{trans}^3}$$

If a single metal atom has a weight m , one can write that

$$\frac{N}{V} = \frac{Nm}{V} \cdot \frac{1}{m} = \frac{M}{V} \cdot \frac{1}{m} = \frac{\rho}{m}$$

Hence one can rewrite the Debye model as:

$$\frac{U}{V} = 3\frac{\rho}{m} k_B T \cdot D_3\left(\frac{T_D}{T}\right) \quad D_n(x) = \frac{n}{x^n} \cdot \int_0^x \frac{t^n}{e^t - 1} dt \quad (3.10)$$

$$T_D = \frac{h \cdot v_{eff}}{2k_B} \cdot \sqrt[3]{\frac{6\rho}{\pi m}} \quad \frac{1}{v_{eff}^3} = \frac{1/3}{v_{long}^3} + \frac{2/3}{v_{trans}^3}$$

Hence all one needs to know about the material is the atomic mass, the mass density and the effective speed of sound.

According to the Debye model, Internal energy density is therefore a function on temperature alone (and some constants that are possibly materialdependent). Hence if one is interested in the time derivative of the temperature due to changes in the internal energy one can state that:

$$\frac{\partial T}{\partial t} = \frac{\partial T}{\partial U} \cdot \frac{\partial U}{\partial t} = \frac{\partial U}{\partial t} / \frac{\partial U}{\partial T} \quad (3.11)$$

Now the fracter $\frac{\partial U}{\partial t}$ is modeled by Plancks blackbody radiation. Notice that U is the absolute internal energy:

$$\frac{\partial U}{\partial t} = \epsilon \sigma T^4 \cdot A \quad (3.12)$$

In equation (3.12), ϵ is the emissivity, σ is the Stefan-Boltzmann constant, T is the temperature and A is the surface that radiates. The factor $\frac{\partial U}{\partial T}$ can be obtained from the Debye model. Notice that this is precisely the specific heat of the material:

$$\frac{\partial U}{\partial T} = C_V = 9 \frac{\rho}{m} V k_B \cdot \left(\frac{T_D}{T}\right)^3 \int_0^{T_D/T} \frac{x^4 e^x}{(e^x - 1)^2} dx \quad (3.13)$$

Hence one obtains:

$$\frac{\partial T}{\partial t} = \epsilon\sigma T^4 \cdot A / C_V = \epsilon\sigma T^4 \cdot A / \left(9 \frac{\rho}{m} V k_B \cdot \left(\frac{T_D}{T} \right)^3 \int_0^{T_D/T} \frac{x^4 e^x}{(e^x - 1)^2} dx \right) \quad (3.14)$$

If this is applied to our diffusion grid, then one obtains:

$$\frac{\partial T_{i,j,k}}{\partial t} = D \cdot \vec{\nabla}^2 T_{i,j,k} - \epsilon\sigma T_{i,j,k}^4 \cdot A / \left(9 \frac{\rho}{m} V k_B \cdot \left(\frac{T_D}{T_{i,j,k}} \right)^3 \int_0^{T_D/T_{i,j,k}} \frac{x^4 e^x}{(e^x - 1)^2} dx + \langle Beam \rangle \right) \quad (3.15)$$

Notice that in equation (??) one always has that $V = h_x \cdot h_y \cdot h_z$ and A is the surface of the mesh that radiates. Hence $A = 0$ for interior points (no radiation) and $A = h_x \cdot h_y, h_x \cdot h_z, h_y \cdot h_z$ or a sum of those depending on where we are on the grid. To solve the beam, a similar procedure can be applied.

As a first to implementing equation (??), let us compute the Debye temperature. This is needed anyway and perhaps allows for a major simplification. Evaluating the integral in equation (??) for all mesh points is a major drain on computational resources. One should therefore try to avoid this. If there is no other way, it is fortunately only a surface effect, meaning that it will not be the dominant drain on computational resources for very fine spatial meshes. Computing the diffusion is proportional to $n_x \cdot n_y \cdot n_z$, while the radiation is proportional to $n_x \cdot n_y + n_x \cdot n_z + n_y \cdot n_z$.

Densimet is composed of 97 % Thungsten (Wolfraam), 2 % Nickel and 1 % iron. Hence one obtains that (for Annealed Thungsten):

$$v_{Thungsten,trans} = 5220 \text{ m/s} \quad v_{Thungsten,long} = 2890 \text{ m/s} \quad \rho_{Thungsten} = 19300 \text{ kg m}^{-3} \quad m_{Thungsten} = 183.85 \text{ u}$$

$$v_{Nickel,trans} = 6040 \text{ m/s} \quad v_{Nickel,long} = 3000 \text{ m/s} \quad \rho_{Nickel} = 8900 \text{ kg m}^{-3} \quad m_{Nickel} = 58.71 \text{ u}$$

$$v_{Iron,trans} = 5960 \text{ m/s} \quad v_{Iron,long} = 3240 \text{ m/s} \quad \rho_{Iron} = 7850 \text{ kg m}^{-3} \quad m_{Iron} = 55.85 \text{ u}$$

Computing weighted averages and knowing that $1 \text{ u} = 1.66053892 \cdot 10^{-27} \text{ kg}$ gives that for Densimet one roughly obtains that

$$v_{eff} = 3791 \text{ m/s} \quad m = 2.9901 \cdot 10^{-25} \text{ kg} \quad \rho = 18978 \text{ kg m}^{-3}$$

This gives a Debye temperature of:

$$T_D \approx 450.21 \text{ K}$$

An overview of the Debye model for Densimet is given by figure 3.2.

Since the Densimet starts at a temperature of $T = 20^\circ\text{C}$ and then heats up to about $T = 700^\circ\text{C}$, one cannot neglect the temperature dependence on specific heat. Hence in short the specific heat can be computed with

$$C_V(T) = 9 \frac{\rho}{m} V k_B \cdot \left(\frac{T_D}{T} \right)^3 \int_0^{T_D/T} \frac{x^4 e^x}{(e^x - 1)^2} dx \quad (3.15)$$

$$T_D = \frac{h \cdot v_{eff}}{2k_B} \cdot \sqrt[3]{\frac{6\rho}{\pi m}} \quad \frac{1}{v_{eff}^3} = \frac{1/3}{v_{long}^3} + \frac{2/3}{v_{trans}^3}$$

Then to model the cooling by radiation, the diffusion equation changes into:

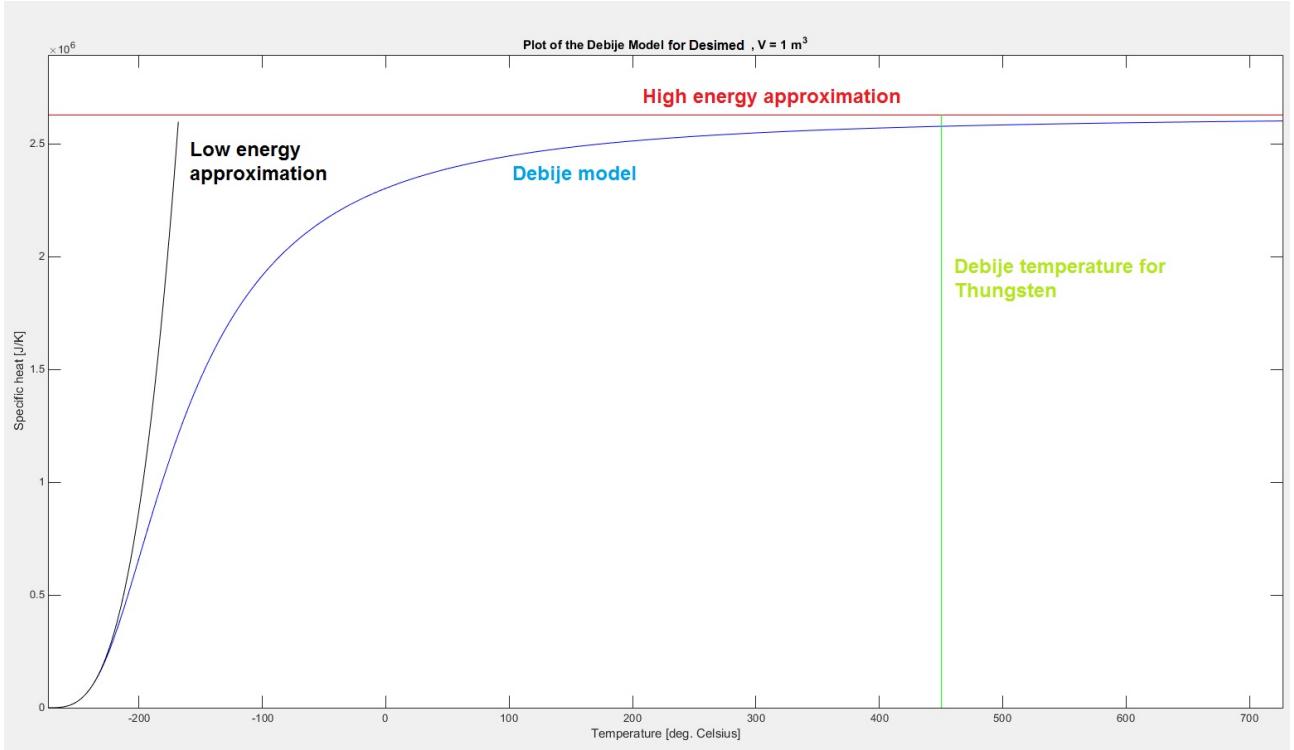


Figure 3.2: Debye model for Densimet

$$\frac{\partial T_{i,j,k}}{\partial t} = D \cdot \vec{\nabla}^2 T_{i,j,k} - \epsilon \sigma T_{i,j,k}^4 \cdot A \cdot \frac{1}{C_V(T_{i,j,k})} \quad (3.15)$$

3.3.3 Implementation of the beam

This is done on a similar basis as the cooling. Suppose $Q_{i,j,k}$ is the absolute power deposited in a single mesh by the beam. The cooling loses energy, so in the case for cooling, one gets

$Q_{i,j,k} = -\epsilon \sigma T_{i,j,k}^4 \cdot A$ But in the case for the beam, $Q_{i,j,k}$ is just prescribed. In our model, a Gaussian beam intensity profile is assumed. It is assumed that the beam is fired in the $+y$ direction on the xy plane located at $y = 0$. The coordinates of x and z at which the beam hits are prescribed as x_{beam} and z_{beam} . The Gaussian standard deviation R_{beam} is also prescribed. Then the Gaussian beam profile is defined as:

$$f(x, z) = \frac{1}{2\pi R_{beam}^2} \cdot e^{-\frac{1}{2} \cdot (\frac{x-x_{beam}}{R_{beam}})^2} \cdot e^{-\frac{1}{2} \cdot (\frac{z-z_{beam}}{R_{beam}})^2}$$

Then if Q is the total beam power, the power prescribed to the meshes equals

$$Q_{i,1,k} = Q \cdot f(x_i, z_k) \cdot h_x \cdot h_z \quad Q_{i,j,k} = 0 \text{ if } j \neq 1$$

In the case that $R_{beam} < \max(h_x, h_y)$ The grid is not capable of representing the Gaussian beam profile accurately, so we simply prescribe that $Q_{i,j,k} = Q$ for the single grid point $j = 1$ and i and k are such that x_i and z_k are closest to x_{beam} and z_{beam} . For all other meshes it is assumed that $Q_{i,j,k} = 0$.

With this model for the beam, the full equation to solve becomes:

$$\frac{\partial T_{i,j,k}}{\partial t} = D \cdot \vec{\nabla}^2 T_{i,j,k} - \epsilon \sigma T_{i,j,k}^4 \cdot A \cdot \frac{1}{C_V(T_{i,j,k})} + Q_{i,j,k} \cdot \frac{1}{C_V(T_{i,j,k})} \quad (3.15)$$

3.3.4 Thermal conductivity

Heat conduction is normally given in terms of thermal conductivity K . In our model we used however the diffusion coefficient D . The relation between them is

$$D = \frac{K}{\rho \cdot c_v} \quad (3.15)$$

In equation (3.3.4), c_v is the specific heat capacity:

$$c_v = \frac{C_V}{M} = \frac{C_V}{V \cdot M/V} = \frac{C_V}{V \cdot \rho}$$

If one takes along the temperature dependence of C_V by the Debye model, the computation becomes extremely expensive, because the Debije integral has to be evaluated for each mesh point and for each timestep, because the temperature changes there. Moreover, K also might be temperature dependent. By doing only a rough evaluation of the Debye integral this effect could be suppressed but is still a major drain on computational resources. Another way is to take a constant value for C_V , the high temperature limit of the Debije model for example. In that case Densimet has the following diffusion coefficient:

$$C_V = 3Nk_B = 3\frac{\rho}{m}k_B V$$

$$c_v = \frac{3\rho k_B V}{V \rho m} = \frac{3k_B}{m}$$

$$D = \frac{K}{\rho \cdot 3k_B/m} = \frac{Km}{3\rho k_B}$$

$$K_{Thungsten} = 173 \text{ W m}^{-1} \text{ K}^{-1} \quad K_{Nickel} = 90.9 \text{ W m}^{-1} \text{ K}^{-1} \quad K_{Iron} = 80.4 \text{ W m}^{-1} \text{ K}^{-1}$$

This gives:

$$K = 170.43 \text{ W m}^{-1} \text{ K}^{-1} \quad m = 2.9901 \cdot 10^{-25} \text{ kg}$$

Hence for Densimet one obtains:

$$D = 6.4830 \cdot 10^{-5} \text{ m}^2 \text{ s}^{-1}$$

So the real model to solve would be:

$$\frac{\partial T_{i,j,k}}{\partial t} = \frac{K \cdot V}{\rho \cdot C_v(T_{i,j,k})} \cdot \vec{\nabla}^2 T_{i,j,k} - \epsilon \sigma T_{i,j,k}^4 \cdot A \cdot \frac{1}{C_V(T_{i,j,k})} + Q_{i,j,k} \cdot \frac{1}{C_V(T_{i,j,k})} \quad (3.15)$$

Where C_V is computed according to equation (3.15): The Debye heat capacity. But instead we will solve equation (??) with $D = 1.2304 \text{ m}^2 \text{ s}^{-1}$ to save computational resources:

$$\frac{\partial T_{i,j,k}}{\partial t} = D \cdot \vec{\nabla}^2 T_{i,j,k} - \epsilon \sigma T_{i,j,k}^4 \cdot A \cdot \frac{1}{C_V(T_{i,j,k})} + Q_{i,j,k} \cdot \frac{1}{C_V(T_{i,j,k})} \quad (3.15)$$

3.3.5 Time integration

In our previous discussion of section 3.3.3 it was stated that forward time integration is used. In this case the size of the time steps is limited to

$$\Delta t \leq \frac{\min(h_x^2, h_y^2, h_z^2)}{8 \cdot D}$$

If however only 10 meshes are used in the x , y and z direction only and $L_x = 0.18$ m, $L_y = 0.25$ m and $L_z = 0.20$ m, this gives a limitation to $\Delta t \approx 10^{-5}$ s!!! Given the fact that the Densimet takes about 40h to reach its steady-state temperature distribution, this is completely unacceptable. Hence we need a different time integrator.

With equation (??), one is able to compute the time derivative $\frac{\partial T_{i,j,k}}{\partial t}$ from the temperature values $T_{i,j,k}$:

$$\frac{\partial T_{i,j,k}}{\partial t}(T_{i,j,k})$$

Hence if one has on a time $t = t_n = n \cdot \Delta t$ a certain temperature distribution $T_{i,j,k}^n$, one can compute the temperature distribution on a subsequent time t_{n+1} by doing:

$$T_{i,j,k}^{n+1} = T_{i,j,k}^n + \Delta t \cdot \frac{\partial T_{i,j,k}}{\partial t}(T_{i,j,k}^n)$$

This is called Forward Euler.

Backward Euler is a slightly different time integrator that has the scheme:

$$T_{i,j,k}^{n+1} = T_{i,j,k}^n + \Delta t \cdot \frac{\partial T_{i,j,k}}{\partial t}(T_{i,j,k}^{n+1})$$

The major advantage from Backward Euler is that there is *no* absolute stability criterion (only that the step size should not be too small!). Hence one can take any time step needed. The disadvantage is that $\frac{\partial T_{i,j,k}}{\partial t}$ depends on $T_{i,j,k}$ in a highly nonlinear way. Hence each time step now requires the solution of a huge non-linear system of equations. Still, this might computationally be less costly than taking timesteps of 10^{-5} seconds.

In order to solve this system, the internal Matlab function *fsolve* is used. This is a well-developed multidimensional variant of Newtons method.

Please notice that one can take arbitrary timesteps regarding absolute stability, but this still does not provide a guarantee for an accurate solution! It only provides a guarantee that the solution is wriggle-free.

3.4 Target system

Chapter 4

Materials and Method of Analysis

4.1 Principles of operation for NeuLAND

In this section the basic principles of the NeuLAND detector are explained in detail. The explanation starts with the particle to detect and ends with the first histograms produced. The software processing is not explained in this section.

4.1.1 Basic principles of scintillators

A scintillator is a material that produces light when it is hit by a charged particle. See figure 4.1.

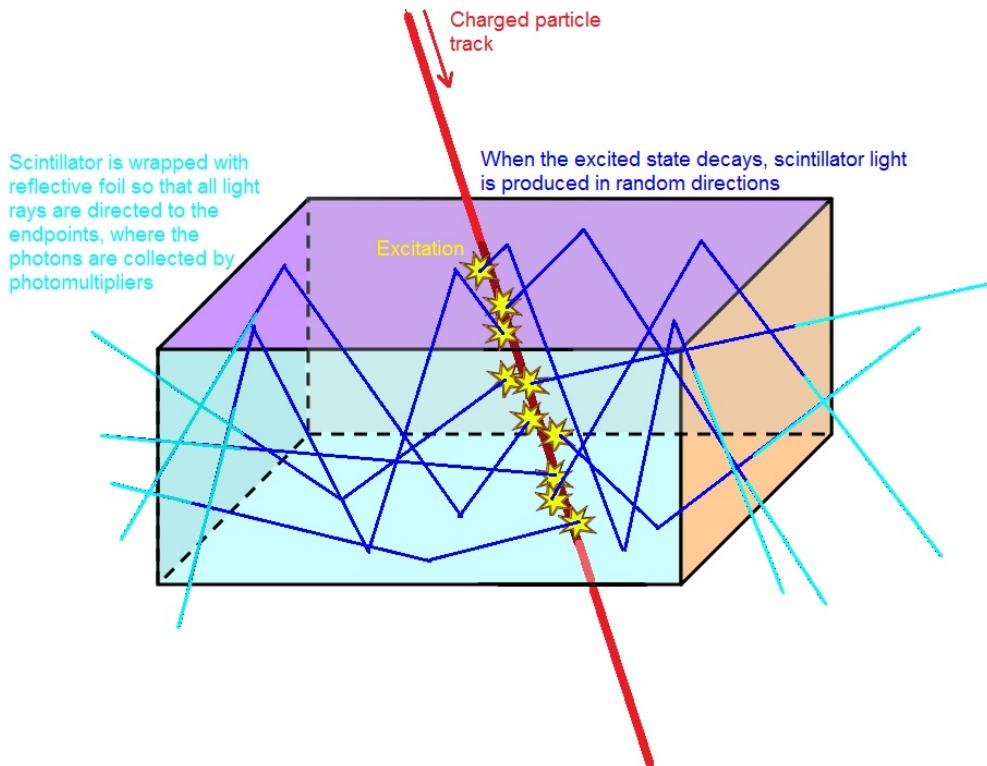


Figure 4.1: Operation principles of a scintillator

When a charged particle travels through the material it produces excitations with its Coulomb force. These excitations must be viewed from a point of Solid State Physics. If one would see it as a picture from interactions with separate atoms, the collision would produce ionizations rather than excitations and these are unlikely to

recombine all. The solid state excitations then decay and produce light. Materials that exhibit these specific solid-state properties are called scintillators. A suitable scintillator must in addition be transparent to its own scintillator light. The scintillator is wrapped with reflective foil to guide the produced light to the endpoints. There the light is collected with photomultipliers. The produced photons carry the energy that the charged particle lost while it was traveling through the scintillator. Hence by detecting these photons the energy loss of the particle can be reconstructed.

A photomultiplier (see figure ??) for its operation principle) is capable of detecting single photons but it is NOT the case that it can only detect one photon at a time. Signals of different photons are simply added when they are detected at the same time. A typical signal from a photomultiplier when detecting scintillator light (consisting of multiple photons) looks like figure 4.3. Hence these signals contain all available information. The additional information that can be exploited as well is the position of the scintillator in space, which is obviously known. This is how position measurements are done.

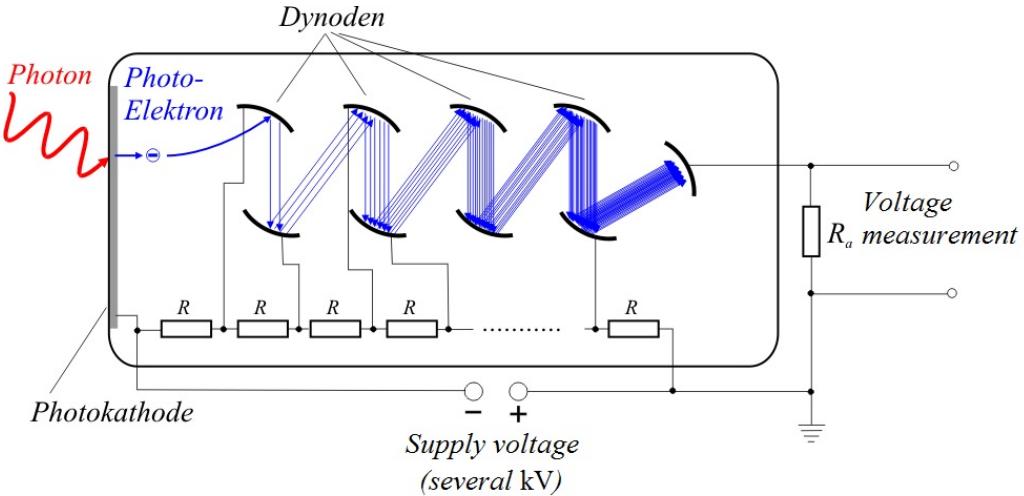


Figure 4.2: Operation principles of a photomultiplier

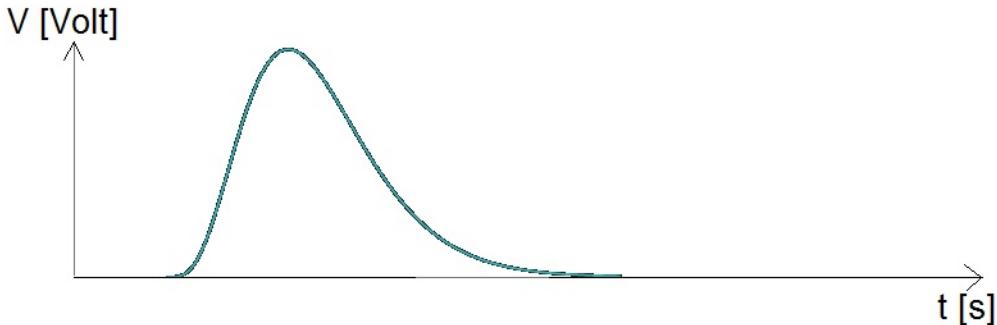


Figure 4.3: typical signal of a photomultiplier

The combination of a scintillator rod used for NeuLAND with 2 photomultipliers at the endpoints is displayed in figure 4.4.

4.1.2 Hardware triggers on the signals

The signal from the photomultiplier is now fed to a discriminator that determines whether there was a signal or not. As soon as $V_{signal} > V_{Threshold}$, a digital block pulse is sent. The height of the digital block pulse is arbitrary and carries no information. The duration is a parameter that is manually set. It is a trade-off between time resolution and detector dead time. This will be explained in more detail later on. The initial time of the block pulse carries the information at which time the signal was detected. Hence these pulses do a time measurement of the particle. See figure 4.5.

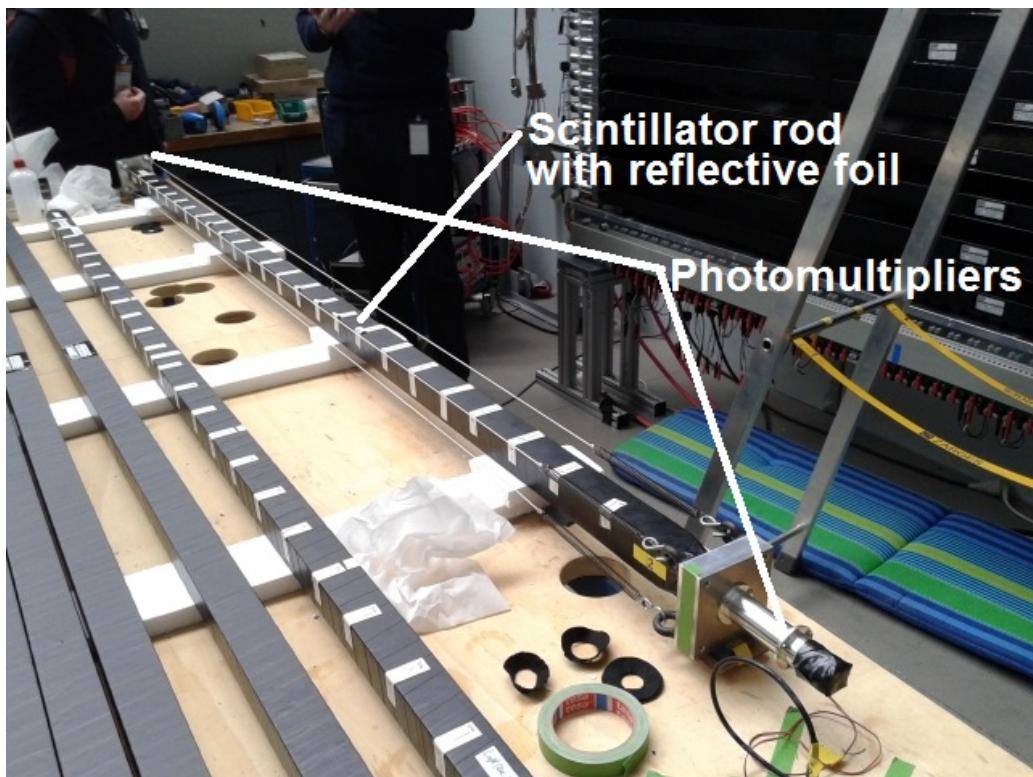


Figure 4.4: typical signal of a photomultiplier

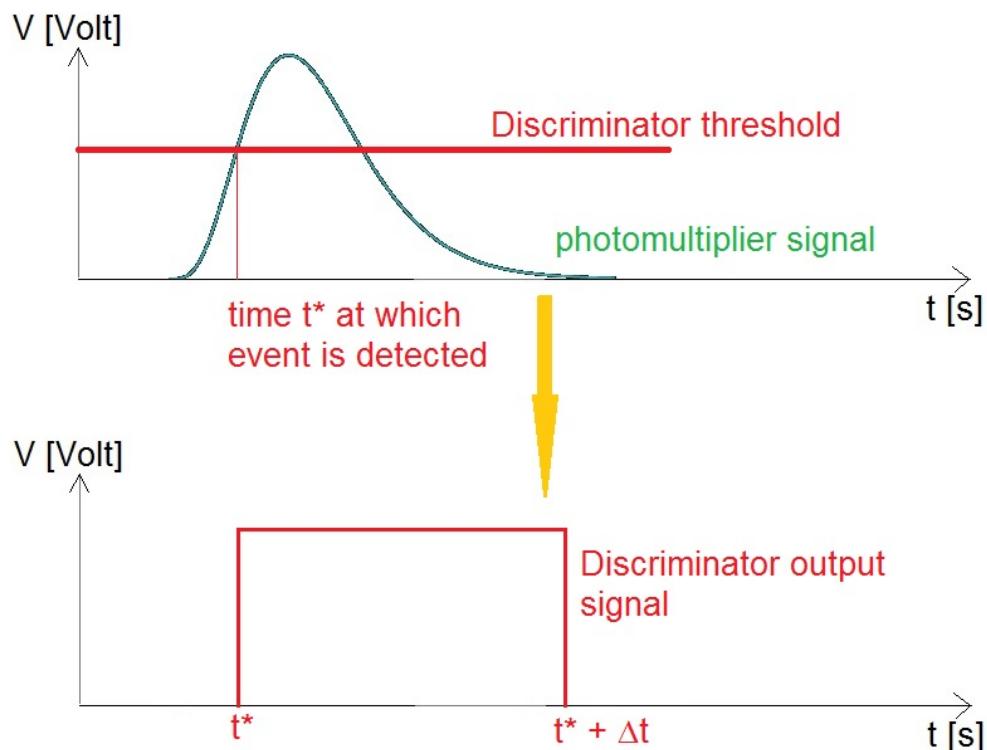


Figure 4.5: Working principle of a discriminator

The discriminator pulses of all photomultipliers are now fed into a logical unit. The output of this logical unit is a single block pulse starting a time $t_{trigger}$ and having a duration time $\Delta t_{trigger}$. This output signal is called the hardware trigger. The purpose of this logical unit is to filter out random noise. The scintillators are hit with much more charged particles (usually from the environment) than the ones you are really interested in. But a particle of interest usually will hit an array of detectors in a specific way. Hence the logical unit can select on these specific ways and suppress random noise. If this would not be done, the amount of background would be way bigger than the signal making any analysis impossible. See figure 4.6.

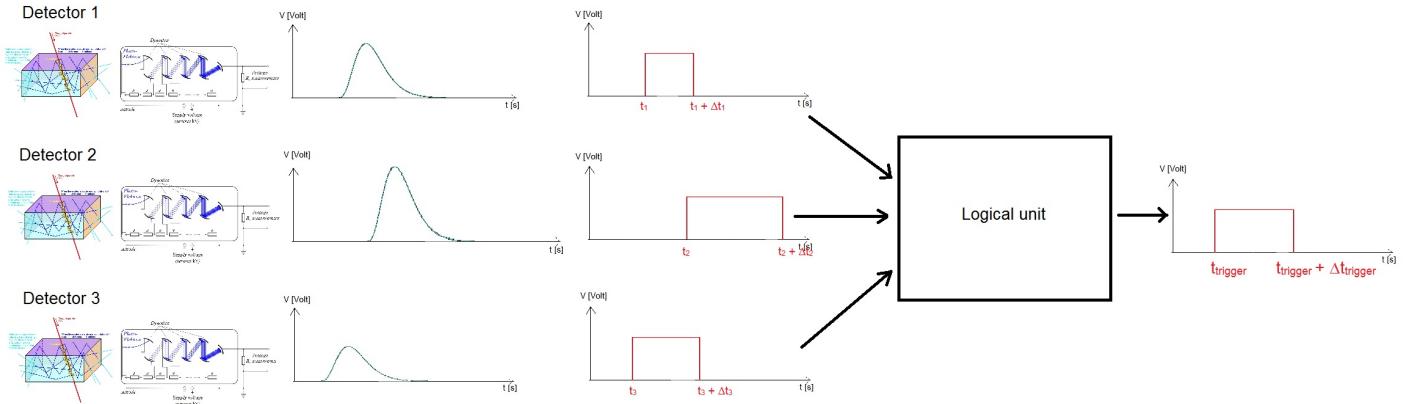


Figure 4.6: Principle of coincidence

The durations Δt_1 , Δt_2 and Δt_3 of the pulses are manually set. They are important for how the signals are processed in the logical unit and which hardware trigger signal is created in the end. If the detectors do not match certain logical conditions, no trigger signal is created meaning that no events are recorded. If there is a trigger signal, data is recorded. This way the coincidence measurement of the logical unit can filter out random noise.

4.1.3 ADC, QDC and TDC

The trigger signal now determines *whether* the signals should be processed or not, but it does not determine how to process them. For this we make two histograms for each photomultiplier. A **TDC** (Time to Digital Converter) histogram of time versus counts and either an **ADC** (Amplitude to Digital Converter) of voltage versus counts or a **QDC** ((Q)charge to Digital Converter) of charge versus counts.

The TDC records the time difference between the starting time $t_{trigger}$ from the trigger block pulse and the starting time t^* of the photomultiplier block pulse. It then adds a single count to the bin corresponding to this time difference.

ADC measures the maximum voltage in the time window between $t_{trigger}$ and $t_{trigger} + \Delta t_{trigger}$ and adds one count to the bin corresponding to this voltage. This voltage represents then the peak of the original photomultiplier signal and hence the amount of energy the particle lost in the scintillator. QDC integrates the photomultiplier signal in the time window between $t_{trigger}$ and $t_{trigger} + \Delta t_{trigger}$ and adds one count to the bin corresponding to this integral. This is also a representation of the amount of energy lost by the particle. It is a matter of preference which is used in which detector.

Hence a single trigger pulse corresponds to a single detected event and hence to a single count in each histogram. So a single TDC histogram and a single QCD histogram (or ACD) are created for each photomultiplier during an entire experiment.

4.1.4 Pulse time and self-triggering

This procedure explains the importance of the pulse duration $\Delta t_{trigger}$, which is determined by all the pulse durations and by the logical unit. If $\Delta t_{trigger}$ is too small, the QCD (or ADC) cannot detect the full energy in the photomultiplier signal. On the other hand, the entire detector array cannot detect a new event during $\Delta t_{trigger}$, because TDC and QDC (or ADC) are based on this single trigger signal. Only when the trigger is

gone, it is possible to detect a new event. So the choice of the pulse durations is a trade-off between detector dead time and the time needed to handle a single measurement.

Notice that all particles (both the ones of interest and the random particles) detected within a single hardware trigger pulse are regarded as a single event. Hopefully this event corresponds to a single particle, but there is no guarantee for that. If there are multiple signals coming from the photomultiplier tube within $\Delta t_{trigger}$, QDC will integrate them all (ADC will find the overall maximum) and regard them as a single particle energy. TDC will add a count to the start of the first signal and ignore the others. These fluctuations give that each measurement gets peaks with a certain width instead of delta-spikes.

The initial time $t_{trigger}$ is usually created from the logical unit by selecting one of the starting times t^* from one of the block pulses coming from one of the photomultipliers. If t^* and $t_{trigger}$ are the same, the TDC of this photomultiplier will add a count to bin zero. This effect is called the self-trigger and is only a reflection of what the logical unit does; it contains no information. Usually the same detector fires first every event, creating a huge delta-spike at zero for that photomultiplier. This is the self-trigger effect and contains no actual information.

4.1.5 Event reconstruction and software triggers

Hence for each event one has only a TDC count containing time information and a QDC (or ADC) count containing energy information. Obviously one knows which counts correspond to which events and which ones belong to each other. In addition one has the position of the detector in space. If one has two photomultipliers on the scintillator rod, one on each side, then one has two QDC values and two TDC values. Then QDC values must obviously be added to obtain the total energy loss of the event, but the two TDC values can be used to reconstruct the place on the rod where the event took place. And hence one can correct the TDC values for the travel time of the light.

NeuLAND is now a detector consisting of planes from scintillator rods which are alternating horizontal and vertical. See figure 4.7 for a picture of one layer of scintillators with photomultipliers from NeuLAND.

Now by combining the information of which scintillators have fired within a single event with the position information of each scintillator, one can reconstruct the particle track. By using the QDC, one can measure the full particle energy. TDC can also help in reconstructing the particles velocity, if the time resolution is good enough. Particle identification is much more troublesome. Usually detectors before and after a bending magnet are used. By fitting the tracks of a single particle together one can reconstruct the particle track through the magnetic field. If the magnetic field is known, the charge-to-mass ratio can be measured which is extremely valuable for the particle identification. Usually much more tricks are used and the final identification is a combination of all these tricks. From energy, track and identification one can reconstruct all kinematic variables belonging to the particle.

All this reconstruction is done by software (usually online). Then a software trigger is simply a decision whether one wishes to keep the event or not immediately after the reconstruction. This is done because a lot of events are produced that do not always allow for a precise reconstruction. So to limit the amounts of data one employs software triggers.

4.1.6 Detecting neutrons with NeuLAND and the VETO-detector

4.1.7 Statistical principles of detector efficiency

Let us first formulate this problem in a statistical way. Suppose we have a detector with an unknown efficiency p . All we know is that $0 \leq p \leq 1$. We now shoot N particles on this detector (either in a simulation, or in real life). Then detector detects n from them. What is now the efficiency p of this detector? And what is the error on p ?

We begin with noting that even if the particles are part of a Monte Carlo simulation, N and n have no error at all. This is because as soon as the simulation is complete, you know exactly how many particles you shot at your detector (N) and you also know how many particles were detected *in that specific simulation*, namely n . Hence after your simulation is complete, both N and n have no error. They might differ from simulation to



Figure 4.7: One plane of scintillators from NeuLAND

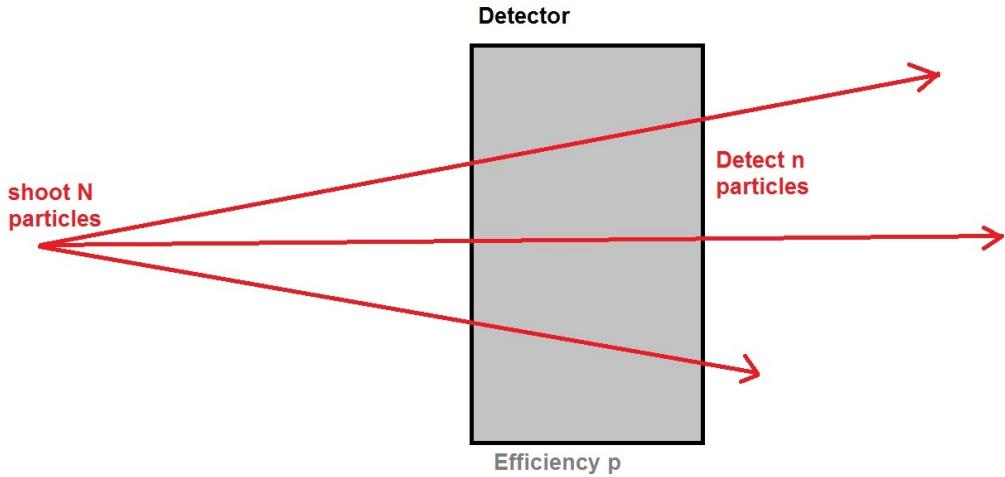


Figure 4.8: General setting on the efficiency of a detector.

simulation (which is where the \sqrt{n} of Monte Carlo enters), but for one specific simulation, they are known and therefore have no errors.

Another thing is that n is always a subset from N . Hence if you would repeat your Monte Carlo simulation (without limits on N), N will have a spread of (\sqrt{N}) between the different simulations, but n does NOT have a spread of \sqrt{n} . This is because it is a subset of N . Even more, if one would compute n/N , the errors do NOT propagate by quadratic addition, because n and N are correlated. Hence getting p and its error is far from trivial.

We therefore limit ourselves to the question of estimating p from a single simulation. As discussed, in this case n and N are fixed known numbers. Hence we want to estimate p from n and N . Since the detector has simply a change p to detect a particle or not, the process of detecting a single particle is a Bernoulli experiment. Hence firing N particles is a binomial experiment. Hence we are dealing with a binomial distribution.

One word of caution: The probability to detect n particles, given that we know we fired N particles and given a known efficiency p is binomial:

$$P(n|p, N) = \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n} \quad (4.0)$$

But in our case we have that n and N are known, while p is unknown. This is the other way around. Hence we will use the likelihood of p , which is by definition the same as the probability distribution of n :

$$L(p|n, N) = \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n} \quad (4.0)$$

Computing the log-likelihood gives:

$$\ln L(p|n, N) = \ln \binom{N}{n} + n \ln p + (N - n) \ln(1 - p) \quad (4.0)$$

Differentiating to p and equating to zero gives:

$$\frac{d \ln L(p|n, N)}{dp} = \frac{n}{p} - \frac{N - n}{1 - p} = 0 \quad (4.0)$$

Solving this gives:

$$\frac{n}{p} = \frac{N - n}{1 - p}$$

$$\begin{aligned}
n(1-p) &= (N-n)p \\
n - np &= Np - npn = Np \\
p &= \frac{n}{N}
\end{aligned}$$

Hence the maximum likelihood estimator for p is

$$\hat{p}_{MLE} = \frac{n}{N} \quad (4.0)$$

which is not too surprising.

The error of \hat{p}_{MLE} is now given by the Fisher information [9] (the $1/\sigma$ is a typo in this reference):

$$\frac{1}{\sigma_{\hat{p}_{MLE}}^2} = -E \left(\frac{d^2 \ln L(p|n, N)}{dp^2} \right) \quad (4.0)$$

Computing the second derivative gives:

$$\frac{d^2 \ln L(p|n, N)}{dp^2} = -\frac{n}{p^2} - \frac{N-n}{(1-p)^2} \quad (4.0)$$

Now equation (4.1.7) is known as the Fischer information. To compute the error, one has to compute the expectancy of this Fischer information. This can be obtained by filling in $p = \hat{p}_{MLE}$ into equation (4.1.7). One obtains:

$$\begin{aligned}
\frac{1}{\sigma_{\hat{p}_{MLE}}^2} &= \frac{n}{p^2} + \frac{N-n}{(1-p)^2} \Big|_{p=\hat{p}_{MLE}} = \frac{n}{(n)^2} + \frac{N-n}{(1-(n/N))^2} = \\
&\frac{nN^2}{n^2} + \frac{(N-n)N^2}{(N-n)^2} = \frac{N^2}{n} + \frac{N^2}{(N-n)^2} = \frac{N^2 n + N^2(N-n)}{n(N-n)} = \frac{N^3}{n(N-n)}
\end{aligned}$$

Hence we obtain that:

$$\sigma_{\hat{p}_{MLE}} = \sqrt{\frac{n(N-n)}{N^3}} = \sqrt{\frac{\hat{p}_{MLE}(1-\hat{p}_{MLE})}{N}} \quad (4.0)$$

Hence this is the correct error for the efficiency of a detector.

Another approach to this problem is to not use MLE estimation of p , but Bayesian statistics. Basian statistics takes the likelyhood of p and interprets this as a probability distribution for p (after normalization). Hence by the Basian approach we get that:

$$p(p|n, N) = \frac{1}{V} \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n} \quad (4.0)$$

In order to get a valid probability distribution, one must divide by the normalization factor V , defined as:

$$V = \binom{N}{n} \int_0^1 p^n (1-p)^{N-n} dp$$

One must however be cautious with Bayesian statistics. They are always a combination of an a-priori distribution and an experiment. The experiment is in this case that we shoot N particles and detect n particles, so that's fine. In order for equation (4.1.7) to hold, one must however assume an a uniform distribution between 0 and 1 for p as a-priori distribution. This assumption is reasonable (the only symmetric distribution if you don't know anything about your detector yet). This assumption is however NOT unambiguous. One could argue that we do have information on NeuLAND, meaning we should have a different a-priori distribution, making equation

(4.1.7) no longer valid. In our case, we will however assume the uniform a-priori distribution of p and work with equation (4.1.7).

Now in order to calculate V , we will first rewrite equation (4.1.7) a bit:

$$p(p|n, N) = \frac{1}{V} \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n} = \frac{1}{V} \binom{N}{n} \cdot p^{\alpha-1} \cdot (1-p)^{\beta-1} \quad \alpha = n+1 \quad \beta = N-n+1$$

Now this is precisely a β -distribution, for which the pre-factor is uniquely fixed. Hence we should have that:

$$\begin{aligned} \frac{1}{V} \binom{N}{n} &= \frac{1}{B(\alpha, \beta)} = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} = \frac{\Gamma(N+2)}{\Gamma(n+1)\Gamma(N-n+1)} \\ \Gamma(x) &= \int_0^1 t^{x-1} e^{-t} dt \end{aligned}$$

For integers we know that:

$$\Gamma(k) = (k-1)!$$

Making our life very easy. We get that:

$$\frac{1}{V} \binom{N}{n} = \frac{\Gamma(N+2)}{\Gamma(n+1)\Gamma(N-n+1)} \Gamma N - n + 1 = \frac{(N+1)!}{n!(N-n)!} = (N+1) \cdot \binom{N}{n}$$

Hence we have obtained that

$$V = \binom{N}{n} \int_0^1 p^n (1-p)^{N-n} dp = \frac{1}{N+1} \quad \forall n$$

Hence our Bayesian probability distribution equals:

$$p(p|n, N) = (N+1) \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n} \quad (4.0)$$

By looking at the beta-distribution once more, we can compute the mean and variance of this probability distribution. This will then give us the Bayesian estimators for p :

$$\hat{p}_{\text{Bayes}} = \int_0^1 p \cdot (N+1) \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n} dp \quad (4.0)$$

$$\sigma_{\hat{p}_{\text{Bayes}}}^2 = \int_0^1 (p - \hat{p}_{\text{Bayes}})^2 \cdot (N+1) \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n} dp \quad (4.0)$$

Since we already know that equation (4.1.7) is exactly a beta-distribution with $\alpha = n+1$ and $\beta = N-n+1$, we already know exactly what the mean and variance are going to be:

$$\hat{p}_{\text{Bayes}} = \frac{\alpha}{\alpha + \beta} = \frac{n+1}{N+2} \quad (4.0)$$

$$\sigma_{\hat{p}_{\text{Bayes}}}^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} = \frac{(n+1)(N-n+1)}{(N+2)^2(N+3)} = \frac{\hat{p}_{\text{Bayes}}(1-\hat{p}_{\text{Bayes}})}{N+3} \quad (4.0)$$

Now the real question comes: which estimation is better? MLE or Bayesian? The good thing about MLE is that it doesn't bother with a-priori probability distributions. On the other hand, if you are 'Unlucky' and detect N or 0 particles, then the MLE estimator will give you also an efficiency of 0 or 1 with an error. But this is not correct, because this would mean theoretically that the efficiency could be 0 - error or 1 + error. This is obviously nonsense, since you know by beforehand that no detector can ever detect more particles than fired on it.

The Bayesian estimator does not have this problem. The minimum efficiency one could obtain here equals $1/(N + 2)$ and the maximum equals $(N + 1)/(N + 2)$. If we now compute the error for this situation, one obtains that

$$\sigma_{\hat{p}_{\text{Bayes}}}^2 = \frac{\hat{p}_{\text{Bayes}}(1 - \hat{p}_{\text{Bayes}})}{N + 3} = \frac{\frac{1}{N+2} \cdot \frac{N+1}{N+2}}{N + 3}$$

In the realistic case when $N \gg 1$, this gives $\sigma_{\hat{p}_{\text{Bayes}}} \approx 1/(N + 2)$. This means that the efficiency \pm its error will always stay within $[0, 1]$. This is of course a much more physical situation than the for the MLE estimator.

Therefore the Bayesian estimator is better for the case of detector simulations. We do point out that in the asymptotic case $N \gg 1$, the MLE and Bayesian estimators become nearly equal. This justifies the assumption of an a-priori uniform distribution for p in the Bayesian approach. Therefore the problem of the Bayesian approach is solved by the MLE estimator and we can safely use the Bayesian estimator. Hence the correct way to compute a detector efficiency is:

$$p = \frac{n + 1}{N + 2} \quad (4.0)$$

$$\sigma_p = \sqrt{\frac{p(1 - p)}{N + 3}} \quad (4.0)$$

As a final aspect it is interesting to try to determine a-priori how many particles N you should fire if you want your efficiency to be known up to 1%. Since a-priority p is unknown, we will use the maximum value of p in the interval $[0, 1]$ as a worst-case scenario:

$$\frac{d}{dp} p(1 - p) = 1 - 2p = 0 \rightarrow p = \frac{1}{2} \rightarrow p(1 - p)|_{p=\frac{1}{2}} = \frac{1}{4}$$

Hence the error on efficiency can at most be:

$$\sigma_p = \sqrt{\frac{1}{4(N + 3)}}$$

So now let us solve

$$\begin{aligned} \sigma_p &= \sqrt{\frac{1}{4(N + 3)}} = 0.01 \\ \frac{1}{4(N + 3)} &= 10^{-4} \\ 4(N + 3) &= 10^4 \\ N &= 2497 \end{aligned}$$

Hence to achieve an error less than 1% in your estimation, at least $N = 2500$ particles are required.

If the efficiency is determined for different energy bins and all bins should have an error of at most 1%, then obviously one requires $N = 2500n_{bins}$. Or more generally:

$$N \geq \frac{n_{bins}}{4\epsilon^2}$$

Where ϵ is the required efficiency per bin. For 1% and 50 bins we obtain $N \geq 125000$. For a 2% efficiency and 40 bins, this would however be a lot better, namely $N \geq 25000$.

4.2 Documentation on R³Broot

<http://web-docs.gsi.de/~kmiki/wiki/index.php?R3BROOT%2FHowToBG>

www.r3broot.gsi

4.2.1 Activating the digitizer

First run an R3Broot simulation as usual. Then go to `<<path to>>/r3broot/macros/r3b/neuland/monitoring`. You will find a macro there called `DigiMonitoring.C`. Modify the input parameters in such a way that it runs on the output `.root`-files of the R3Broot simulation. Then run this macro AFTER the usual simulation. This creates a second `.root`-file that contains also a `cbmsim`-tree. This tree contains the output of the Digitizer as leafs. for up to date information, `git branch` to see the branches and `git checkout master` to switch to the master branch, `git checkout dev` switches to development branch.

The leafs are named LandDigi.<something>. Labels are:

```

// R3B NeuLAND digitizer documentation.
// Information is given to me by Jan Mayer, the writer
// of this software.

Int_t const N_hits = >>> // number of hits registered in entire NeuLAND during one single event. It varies from event-to-event.

UInt_t LandDigi_funiqueID[N_hits]; // Standard output of R3Broot ==> No physical meaning
UInt_t LandDigi_fhits[N_hits]; // Standard output of R3Broot ==> No physical meaning
Int_t LandDigi_fpdmchid[N_hits]; // Number of the bar from NeuLAND that registered the hit. Remember the mapping table of NeuLAND!
Double32_t LandDigi_fddc[N_hits]; // TOC output of the first PMT as coming from TACUILA [ns]
Double32_t LandDigi_fTdc[N_hits]; // TOC output of the second PMT as coming from TACUILA [ns]
Double32_t LandDigi_fns[N_hits]; // Reconstructed time at the hit: (leftPMT.GetTDC() + rightPMT.GetTDC()) / 2. - fPaddleHalfLength / fcMedium; [ns]

Double32_t LandDigi_fOdc[N_hits]; // ODC output of the first PMT as coming from TACUILA [MeV, but uncalibrated!]
Double32_t LandDigi_fOdc[N_hits]; // ODC output of the second PMT as coming from TACUILA [MeV, but uncalibrated!]
Double32_t LandDigi_fN[N_hits]; // Reconstructed energy deposition at the hit: It is computed as sqrt(leftPMT.GetEnergy() * rightPMT.GetEnergy()) [MeV, nut uncalibrated!]

Double32_t LandDigi_fx[N_hits]; // Hit X-coordinate determined with TOF information from TDC: Computed in Paddle Structure as (rightPMT.GetTDC() - leftPMT.GetTDC()) / 2.*fcMedium; [cm]
Double32_t LandDigi_fy[N_hits]; // Hit Y-coordinate determined with TOF information from TDC: Computed in Paddle Structure as (rightPMT.GetTDC() - leftPMT.GetTDC()) / 2.*fcMedium; [cm]
Double32_t LandDigi fz[N_hits]; // Hit Z-coordinate determined with TOF information from TDC: Computed in Paddle Structure as (rightPMT.GetTDC() - leftPMT.GetTDC()) / 2.*fcMedium; [cm]

```

Figure 4.9: NeuLAND Digitizer documentation

In order to run the clusterfinding algorithm for NeuLAND, go to the `DigiMonitoring.C` macro and after the lines `run->AddTask(new R3BNeulandDigitizer());` and `run->AddTask(new R3BNeulandDigiMon());`, add the following lines: `run->AddTask(new R3BNuLandClusterFinder(0));` and `run->AddTask(new R3BNeutronCalibr2D());`. Now the root-file that is created will contain two separate trees: one containing the Digitizer leafs, and one containing a new set of Cluster-leafs. Notice that you can only run this algorithm AFTER you ran the digitizer (or on real data...). The documentation on the names are:

Figure 4.10: NeuLAND ClusterFinder documentation

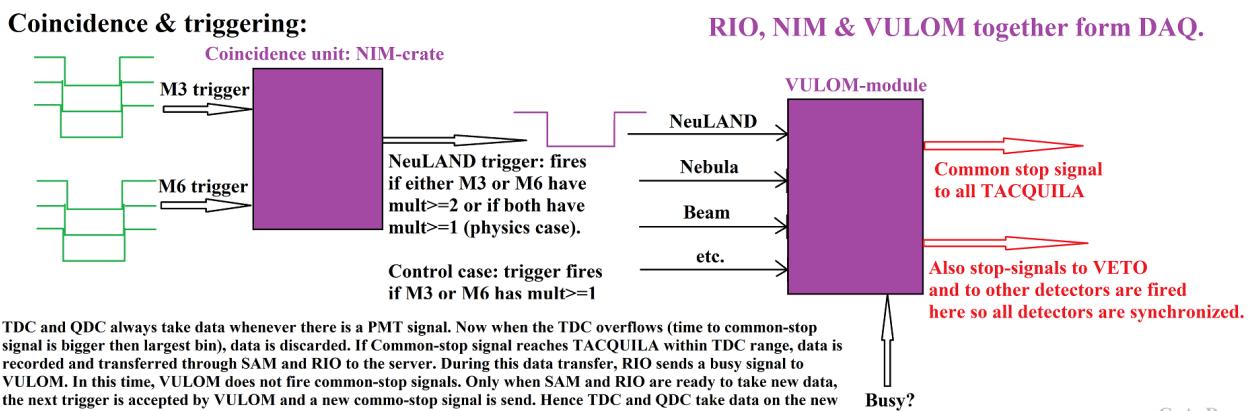
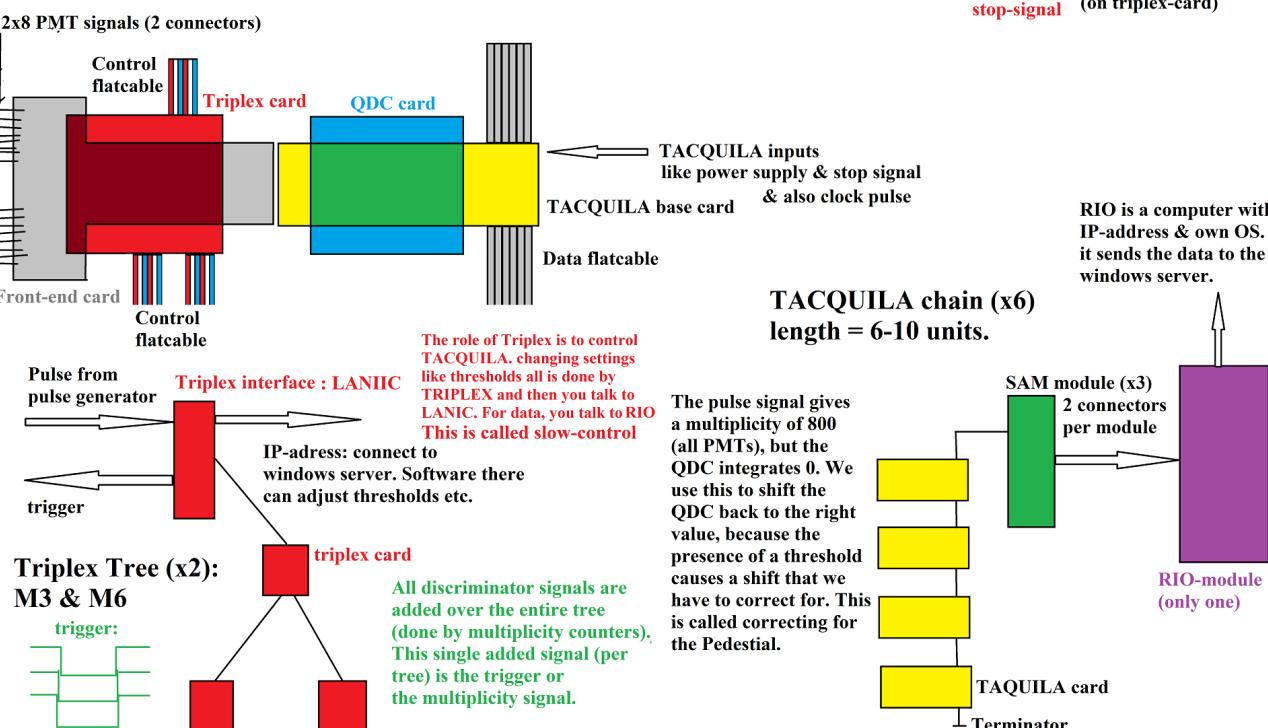
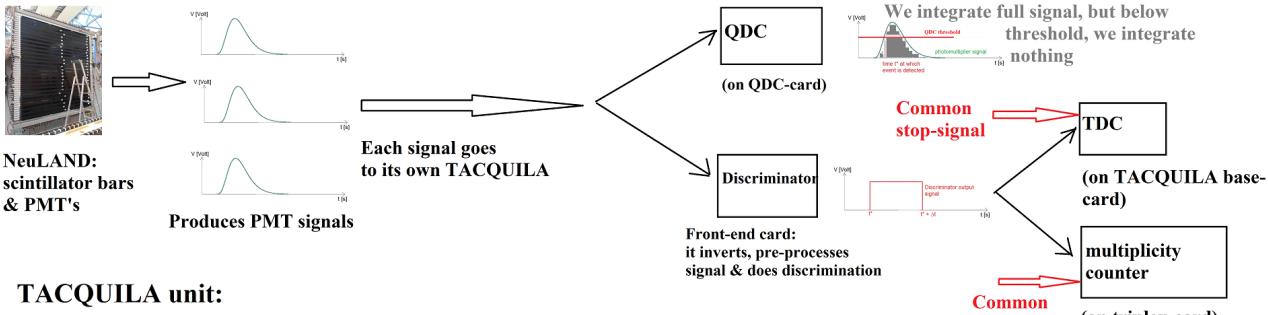
4.3 Full infrastructure of the NeuLAND detector

Most of this content is also explained on the SAMURAI wiki page. You can log in to the general SAMURAI wiki page with username `samurai21` and password `doublymagic28o`. In order to be able to log in to the NeuLAND SAMURAI wiki page, you have to ask Mr. Kondo for an account. My account has username `Christiaan` and the same password as my Linuxslaaf virtual machine with Ubuntu 14.01 LTS.

4.3.1 Electronic infrastructure of NeuLAND

This infrastructure is best illustrated with figure 4.11.

NeuLAND only produces photomultiplier signals. These signals are read out by TACQUILA units. a TACQUILA unit consist of a front-card, a TRIPLEX card, a QDC card and a base card. first the front-end card pre-processes the signal (inverts it from negative to positive, etc). Then it performs the discrimination. It then sends the analogue signal to the QDC-card to perform the QDC and the discriminated signal is send to the base-card for the TDC and to the triplex-card for multiplicity counting. QDC-card does only QDC. base card does TDC and data readout. TRIPLEX is used to control the TACQUILA and it does the multiplicity counting.



C. A. Douma

Figure 4.11: Electronic infrastructure of NeuLAND

All TRIPLEX cards are organized in a tree-structure. On top there is a TRIPLEX interface that talks to the windows-server called **ribfana**. We have 2 separate trees, each with its own interface. The multiplicity signal is now added up over all TRIPLEX cards in a tree and this added signal (with is same as discriminator signals and height is proportional to the number of PMTs that fired) is called the multiplicity signal. This signal is output of the TRIPLEX interface. inputs are a pulse generator and the slow-control: commands from the windows server to modify the TACQUILA settings like thresholds.

The pulse signal from the TRIPLEX interface is passed down the tree to all QDC cards. A QDC-card integrates the entire signal, but only if the peak comes above the threshold. If it stays below the threshold, nothing is integrated. Therefore it is unclear what integral 0 actually means. Therefore we fire a pulse to activate QDC but integrate nothing. This should correspond to a QDC of 0 and all other signals have to be shifted accordingly. This is NOT performed on the TACQUILA cards. Actually, TACQUILA treats the pulses from TRIPLEX as photomultiplier signals. Hence these signals will be part of the data. They will have multiplicity 800 (the number of bars), so after the experiment one can recognize these events and then use them to perform the QDC shift. This is called subtracting the pedestal.

All TACQUILA base cards are organized in chains of length 6, 8 or 10. Data is passed along the chain, until it reaches the end, where a so-called SAM-module is connected. Each SAM catches data from 2 chains. We have 3 SAMs in total. the SAM-modules catch the data and pass it to the RIO computer (only one in the setup), which passes it to the windows server.

The multiplicity signal from both trees is fed into a coincidence unit. This unit only fires if the multiplicity is ≥ 2 , hence at least 2 PMTs have fired. This can happen either if one tree has multiplicity ≥ 2 and then the other doesn't care, or if both trees have multiplicity ≥ 1 . If this condition is met, a master trigger signal is fired. This signal is fed into a VULOM module, which catches all master triggers from all detectors involved. It also catches so-called *busy?* signals. If the modules catches a trigger and no *busy?* signals are registered, it fires a common-stop signal to he TAQCUILA base cards.

Now, TDC and QDC on TACQUILA always take data if there is a PMT signal. However, if the TDC of the TACQUILA measures time until it overflows (if the max. number of bits available is insufficient to count the time), both the TDC data and the QDC data is discarded. If, on the other hand, a common-stop signal fired by the VULOM is registered, TDC stops measuring time (which is before it overflows) and only then, the data is transferred through the chain and recorded by SAM and RIO and written to the server.

Now if SAM and RIO are busy recording data, RIO gives a *busy?* signal to VULOM, so that no common stop signal is fired and new data is discarded. When recording the previous data is done, SAM and RIO are ready to take new data and no *busy?* signal is sent. Then at the next trigger, a common stop signal is send and new data is red out. Hence, there is a dead-time in this detector: the time RIO and SAM need to process the data.

Notice also that all detectors give master trigger and *busy?* signals to a single VULOM-module. Only when all of them are ready to take new data, new common-stop signals are fired again. This makes sure all detectors are perfectly synchronized.

The box left to NeuLAND has 4 sets of 6 TACQUILAs. They are organized in 3 chains and one TRIPLEX tree. The box right to NeuLAND has 2 boxes of 10 TACQUILAs and one of 8. They are organized in one tree and 3 chains. All the other electronic modules are also at the right of NeuLAND: 3 SAM-modules which each take 2 chains, 2 triplex interfaces (one is on the left at its own tree), one RIO module, a coincidence unit and a VUMON module. VULOM coincidence (NIM-crate) and RIO are together called DAQ. Notice that the VULOM also talks to the other detectors, while RIO and coincidence do not.

Each TACQUILA unit can process the signals of 16 photomultipliers. All photomultipliers, tree structures and chain structures are specified in the NeuLAND mapping tables.

Separate from this infrastructure, each photomultiplier is also connected to a high-voltage channel that powers the tube. The high-voltage is controlled by HV-modules that each power 25 photomultipliers. The modules are connected to the windows server through an application called **telnet**. This way, one can adjust the high voltage of each channel separately. One can also switch the voltage on and off through this server. In this case, the voltage will be *ramping*: slowly turning it on or off. This is because a sudden jump will damage the photomultiplier.

After the data arrives at the windows server, this server writes it to an .lmd-file. The next subsection discusses how the TDC on the TACQUILA precisely works. The subsection after that discusses the software stage that

comes after this hardware infrastructure.

4.3.2 Operation principles of the TDC and QDC on TACQUILA

The photomultiplier tubes are grouped in sets of 16 and then connected to a single TACQUILA card. This card performs 16 TDC and 16 QDC tasks. A mapping table is then used to guide all photomultiplier tubes to their respective TACQUILA card in these groups of 16.

Now, A block pulse is send to all TACQUILA units (to the base-cards) with a period of 25 ns. This common pulse is used to measure time so that the TDC can be generated. It works according to figure 4.12.

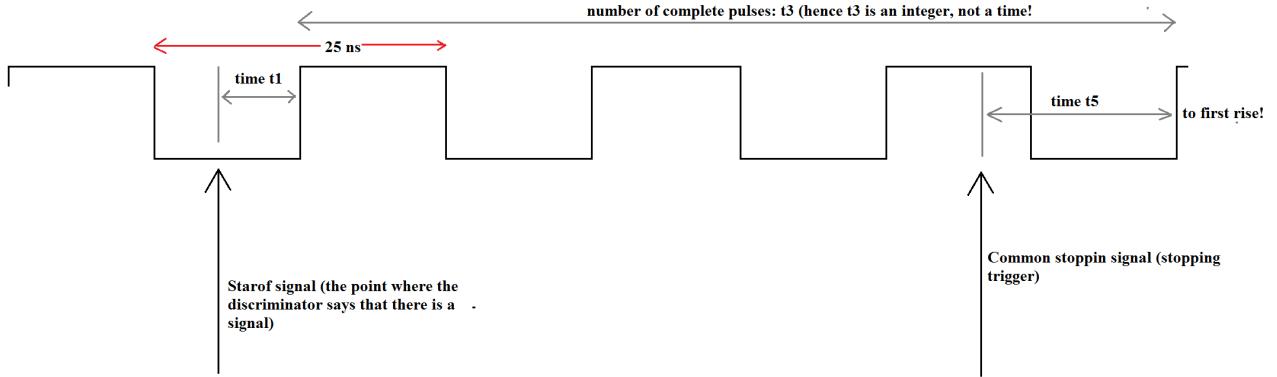


Figure 4.12: Time measurement by TACQUILA

In figure 4.12 the times t_1 , t_3 and t_5 are used for the photomultipliers on the bottom and the times t_2 , t_4 and t_6 are used for the photomultipliers at the top. For horizontal planes, we use t_1 , t_3 and t_5 for the right multipliers and t_2 , t_4 and t_6 for the left ones. This is because the NeuLAND labeling is from bottom to top and from right to left (if you stand with your face towards the first plane).

If the common-stop signal arrives before the TDC overflows, then a single number for t_1 , t_3 and t_5 (and also for t_2 , t_4 and t_6) is produced and processed. This is called one event. This is repeated for each PMT that is hit. This is the full TDC output. Note that t_1 , t_2 , t_5 and t_6 range from 0 to 25 ns because the starting and stopping signals will be uniformly distributed over this pulse. Hence their spectra will be square blocks. For t_3 and t_4 it is more complicated. They will not be square blocks and they are integers. They contain the amount of pulses between start and stop and their range depends on the dead time of all detectors involved and on the TDC range.

In the root-files, these branches are therefore labeled as `Nnp03t05` for example. This means that we talk about single-plane 3 and time quantity t_5 . Of course one should notice that one single plane still contains 50 bars, so each of these branches contains an array of size 50 for each event.

For the QDC (TACQUILA uses QDC, not ADC), the signal from the photomultiplier is integrated between the moment the TDC starts counting time and the common-stop signal.

TACQUILA has a certain energy threshold (or voltage threshold if we measure the photomultiplier signal in volts) and only if the signal comes above the threshold, it is integrated. In this case the full signal is integrated. Below the threshold, nothing is integrated. Then the pulses from the TRIPLEX are used to determine the zero-point. This is explained in the previous section.

The branches from QDC are labeled `Nnp03e01` and `Nnp03e02` for example. Again `Nnp03` stands for single plane 3 and `e01` stands for the first photomultiplier.

There is also a quantity called `Nnp03i` which states which bar in the single plane was hit. This is also an array of size 50. The label `Nnp03n` stands for the multiplicity of that signal. This is only a single number. A multiplicity of unity means that a single bar is hit (hence 2 photomultipliers were hit).

Notice that all these quantities are written directly to the `.lmd`-file without performing any correction on it or what so-ever. This is purely what the photomultiplier saw.

4.3.3 Software structure of NeuLAND

The software structure we discuss here is for NeuLAND running stand-alone. In this case, it takes data on cosmics. This is useful to test the detector. In this case, the own VULOM-module on NeuLAND is used, which responds only to the RIO *busy?* signal and to the NeuLAN master trigger. In case of the experiment, a different VULOM-module is used that controles the entire experiment and keeps the detectors synchronized. In this case, however, the software structure is different because RIKEN uses its own system instead of ROOT. Hence we will discuss the ROOT-related structure for NeuLAND stand-alone here.

The most raw data coming directly from RIO is written in .lmd-files on the windows server. Now one can simply connect to this server with a linux PC and run scripts to process this data. In order to connect to this server, one needs the following information: **username = neuland**

computer name = ribfana02.riken.jp

If you use a guest account at RIKEN, use only **ribfanaas** computer name. If you use **eduroam**, use the full computer name.

password = samuraineuland

You can login from your personal Linux machine with the following command: **ssh -Y neuland@ribfana02.riken.jp**

Then enter the appropriate information you need.

In order to log in at the Samurai NeuLAND wiki page, just google the page. Use **Christiaan** as username and use the same password as you use for your own Linux Ubuntu 14.04 LTS virtualbox machine.

Taking data is done with the program called **empty** which is located on the server at **/home/neuland/empty/empty**. You can run the program with the following command:

./empty --trans=sdaq28 --max-events=1000 --output=/home/neuland/.../rawdata.lmd

The option **--trans** tells the program **empty** which computer to connect with **sdaq28** is the RIO-module from NeuLAND. Hence this option tells the program to take data from NeuLAND. The option **--max-events=1000** specifies the maximum number of events to take data for. In this case: 1000 events. **-output** specifies the name of the .lmd-file where the data should be saved in. Notice that the entire path on the server should be specified.

After taking the data, the first step is to unpack the data. The command to unpack the data is the following:
./paw_ntuple rawdata.lmd --paw-ntuple=RAW, NNP, unpacked.root

Of course **rawdata.lmd** is the .lmd-file you want to unpack. **unpacked.root** is the filename of the .root-file the unpacked data is written to. The option **RAW** means that the .lmd-file is only converted to a .root-file without doing anything to it. This .root-file will then contain a ROOT-Tree where the branches have the labels discussed in the previous section. One also has the options **TCAL** and **SYNC** instead of **RAW**. These options will be discussed in a minute. The option **NNP** specifies that we should unpack data from NeuLAND. This is because the unpacker can also handle a lot of different detectors. The unpacker is named **paw_ntuple** and has file-location **/home/neuland/land02/s438c/paw_ntuple** on the windows-server.

Note however that if the path to the lmd-file does not contain the name of an experiment (say a directory like **s483c**), the unpacker will not work. This is just a stupid issue of the unpacker.

If the option **TCAL** is called, then all functions from **RAW** are performed. In addition, the signals with multiplicity 800 are removed from the data and the QDC-signals are shifted according to what these removed signals tell us. Hence the pedestal subtraction is now performed. Also the times *t3* and *t4* are converted from integers to nanoseconds.

If the option **SYNC** is called, then in addition to everything **TCAL** does, the different times *t1-t6* are combined into a single time for each photomultiplier. Hence the computation that is performed is:

$$t = t1 + t3_{\text{converted to ns by TCAL}} - t5$$

And a similar computation is performed for *t2*, *t4* and *t6*. The results are stored in *t1* and *t2*. The other times *t3,..,t6* are left unchanged by this program.

At this point the unpacker is done. The idea is to add more options to the unpacker in the future, like the synchronization of the different PMTs within NeuLAND. At this time these options however do not work well

yet. Therefore one now has to load the following script:

`/home/neuland/Nearline/neuland2.C`

This script contains several functions to process the data further, among which are functions to synchronize all the different PMTs.

4.3.4 Synchronization of the photomultipliers of NeuLAND

For each bar, the times registered by the photomultipliers (the TDC values) obey the following formulas:

$$t1 = t_{interaction} + x_{interaction}/v_{signal} + t_{offset,1}$$

$$t2 = t_{interaction} + (L_{bar} - x_{interaction})/v_{signal} + t_{offset,2}$$

Note that the t_{offset} can be different for each photomultiplier. Now synchronization of all photomultipliers means that we have to correct for all these different offsets.

We begin with synchronizing the two photomultipliers belonging to one bar with respect to each other (the so-called T_{diff} calibration). We repeat this for all bars. Given the times $t1$ and $t2$ one can make the following computations:

$$\begin{aligned} T_{pos} &= \frac{t1 - t2}{2} = x_{interaction}/v_{signal} + \frac{1}{2}t_{offset,1} - \frac{1}{2}t_{offset,2} \\ T_{int} &= \frac{t1 + t2}{2} = t_{interaction} + L_{bar}/v_{signal} + \frac{1}{2}t_{offset,1} + \frac{1}{2}t_{offset,2} \end{aligned}$$

Now if we add a parameter T_{diff} to the times $t1$ and $t2$ in the following way:

$$t1_{syn} = t_{interaction} + x_{interaction}/v_{signal} + t_{offset,1} + T_{diff}$$

$$t2_{syn} = t_{interaction} + (L_{bar} - x_{interaction})/v_{signal} + t_{offset,2} - T_{diff}$$

Then we obtain that

$$\begin{aligned} T_{pos,syn} &= \frac{t1 - t2}{2} = x_{interaction}/v_{signal} + \frac{1}{2}t_{offset,1} - \frac{1}{2}t_{offset,2} + T_{diff} \\ T_{int,syn} &= \frac{t1 + t2}{2} = t_{interaction} + L_{bar}/v_{signal} + \frac{1}{2}t_{offset,1} + \frac{1}{2}t_{offset,2} \end{aligned}$$

Then we can choose T_{diff} in such a way that it cancels all offsets in $T_{pos,syn}$. We do this by plotting a histogram of $T_{pos,syn}$ versus counts of all events that we have for this bar and then choose T_{diff} in such a way that the spectrum ranges exactly from 0 to L_{bar}/v_{signal} . (Or to be symmetric around zero, depending on which coordinate system you use). Note that the spectrum should be approximately a square block. In our formulas this means to set $T_{diff} = -\frac{1}{2}t_{offset,1} + \frac{1}{2}t_{offset,2}$. This will give:

$$t1_{syn} = t_{interaction} + x_{interaction}/v_{signal} + \frac{1}{2}t_{offset,1} + \frac{1}{2}t_{offset,2}$$

$$t2_{syn} = t_{interaction} + (L_{bar} - x_{interaction})/v_{signal} + \frac{1}{2}t_{offset,1} + \frac{1}{2}t_{offset,2}$$

This means that the times now have a common offset. Hence the photomultipliers are now synchronized with respect to each other. For the compound times we obtain that:

$$\begin{aligned} T_{pos,syn} &= \frac{t1 - t2}{2} = x_{interaction}/v_{signal} \\ T_{int,syn} &= \frac{t1 + t2}{2} = t_{interaction} + L_{bar}/v_{signal} + \frac{1}{2}t_{offset,1} + \frac{1}{2}t_{offset,2} \end{aligned}$$

Since we know that $x_{interaction}$ ranges from 0 to $L_{bar} = 250$ cm, we can now fit the spectrum for $T_{pos,syn}$ and obtain v_{signal} in the bar: the effective speed the scintillator light signal travels through the bar (smaller than c due to all reflections).

Notice also that the offsets of the photomultipliers also take different cable lengths etc. into account. Hence this is also synchronized now.

4.3.5 Synchronization of the different bars

Now that the photomultipliers of one bar are synchronized with respect to each other, we have to synchronize the different bars with respect to each other. In order to do this, consider a single double-plane of NeuLAND. Let $T_{int,syn,hor}^i$ be the interaction time of the i -th bar in the horizontal plane of the double plane. Let $T_{int,syn,vert}^j$ be the interaction time of the j -th bar in the vertical plane of the double plane. Then define the following matrix:

$$A_{ij} = T_{int,syn,hor}^i - T_{int,syn,vert}^j$$

Notice that the matrix A_{ij} is only defined for coincidence events in the respective bars i and j . We do however use all events of the experiment at once to get good peaks.

Now if we want, for example, to synchronize the vertical bars in this double-plane, we compute the following matrix:

$$V_{ij} = A_{ij} - A_{i,25}$$

Since 25 is defined to be the reference bar. Now we average the matrix V_{ij} over all horizontal bars:

$$F_j = \frac{1}{50} \sum_i V_{ij}$$

And then the vector F_j contains the offsets of all vertical bars that should be subtracted from all the times $t1$ and $t2$ of all photomultipliers in that single plane.

Of course all horizontal planes can also be synchronized with the same procedure. So now all that is left is to synchronize all single planes with respect to each other. For this we use coincidence hits between successive bar-25 of single plains to make sure that all offsets are exactly the same. Then the entire NeuLAND detector is synchronized.

Notice that for all cosmics coming from one way there will always be another coming from the other way, so one can synchronize the peak position at exactly zero, which is the procedure that we described. The peak is formed by all events from the experiment. This will however give a little wrong results for bars at the end of a plane, because some of those cosmics from the other way cannot be detected because the detector ends there. This effect is however small.

If, however, you use beam data, then you can still use the same procedure for synchronizing the single planes. But now you cannot center the time peaks of coincidence hits around zero. You have to set the peak at $5 \text{ cm}/v_{beam}$ and v_{beam} must be given to you. This is because coincidence hits are in different planes, hence the beam needs travel time between the planes.

The synchronization of the bars is illustrated in figure 4.13.

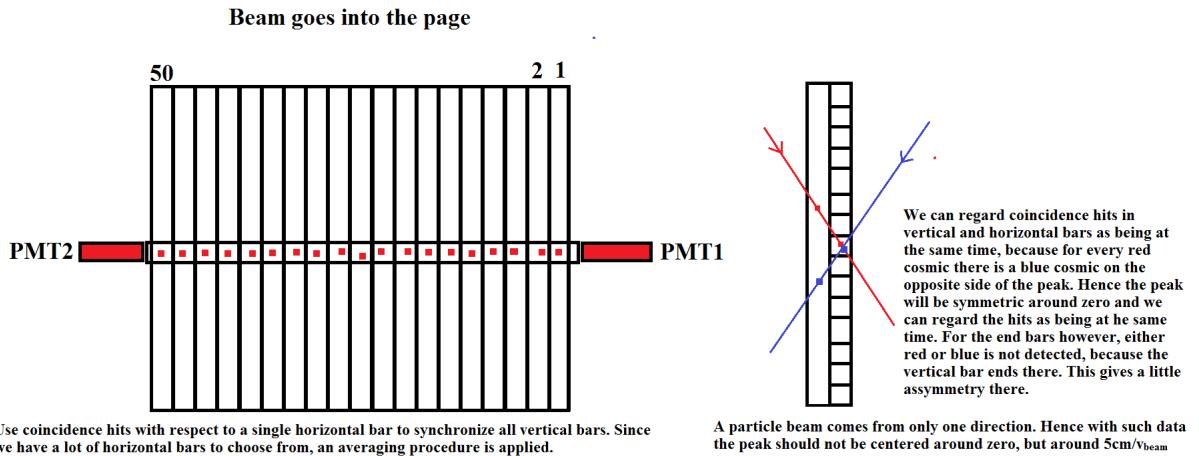


Figure 4.13: Coincidence hits for the synchronization process.

Of course NeuLAND is now synchronized. But the common offset of the entire detector is not accounted for yet. This has to be done with calibration. The synchronization of the different bars (performed on the actual data!) is performed by the function `calibrate_tsync()` in `neuland2.C`.

4.3.6 Correcting QCD for the light attenuation in the scintillator bar

After the synchronization of the data is done, one has to correct for the light attenuation in the bars. Some of the light is lost while it travels from the particle hit to the photomultiplier and we have to correct for that. From T_{pos} one can get the exact position x inside the bar. From fitting the entire spectrum (of that single bar) of T_{pos} at once, one obtains the propagation speed of the signal and then for a single event one can recover x from T_{pos} . Hence the position of the hit inside the bar is known.

The procedure of correcting for this is illustrated in figure

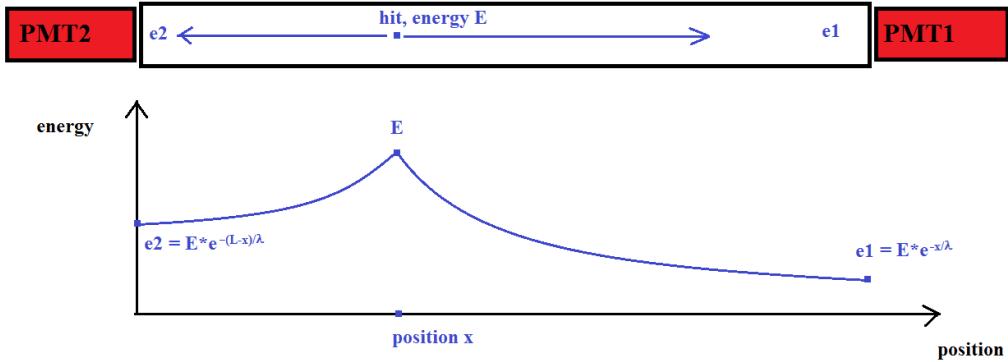


Figure 4.14: Correction process for light attenuation.

If the position x is known, correction for the attenuation is actually pretty easy. It is known that the attenuation of light inside a scintillator bar is exponential. Hence if the particle lost an energy E with its interaction, one will measure

$$e1 = E \cdot e^{-x/\lambda} \quad e2 = E \cdot e^{-(L-x)/\lambda}$$

Now the attenuation length λ is not known, but x is. Also the scintillator bar length L is known. And of course we have $e1$ and $e2$ from our QDC. Hence we have two equations and two unknowns: λ and E . We can solve for this and obtain the real energy lost by the interaction of the particle.

Unfortunately not all PMT tubes and QDC have the same sensitivity. We know by hand however that if a hit has taken place in the middle of a scintillator bar, the attenuation to both sides must be equal. Hence we first multiply $e1$ by a number E_{diff} and divide $e2$ by this number. This number is determined in such a way that $e1$ and $e2$ become exactly equal for hits in the middle of the bar. Coincidence hits with vertical bars are used for this procedure. This procedure is called E_{diff} calibration.

After this procedure there is probably also something similar to synchronizing the different bars: making the sensitivity for the different bars equal. I do not know this for sure and if this is the case, I do not know the details. Just check it! Aligning this sensitivity is not part of E_{diff} calibration, but if it is performed, there is probably some software for this, probably in `neuland2.C`.

After this is done, we can make the fit

$$e1 = E \cdot e^{-x/\lambda} \quad e2 = E \cdot e^{-(L-x)/\lambda}$$

and solve for E and λ . Notice that during E_{diff} calibration, the factor $e^{-L/\lambda}$ is cancelled out so that we obtain that $E = \sqrt{e1 \cdot e2}$. A function for this entire procedure exists in `neuland2.C`.

Then after synchronization and attenuation correction, we have for each hit the position x inside the bar, the time of the hit t and the energy deposition E . Of course we also know which bar fired and what its coordinates in space are. The hits are organized event-by-event (one event is one trigger!). Now the analysis can begin with particle identification, track reconstruction, etc.

4.3.7 Operation principles of TAMEX

At the ImPACT experiment, QDC and TDC are performed by TAQUILA electronics. at FAIR, this will be replaced by TAMEX. TAMEX does not simply perform QDC and TDC, but it records both the starting time

of a PMT signal (which is the same as TDC does) and the end time of the PMT signal. Since the length of a PMT signal is related to the integral, this information provides both TDC and QDC.

The advantage of this method is that this way, TAMEX can store multiple PMT pulses within a single event (which means: between two triggers). This gives TAMEX the ability to identify multi-hits, while TACQUILA can only store the first PMT pulse that arrives after the trigger (or the non-overflowing TDC signal for a stopping trigger). Probably TAMEX will also be equipped with a QDC function for calibration purposes and to establish the precise relation between the pulse integral and length. Unfortunately TAMEX is still under development at FAIR. Hence we use TACQUILA for now.

4.3.8 Monitoring of NeuLAND during IMPACT experiment

first you log into the ANAROOT server with the following command: `ssh -Y tm1510@ribfana02.riken.jp`. Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. The password is `obakiller`. Then you give the command `anarootlogin`. Then input your name. My user name is `Christiaan`. If this is a new user, it will say this and create a new directory for you if you want. If you are an existing user, it will not report anything. No password is required. for login into anaroot.

Then go to the directory `/home/tm1510/exp/exp1510_tm/users/anaroot/hanstt/` which is the directory of Hans. Then start root with giving the command `root -l`. Then load the monitoring macros by giving the following command: `.L macros/OnlineMonitor_neuland.cc+` and fire the macros with the command `OnlineMonitor_neuland f`. The last letter `f` is just the name of the class you declared and it doesn't really matter. Any name will do just fine.

Now 8 histogram windows are opened that are auto-updated every 2 seconds. You should just look at those histograms to make sure all of NeuLAND is performing well. Especially the T_{diff} histogram is important. It takes a while to generate the histograms on startup, so do not be alarmed if it doesn't seem to work. Wait at least 60 seconds.

In order to monitor the high voltage slow-control, log into the windows server with the command `ssh -Y neuland@ribfana02.riken.jp`. Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. Go to the directory `/home/neuland/NEULANDHV/` and run the script `startGUI.sh` which is located there. This will fire up the GUI for slow-control of high voltage and it is pretty easy to use.

In order to watch the triggers and dead times, go into the windows server with the command `ssh -Y neuland@ribfana02.riken.jp`. Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. Go to the directory `/home/neuland/` and give the command `screen -x daq`. This will give you a terminal output screen to monitor triggers and dead times. By itself, the screen is in view mode. By selecting `CTRL+a` you go to control mode. Then you can press a number between 0 and 5 to switch between the different screens. also option `q` and `esc` are useful to remember to return to the default view mode. After you selected a screen, you are back in view mode again. Screen number 1 is used for monitoring triggers and dead times.

If triggers or multiplicities become zero and/or dead time becomes infinite this means no data is taken and an expert should be called **IMMEDIATELY!**. If a channel in, for example, the T_{diff} histogram does not take data, try to reboot the high voltage. If this does not help, not much can be done about it. Hence it is not required to call an expert immediately (because he/she cannot do much about it) but the problem should be reported in the official log.

In order to watch the oscilloscope, open Internet Explorer (in Windows!) and enter in the address line: 172.27.230.102. You can compare this with the older signals by logging into the windows server with the command

`ssh -Y neuland@ribfana02.riken.jp`. Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. The password is `samuraineuland`. Then go to `/home/neuland/NEULANDLV/logger/`. Older pictures are saved here. Compare the pictures on a regular basis and if they are not more or less the same, Hans or Igor should be called **IMMEDIATELY!**

If the histogram `TDiff VS bar` has jumps in it, try to increase the energy threshold by 5 units in that channel. This can be done in the following way: log into the windows server with the command `ssh -Y neuland@ribfana02.riken.jp`. Again, the `riken.jp` is not required for the guest-network, but it is re-

quired for eduroam. The password is `samuraineuland`. Then navigate to the following path: `/home/neuland/neuland/` Then start root with `root -l`. Then load the macro for TRIPLEX control with the following command: `.L triplex.C++` Then you actually connect to the TRIPLEX tree with the command `read_Mapping()` A lot of stuff will be printed that you do not need, but the connection you do need. That is why you have to fire this function first. Then you can obtain the current value of the energy threshold by giving the command `get_thr_single(5,31,2)` This means that you access layer 6 and bar nr. 31 in that layer and PMT nr. 2 Of course you can change the numbers according to your needs. Then the value of the energy threshold is printed. Now to change this value, give the command `set_thr_single(5,31,2)` The program will now ask for typed input. You can then type the new value and by hitting enter the new value will be set. **BE CAREFUL!** Only do this is if you have to and make a note in the log! You will see old and new value printed on the screen once more. That's it!

The official logbook of the experiment is kept electronically on: <http://ribf-exp.riken.jp/elog/SAMURAI-21/?fail=1> Log in with `Christiaan` and Linux password. This also is the same for the Samurai NeuLAND wiki page. Write any special things there. This is, but for the entire experiment.

For quick analysis on a `root[11] TTree* t = f->Get("nl")` you can do the following to get quickly a picture:

```
root [12] TCut t = "nl_bar[0] == 25"
root [13] nl->Draw("(nl_tcal0[0]+nl_tcall1[0])/2 - (sbt1_t0[0]+sbt2_t0[0])/2 : (nb_tcal0[0]+nb_tcall1[0])/2 - (sbt1_t0[0]+sbt2_t0[0])/2>>h(1000,-650,-450,1000,-1500,-1000)",t&& "", "colz")
// colz=plotting option.
// t&&="" ==> Gives you the cut condition.
// >>h() gives you the histogram with bins and ranges you put the plot in.
// first string is what should be on the axis.
// You should give indices to the leafs! else, you will just add up everything and you do not know what you are looking at anymore.
```

Figure 4.15: Terminal 1 line analysis on TTree

4.3.9 Use of the upgraded algorithm for monitoring histograms

Now the methods stated above work fine, but the histograms are quite slow and have poor resolution. Therefore the algorithm to plot the histograms is replaced by an upgrade. Hence the old method probably no longer works, but we will now explain how to install and use the new algorithm.

The first thing required to use this program is to add the public ssh key of your Linux system to the ANAROOT-server. If you do not have such a key, you have to generate one first. In order to do this, first give the command `cd` to return to the default directory. Then give the command `ssh-keygen`. Some questions will be asked. Press `ENTER` on all of them to use the default settings. Your public ssh key is generated and saved in the file `~/.ssh/id_rsa.pub`. You also generated a private key, but you are not supposed to do anything with that.

Public key is for others to decript your files and Private is for you to encrypt your files. There is no worry that someone else can log into your PC if he possesses your ssh keys. This is impossible. If someone else wants to log into your computer, **you** have to add **their** public ssh key to **your** allowed list. And even then, this other person still needs the password. But as long as you do not add an ssh key to your own list of allowance, your own computer is safe.

Now log into the ANAROOT server with the following command: `ssy -Y tm1510@ribfana02.riken.jp`. Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. The password is `obakiller`. Go to the directory `/home/tm1510/.ssh/` and open the file `/home/tm1510/.ssh/authorized_keys` with your favorite text editor. `emacs` performs nicely. Open a second terminal and go (in your own PC) to the file `~/.ssh/id_rsa.pub` and open it with your favorite text editor. Then copy all content from the file `~/.ssh/id_rsa.pub` and add it onto a new line inside the file `/home/tm1510/.ssh/authorized_keys`. Then save the file `/home/tm1510/.ssh/authorized_keys`. Close all editors and all terminals. If you now open a new terminal and try to log into the ANAROOT server, you should no longer need a password.

For setting up an ssh link to an LWP computer, you have to add the public key to the file `/var/lwp/ssh/p261006/authorized_keys` instead (replace the personal number with your own number). This is due to the LWP structure. Then with the command `hostname -f` you will get the full hostname of the LWP computer. Your p-number will be our user name. Hence in my case the output of `hostname -f` will be `kvi-cart-81-217.kvi-cart.rug.nl` Hence the full ssh-command becomes:

```
ssh -Y p261006@kvi-cart-81-217.kvi-cart.rug.nl
```

Notice that you do not need a password to log into the PC but due to the LWP structure you WILL need your central RUG-password to log into your account.

Notice that if you run a shell script `script.sh` through the ssh-link with the command `./script.sh` then breaking the ssh-link and closing your own laptop will terminate the script. This is not what you want because you want to be able to shut down your own laptop and keep the script running. In order to achieve this, run the script with the command `./script.sh &bg disown`. If no output is generated to the terminal screen, the application will keep running.

In case that there is terminal screen output, you should use the `screen` command. The commands to use (in sequence) are:

```
ssh -Y p261006@kvi-cart-81-217.kvi-cart.rug.nl
screen
./script.sh
CTRL+a
d
```

The command `screen` opens a virtual terminal screen within your terminal. Now you can run your script within this virtual screen. If you hit the keys `CTRL+d` the virtual screen is terminated and the computation is lost. If you however hit the keys `CTRL+a` and then press the `d`-key **afterwards**, the virtual screen is detached. This closes the screen, but keeps the computation running. If you want you can now shut down your system and the remote session will keep running.

In order to re-connect to your computation, first log in with ssh and then give the command `screen -r`. If you only have one virtual screen, this will re-attach your screen and link you back to the application. It will still be running! In case of multiple virtual screens, the command `screen -r` will produce a list of all virtual screens available. You then have to type the name of the screen you want in order to re-attach that screen.

Now that your computer can log into the ANAROOT server without a password, we can install the Online Monitor for NeuLAND. create a directory `<path to>/OnlineMonitor/` at the location of your choice (at your own PC!) and navigate to it in your terminal. Then give the following command:

`git clone https://bitbucket.org/hanstt/samurai_neuland_online/`. The program is downloaded into a subdirectory of the directory `<path to>/OnlineMonitor/`. It will be called `<path to>/OnlineMonitor/tm1510_neuland_online/`. Note that the program is installed at your own PC and it will also run on your own PC;. But the program needs to be able to log in without a password in order to collect the data.

Now to run the program, open 3 different terminal windows. In each window, navigate to the directory `<path to>/OnlineMonitor/tm1510_neuland_online/`. In the first window, give the command `./fetcher.sh` in order to connect to the ANAROOT server and collect the data. Then, in the second window, start root with the command `root -l` (So you need to have root installed at your own PC). Then load the plotter-macro with the command `.L plotter.C+`. Run the macro with the command `plotter()`. The histograms will now open. All of them will be auto-refreshed continuously as long as there is data. Hence if they do not change, it just means that there is no new data to see. Nothing is wrong.

If you want to make a screenshot of the histograms, go to your third terminal window and give the command `./do_saveall.sh`. The histograms that exist at the moment you fire this script, are saved as `.png`-files. A directory `<path to>/OnlineMonitor/tm1510_neuland_online/online_saved/` is generated if it did not exist yet. Then a subdirectory is generated in it which is named after the time index at which you ran the script. Inside this subdirectory you will find your `.png`-files.

Notice that the program might be updated during the experiment, or that there are some bug fixes, so the steps from the `git`-command on might have to be redone when necessary (also keep the old folder just to be save). Setting the ssh keys however only needs to be done once.

In the newest version of the program all you have to do is to give the `git`-command. Then go to the directory `<path to>/OnlineMonitor/tm1510_neuland_online/` and run the command `make`. Then fire up the online monitor with the command `./plotter`. All histograms will fire and be automatically updated. When the update fails for some reason (no refreshment of data comes withn 30 seconds), the background of the histograms turns from white to brown. Hence white means everything is OK and brown means you should check something. The beam, the DAQ, the monitor itself, etc. Both ROOT and the `./fetcher.sh` script are run automatically in the background as well, so you do not need to do anything.

For the $4n$ -experiment, the procedure is similar. However, one should log now into the server `s034@ribfana02.riken.jp` with password `samurai_4n` instead.

Only if you want to save .png-files from the histograms, open a second terminal and fire the script `./do_saveall.sh` for a single saving of the histograms of that time instance.

4.3.10 Using the SAMURAI online monitor

Log into the ANAROOT server with the following command: `ssh -Y tm1510@ribfana02.riken.jp`. Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. The password is also here `obakiller`. Then log into anaroot, this time with the command `anarootlogin shift`. Then start root with `root -l`.

Load the SAMURAI online monitor with `.L macros/OnlineMonitor.cc+g`. Then create an OnlineMonitor-type class with `OnlineMonitor* mon = new OnlineMonitor()`. An empty canvas will appear where your spectra will appear later. Return to the terminal and activate the monitor with `mon->Run()`. A lot of text will appear on your terminal and after some time, your spectra will appear on the canvas. So wait for 1 or 2 minutes. Once the spectra appear, they will be automatically refreshed.

When you want to stop analyzing online data, press `CTRL+c`. The monitor will stop, but ROOT is still running (it needs a little time to stop). Then execute the writer with `.x macros/PrintRootFile.cc`. The program will ask for a filename with a path. Hence enter something like `ps/93Zr110MeV/MyFile.ps` and press `ENTER`. A new canvas will appear. This canvas is going to contain spectra, but this will take some time again. In the terminal the following line will appear: `[q]: quit, others: continue >`. You can now press `ENTER` to add the current canvas to your file `MyFile.ps`. Again, each time you want to do this, it is going to take time, so do not press `ENTER` more than once. The command verb`[q]: quit, others: continue` will re-appear once the computer is done. After you added enough canvas to your file, enter`q+ENTER`. (`q` first). The macro will take some time to finish up. Then close ROOT with `.q`.

Now navigate to `/home/tm1510/exp/exp1510_tm/anaroot/users/shift/ps/93Zr110MeV/`. This path is because you entered `ps/93Zr110MeV/MyFile.ps` as input of `PrintRootFile.cc`. If you give here the command `lpr MyFile.ps`, the content of this file will be printed on the printer in the control room.

In order to get the file yourself, use some `exit`-commands to log out of ANAROOT and of the ANAROOT server. If you are back at your own PC, give the following command:

```
scp -C tm1510@ribfana02.riken.jp:/home/tm1510/exp/exp1510_tm/anaroot/users/shift  
/ps/93Zr110MeV/MyFile.ps /home/christiaan/Desktop/NeuLAND/MyFile.ps
```

This will copy the file `MyFile.ps` from the server to your own PC. The command requires the server password if you did not install an ssh-key. Notice that you have to adapt the second path (the target path) to your needs.

Notice that you have to press `ENTER` enough times when you use `PrintRootFile.cc` because the data of NeuLAND is only in the third and fourth canvas!

When you see jumps in the data (large values in `Tdiff` versus bar number), one could try to increase the energy threshold by 5 units. The first thing you need is to know which PMT is the problem. For that, look at the online monitor `T 1` versus bar and `T 2` versus bar. Then you know the bar number and the PMT number. Next, login to the neuLand server with `ssh -Y neuLand@ribfana02.riken.jp` with password `samuraineLand`. Next, go to the path `/home/neuland/neuland/` and start ROOT with `root -l`. Next, load the TRIPLEX controller with `.L triplex.C+` to control the TRIPLEX card. Next, load the mapping table with `read_mapping()` to get access to individual NeuLAND bars. Then give the command `get_thr_single(PLANE,BAR,PMT)` to get the current value of the threshold. `PLANE` is the number of the plane where the bar is located in (starting from 1). `BAR` is the bar number you are interested in, ranging from 1 till 50. `PMT` is either 1 or 2: the number of the PMT you are interested in. (corresponding to the `T 1` and `T 2` online monitor histograms). You get the threshold value out. To change it, give the command `set_thr_single(PLANE,BAR,PMT)` and the prgram will ask you to enter a new value. Enter it and now you are done.

4.3.11 Restarting a crashed DAQ

Here we explain how to restart the DAQ if it crashes. The first thing to notice is that you cannot do anything if the full DAQ of the entire experiment crashes. Then the people from RIKEN should restart the DAQ and all you can do is ask them to. You can only do something yourself if it is not the master DAQ that crashes, but only the NeuLAND part of the DAQ.

Log into the windows server with the command

`ssh -Y neuland@ribfana02.riken.jp.` Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. The password is `samuraineuland`. Give the command `screen -x daq` to monitor the triggers. By selecting **CTRL+a** you go to control mode. Then you can press a number between 0 and 5 to switch between the different screens. Now use this method to go to screen number 0.

press ENTER one or two times. The command prompt should appear and start with `mbs>`. If this does not work an expert should be contacted **IMMEDIATELY!** The command line `mbs>` might however take some time to appear, so wait a short time. Now in order to see if the DAQ works, you can give the command `sho acq`. This command does involve a little danger, since invoking it often might actually cause a crash of the DAQ! So use it only when you absolutely have to!

If you give this command, a lot of output is generated, with the event rate in number of events and in kHz at the bottom. If these are nonzero, everything is OK. If they are zero, no data is taken, but since the output is a single snapshot that is NOT refreshed, it can also be bad luck. For that purpose, the command `sho acq 5` can be given. This command takes 5 seconds (because you entered 5) and creates an average snapshot over 5 seconds. The non-zero check is then much more reliable. But still, the rate is checked by file size and if the file output is below a threshold, you see zero while nothing might be wrong.

Now if you are convinced that something is wrong, stop the DAQ with the command `@s`. Then you can restart the DAQ with `@t` and everything should work again. If this is not the case, an expert should be contacted **IMMEDIATELY!** Also the commands `sto acq` is there to stop the DAQ and `sta acq` to start the DAQ, but `@s` and `@t` are better.

4.3.12 How to get access to the real data of NeuLAND in coincidence with the beam

The SAMURAI joined DAQ system writes data to `.ridf`-files. To access these files, log into the `s021` server with the following command: `ssh -Y s021@ribfana02.riken.jp.` Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. Password is `samurai_oxygen28`. Then go to the path `/home/s021/ridf/sdaq02/`. All the `.ridf`-files are located here. With the command `ls -1Ah` you get the file sizes and time indices of all files. **NEVER** touch the newest file of a particular kind! This is because the DAQ is still writing to this file. The older files are closed, so you can unpack them if you want. But still you should not make any changes to them! The RAW data should be conserved!

To access the unpacker, log into the `s021` server with the following command: `ssh -Y s021@ribfana02.riken.jp.` Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. Password is `samurai_oxygen28`. Then give the command `anarootlogin neuland`. Then go to the directory `/home/s021/exp/exp1511_s021/anaroot/users/neuland/macros/`. In this file there is a macro called `maketree_neuland.cc` This macro will unpack the data to a rootfile, normally located in `/home/s021/exp/exp1511_s021/anaroot/users/neuland/macros/rootfiles`. Now the rootfile that is given as output contains data trees that you can analyze in any way you want. The macro `maketree_neuland.cc` also contains some documentation on what the names of the Leafs mean.

Notice that the unpacker needs several headers (probably ANAROOT headers) to perform the unpacking process. Hence you probably should do the unpacking on the server, not on your own laptop. Then you can `scp` the roottree, because that one can be accessed with just your own ROOT-system.

The leafs in the tree carry labels that begin with `n1_` for NeuLAND, `n1v_` for the NeuLAND Veto detector (as in ImPACT geometry: 8 vertical bars of 32 cm width and 192 cm long, each with an overlap of 1 cm and a thickness of 1 cm in central position in front of NeuLAND), `nb` for NEBULA and `nbv` for the NEBULA Veto detector (same geometry as for NeuLAND). We first have the NeuLAND Veto, then NeuLAND, then the NEBULA Veto and then NEBULA. There are also a few other leafs. The labels of the NeuLAND leafs are explained in the following picture:

The leafs for NEBULA and both Veto detectors have similar meanings. The distance between the Veto detector and NeuLAND (edge to edge) is 33 cm. NeuLAND is located 11 m behind the target. NeuLAND consists of 400 scintillator bars of 5 cm by 5 cm by 250 cm (crystal, foil and tape together). The foil is 0.2 mm thick and the tape is 0.3 mm thick. The dimensions of the VETO bars are also for crystal, foil and tape together.

```

// Documentation of NeuLAND ROOT-files. Information is given to me by Igor Casparic,
// staff member of the NeuLAND workgroup and project leader of NeuLAND@SAMURAI.
// Int_t NL_BAR_NUM; >>> // Number of different bars that registered a hit in NeuLAND. Hence the index has nothing to do with which bar was actually hit.
// This is obtained through the nl_bar leaf. NOT the nl_rbar leaf, that is something different! This NL_BAR_NUM is therefore just
Double_t nl_tms; >>> // Master TDC signal t1+t2-t3 & converted to [ns]. The electronics generate a time delay in processing TDC signals. This is uniform distributed
// between 0 and 10 ns. In order to correct for it, we interface the trigger also directly to TACQUILA so we can determine the electronics
// response exactly and correct for it later. This leaf tells you the data on this channel. Obviously there is only one TACQUILA that does
// this, so we have only one number per event.
Int_t nl_tms1; >>> // RAW TDC TACQUILA output in [channel nr.] of Master Time signal. Time t1 in TACQUILA clock pulse.
Int_t nl_tms2; >>> // RAW TDC TACQUILA output in [channel nr.] of Master Time signal. Time t2 in TACQUILA clock pulse.
Int_t nl_tms3; >>> // RAW TDC TACQUILA output in [channel nr.] of Master Time signal. Time t3 in TACQUILA clock pulse.
Int_t nl_rbar; >>> // No idea what this is...
Int_t nl_bar [NL_BAR_NUM]; >>> // NeuLAND bar number according to the mapping table. Hence if hit index 4 corresponds to a hit in bar number 25 of single layer 3, then
// nl_bar->GetValue(4) = 2*5+25 = 25 so bar 125 in entire detector. Hence this is where the hit took place.
Double_t nl_qraw0 [NL_BAR_NUM]; >>> // RAW QDC TACQUILA output [MeV, but uncalibrated!] of PMT 1 of the bar in question.
Double_t nl_qraw1 [NL_BAR_NUM]; >>> // RAW QDC TACQUILA output [MeV, but uncalibrated!] of PMT 2 of the bar in question.
Double_t nl_qcal0 [NL_BAR_NUM]; >>> // nl_qraw0 but now pedestals have been subtracted, PMT sensitivities of one bar have been aligned (Ediff calibration) and sensitivities
// have been tuned so that Edep_hiipoint = sqrt(nl_qcal0*nl_qcal1). Units are now [MeV].
Double_t nl_qcal1 [NL_BAR_NUM]; >>> // nl_qraw1 but now pedestals have been subtracted, PMT sensitivities of one bar have been aligned (Ediff calibration) and sensitivities
// have been tuned so that Edep_hiipoint = sqrt(nl_qcal0*nl_qcal1). Units are now [MeV].
Int_t nl_t1 [NL_BAR_NUM]; >>> // RAW TDC TACQUILA output in [channel nr.] of PMT 1 of the bar in question. Time t1 in TACQUILA clock pulse.
Int_t nl_t2 [NL_BAR_NUM]; >>> // RAW TDC TACQUILA output in [channel nr.] of PMT 2 of the bar in question. Time t2 in TACQUILA clock pulse.
Int_t nl_t3 [NL_BAR_NUM]; >>> // RAW TDC TACQUILA output in [channel nr.] of PMT 1 of the bar in question. Time t3 in TACQUILA clock pulse.
Int_t nl_t4 [NL_BAR_NUM]; >>> // RAW TDC TACQUILA output in [channel nr.] of PMT 2 of the bar in question. Time t4 in TACQUILA clock pulse.
Int_t nl_t5 [NL_BAR_NUM]; >>> // RAW TDC TACQUILA output in [channel nr.] of PMT 1 of the bar in question. Time t5 in TACQUILA clock pulse.
Int_t nl_t6 [NL_BAR_NUM]; >>> // RAW TDC TACQUILA output in [channel nr.] of PMT 2 of the bar in question. Time t6 in TACQUILA clock pulse.
Double_t nl_traw0 [NL_BAR_NUM]; >>> // t1+t3-t5 (PMT1) and converted to [ns]. Nothing more.
Double_t nl_traw1 [NL_BAR_NUM]; >>> // t2+t4-t6 (PMT2) and converted to [ns]. Nothing more.
Double_t nl_tcal0 [NL_BAR_NUM]; >>> // nl_traw0 but now both PMT have been synchronized (Td iff calibration), all bars have been synchronized (as discussed in the Log under
// the procedure of synchronizing all different bars w.r.t. each other. Also master signal nl_tms has been subtracted (corrected for).
Double_t nl_tcal1 [NL_BAR_NUM]; >>> // Notice that TOF_hiipoint = (nl_tcal0-nl_tcal1)/2 and that x_coordinate_hiipoint = (nl_tcal0-nl_tcal1)/2 ==> corrected for velocity of signal. Unit is [ns]
// nl_traw1 but now both PMT have been synchronized (Td iff calibration), all bars have been synchronized (as discussed in the Log under
// the procedure of synchronizing all different bars w.r.t. each other. Also master signal nl_tms has been subtracted (corrected for).
Bool_t nl_fired0 [NL_BAR_NUM]; >>> // Notice that TOF_hiipoint = (nl_tcal0-nl_tcal1)/2 and that x_coordinate_hiipoint = (nl_tcal0-nl_tcal1)/2 ==> corrected for velocity of signal. Unit is [ns]
Bool_t nl_fired1 [NL_BAR_NUM]; >>> // States if PMT 1 has fired or not for that specific hiipoint.
Bool_t nl_fired2 [NL_BAR_NUM]; >>> // States if PMT 1 has fired or not for that specific hiipoint.
Double_t nl_x [NL_BAR_NUM]; >>> // x-position of hit in a bar. Can only be resolved up to 5 cm unless it is in the direction of the bar, in that case, see nl_tcal0 for calculation. Unit is [mm]
Double_t nl_y [NL_BAR_NUM]; >>> // y-position of hit in a bar. Can only be resolved up to 5 cm unless it is in the direction of the bar, in that case, see nl_tcal0 for calculation. Unit is [mm]
Double_t nl_z [NL_BAR_NUM]; >>> // z-position of hit in a bar. Can only be resolved up to 5 cm unless it is in the direction of the bar, in that case, see nl_tcal0 for calculation. Unit is [mm]
Int_t nl_mul; >>> // multiplicity of the entire event [nr. of bars]
Int_t nl_muh; >>> // multiplicity of all horizontal bars [nr. of bars]
Int_t nl_muv; >>> // multiplicity of all vertical bars [nr. of bars]
Int_t nl_mulv1; >>> // multiplicity of the first layer of 50 bars (horizontal layer) [nr. of bars]
Int_t nl_mulv2; >>> // multiplicity of the second layer of 50 bars (vertical layer) [nr. of bars]
Int_t nl_mulv3; >>> // multiplicity of the third layer of 50 bars (horizontal layer) [nr. of bars]
Int_t nl_mulv4; >>> // multiplicity of the fourth layer of 50 bars (vertical layer) [nr. of bars]
Int_t nl_mulv5; >>> // multiplicity of the fifth layer of 50 bars (horizontal layer) [nr. of bars]
Int_t nl_mulh2; >>> // multiplicity of the sixth layer of 50 bars (vertical layer) [nr. of bars]
Int_t nl_mulh3; >>> // multiplicity of the seventh layer of 50 bars (horizontal layer) [nr. of bars]
Int_t nl_mulh6; >>> // multiplicity of the eighth layer of 50 bars (vertical layer) [nr. of bars]

```

Figure 4.16: NeuLAND ROOT Tree documentation. Note that due to a problem with NEBULA, the Neutron-leafs only contain NeuLAND information at this moment. This will be corrected in the future. NEBULA multiplicity is therefore empty at this moment. the IsNEBULA distinguishes between NeuLAND and NEBULA. For each hit one already knows it is a hit, hence it has to be in either one of them. This boolean then distinguishes. Also note that the Neutron ID ranges from 1 to 400 (NeuLAND). NEBULA ranges from 1 to 120. For each hit you know which to take due to IsNEBULA. But a histogram from only *NeutronID* might therefore look very strange...

The leaf `coin_ch` tells you which detectors fired (registered something) in that particular event. It is a big integer. Then you have to identify which powers of 2 are in that number. Those powers represent the numbers of the detectors that fired. This leaf is generated based on trigger signals.

In order to fire the unpacker, log into the `s021` server with the following command: `ssh -Y s021@ribfana02.riken.jp`. Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. Password is `samurai_oxygen28`. Then give the command `anarootlogin neuland`. Now, STAY in this directory and fire ROOT here with `root -1`. Otherwise the required headers of the unpacker cannot be found. Load and compile the unpacker with `.L ./macros/maketree_neuland.cc++` You will get some warnings, but everything should work fine. Then Unpack by giving the command

`Run("<<Name of the .ridf file with extension and its full path>>")` You have to give the name of the `.ridf`-file without extension. The unpacker will know where the directory of the `.ridf`-files is. Then the unpacker will work and create a root-file in the `rootfiles`-directory. And that's it.

Now that's for the NeuLAND and NEBULA data. To get access to the other data, log into the `s021` server with the following command: `ssh -Y s021@ribfana02.riken.jp`. Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. Password is `samurai_oxygen28`. Then give the command `anarootlogin neuland`. the go to the directory

`/home/s021/exp/exp1511_s021/anaroot/users/Analysis/rootfiles/samurai` The Oxygen experiment datafiles have a name like `run0129.root`. Make sure you have corresponding names between NeuLAND and The rest of the experiment when you analyze. The unpacker for this data is called `TreeConverter.C` and is located at `/home/s021/exp/exp1511_s021/anaroot/users/convert/`. It takes the same `.ridf`-files as for the NeuLAND data, but it just generates other leafs holding other data.

Now there is also a macro called `analysis_convert.cxx` located at

`/home/s021/exp/exp1511_s021/anaroot/users/convert/macros/`. This macro converts a tree called `run0129.root` located at

`/home/s021/exp/exp1511_s021/anaroot/users/Analysis/rootfiles/samurai` to a new tree called `run0155_Analysed_Z=9_A=27_shift-0.012000.root` located at

`/home/s021/exp/exp1511_s021/anaroot/users/convert/macros/analysed_rootfile/`. This macro adds

leafs to the already existing tree. These new leafs are holding information like particle A and Z and momentum. The macro has two input arguments: A and Z . All data is analyzed and processed, but energy and momentum reconstruction is performed with masses etc based on the A and Z of your input. Hence only for the fragment you are interested in, the reconstruction works OK. But all data is still there and you have to cut out your own fragment. This conversion macro is still being updated and then all `run0XXX_Analysssed.root`-files have to be updated. Therefore, keep a look on

`/home/s021/exp/exp1511_s021/anaroot/users/convert/RunConversion.ods` This is a table and only the green files are up to date. (except for the A and Z cuts).

4.4 SAMURAI Analysis procedure

4.4.1 Performing analysis on SAMURAI data

In this section we discuss the documentation on the leafs of the `run0XXX_Analysssed.root`-files. It is given in the following picture:

```
// Information is given to me by the author of analysis_convert.cxx
// All leafs have the same size of [1].

// One event = one particle in the beam ==> one fragment (in the rare case of 2 fragments, one will be deflected so much/so little it will not hit the HODF. Physics limits that F29--> 024 + 025 KN).
// but you can have multiple protons/neutrons in one event.

Int_t CoincidenceTrigger;
Double_t BEAM_A;           // beam A before hitting MINOS target. The analysis software makes a particle ID and then stores the A and Z of this ID inside these leafs.
Double_t BEAM_Z;           // if the particle ID had not been programmed in, you will get -9999 for these leafs. In this case, the reconstruction where you need
                           // to reconstruct the particle will be done for the whole event, but the other leafs will be fine.

Double_t BEAM_Z;           // beam Z before hitting MINOS target. Same story as for BEAM_A.

Int_t fra_Az;             // fragment Z before hitting MINOS target. Same story as for BEAM_A.

Int_t fra_Zz;             // fragment Z before hitting MINOS target. Same story as for BEAM_A.

Double_t F3_Charge;        // F3, F5, F7 and F13 are in-beam detectors: plane scintillators with 2 PMTs in total. They are located at different positions of
                           // the BigRIPS fragment separator. Hence they represent different focal planes. They are not part of the SAMURAI setup.

Double_t F3_Time;          // Time = TOF = (TDC of PMT1 + TDC of PMT2)/2 - Tstart + corrections

Double_t F3_TimeDiff;       // Charge = sqrt(RawQDC 1*RawQDC 2) where RawQDC 1 or 2 are the rawQDC data of the 2 PMTs of the detector. ==> Charge = Raw Energy deposition.

Double_t F5_Charge;        // Time = TOF = (TDC of PMT1 + TDC of PMT2)/2 - Tstart + corrections

Double_t F5_Time;          // Time = TOF = (TDC of PMT1 - TDC of PMT2) + possibly corrections. Time = [ns], TimeDiff = [ns]. Charge = [channel number]

Double_t F5_TimeDiff;       // TimeDiff = TDC of PMT1 - TDC of PMT2 + possibly corrections. Time = [ns], TimeDiff = [ns]. Charge = [channel number]

F7_Charge;                // Charge = sqrt(RawQDC 1*RawQDC 2) where RawQDC 1 or 2 are the rawQDC data of the 2 PMTs of the detector. ==> Charge = Raw Energy deposition.

Double_t F7_Time;          // Time = TOF = (TDC of PMT1 + TDC of PMT2)/2 - Tstart + corrections

Double_t F7_TimeDiff;       // TimeDiff = TDC of PMT1 - TDC of PMT2 + possibly corrections. Time = [ns], TimeDiff = [ns]. Charge = [channel number]

Double_t F13_Charge;        // Charge = sqrt(RawQDC 1*RawQDC 2) where RawQDC 1 or 2 are the rawQDC data of the 2 PMTs of the detector. ==> Charge = Raw Energy deposition.

Double_t F13_Time;          // Time = TOF = (TDC of PMT1 + TDC of PMT2)/2 - Tstart + corrections

Double_t F13_TimeDiff;       // TimeDiff = TDC of PMT1 - TDC of PMT2 + possibly corrections. Time = [ns], TimeDiff = [ns]. Charge = [channel number]

BetaF7F13;                // relativistic beta of the beam particles before entering MINOS. It is based on a time difference between F7 and F13.

Double_t BDC1_X;            // reconstructed X position at the incident window of the BCD1 detector (far from SAMURAI magnet) All distances are in [mm].
Double_t BDC1_Y;            // reconstructed Y position at the incident window of the BCD1 detector (far from SAMURAI magnet) All distances are in [mm].
Double_t BDC1_ThetaX;        // Angle in XZ-plane between beam design orbit and actual orbit, measured at BCD1 X and BCD1 Y. Hence it is the direction of the particle. [rad].
Double_t BDC1_ThetaY;        // Angle in YZ-plane between beam design orbit and actual orbit, measured at BCD1 X and BCD1 Y. Hence it is the direction of the particle. [rad].
Double_t BDC2_X;            // reconstructed X position at the incident window of the BCD2 detector (close to SAMURAI magnet) All distances are in [mm].
Double_t BDC2_Y;            // reconstructed Y position at the incident window of the BCD2 detector (close to SAMURAI magnet) All distances are in [mm].
Double_t BDC2_ThetaX;        // Angle in XZ-plane between beam design orbit and actual orbit, measured at BCD2 X and BCD2 Y. Hence it is the direction of the particle. [rad].
Double_t BDC2_ThetaY;        // Angle in YZ-plane between beam design orbit and actual orbit, measured at BCD2 X and BCD2 Y. Hence it is the direction of the particle. [rad].
Target_X;                  // reconstructed X position at the incident window of the MINOS target. All distances are in [mm].
Target_Y;                  // reconstructed Y position at the incident window of the MINOS target. All distances are in [mm].
Double_t FDC1_X;            // reconstructed X position at the incident window of the FDC1 detector. All distances are in [mm].
Double_t FDC1_Y;            // reconstructed Y position at the incident window of the FDC1 detector. All distances are in [mm].
Double_t FDC1_ThetaX;        // Angle in XZ-plane between beam design orbit and actual orbit, measured at FDC1 X and FDC1 Y. Hence it is the direction of the particle. [rad].
Double_t FDC1_ThetaY;        // Angle in YZ-plane between beam design orbit and actual orbit, measured at FDC1 X and FDC1 Y. Hence it is the direction of the particle. [rad].
Double_t FDC2_X;            // reconstructed X position at the incident window of the FDC2 detector. All distances are in [mm].
Double_t FDC2_Y;            // reconstructed Y position at the incident window of the FDC2 detector. All distances are in [mm].
Double_t FDC2_ThetaX;        // Angle in XZ-plane between beam design orbit and actual orbit, measured at FDC2 X and FDC2 Y. Hence it is the direction of the particle. [rad].
Double_t FDC2_ThetaY;        // Angle in YZ-plane between beam design orbit and actual orbit, measured at FDC2 X and FDC2 Y. Hence it is the direction of the particle. [rad].
SAMURAI_Brho;              // Br*rho product [TeV] of SAMURAI magnet.

Hodo1_Charge;               // The HODF detector is a single layer of 24 vertical scintillator bars. We only measure TOF and Energy deposition of the particles here.
Hodo1_Time;                 // We are NOT interested in the position of the hit, since we can get that from the FDC2 detector. The Time leafs are computed as
                           // Time = TOF = (TDC of PMT1 + TDC of PMT2)/2 - Tstart + corrections and Charge = sqrt(RawQDC 1*RawQDC 2) of the 2 PMT of the bar that is hit.
Hodo2_Charge;               // Hence Charge is the raw energy deposition in that specific bar [channel number]. It is uncalibrated.
Hodo2_Time;                 // Now we have leafs like Hodo3_Time meaning that we are talking about the TOF for bar nr. 3. We have leafs
                           // from 1 till 24 here, that all mean the same. Time leafs are in [ns] and Charge is in Atomic number. But it has a resolution. Thats why it is a Double_t.

...
Int_t Hodo_ID;              // Number of the scintillator bar in HODF that fired [Int: Bar_ID]. Since we always have a single fragment, we never have more than 1 hit in the entire HODF. [bar nr.]
Hodo_Charge;                // Charge-leaf from the appropriate bar [channel number].
Hodo_Time;                  // TOF-Time-leaf from the appropriate bar. [ns]

Double_t Hodo_Z;             // Z-number of the fragment that hits the HODF. It is the same as Hodo_Charge, but corrected for several artifacts. Hence use this! [Atomic Number]

Double_t Hodo_Verz;           // AZ-number of the fragment that hits the HODF. It is the same as Hodo_Charge, but corrected for several artifacts. Hence use this! [Atomic Number]

vector<double> Neutron_X;   // hit X-position of a possible neutron [mm]. We are not sure whether it is a neutron yet. [mm]. You can have several hits per event, hence this is a vector<double>.
vector<double> Neutron_Y;   // hit Y-position of a possible neutron [mm]. We are not sure whether it is a neutron yet. [mm]. You can have several hits per event, hence this is a vector<double>.
vector<double> Neutron_Z;   // hit Z-position of a possible neutron [mm]. We are not sure whether it is a neutron yet. [mm]. You can have several hits per event, hence this is a vector<double>.
vector<double> Neutron_Time; // hit TOF of a possible neutron [ns]. We are not sure whether it is a neutron yet. [mm]. You can have several hits per event, hence this is a vector<double>.
vector<int> Neutron_ID;        // Number of the bar that fired. [bar nr.]
vector<double> Neutron_ChargeU; // Raw QCD output of PMT2 at NeuLAND/NEBULA scintillator bar [uncalibrated MeV].
vector<double> Neutron_ChargeD; // Raw QCD output of PMT1 at NeuLAND/NEBULA scintillator bar [uncalibrated MeV].
vector<bool> Neutron_ISNEBULA; // Hit in NEBULA yes/no. (NEBULA VETO = KTRUE as well). [Boolean]
Int_t Neutron_Multiplicity; // Total number of hits for all Neutron detectors: this matches the size of the vectors in the leafs!
Double_t NEBULA_VetoSize;     // Total number of hits for all VETO detectors: size of vector, since there are 2 VETOS, NeuLAND and NEBULA together!
Bool_t NEBULA_Veto Fired;    // if even one of the VETO bars of NEBULA fired, this is KTRUE, else KFALSE.
Int_t NEULAND_Multiplicity; // Total number of hits for NeuLAND: NOT size of vectors, since that is 2 VETOS, NeuLAND and NEBULA together!
Bool_t NEULAND_Veto Fired;  // if even one of the VETO bars of NeuLAND fired, this is KTRUE, else KFALSE.

Int_t DALI_Multiplicity;    // DALI is a sphere of Gamma NaI crystal detectors around MINOS target. Hence we can have several gammas: This leaf = number of crystals hit: hence size of the vectors.
DALI_ID;                   // Which crystal is hit: ID of crystal. Total number of crystals = 142. last 2 are interfaced on an electric pulse to synchronize and 140 are for detection.
vector<double> DALI_Energy; // Energy deposition in crystal [keV]
vector<double> DALI_Time;   // Time of crystal hit [ns]
vector<double> DALI_CosTheta; // Angle between beamline and detector that fired considered from MINOS center (so not the collision point!) [dim. less: it is a CoSine!]
vector<double> DALI_EnergyDopplerCorrected; // DALI Energy corrected for doppler shift. We simply take DALI_Energy and BETA_BEAM and perform the doppler correction. [MeV].
Double_t MINOS_X;           // Reconstructed X-position [mm] of the collision in the MINOS liquid Hydrogen target.
Double_t MINOS_Y;           // Reconstructed Y-position [mm] of the collision in the MINOS liquid Hydrogen target.
Double_t MINOS_Z;           // Reconstructed Z-position [mm] of the collision in the MINOS liquid Hydrogen target.
Double_t MINOS_Radius;       // sqrt(x*x+y*y) for MINOS_leaves [mm]
Double_t MINOS_D_min;        // minimum distance between proton tracks [mm]. This is used to say whether the protons tracks touch or not: tracks belong together yes/no.
Int_t MINOS_NumberTracks;   // Number of distinct proton tracks in MINOS. ==> This is number of protons detected. [nr. of particles]
Double_t Angle_frag;         // Angle between position in MINOS and in FDC1 [rad].
Double_t BETA_f;             // reconstructed relativistic beta of the fragment. [dim. less] This is computed from TOF, so it is always correct!
Double_t GAMMA_f;            // reconstructed relativistic gamma of the fragment [dim. less] This is computed from TOF, so it is always correct!
Double_t MOM_f;              // Fragment reconstructed total momentum: p = sqrt(px*px+py*py) [MeV/c]. NOTE: if input of analysis_convert.cxx is wrong, this leaf is wrong, others are fine!
Double_t MOM_f_X;            // Fragment reconstructed momentum in X direction [MeV/c]. NOTE: if input of analysis_convert.cxx is wrong, this leaf is wrong, others are fine!
Double_t MOM_f_Y;            // Fragment reconstructed momentum in Y direction [MeV/c]. NOTE: if input of analysis_convert.cxx is wrong, this leaf is wrong, others are fine!
Double_t MOM_f_Z;            // Fragment reconstructed momentum in Z direction [MeV/c]. NOTE: if input of analysis_convert.cxx is wrong, this leaf is wrong, others are fine!
vector<double> BETA_n;       // relativistic beta of the beam particle [dim. less]
vector<double> GAMMA_n;       // 
vector<double> MOM_n;          // We assume here that the fragment is a decay product of a bigger fragment and one neutron. We now reconstruct the momentum of this bigger
                           // fragment. Leaf MOM_n and MOM_f have the same meaning, only the MOM_n is for this bigger fragment.
vector<double> MOM_n_X;        // Fragment. Leaf MOM_n and MOM_f have the same meaning, only the MOM_n is for this bigger fragment.
vector<double> MOM_n_Y;        // NOTE: if input of analysis_convert.cxx is wrong, this leaf is wrong, others are fine!
vector<double> MOM_n_Z;        // NOTE: if input of analysis_convert.cxx is wrong, this leaf is wrong, others are fine!
Double_t BETA_BEAM;          // reconstructed relativistic beta of the beam particle at the collision point in MINOS. It lost some energy in the target, so this beta is smaller the betaF7F13!
Double_t Energy_rel_In;       // NOTE: if input of analysis_convert.cxx is wrong, this leaf is wrong, others are fine!
                           // Same as MOM_n only now we reconstruct not the momentum but the energy [MeV]. NOTE: if input of analysis_convert.cxx is wrong, this leaf is wrong, others are fine!
```

Figure 4.17: Documentation of the SAMURAI data tree. Information was given by Aldrich Revel.

4.4.2 Performing particle ID on the SAMURAI NeuLAND setup

```
Fabia's plots: Hodo_AoverZ:Hodo_Z {((F5_Time+F5_TimeDiff*-2.25)/-58.5+14.812)<3.3
&& ((F5_Time+F5_TimeDiff*-2.25)/58.5+14.812)>3.151}
(F5_Time+F5_TimeDiff*-2.25)/58.5+14.812:(TMath::Sqrt(F7_Charge)-20.23)/0.75+10.0
```

Charge is measured by energy loss: Bethe Bloch. TOF gives you β . Then you also know γ . And β and Energy loss gives you the charge from Bethe Bloch formula. Then from position measurements you can get the ρ . Now we know that:

$$B \cdot \rho \propto \frac{A}{Z} \cdot \beta \cdot \gamma$$

Hence by plotting $B\rho$ versus $\gamma\beta$ we can obtain A/Z (after calibration). Notice however that a precise knowledge on the geometry of the detectors is required for reconstruction of ρ . The same holds for knowing the relation between TOF and β .

This procedure is already done in the `Hodo_Z` and `Hodo_AoverZ` leafs. However, one needs to make a particle identification on both the incoming beam and the outgoing fragment. For the incoming beam it is a slightly different story. for this you get β from the TOF. Then you can compute γ from β . From the QDC you get energy loss. From energy loss you obtain Z by Bethe Bloch formula. But we also would like to know the mass for proper particle ID. One could use the Time Difference (horizontal position in the in-beam scintillator) for this. This is because the magnets from BigRIPS will cause particles with different A to arrive at slightly different positions in the in-beam scintillator. Hence plotting the position in the scintillator versus Z is like plotting A versus Z .

However, since

$$B \cdot \rho \propto \frac{A}{Z} \cdot \beta \cdot \gamma$$

One gets out A/Z by:

$$\frac{A}{Z} \propto B \cdot \rho \cdot \frac{1}{\beta} \cdot \frac{1}{\gamma} \propto B\rho \cdot TOF \cdot \frac{1}{\gamma}$$

Now the position or time difference is something like:

$$F5_TimeDiff \propto position \propto \rho - \rho_0$$

Where ρ_0 represents the design orbit. Hence we obtain:

$$\frac{A}{Z} \propto B(\rho_0 + position) \cdot TOF \cdot \frac{1}{\gamma} = \text{const} + \text{const} \cdot \Delta t \cdot TOF \cdot \frac{1}{\gamma}$$

Hence we can absorb both constants into the linear calibration and obtain:

$$\frac{A}{Z} \propto \Delta t \cdot TOF \cdot \frac{1}{\gamma}$$

Now the question remains what to do with γ .

$$\frac{1}{\gamma^2} = 1 - \beta^2 = 1 - \frac{TOF^2}{pathlength^2}$$

Now this cannot be absorbed into the linear calibration. Hence we should think up something else. We know that the energy spread of the beam is only of the order of 0.1%. The energy is 250 MeV/u. Hence if we neglect the energy spread and assume $M_p = M_n$ then we get that $\gamma \approx 1.266$. Now due to the little spread we can assume this is constant for all particles. Hence we can absorb γ into the linear calibration constants. Hence we remain with:

$$\frac{A}{Z} \propto \Delta t \cdot TOF$$

However, TOF is NOT equal to `F5_Time`. This is because TOF is the difference between the timing indices of two in-beam detectors. But this would give us:

$$\frac{A}{Z} \propto F5_TimeDiff \cdot (F5_Time - F3_Time) \approx \text{const} \cdot F5_TimeDiff$$

Due to the little energy spread in the beam. Hence, we get out that A/Z is represented by `F5_TimeDiff`.

Fabias plots on the incoming beam are obtained from plotting F5_Time (TOF) versus F5_TimeDiff (position in scintillator). This already separates the beam. But the blobs representing the different nuclei will be correlated between x and y . Hence we plot F5_Time+F5_TimeDiff*-2.25 against F5_Time (TOF) or F5_TimeDiff so that the blobs canNOT be separated with respect to one axis and can be perfectly separated with respect to the other axis. Hence we remove the correlation with this constant. The good axis will be F5_Time (TOF) versus F5_TimeDiff. Then now we plot this axis against TMath::Sqrt(F7_Charge), since $dE/dx \propto Z^2$ by Bethe-Bloch. (Due to little energy spread, β is just a calibration constant) This gives us then (sort of) A/Z versus Z . This is how the incoming beam is identified. The other constants are just calibration coefficients.

F5 has the best time resolution and F7 has the best energy resolution. Hence that is why we chose these in-beam detectors. Hence plotting F5_Time+F5_TimeDiff*-2.25 is just about removing unwanted correlations. Nothing more. See the next picture:

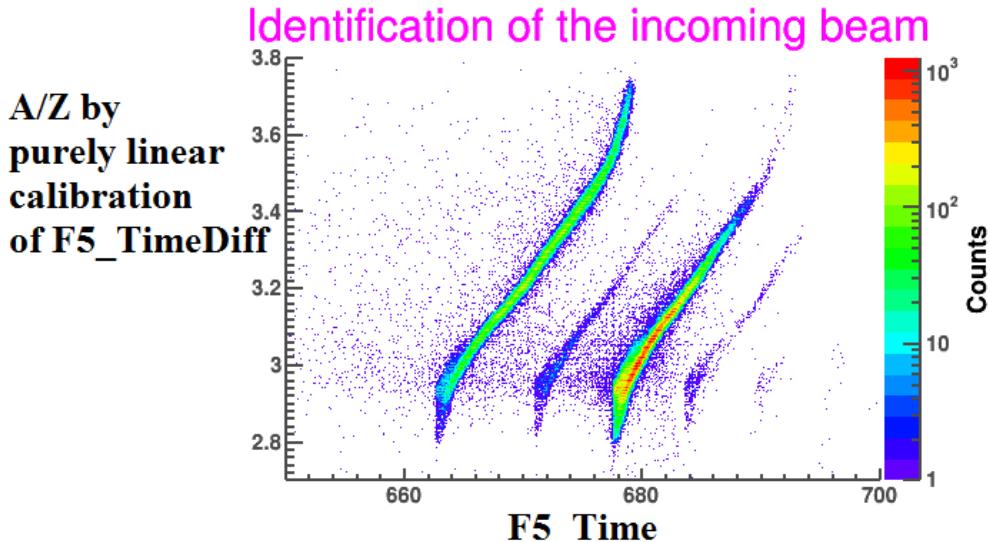


Figure 4.18: Correlation between TOF and position in in-beam detector F5.

Hence if we would separate the particles ONLY by A/Z , we would project this picture on the y -axis and it is then clear that you cannot separate that much. But if you project along the line of the correlations, you can separate much better. Hence we separate not only on A/Z and Z , but also on time of the hit. Hence removing the correlations allows us to select on more then just A and Z and therefore enhances our resolution considerably. That is why we plot F5_Time+F5_TimeDiff*-2.25 instead of F5_TimeDiff. NOTE: using the time of the hit means that we separate the beam in the longitudinal direction as well.

4.4.3 SAMURAI setup

When the beam enters the SAMURAI area [10], it will first come through 2 STB detectors: single scintillator plates. Then through an ionization chamber: ICB. The ICB performs a ΔE measurement. Then we go through 2 BDC detectors: drift chambers that perform position and angle (=direction) measurement before the target. Then we go to the target: MINOS. MINOS is a thick liquid hydrogen target combined with a TPC detector for proton tracking. It is surrounded by DALI: a gamma spectrometer. Then after the target we go to FDC1: a wire chamber for position and angle measurement before entering the dipole magnet. After the magnet we have an FDC2 detector. Both FDC detectors are wire chambers: very suitable for position measurement. They consist of ionized wires arranged in layers: horizontal, diagonal, vertical, etc. After the FDC2 we get to HODF: an array of scintillator bars (one vertical layer and one horizontal layer). They are used for TOF and ΔE measurements. One can then perform a particle ID by combining the information of FDC1, FDC2 and HODF together with the magnetic field information: you then track the particles through the magnet and fit their curved trajectory. For nuclear physics this is enough, since everything has a non-zero Z/A fraction except neutrons. These fly straight into NeuLAND and NEBULA where they are tracked, reconstructed, etc.

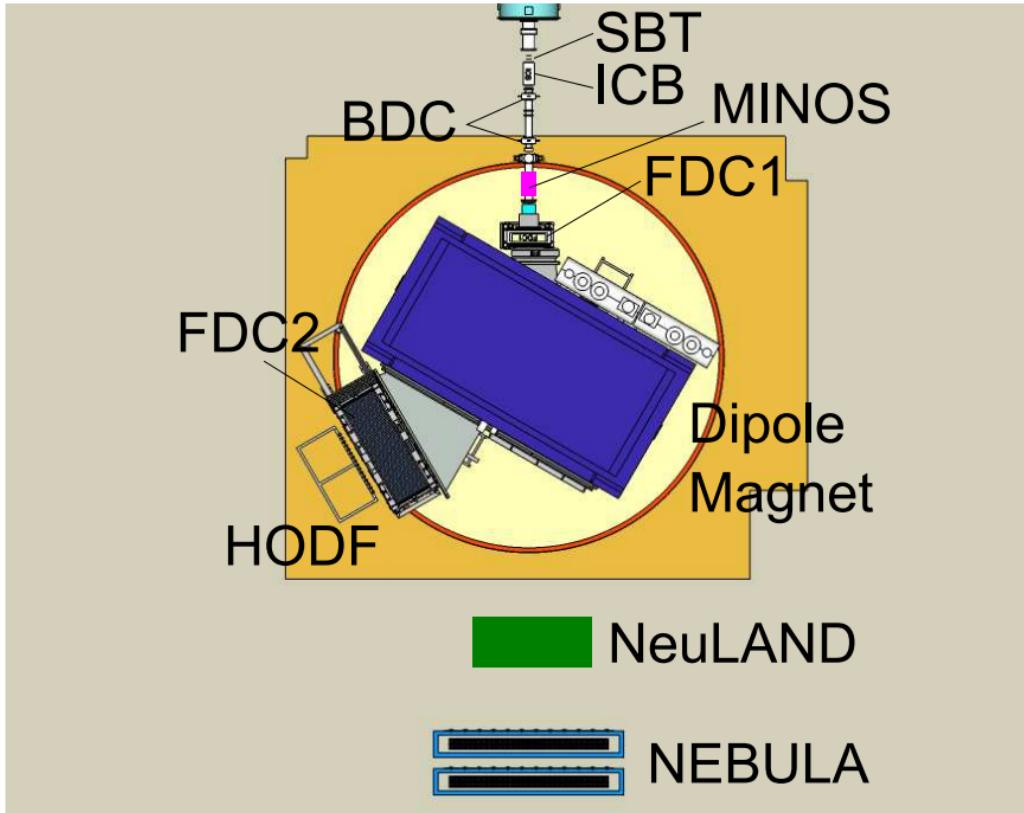


Figure 4.19: Overview of the SAMURAI setup

4.4.4 MINOS

The MINOS detector is best illustrated by the following picture:

MINOS consists of a thick liquid hydrogen target surrounded by a gas detector. see figure 4.20. The endpoints have a potential difference. Electrons produced by the ionizations of a proton that passes through the gas drift to the bottom point due to this potential difference. The bottom detector plate is well segmented, so you get position information. From the time it takes the electrons to reach the bottom plate you can get the z -coordinate.

Time calibration is performed by adding all drift times of all events. You then get a uniform distribution. You know that the start point of the distribution corresponds to $t = 0$ and the endpoint to a distance of 30 cm. Hence you then know the electron velocity (or acceleration) and you can extract z from the time of the hit. Hence each ionization gives you x , y and z information. The detector pads can also detect energy, which is used as a weight factor in calculating the precise positions. The energy distribution has a mean corresponding to the z (hence the amount of energy the electron picked up from the potential) and a sigma corresponding to the energy deposition of the proton. Hence one could measure the energy deposition of the proton and then reconstruct its total energy from the Bethe Bloch formula. But the resolution is about 50% of the total proton energy, so this is pointless.

Hence from tracking the protons you will know the amount of protons coming from the collisions and their direction, but not their energy. Hence one has no choice but to assume that the momentum transferred to the protons is negligible and to put it to zero.

MINOS is surrounded by a gamma spectrometer: DALI. It consists of 142 NaI crystals in a spherical shape around the target. the last 2 crystals are interfaced to the electronics directly for synchronization purposes. The other 140 take data.

To get access to the MINOS rootfiles, log into the s021 server with the following command:

`ssh -Y s021@ribfana02.riken.jp`. Again, the `riken.jp` is not required for the guest-network, but it is required for eduroam. Password is `samurai_oxygen28`. Then give the command `anarootlogin neuland`. Then go to the directory `/home/s021/exp/exp1511_s021/anaroot/users/minos/rootfiles/minos` This is

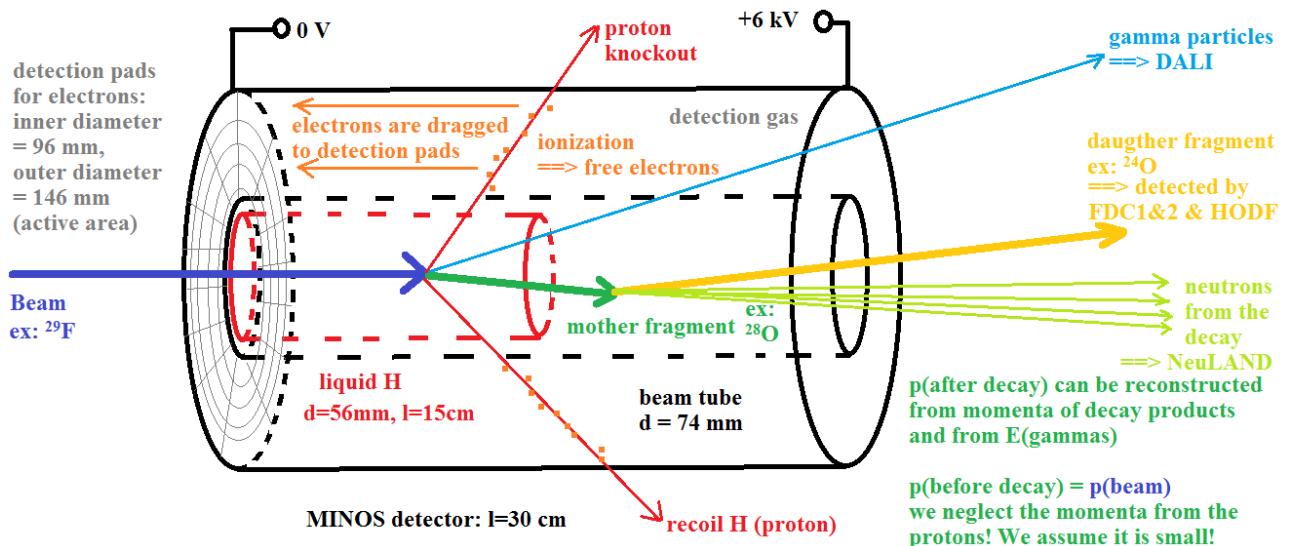


Figure 4.20: Principles of operation for the MINOS detector, as explained by Alexander Obertelli.

where the MINOS rootfiles are located. Now some rootfiles are called `run0XXX_minos.root` and some are called `run0XXX_minosOnline.root`. The Online-files used a very simple tracking algorithm to allow for online processing. The other files contain the proper protontracking.

For documentation on the MINOS leafs, see the following picture:

```

N_hits      >> fUniqueID[N_hits];>> // Number of pads in XY-bottom plane that fired within this event.
UInt_t       fitdata_fUniqueID[N_hits];>> // No real physics information
UInt_t       fitdata_fBits[N_hits];>> // no real physics information
Double_t     fitdata_x_mm[N_hits];>> // x-coordinate of the pad that fired [mm]. You can have multiple hits within one event, hence this leaf has a size.
Double_t     fitdata_y_mm[N_hits];>> // y-coordinate of the pad that fired [mm]. You can have multiple hits within one event, hence this leaf has a size.
Double_t     fitdata_t_ns[N_hits];>> // Electron drift TOF of the pad that fired [ns]. You can have multiple hits within one event, hence this leaf has a size.
Double_t     fitdata_z_mm[N_hits];>> // computed as fitdata_t_ns/VDrift ==> see other leafs. units is [mm].
Double_t     fitdata_ChargeMax[N_hits];>> // ADC value of the pad that fired [unit=?]
Double_t     fitdata_n_Cluster[N_hits];>> // index of the cluster the fired pad belongs to (after cluster finding has been performed).
Double_t     fitdata_n_Pads[N_hits];>> // either number of the fired pad, OR number of pads in the cluster that this pad belongs to ==> check it NOTE!
Double_t     fitdata_Chi2[N_hits];>> // Probably the Chi^2 of the single hit point with respect to the fitted track. Not sure!
Bool_t       MINOSOnline; >> // kTRUE means that simple online tracking is used. kFALSE means that proper tracking is used.
Int_t        EventNumber; >> // Event number that is analyzed, as registered by MINOS individual DAQ.
Int_t        Ridf_EventNumber; >> // Event number that is analyzed, as registered by entire SAMURAI DAQ. The event numbers should be equal: this is a check.
Int_t        NumberTracks; >> // Number of different protontracks as identified by MINOS.
Double_t     VDrift; >> // Calculated drift velocity: based on data of an entire run: no individual V_drift of electrons can be computed.
Double_t     DelayTrig; >> // Electronics value, not used in analysis.
ULong64_t  TimeStamp; >> // some time ID... ?
vector<double> chargeTot; >> // Sum of ADC values belonging to one track. The leaf has a size: length = the amount of different tracks.
vector<double> lenght; >> // length of each proton track.
vector<double> parFit1; >> // Fitting parameter for tracking the protons. The length of the vector<double> is the number of tracks in the event.
vector<double> parFit2; >> // Fitting parameter for tracking the protons. A track is defined by the equations x = parFit1 + z*parFit2
vector<double> parFit3; >> // Fitting parameter for tracking the protons.
vector<double> parFit4; >> // Fitting parameter for tracking the protons.
Double_t    Trigger_charge; >> // ?
Double_t    Trigger_charge2; >> // ?
vector<int> MyEventInfo_fBit; >> // No real physics information

```

Figure 4.21: Documentation on MINOS leafs. Information is from Alexander.

4.4.5 NEBULA

NEBULA consists of 120 scintillator bars, each having dimensions of $180\text{ cm} \times 12\text{ cm} \times 12\text{ cm}$. VETO scintillators (just like NeuLAND) have dimensions $190\text{ cm} \times 32\text{ cm} \times 1\text{ cm}$. We then have 2 layers of 30 scintillator bars each (all vertical!) and a VETO wall of 12 paddles in front of it. Then we have 2 such configurations placed behind each other (hence we have 2 VETO detectors of 12 paddles each)! See the following picture for NEBULA geometry:

NEBULA

(Neutron-detection system for Breakup of Unstable-Nuclei with Large Acceptance)

- Design
 - 240 Neutron counters
 - 48 VETO counters
 - arranged into 4 stacks
 - each stack
 - 60 neutron counters
 - 12 VETO counters
- Funded (Current version)
 - 120 Neutron counters (half)
 - 48 VETO counters

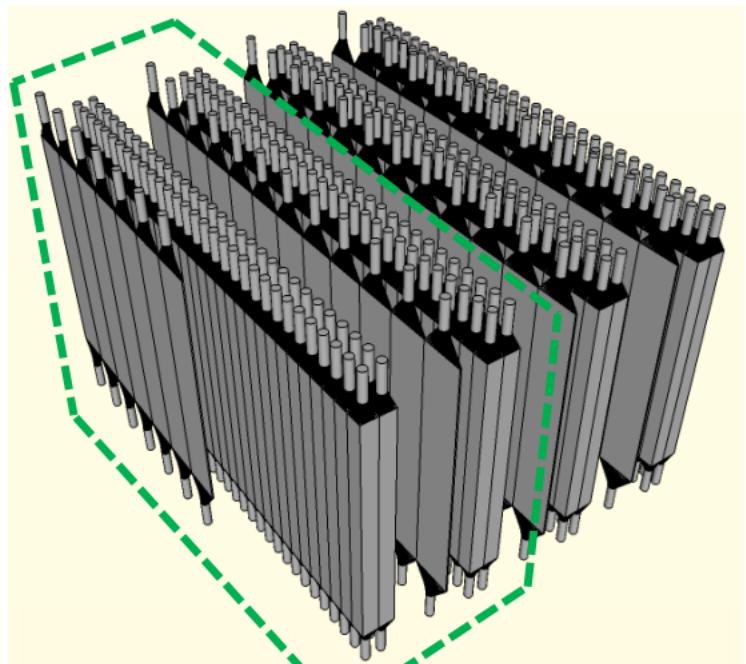


Figure 4.22: NEBULA detector geometry (only the funded part is in the setup now).

Chapter 5

Experiments at the RCNP facility

5.0.1 Overview of RCNP

An overview of the RCNP facility at Osaka University is given in figure 5.1.

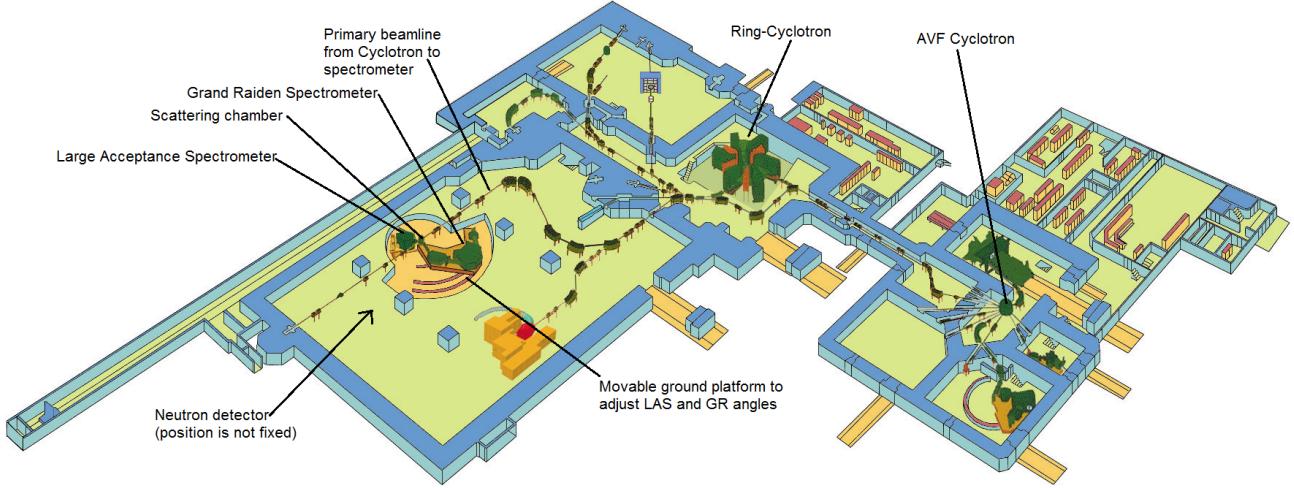


Figure 5.1: An overview of the experimental setup at RCNP. LAS (Large Acceptance Spectrometer) and GR (Grand Raiden spectrometer) are the key components here.

The RCNP building is located at Osaka university, see figure 5.2.

The idea of the experiments here is to generate a primary beam. This beam is first pre-accelerated with AVF cyclotron and then accelerated further with the Ring-Cyclotron. This primary beam is then guided to the scattering chamber, where it collides with a fixed target. The reaction products are the analyzed with either (or both) of the two spectrometers. If neutrons are produced, a neutron detector can be carefully positioned and used as well. For the charge exchange reactions no neutrons are produced, hence this detector is not used in our case.

Both LAS and GR are optimized to measure light ions only. Hence one usually uses a light-ion beam (low Z) and a heavy (high Z) target. Then the reaction products will be usually one light ion and one heavy ion. The light ion is then measured with either LAS or GR and the heavy ion is not measured at all.

LAS has a poor energy resolution, but a high acceptance. Hence it can easily analyze a large number of ions and therefore obtain excellent statistics. GR has a very good energy resolution [11], [?], [?], but a worse acceptance.

Notice that contrary to RIKEN and FAIR, we use a primary beam here for the collision. This has several advantages. The first is that the beam energy is known in advance very very precisely. It is just the amount of cycles the particles has spend inn the cyclotron before extraction. This is known from the operation specifications of the Cyclotron. One also known that this beam is in the pure z -direction. Hence knowing the particle you

use (and you always do, again from cyclotron operations), the 4-momentum of the incoming beam is always precisely known in advance! No inbeam detectors are required for this, like in RIKEN.

Secondly, the Cyclotron RF-frequency (used to accelerate the particles) gives you a very precise reference time. Hence one can always measure TOF with respect to this time reference. No in-beam detectors or anything are required for this either.

Hence if one is able to identify the outgoing light ion from the collision and completely reconstruct its 4-momentum, one can also obtain the 4-momentum of the outgoing heavy ion by momentum conservation (the 4-momentum from the beam is known and the 4-momentum from the target is zero (expect for the rest mass)).

In the case of the ${}^3\text{He}, t$ -exchange reaction, one uses a stable isotope as a target (like ${}^{116}\text{Sn}$) and a ${}^3\text{He}$ -beam. After the collision, one is left with a triton ${}^3\text{H}$ and a heavy ion like ${}^{116}\text{Sb}$. One then obtains the 4-momentum of the outgoing ${}^{116}\text{Sb}$ after the collision. This 4-momentum will have an invariant mass $m = m_0 + E^*$ where m_0 is the mass of the ground state and E^* is the excitation energy. m is obtained from the 4-momentum. m_0 is obtained from the literature, since it is known very precisely. Then one known E^* , which is what this experiment is all about.

5.0.2 Practical information on the experiments

There are 3 important internet networks at RCNP. The first one is eduroam, the same as RUG. it works everywhere and you can use it for about anything. You can however NOT use it to log into the servers for Grand Raiden experiments. For this one needs to use either RCNP-GP or RCNP-VISITOR. RCNP-VISITOR is unprotected, but will sometimes malfunction. RCNP-GP is the one to go here, but one needs to login with a username and a password for this network. Also, be sure to put the authentication method to PEAP for RCNP-GP.

One can use a temporary login with **username: diamghus** and **password: Gag3Dma4Vmu4** This might work, but is still not ideal. In order to get ideal access, one needs to create an RCNP-account. One can do this at <https://www.rcnp.osaka-u.ac.jp/usermanage1/> Note that it will take a few days before this account is processed. After creating an account (with a username of your choice), one will receive an email with account information like a password. With your chosen username and the password send to you you will be able to login to the RCNP-GP internet network. The information on my account is all written on the next page.

ユーザー管理システム- User Management System A

トップメニュー
Top Menu (/usermanage1/menus)

アカウント新規申請 - Application for a new account

希望ユーザー名1 Desired username 1	cad		
希望ユーザー名2 Desired username 2			
希望ユーザー名3 Desired username 3			
ログインシェル Login shell	/bin/bash		
氏名(English) Name in English	*姓 Last	ミドルネーム Middle	*名 First
	Douma	Alwin	Christiaan
氏名(Japanese) Name in Japanese	*姓 Last	ミドルネーム Middle	*名 First
国籍・居住地 Nationality, Residence	国籍 Nationality	Netherlands - オランダ王国	
	居住地 Residence	Netherlands - オランダ王国	
	居住開始日 Start date	1990年(years)	
所属区分 Affiliation Section	その他研究機関 - Other Research institute		
所属 Affiliation	University of GroningenKVI-CARTNuclear and Hadron Physics		
身分 Position	博士課程院生 - Graduate Student Doctor Course 2年生 - 2nd		
所属組織の住所 Affiliation address	9747 AA Zernikelaan 25, Groningen, Netherlands		
所属組織の電話番号1 Affiliation phone number 1	+31 50 363 3600		
所属組織の電話番号2 Affiliation phone number 2			
携帯電話番号 Mobile Phone number			
その他電話番号 Other Phone number			
所属組織のメールアドレス Affiliation e-mail address	c.a.douma@rug.nl メーリングリスト - Mailing list 参加する - Join		
その他メールアドレス1 Other e-mail address 1			
その他メールアドレス2 Other e-mail address 2			
コンタクトパーソン氏名 Contact person name	Akimune, Hidetoshi		

指導教員氏名・メール Supervisor Name, e-mail	その他の人を指定 - Specify the other person		
	氏名 Name	Nasser Kalantar	
	メール E-mail	nasser@kvi.nl	
利用期限日 Expiration date	2017/04/30		
	メール受信方法 Mail receiving method	利用する - Use	
		他センターのアカウント Other center account	
申請(確定) Submit			入力画面へ to edit page

ユーザー管理システム- User Management System A

トップメニュー
Top Menu (/usermanage1/menus)

👤 アカウント新規申請 - Application for a new account

アカウント申請処理が完了しました。
受付完了のメールを指定されたメールアドレスに送付しました。もし届かない場合は、申請が正常に受け付けられていないか、記入したメールアドレスが誤っている可能性があります。
その場合は、下記にお問い合わせください。
申請の審査結果はメールでお知らせ致します。代理人申請の場合は代理人にメールでお知らせします。
以下の「受付番号」「Password」を記録して下さい。

受付番号は問合せ時に、Passwordはアカウント発行後システムにログインする時に必要となります。

Your application form for the new account has been received.

We will inform you the result of the examination of your qualification by e-mail.

Please write down the following 'application number' and 'Password'.

The application number will be required for contacting us about your application.

The Password will be required for logging in after the account is issued.

■あなたの受付番号(Your application number):A0001135

■あなたのPassword(Your password):Fpz4-Tyq8-Hun3 <<-必ずメモ!(Note down!)

*Passwordが分らなくなったり、照会に時間がかかる場合がありますのでご注意ください。

NOTICE. If you lost your password, it may take a long time to reissue.

◆申請に対するお問い合わせ先(To contact us about your application):

consult@rcnp.osaka-u.ac.jp



Christiaan Douma <christiaan90518@gmail.com>

Your application for an account on the RCNP computer has been accepted.

1 bericht

apply-answer@rcnp.osaka-u.ac.jp <apply-answer@rcnp.osaka-u.ac.jp>

24 april 2016 18:08

Aan: c.a.douma@rug.nl

Cc: akimune@konan-u.ac.jp, nasser@kvi.nl

Your application for an account on the RCNP computer has been accepted as follows.

If you have any questions, please send an E-mail to the following address with your application number. Your application number is A0001135

Your login password was already issued at application time.

consult@rcnp.osaka-u.ac.jp

Please refer to the following one of URL to use RCNP computer.

https://www.rcnp.osaka-u.ac.jp/usermanage1/user_logins

https://www.rcnp.osaka-u.ac.jp/usermanage2/user_logins

You can use either because it is redundant system.

User name	:	cad
User id	:	52586
Group id	:	51000
Home directory	:	/home/cad
Login Shell	:	/bin/bash
Name (in English)	:	Douma, Alwin, Christiaan
Name (in Japanese)	:	
Nationality	:	Netherlands - オランダ王国
Residence	:	Netherlands - オランダ王国
Residence start date	:	1990年 - year
Affiliation postal address	:	9747 AA Zernikelaan 25, Groningen, Netherlands
Affiliation phone number 1	:	+31 50 363 3600
Affiliation phone number 2	:	
Mobile phone number	:	
Other phone number	:	
Affiliation e-mail address	:	c.a.douma@rug.nl
Other e-mail address 1	:	
Other e-mail address 2	:	
Affiliation Section	:	その他研究機関 - Other Research institute
Affiliation	:	University of Groningen KVI-CART Nuclear and Hadron Physics
Position	:	博士課程院生
Name of the contact person (in English)	:	Akimune, Hidetoshi
Type of the Supervisor	:	その他の人を指定 - Specify the other person
Name of the Supervisor (in English)	:	Nasser Kalantar
Email of the Supervisor	:	nasser@kvi.nl
Affiliation of the Supervisor	:	University of Groningen KVI-CART Nuclear and Hadron Physics
Date of issue (YYYY/MM/DD)	:	2016/04/24
Expiration Date (YYYY/MM/DD)	:	2017/04/30
Mail alias	:	Christiaan.Alwin.Douma@rcnp.osaka-u.ac.jp
Mail receiving method	:	利用する - Use

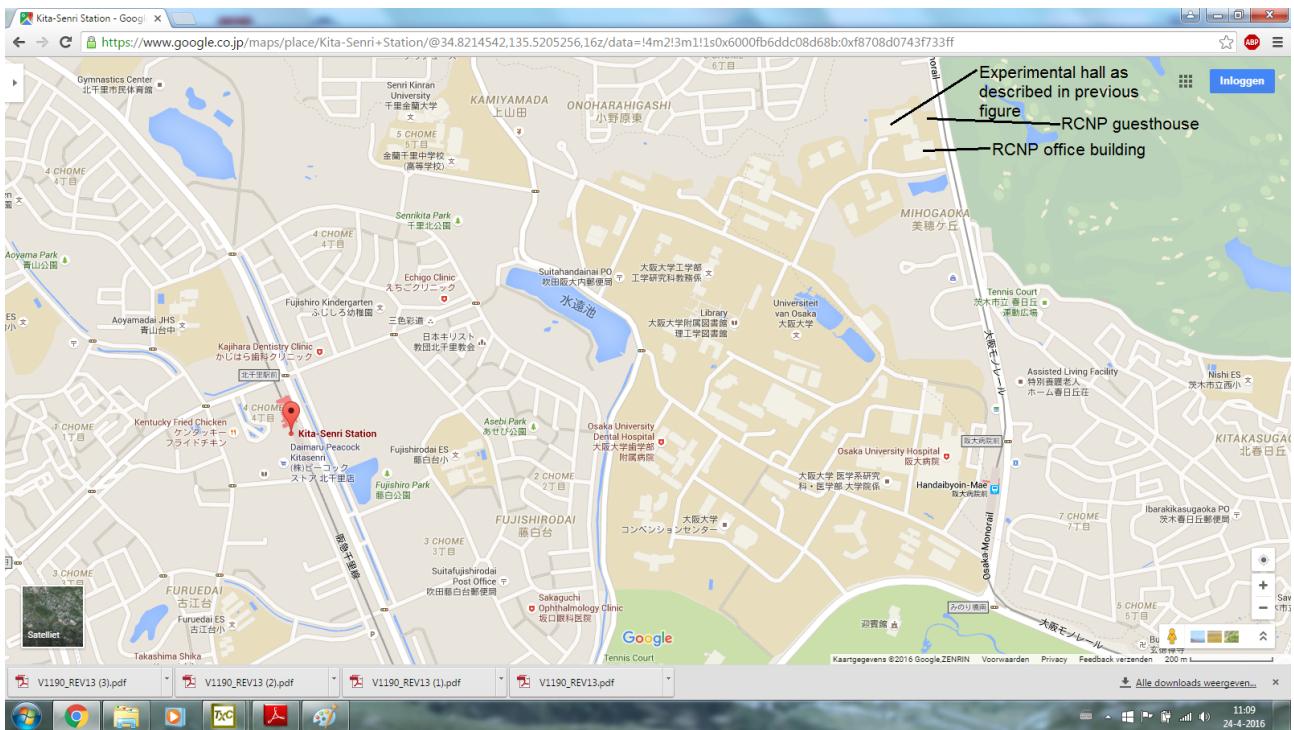


Figure 5.2: Location of RCNP facility.

the important thing to remember is **username**: cad and **password**: Fpz4Tyq8Hun3 to get access to RCNP-GP network. In case this does not work, you can use Heguri his account. For this account, use **username**: heguri and **password**: pMR9VUAb to log into RCNP-GP.

Now to login to the Experiment Server, open a linux terminal and give the following command:
`ssh -Y e451@aino-1.rcnp.osaka-u.ac.jp` and use the password **password**: p2HeE451 to gain access. The raw data files of the experiments are located in `/np1a/v05/e451/data/` (you arrive in the folder `/home/e451/` after you log in). Or use `ssh -Y e451@login-1.rcnp.osaka-u.ac.jp` or `ssh -Y e451@miho-1.rcnp.osaka-u.ac.jp` or `ssh -Y e451@ftp-1.rcnp.osaka-u.ac.jp` this will all connect you to the same server and to the same `/home/e451/` directory. The password is always the same. The addition `-1` can also be `-2` and then the passwords are different because this is a different server. This `-2` is better because the DAQ does not write to this server. You can also change the `e451` to a different experiment. `e425` for example. Different experiments have a different password. for `e425` the password is `E425daq1` Notice that you will get to the same server but now in directory `/home/e425/` instead. The data (of all experiments) is still located in `/np1a/v05/e451/data/`

Notice that in order to be able to ssh to the server, the server has to know your mac address. Hence before the first time you log in, you have to register this MAC address at <https://www.rcnp.osaka-u.ac.jp/netmanage1/> and log in with either your own RCNP account or with Heguri his account. Then push the button 'general user'. After this is done, you will see a screen like figure 5.3.

The box with **IP Adres** can be left blank. The box with **host/IF Name** should be filled preferably with your computer name (the computer name of your OS installation). **MAC Adress** is most important box. Click on the connection symbol in Ubuntu in your upper right corner. Then choose 'edit connections' and choose the connection you want to add. There, click the 'Ethernet' tab. Duplicate the content of the **MAC Adress** box here into the box in figure 5.3. The box **Operator** should be filled with the user name of the account you used to log into this web page. **Administrator** should always be filled with **Tamii** and the box **Model Number** should preferably be filled with the model number of your computer. The box with **Comment** can be left blank. Then register this addition and then log out. Wait for a few minutes and then the new connection should work.

Now the raw data files (not yet unpacked) have the following structure. They all have a filename, which contains the run number. Then they have a certain extension. There are always 6 different extensions for each run. The run is only complete when you have all of these 6 files. The extensions are:

NOTE: All of these files are directly written by the DAQ. The DAQ does NOT perform any processing or computations on the data, it just writes the data to these files. The event reconstruction means that the DAQ

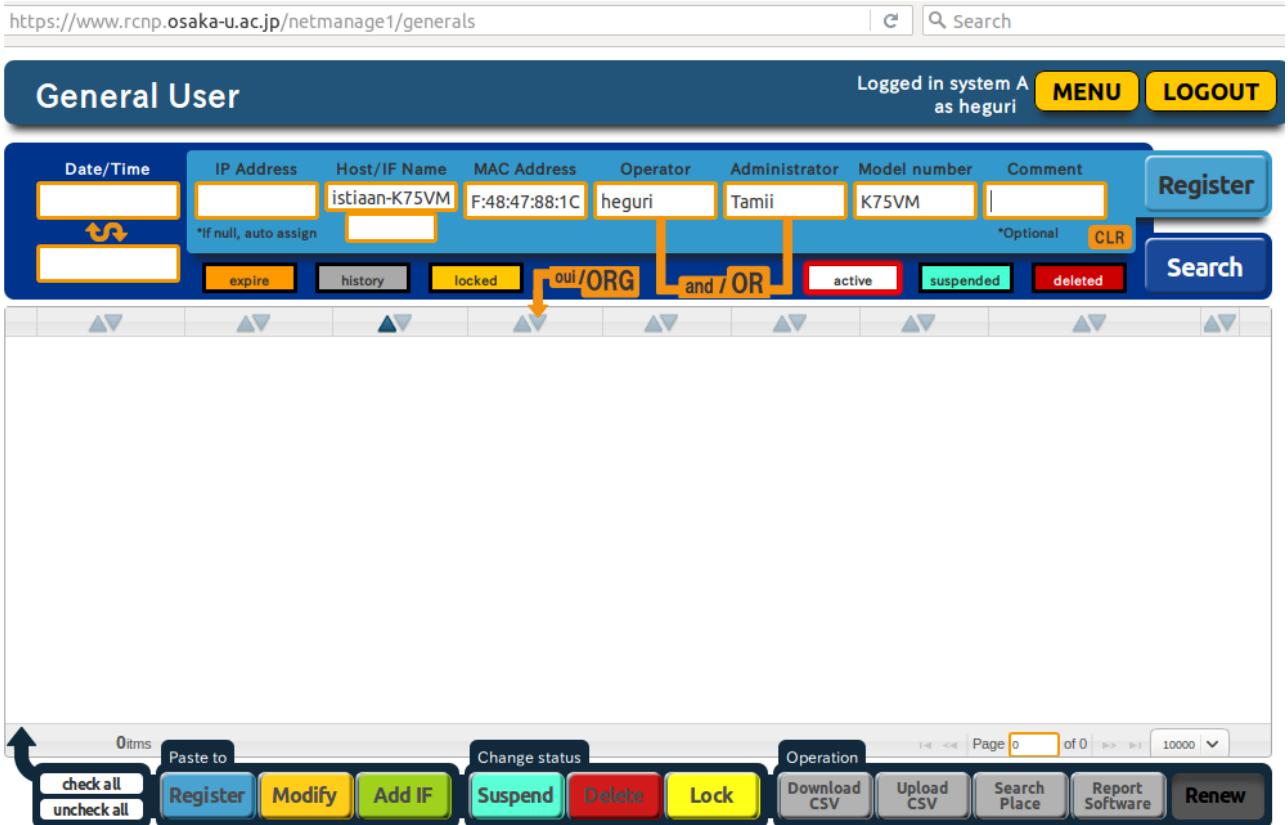


Figure 5.3: How to add your mac address on RCNP.

.lv	All Raw Data of the LAS spectrometer. Both Drift chambers and scintillators
.gv	Data of the GR spectrometer on the Drift chambers.
.gr	All Raw Data of the GR spectrometer. Both Drift chambers and scintillators
.blp	Primary Beam parameters. Usually this file is (almost) empty.
.grb	intermediate writing step. This file is generated from the .gr file
.bld	Final writing step (build-file). All event reconstructed data is contained in here.

Table 5.1: The file extensions of a single run.

simply decides which pulses of GR and LAS belong to which events. Events are designated based on the RF frequency reference time of the cyclotron. This information is NOT stored in the other files.

The DAQ first creates the .lv, .gv, .gr and .blp files. Then it generates the .grb-file from the .gr-file. Finally, it generates the .bld-file from the .lv, .gv, .grb and .blp files. In this step it performs the event reconstruction. In principle, the .bld file now contains all information on all detectors. Hence this file is what you need to perform your analysis at home. But it is of course better to possess all the files.

Now in order to unpack the data, one needs a piece of software that is called 'the analyzer'. It does not only unpack the data, but also performs track reconstruction, cuts, etc. This is explained in the next section. The analyzer can be obtained from

<http://vmeserver.rcnp.osaka-u.ac.jp:8080/ExpDBBrowser/#Rate%20Meter> Here one can also view some progress on the experiment. For questions on the analyzer, ask Gey Guillaume, one of the constructors on this software (email: gey@rcnp.osaka-u.ac.jp). The analyzer is simply a .tar.gz-file. Hence put in in a directory of your choice and untar it there with tar -xzvf <>filename>>. To get the analyzer, log into the server with ssh -Y e451@aino-1.rcnp.osaka-u.ac.jp and password p2HeE451 then go to the path /home/e451/ana/gey/ and copy the file analyzer_20160423.tar.gz this file is all you need!

Now in order to be able to perform your analysis, you have to perform the following tasks. The first one is to run the command sudo apt-get install cernlibs or sudo apt-get install paw to install the CERN Libraries and the PAW analysis program. PAW and CERN Libraries are the predecessor of the current ROOT program. It is now no longer maintained by CERN, but fortunately Debian took over. Hence the physics is no longer maintained, but code optimization and making is suitable for modern operating systems etc. is still done. And because Debian now does it, no complicated installation and compilation procedures are required. Instead, standard linux apt-get install can now do everything automatically. That's great! the program PAW can now be started by simply typing the command paw It then requires a startup condition as input. Type 0 and hit enter. Now you are ready to go!

Now to get the analyzer working, put in in a directory of your choice and untar it there with tar -xzvf <>filename>>. Then go to the subdirectory ./analyzer/ and run the command make there to compile the analyzer. It should finish with a lot of warnings, but without errors. If not, you probably just miss a few linux packages that are required. Check the log and install them then try again. An up-to-date Ubuntu 15.04 system with FairRoot installed on it should have all required packages by default.

Next, go back to the directory where you untarred the analyzer and then open the offline-script. Scroll a bit down and modify the blddir-variable. This should become the absolute path to the directory where all your raw datafiles are located (this directory is allowed to hold multiple runs). Now open the ./chanruns.sh script and adapt the run number in first line of this script to the run you need (and make sure all those files are in the blddir-directory!) Finally, go back to the directory where you untarred the analyzer and run the command chmod -R +x ./* to make all files executable. Now run the analyzer with the command ./chainruns.sh It should work without flaws. If it doesn't, send an email to Gey Guillaume (email: gey@rcnp.osaka-u.ac.jp) and ask him for help. The log can be retrieved with the command less .child00.stderr in case of a crash.

Data for a test run of the analyzer can be obtained from:

```
ssh -Y cagragr@miho-1
```

```
password: CagfDaq1
```

```
Data location: /np1a/cagra/gr/tests_daq/gey/datatest/
```

```
Copy run number 1001. Use an scp command:
```

```
scp cagragr@miho-1:/np1a/cagra/gr/tests_daq/gey/datatest/*1001* ./
```

This copies the data for the test run to the directory where you are when executing the scp-command.

5.0.3 Operation principles of the Grand Raiden Spectrometer

A lot of useful information is contained in [12], [13] and [14]. An overview of Grand raiden is given in figure 5.4.

First the primary beam comes into the scattering chamber to collide with a fixed target. The heavy ion is not measured and the light ion comes into the Grand Raiden spectrometer. It first passes through the Q1, Q2 and SX arrays. Q1 is the QR-magnet. A big quadrupole for focussing. SX is a sextupole magnet and Q2 is again a dipole magnet. All for focussing. MP also provides focussing of the beam. Between QR magnet and the scattering chamber is a place where one can place a beam collimator if desired.

Grand Raiden Overview

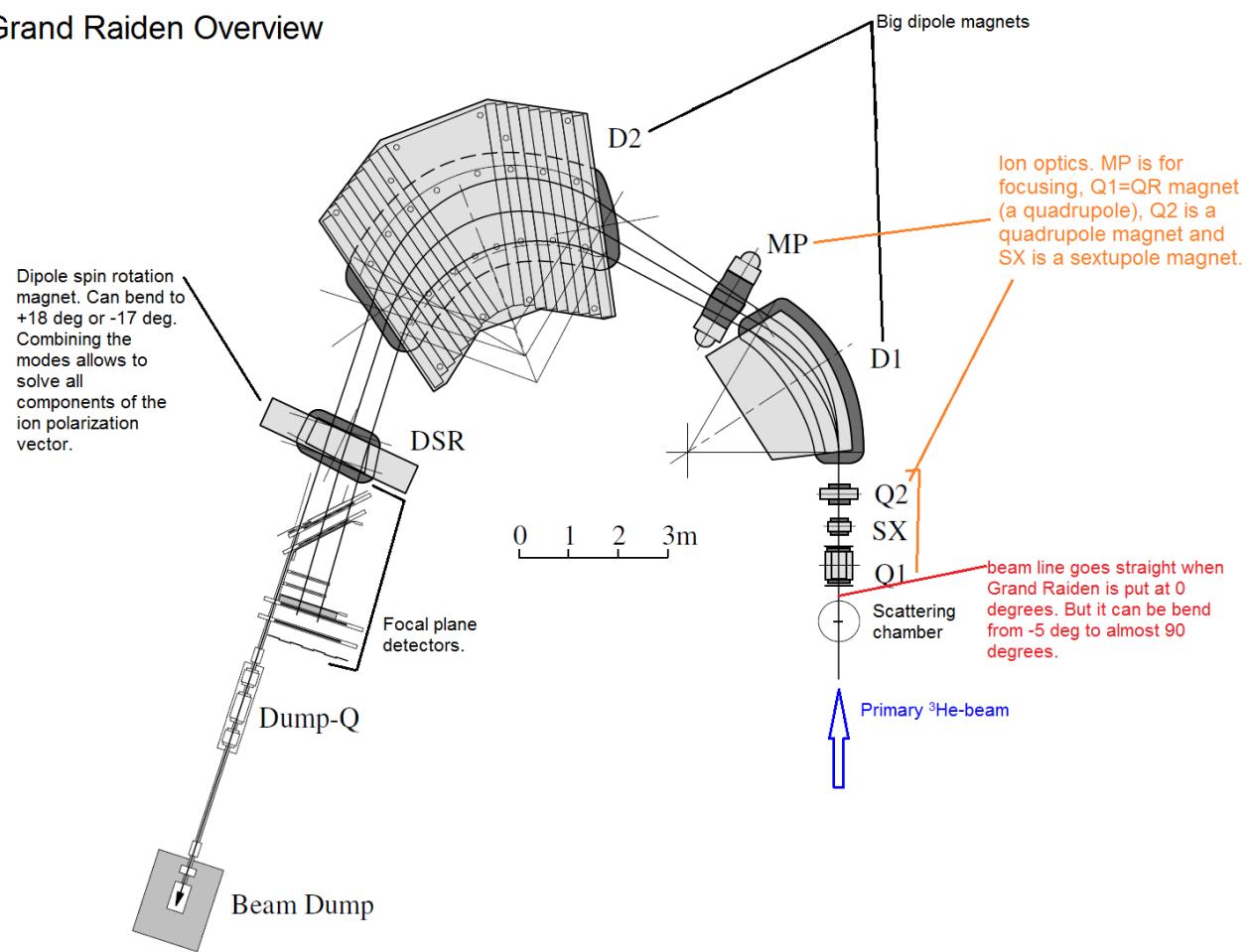


Figure 5.4: Overview of the Grand Raiden Spectrometer.

Then the light ion passes through the big dipole magnets. It are actually these dipole magnets that provide the separation and therefore the spectroscopy. The ions will all have a slightly different initial angle and slightly different initial momentum. Hence they are all bend in a slightly different way. Hence they arrive at different positions (at different bending radii) at the focal plane detectors. These focal plane detectors can then measure the position of the ions. With this information one can then reconstruct the initial momentum, exactly what we want. The angular acceptance of Grand Raiden is ± 20 mrad in the x -direction and ± 40 mrad in the y -direction. the aperture is elliptic.

After the dipole magnets, but before the focal plane detectors, one has the DSR dipole magnet. This magnet can bend protons either $+18^\circ$ furter, or -17° back. By changing this angle, one can obtain all components of the ion polarization. Think about the spin dipole experiments at first year Quantum Mechanics courses! That is what is happening here!

Then we arrive at the focal plane detectors. An overview of this setup is given in figure 5.5.

First we pass through two wire chambers: VDC1 and VDC2. Each chamber consists of 2 layers of wires: one vertical and one diagonal. Then we pass through an array of horizontal scintillator bars: PS1. Note that VDC1, VDC2 and PS1 are not perpendicular to the beam z -axis! Close by the design orbit (when an ion is exactly leaving the scattering chamber at 0° and with exactly the energy we intended), we also have a few beam monitors.

Ciombining the information of both drift chambers one can extract the position and the direction of the ion track through the drift chamber. On actually obtains (x, y, θ, ϕ) at the reference focal plane. x and y are the coordinates in this plane and θ and ϕ specify the direction out (shperical coordinates). Notice that the wirs fired in the chamber also have TDC information which is used to compute (x, y, θ, ϕ) very precisely.

Then the Scintillators PS1 give us TDC and QDC. The scintillator TDC is particulary useful for the trigger conditions. The QDC gives us the ΔE of the ion in the scintillator, hence with Bethe-Bloch this can tell us something about the total energy. We usually get the energy from time of flight from either wire chamber or scintillator TDC. Then we know position and detector geometry, hence we know the TOF and the travel distance. This gives β from Relativity. From particle identification we get the mass, hence then the energy is given. This more accurate than from Scintillator QDC, but this QDC is still nice for making analysis cuts as we will see later on.

Then we pass again though 2 drift chambers, a scintillator, a carbon slap, another two drift chambers and finally Hodoscope. The principles of the PS2 scintillators are the same as for the PS1, but these 4 drit chambers are different. VDC has one vertical plane and one diagonal plane. MWPC has one vertical plane, a diagonal plane and another diagonal plane in the other orientation.

There is also a carbon slap in between, which is ment to scatter the ion. Then the MWPC 1 and 2 and PS2 measure the ion before the scattering and MWPC 3 and 4 and the Hodoscope HS-X measure the ion after the scattering. HS-X is just like PS 1 and 2, only these scintillators are vertical, not horizontal. And there are 8 of them. We scatter the ion to obtain its polariarion. We know the track energy and position before and after the scattering, hence the collision can be reconstructed. If we now do this for both the $+18^\circ$ and the -17° bending of DSR, one can reconstruct the polarization.

Hence the VDC 1 and 2 and PS1 are simply to obtain the track position and energy as it comes out of the scattering chamber. Them MWPC 1-4, PS2 and HS-X are there to obtain the polarization of the ion. In our experiment we use unpolarized beam, so we do not need (and do not use) MWPC, carbon slab and hodoscope.

for the next experiment, log into `ssh -Y e425@aino-1` with password `E425daq1`

The precise specifications on Grand Raiden and the wire chambers etc are given in figures 5.6, 5.7, 5.8.

5.0.4 Meeting report 25 April

The wiki page of the experiments is <http://www.rcnp.osaka-u.ac.jp/Divisions/np1-a/wiki/index.php?E425>

Focal plane detectors have been upgraded to work with b1190V now [12]. Hence the data system is no longer compatible with LR3377. The analyzer now uses the new system.

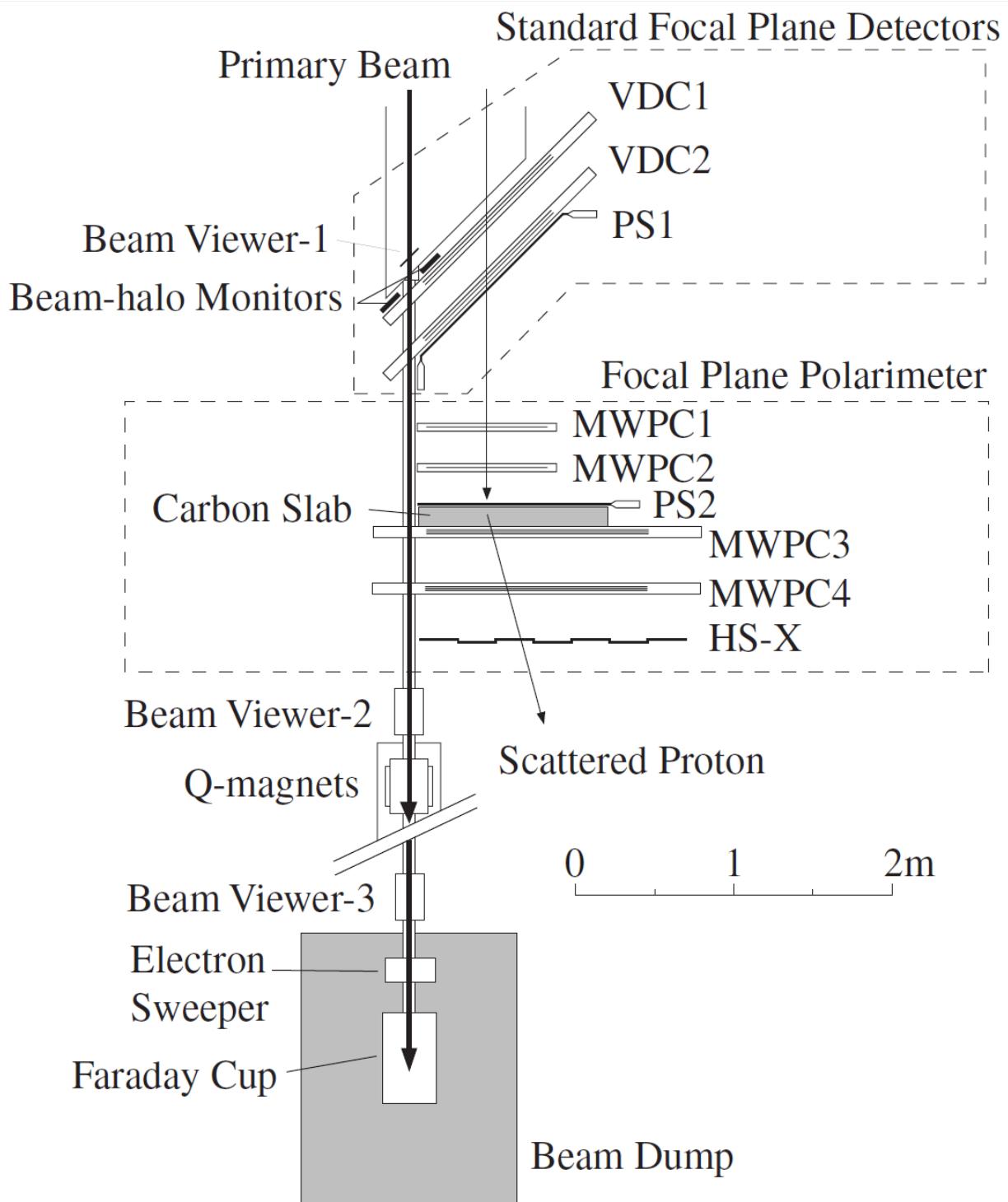


Figure 5.5: Overview of Focal Plane Detectors in the Grand Raiden Spectrometer.

Design specification of the *Grand Raiden* spectrometer.

Configuration	QSQDMD(+D)
Mean orbit radius	3 m
Total deflection angle	162°
Focal plane length	150 cm
Focal plane tilting angle	45.0°
Maximum particle rigidity	54kG-m
Momentum resolving power $p/\Delta p$	37,000
Momentum broadness	5%
Horizontal magnification ($x x$)	-0.417
Vertical magnification ($y y$)	5.98
Momentum dispersion ($x \delta$)	15451 mm

Figure 5.6: Grand Raiden Specifications

Specification of the VDCs.

Wire configuration	X(0°), U(-48.2°)
Active area	1150 ^W × 120 ^H mm
Number of sense wires	192 (X), 208 (U)
Anode-cathode gap	10 mm
Anode wire spacing	2 mm
Sense wire spacing	6 mm (X), 4 mm (U)
Anode sense wires	20 μm gold-plated tungsten wire
Anode potential wires	50 μm gold-plated beryllium copper wire
Cathode film	10 μm carbon-aramid film
Applied voltage	-5600 V (cathode), -300 V (potential), 0 V (sense)
Gas mixture	argon:iso-butane:iso-propyl-alcohol = 70:30: ¹
Pre-amplifier	LeCroy 2735DC
Digitizer	LeCroy 3377 drift chamber TDC

¹Mixed with the argon gas in 2°C vapor pressure.

Figure 5.7: VDC Specifications

Specification of the MWPCs.

	MWPC1,2	MWPC3	MWPC4
Wire configuration	X(0°)	X(0°), U(-45°), V(+45°)	
Active area	760 ^W × 200 ^H mm	1400 ^W × 418 ^H mm	1400 ^W × 600 ^H mm
Number of wires	384	704 (X), 640 (U,V)	704 (X,U,V)
Anode-cathode gap	6 mm		
Anode wire spacing	2 mm		
Anode wires	25 mm ϕ gold-plated tungsten wire		
Cathode film	10 μm carbon-aramid	6 μm aluminized mylar	
Cathode voltage	-4900 V	-4700 V	
Gas mixture	argon:iso-butane:freon:iso-propyl-alcohol = 66:33:0.3: ¹		
Pre-amplifier	LeCroy 2735PC and Nanometric N277-C3		
Digitizer	LeCroy PCOS III		

¹Mixed with the argon gas in 2°C vapor pressure.

Figure 5.8: MWPC specifications

The installation of Sieve-Slits and Faraday-Cups (FC elements) were discussed. See figure 5.9.

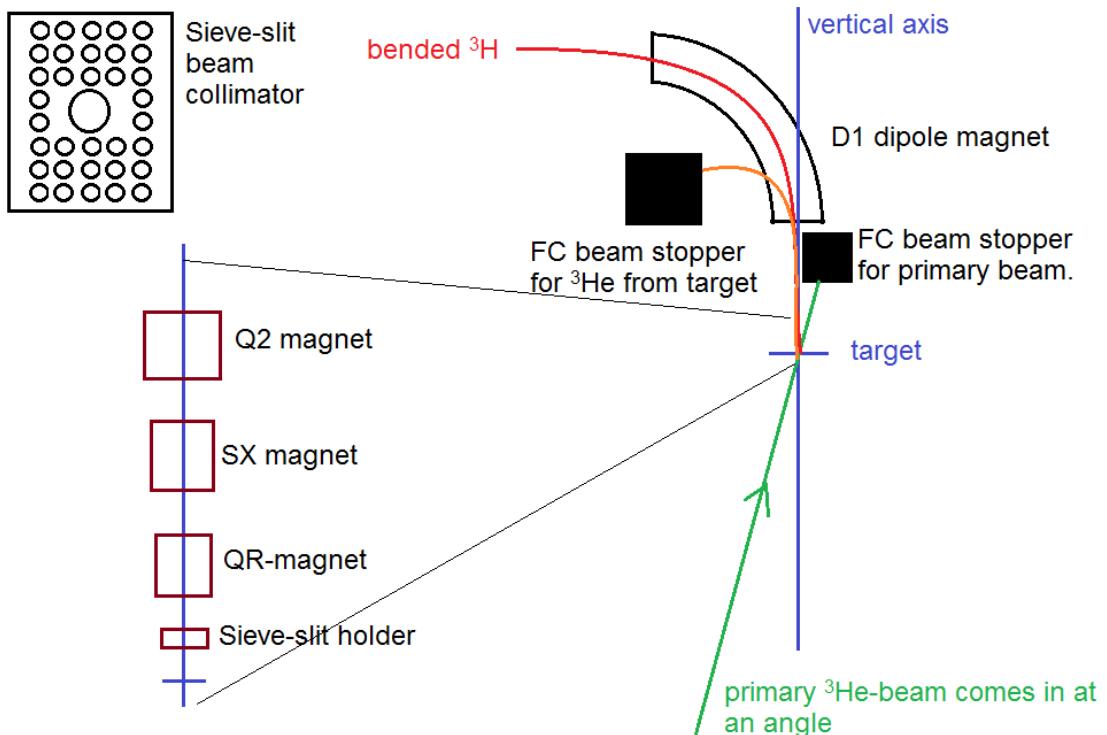


Figure 5.9: Installation of Faraday-Cups and Sieve-Slit.

Faraday cups are needed as beam stoppers (just blocks of metal, usually Aluminium). The Sieve-Slit is put in during beam tuning. Then one obtains a distorted image of the Sieve-Slit pattern on the focal plane detectors. Since one knows the original Sieve-Slit and the image, one knows the aberrations and distortions of the spectrometer and one can then correct for it. Once we know this correction, the Sieve-slit is removed and only after that the real experiment is done.

One uses 1 mm, 3 mm and 10 mm thick scintillators. Thin is better, since a $^3\text{He}^{1+}$ ion might convert to ^3H in the scintillator. $^3\text{He}^{1+}$ just bends like ^3H and is therefore not separated by the dipoles. Since in the end you have in both cases ^3H , you do not know which ions came from the target. Hence a thin scintillator keeps the probability for this conversion very low.

If the spectrometer runs at 0° , H and He are well-separated by the dipoles. At bigger angles, this separation becomes less good. Gives you also background.

^3He elastic scattering is good to measure the energy beam spread.

Q -value of an ion is the difference (in energy) between the ^3He and the ^3H peak. Large Q -value is good for calibration purposes.

faint beam means that a human could count the particles one-by-one.

The use of achromatic and dispersive beam transportation during the experiment is explained in figure 5.10.

Sieve Sli has 0.49 deg vertical between the holes and 0.39 deg horizontal. The hole size is about 3 mm or 2 mm. The distance to the target is about 598 mm.

The idea is that we artificially give the beam profile different energies and different positions to hit the target, so that the entire beam converges to a single point. This is called a dispersive beam pattern. However, if one of the ions will now have a slightly different energy loss in the target (during the collision) than what we anticipated, then this ion will of course hit a different spot on the spectrometer than the design orbits. This way the position of the focal plane tells you very accurately about the energy loss in the target and therefore gives you a very high energy resolution. Exactly what we need for these experiments.

The targets that we are going to use are given in figure 5.11.

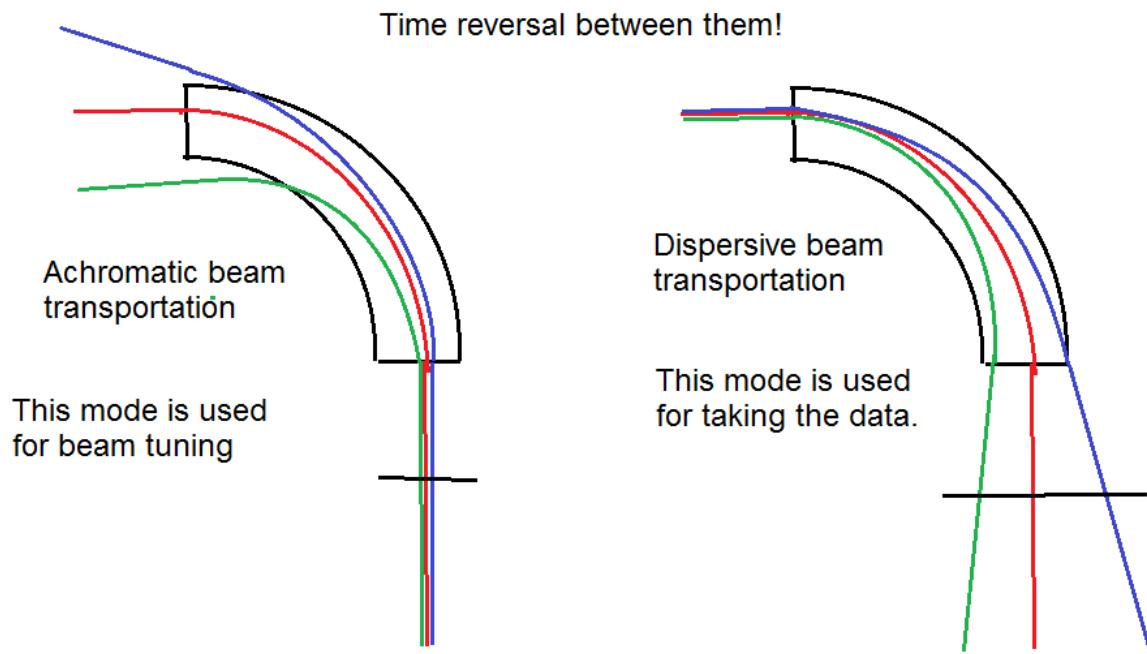


Figure 5.10: Beam transportation during the experiment.

	Areal density	Aim	Place, "Target box label"
natMg	1.2 mg/cm ²	Ex calib	左上棚, "natMg, 13CH2 ..."
Mylar	9 μm^3	Ex calib	棚の上, "13CH2, natCH2, ..."
13CH2	70 μm	Angle calib	? , "natMg+B"
197Au	1.68 mg/cm ²	Elastic tuning	左上棚、"Sn, Al, Au"
118Sn	1.4, 1.24, ~2.2, ~4.7 mg/cm ²		左上棚, "Sn, Al, Au"
120Sn	2.7 mg/cm ²		左上棚、"Sn, Al, Au"
30Si	~5 mg/cm ²	Data?	右上棚、"32S, 30Si, ..."
112Sn	2.7 mg/cm ²		"HS/AT/YF"
114Sn	7.51 mg/cm ²		"HS/AT/YF"
116Sn	4.96, 3.7 mg/cm ²		"HS/AT/YF"
118Sn	4.5(3) mg/cm ²		"HS/AT/YF"
120Sn	6.5 mg/cm ²		"HS/AT/YF"
122Sn	5.2 mg/cm ²		"HS/AT/YF"
124Sn	5.0, ~7.8?, ~2 mg/cm ²		"HS/AT/YF", 7.8 is on non-RCNP frame
208Pb	4.77 mg/cm ²		アクリル板 サンドイッチ
208Pb	5.2 mg/cm ²		GSI 208Pb box

Figure 5.11: Targets to be used during the experiment.

5.0.5 Tasks and operation principles of the Analyser

The analyzer unpacks the data and executes some pre-processing. For the pre-processing, the analyzer first looks at the wires of both planes in each of the wire chambers. If the number of wires that has fired is below or above a certain boundary, the entire event is rejected (and not saved in the output of the analyzer). Only events where EACH plane in EACH drift chamber is between the two boundaries, the event is saved.

For each wire we record if it fired yes/no and we record the TDC of the pulse (QDC is not recorded). With these TDC and knowledge of the drift velocity, the exact position where the particle has passed can be reconstructed (take all fired wires along). This is illustrated in figure 5.13.

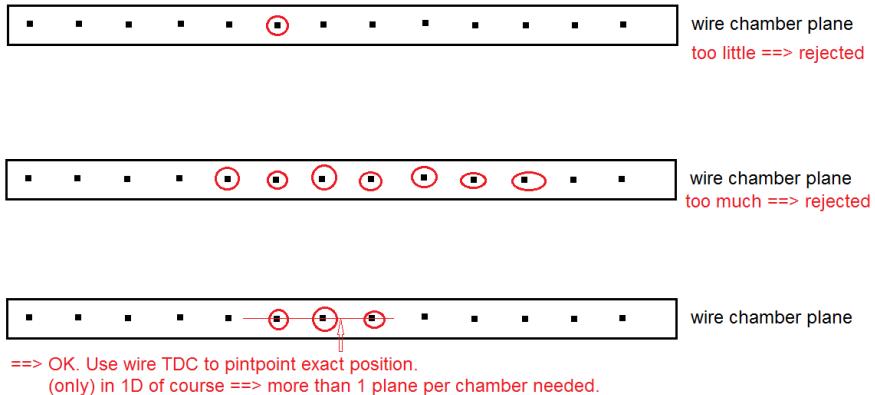


Figure 5.12: Reconstruction in the wire chambers.

The next step the analyzer does is now to reconstruct the tracks through the VDC chambers. See figure

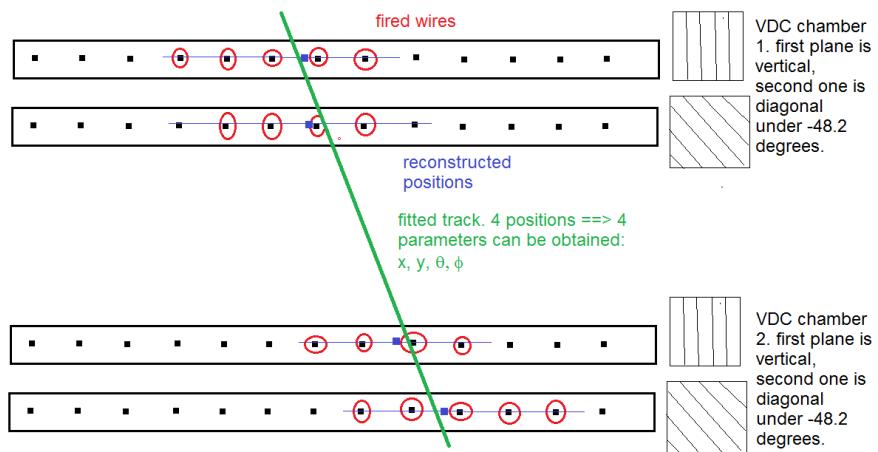


Figure 5.13: Reconstruction of the particle track.

both of the VDC chambers each have two planes. These are 4 planes. Hence 4 positions can be obtained as described above. Then a straight line is fitted through these positions. 4 positions can give us 4 parameters of the line. They are the x and y position on the focal plane and the θ and ϕ angles that specify the direction through the focal plane. This is done like accelerator physics coordinates: (x, y, θ, ϕ) with respect to the design orbit of the particle (see figure 5.5). θ is in the horizontal plane and ϕ is in the vertical plane, hence NOT spherical coordinates! But what is precisely the design orbit? See the section about Grand Raiden. ϕ is the angle between y -axis and trajectory, θ is the angle between x -axis and trajectory. The y -axis runs vertical. The detector Focal plane (where we measure x and y) is the plane defined by VDC1 detector.

The MWPC wire chambers and the carbon slab, etc. are not used in our experiment, since we used unpolarized beam.

Now from the fitted line and precise knowledge on the geometry of Grand Raiden and the $B\rho$ of the dipole magnets, one can track back the particle through the spectrometer all the way back to the target. Then

one knows the scattering angle of this particle. Also, one can use the TDC of the wire chambers and/or the scintillators (PS1 and PS2) to obtain the TOF of the particle (Use Cyclotron RF as reference time). From the tracking through the spectrometer, one also knows the flight distance of the particle. Hence one can obtain β from special relativity. Then since one has to tune the dipoles precisely onto a single isotope of ions, one also knows which particle came through the VDC. Since Grand Raiden is only meant for very light ions, one cannot possibly miss-ID the particle. Then from the particle ID one gets the ground-state mass. From this mass and β one can obtain the energy of the particle.

Hence from this energy and the direction of the track (or the scattering angle), one can make a complete reconstruction of the 4-momentum of the particle. This kinematic reconstruction is also done by the analyzer.

If at some point during the track reconstruction, the analyzer finds more than 1 track within a single event, the entire event is also discarded.

The analyzer assigns a binary variable `GR_RAYID` to each event. If this variable equals zero, everything of the track reconstruction went OK and the full event is saved in the output. If this variable is anything else than zero, the event is not saved and only this variable is saved. Hence during analysis one should NOT make a cut on `GR_RAYID`. This cut is already made for you since the wrong events are simply not there in the output.

The other values of `GR_RAYID` are in table 5.2.

0	Everything OK!
1	VDC1 vertical plane failed to reconstruct position.
2	VDC1 diagonal plane failed to reconstruct position.
4	VDC2 vertical plane failed to reconstruct position.
8	VDC2 diagonal plane failed to reconstruct position.

Table 5.2: The file extensions of a single run.

Note that if `GR_RAYID`=3 then both planes of VDC1 failed, and so on. The table is , since `GR_RAYID` can have values up to 50, although those values are very rare.

Notice that the analyzer does NOT do any synchronization procedures on the TDCs of the wire chambers or the scintillators. Synchronization procedures like in the PMTs of NeuLAND are not necessary since we use primary beam here. Hence the Cyclotron RF provides us with excellent reference time to put all raw TDC against. This is not the case for NeuLAND, since there one uses a secondary beam. Hence there one has to synchronize everything since a clear reference time is not easily available there.

5.0.6 Analysis tricks you can do yourself (NOT done by the analyzer)

The scintillators record both TDC and QDC. TDC can be used for TOF information. However, the best use for the scintillators is to generate the triggers. The energy is obtained from TOF, not from QDC of the scintillators. This could be done. $QDC = \Delta E$ and through Bethe-Bloch this gives the particle energy. But this is inaccurate. However, the QDC and TDC is useful for triggers and for analysis cuts.

One could for example plot ΔE of PS1 scintillators against ΔE of PS2 scintillators and put a 2D-gate on this to select a single type of particles. See figure 5.14.

One could also plot the exitation energy (fully reconstructed!) against the RF-time of the cyclotron in figure 5.15.

Since there is TOF on the x -axis there will be no pile-up here. Hence one can again select a blob by putting a 2D gate on this plot.

There are many other tricks. Report them here as you learn them...

5.0.7 How to do Online Monitoring

To monitor trigger rates and detector succes etc, go to `vmeserver:8080/ExpDBBrowser/#Rate%20Meter` and check out the data there. To see more precise data, do the following steps.

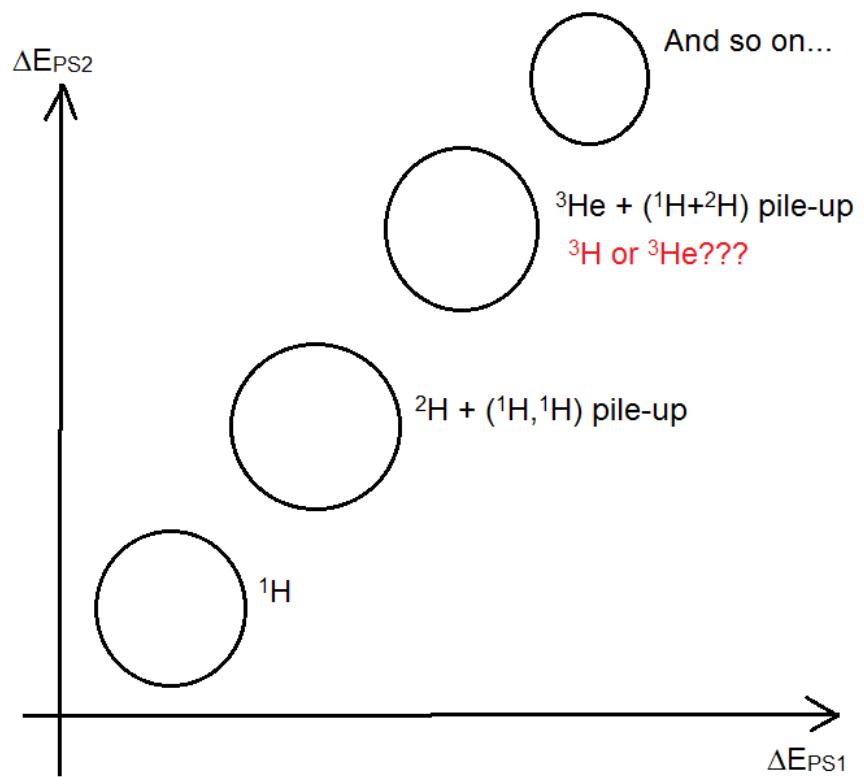


Figure 5.14: Possible cuts on ΔE of scintillators.

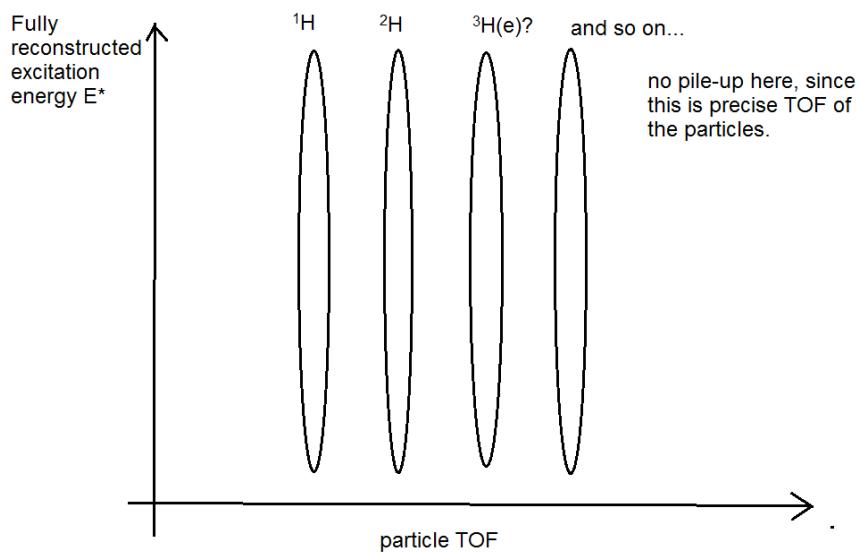


Figure 5.15: Possible cuts on particle TOF.

One version of the online monitor is located in `/home/e451/ana/online_16_Apr` (Log in to `ssh -Y e451@aino-1.rcnp.osaka-u.ac.jp` server with password p2HeE451).

need 3 experimental accounts for 3 experiments. Each account gets a directory for me. STRICTLY FORBIDDEN to duplicate ata from one account into another or to your own PC (unless you have permission to analyze that data).

online monitor means fire the analyzer on online data ipv offline. It works the same, but the online version has no root option, so you have to use PAW instead. (online and offline work exactly the same, except for the conversion to ROOT).

go to `/home/e451/ana/gey` after loggin into the `ssh -Y e451@aino-1.rcnp.osaka-u.ac.jp` server with password p2HeE451. This number will be different for different experiments. For example for E424: go to `/home/e424/ana/gey` after loggin into the `ssh -Y e442@aino-1.rcnp.osaka-u.ac.jp` server with password (...) or use path `/home/e424/ana/cad` once you have your own directory.

Then first start the online monitor with `./online` This calls the analyzer to process the run in progress now!

Then to view the outcome of the online analysis, start PAW with `paw` and then give command `glob bob` The command is `glob` and then `bob` is the shardmemory name for the DAQ (online analysis!). To check the name, give `grep -rn 'BOB' ./` If the first line ends with BOB (and is not a temprary ~-file, be careful!), then this is the shell name. You can also open the file `/home/e451/ana/gey/analyzer/include/freadin.h` and check line 39: `#define SHMNAME "BOB"`

`glob bob` connects to the shared memory of the DAQ. I no error message shows up, it worked. IMPORTANT: never run more then 1 online job at the same time. The problem is if multiple people connect to the same shared memory name. So only one person can really do the online stuff! other people should go offline with the previous run.

now run `hfb temp 9030`. `hfb` is the command. `temp` is a user filename tag of the output. It is created when you give this command. So you can take whatever you want. 9030 is the run number. This will give a message on unit 20 that you can ignore but no other errors should show up. Then give command `h1 -histolist` This produces a list of all histograms in the `.hb` file produced by the analyzer. All of these histograms are also loaded. Then with the command `hp 134` you plot a histogram. 134 is the number from the histogram list that you want to plot. If you want a zoomed plot for the *x*-axis, give `hp 134(1200.0:1600.0)` Notice that both arguments should be double C++ arguments and they are separated by : Zooming in *y* axis is more difficult. Set log-scale by `opt logy` and then re-plot. Set linear scale back by `opt liny`

To plot multiple histograms in one canvas, give `zone 2 2` (2 vertical, 2 horizontatl, hence 4 in total). Then fill the canvas 1 histogram at a time by giving `hp`-commands.

in `/home/e451/ana/gey/macro/` are some useful PAW macros. make sure to duplicate them! Look for extension `.kumac` There you will find the PAW code.

file `/home/e451/ana/gey/hist.def` contains all the definitions of histograms for PAW. There is oone big line in here that contains all variables that are written to the ROOT-tree in the end. Some names come directly from the DAQ and others are defined just below that long line (for example to split array variables into single variables. After writing, the arrays are rconstructed inside the leafs again). After line 1150 the histograms are defined. Wen editing this file, respect the position of ALL blank chararcters, otherwise it crashes! Lines with a <<->> in front are commented and will not be processed to the `.hb`-file. A single <<->> after a 2 for example means that every line is commented untill the next 2 (or any lower number).

in `/home/e451/ana/gey/` there is a program `hm` that can read `hist.def` Give: `hm hist.def` command. Then you see a better structure than with kwrite. But it is slow... It also does not work very stable, so you better not use it.

In order to add a histogram, just duplicate one of the line. Take for example:

```
5 hist1 GADC1L; "GR ADC 1L"; GR_ADC[1]; 0; 2048; 512; "ADC [ch]"; "Counts/ch"; EVENT_GR_COIN
```

And modify the duplicated line to your needs. `hist1` is the histogram type. This one is a 1D histogram (`hist2` would be 2D) That are all types that exist in PAW. `GADC1L` is the pointer to the histogram that you use to access it inside your code. "GR ADC 1L" is the histogram title (displayed in the plot). `GR_ADC[1]` is the variable name: the content of the histogram. 0; 2048; 512 define histogram boundaries. It will go from 0 to 2048 with 512 bins. "ADC [ch]" is the *x*-axis label. "Counts/ch" is the *y*-axis label (counts axis). `EVENT_GR_COIN` is the

condition. It is a cut on the variable that you plot (only 1D cuts are possible). No argument here gives you no cuts and you get everything.

```
def GR_RFGate = gate(GR_RF,526,658) || gate(GR_RF,1045,1205)
```

This is how to define a cut. `def` means that the cut should be defined. You should always start with this for defining a cut. `GR_RFGate` is the name of the cut. `gate` is the function that defines the cut. `GR_RF` is the variable name the cut should be imposed on. Then the two numbers are the lower bound and higher bound. You only keep everything between the bounds. That is basically all. Then one can do boolean operations in defining a cut, like the `||` which is a C++ OR-operation. This way one can define more complex cuts.

After you defined the cut you can give the name of the cut as input to the histogram definition discussed above. NOTE: The variable you plot in the histogram and the variable you use in the cut do NOT have to be the same variable. Hence if you have data pairs (x, y) you can put x with a cut on y . Exactly what you need for data analysis!

The decoder or unpacker is `/home/e451/ana/gey/analyzer/scr/fread_rcnp.c` look only for the place of the line `case ID_V1190` that is the system we currently use. `case ID_FERA_ADC` and `ase ID_FERA_TDC` are also useful since they deal with the scintillators. In this file the raw variables are defined. (FERA does not have trivial names. Hence those names will be written to the ROOT-files with a different name).

One final important remark. For offline analysis, the entire procedure is EXACTLY the same. Only now you run the command `./offline -R "" temp 9030` in the beginning instead of the command `./online`. Here 9030 is the run number you want to analyze. `temp` is a nametag for the output files. You can use anything you like. `""` Is an argument that should always be this and always be there. The option `-R` will write a `.root`-file with a ROOT-tree as output (besides the `.sca` and `.hb` files that are the standard output. In the main directory of the analyzer a directory `hb` is created for the `.hb`-files, the same for the `.sca`-files and the `.root`-files are saved in the `rootfiles`-directory (inside the main directory of the analyzer).

In the `hist.def`-file are some conditions that do not apply to our experiment. This is why a lot of events have empty data in the ROOT-files. This still has to be fixed. The Wire TDCs will be reconstructed to x -position and so on. This is done by drift velocities etc. Due to the bad conditions in the `hist.def`-file this reconstruction based on drift velocity fails much more often than it should. This still should be fixed.

The meaning of the variables in the outputs are displayed in figure 5.16.

```
// NOTE: Each variable has its own content version and a mult-version. Example: GR_X yields the content and GR_X_mult yields the array size of that event. In most
// cases this array size is just 1.

// NOTE: All Double_t quantities are still in [Number of Channels]. This means that they come out of the electronics as raw data. Nothing is calibrated yet. Hence the
// units are still unclear. They come out as raw binary data and still are in that form. You yourself have to calibrate. Use the calibration runs for that. Make an
// analysis of the data for a target that is already published by someone else. Compare your output to the paper and use that to define your calibration and hence unit definition!

GR_RAYID // Binary result variable of track reconstruction. This is explained in the log. Also see the BinaryCode.txt
GR_RF // Radio Frequency: Time difference between GR trigger and Beam frequency (probably obtained from beam scintillator). [Number of channels]
GR_X // Measure between trigger start and the point where we have maximum beam intensity. Plot GR_X versus GR_RF, you will get multiple blobs. The difference
// between two successive (in RF-direction) blobs corresponds to 1-T of the cyclotron frequency. Than you can calibrate. Usually you just use it for cuts.
// X-position on the Grand Raiden Focal plane. In Tamii his analyzer this focal plane is the middle of VDC-1. [mm, but Uncalibrated]. This X-position is
// directly proportional to GR excitation energy (=Energy loss by collision). In Dispersion mode all primary beam without collision arrives at a single
// point on the focal plane. Then GR_X is the difference on the Focal plane between this point and the particle track. This is directly proportional to
// the energy loss in the target.
GR_Y // Same as GR_X. Only this is in the vertical direction and GR_X in the horizontal direction (on the focal plane. this plane is vertical in the LAB-frame,
// but not perpendicular to the beam. See GR specs. [mm, but Uncalibrated]
GR_TH // theta-angle. angle in the horizontal plane between the particle trajectory and the design orbit (beam that did not scatter). It could also be between
// the orthogonal of the VDC-1 and the particle trajectory. Carefully read Tamai his Thesis to be sure. [Number of Channels]
GR_PH // phi-angle between the particle trajectory and the vertical axis. [Number of Channels].
GR_ADC // GR has 6 photomultipliers reading out the scintillators. Mostly one just uses the first 4 of them. The GR_ADC variable gives you the pulse content in
// [Number of Channels]. GR_ADC_CHAN gives you the number of the photomultiplier that fired the pulse. GR_ADC_mult gives you the total array size of that event.
GR_TDC // Same as GR_ADC (also only for the scintillators), but now the content is TDC, not ADC [Number of Channels].
GR_TPPOS // Position in the scintillator of the hit (Toiff-calibration). [Number of Channels] There are 3 scintillators (3 planes). Hence this array has a size up to 3.
// The GR_TDC arrays has a size up to 6 photomultiplier. PMT 1 and 2 form plane 1 together, plane 2 = PMT 3 and 4 and plane 3 is PMT 5 and 6.
// So basically just the same as TDC_size = 6 * number of channels. This is literally programmed as TDC2 - TDC1 Hence uncalibrated.
GR_MADC // Raw data after the first deposit at the hit. (Correcting for attenuation). Just like in NeuAND. Format is exactly the same as GR_TPPOS, only this one is with ADC.
// It is defined as GR_MADC = sqrt(GR_ADC*GR_ADC2).
GR_WIRE_X1 // Number of the wire that fired in the X (vertical) plane of VDC1 drift chamber. [wire number]
GR_WIRE_X1 // This is an array. GR_WIRE_X1 mult holds the array size and then the content are the wire numbers that fired in that event.
GR_WIRE_U1 // Number of the wire that fired in the U (diagonal) plane of VDC1 drift chamber. [wire number]. Same principle as GR_WIRE_X1
GR_WIRE_X2 // Number of the wire that fired in the X (vertical) plane of VDC2 drift chamber. [wire number]. Same principle as GR_WIRE_X1
GR_WIRE_U2 // Number of the wire that fired in the U (diagonal) plane of VDC2 drift chamber. [wire number]. Same principle as GR_WIRE_X1
GR_TDC_X1 // TDC of the wire that fired in the X (vertical) plane of VDC1 drift chamber. [Number of Channels]
GR_TDC_X1 // This is an array. GR_TDC_X1 mult holds the array size and then the content are the TDC of the fired wires. Element GR_WIRE_X1[2] = 5 and GR_TDC_X1[2] = 2134 then correspond
// to each other (same index). This means that wire number 5 fired with a TDC-value of 2134 [number of channels].
GR_TDC_U1 // TDC of the wire that fired in the U (diagonal) plane of VDC1 drift chamber. [Number of Channels]. Same principle as GR_TDC_X1
GR_TDC_X2 // TDC of the wire that fired in the X (vertical) plane of VDC2 drift chamber. [Number of Channels]. Same principle as GR_TDC_X1
GR_TDC_U2 // TDC of the wire that fired in the U (diagonal) plane of VDC2 drift chamber. [Number of Channels]. Same principle as GR_TDC_X1
GR_WTDC_X1 // Result of previous versions of the software. Do not use this variable. It holds no information!
GR_WTDC_U1 // Result of previous versions of the software. Do not use this variable. It holds no information!
GR_WTDC_X2 // Result of previous versions of the software. Do not use this variable. It holds no information!
GR_WTDC_U2 // Result of previous versions of the software. Do not use this variable. It holds no information!

// GR = Grand Raiden, LAS = LAS Large Acceptance Spectrometer. Hence all variables starting with LAS are about LAS and all variables starting with GR are about GR.

// NOTE: MWPC, carbon slab, etc. We do NOT use MWPC-chambers, carbon slab, Hadoscope, etc. For our charge exchange reactions. This is used to measure polarizations and that is meaningless when we
// do not use a polarized beam. We do not use that here, so this part is meaningless. Hence the variables for MWPC and so on are NOT in our current version of the analyzer.
```

Figure 5.16: Explanations of the Analyzer output variables.

NOTE::: Tamai send an email that it might be necessary to rebuild the `.bld`-files. Be sure to check this!

5.0.8 Planning of my PhD Thesis

The angular acceptance of Grand Raiden is ± 20 mrad in the x -direction and ± 40 mrad in the y -direction. The aperture is elliptic. hence you can divide your data in bins of 0.5° . See proposal, histogram at different colors. First you get histograms in number of counts. This can be converted to absolute cross sections by knowing particle charge and beam current intensities, etc. See Phys. Rev C 86 014304 (2012). This step is called Analysis I (extracting cross sections and excitation energy spectrums). Analysis I should take 6 to 10 Months. Analysis II should take a few Months.

Analysis II Do some theoretical calculations. Fit the data by the right functions (angular distributions). Then with this fits do physics interpretations concerning double beta decay nuclear matrix elements and the consequences to neutrino (astro) physics.

These kind of experiments are already done for many other targets, so look at the papers!

E425 is spin-dipole states, most relevant to neutrinoless double beta decay. E452 2 neutrino double beta decay and Gamov-Teller resonance.

Look at paper [15] and get references 20-26 in that paper. They are about the RCNP experiments.

1^+ state of Cadmium is very relevant. Spin-dipole is important for neutrinoless double beta decay. Gamov-Teller is important for 2 neutrino double beta decay. 1^+ state is then very important. Look at [15] references 1, 2 and 4. Very good introduction to double-beta decay. 2 neutrino double beta decay, usually the neutrino takes 1 MeV with it. That very low. Neutrinoless double beta decay is 2^- state 3^+ states.

Cd-116 you mainly study 1^+ states.

experimentally you always get both Gamov-Teller and Spin-dipole. Always one of the two is dominant.

Hattori does Te-126 and Ge-174. I will do (Sn-122 and Sn-116) and/or (Cd-116 and Cd-natural). Probably. This might still change. Cd is for Gamov-Teller and Sn is for spin-dipole.

Sn isotopes. We sjould do low (0-30 MeV) and high energy measurements (above 30 MeV)

You can call prof. Hiro Ejiri with his skype name `ejiri.hiro`

5.0.9 Michael his analyzer

Go to `Michael/orz_e322.sh` and edit the fourth line, the `bldfile` that tells you where the build file to analyze is located. Give the entire build-file name including the absolute full path. Also change the `key` variable and the `nruns` variable both to the appropriate run number (not the entire file name).

Use the Find-option to search for `dragon` to see chainges in `orz_main.cpp` done by Michael. The file `Michael/orz/orz_main.cpp` is the main file of this analyzer. Look for `Christiaan` to see your own tips.

Notice that the analyzer from Gey and from Michael use a different file-format in the `.bld`-files! Hence only one of the analyzers can read a specific `.bld`-file.

After editing `Michael/orz_e322.sh`, one can run this script to read the `.bld`-file. (Use `make` to compile the `orz`-program first). The output is then located in `Michael/root/` Take the right run-number. However, the output only contains histograms, not a ROOT tree. In order to get this one has to run `wurzel` AFTER one has run `orz`

Then one goes to `Michael/wurzel/` and compile the `wurzel`-program with `make` Then open the file `Michael/wurzel/wurzel_numbers2do` And modify the run number to the number you just analyzed with the `orz`-program. Then give the command `./wurzel82Se` and be patient. Then the resulting tree will be stored in `Michael/wurzel/results/82Se_wurzel.root` The `wurzel`-program knows how to do this because `orz` created not just ROOT-histograms, but also a `.dat`-file stored in `Michael/tracks/` that contains the tree information.

Michael his email address is `michael.holl@uni-muenster.de` And he has a complete backup of all data of the experiment!!!

5.0.10 Binary Code from the wiki page

5.0.11 Accelerator

AVF: Particle: ${}^3\text{He}^{2+}$. Energy 87.8 MeV. Frequency: 11.650070 MHz.

RING: Particle: ${}^3\text{He}^{2+}$. Energy 420 MeV. Frequency: 34.950210 MHz. 6th harmonic.

One circle is roughly 30 keV. 1 mm \approx 35 keV at GR_X When the x -plane is diagonal (VDC-1).

5.0.12 Summary of the Experimental Log

Distance between Sieve Slit and target: 598 mm. Only θ (at the focal plane) gives you θ at the target. x at the focal plane gives you the energy. In the y -direction, we put the magnets at overfocus mode. We tune it at exactly 2.5λ so to speak, so that the particles arrive straight. Then only y at the focal plane determines ϕ at the target. So plot θ -measured against θ at target (known from Sieve-Sli geometry). But this is different for different energies. And simple binning in bins of x does not work, since x and $|\theta|$ at focal plane are correlated. Hence you should make 2D plots of X versus θ -plane. and a plot of x versus θ -target. This is the first step. Then we can deal with y later on.

Collimator is always open, unless we use the Sieve Slit.

Run 5025 and lower: Achromatic beam tuning. Then to Dispersive mode (Except for the explicit beam tuning)
D1 means Faraday Cup D1FC is used (inside the dipole magnet)

SC means Faraday Cup SCFC is used (inside the scattering chamber)

Q1 means Faraday Cup Q1FC is used (inside the QR-quadrupole magnet right after the scattering chamber)

Beam intensity is in Electro-nano-Amp.

Angle (deg) ==> of GR spectrometer.

Run 5030 to run 5085 have a low efficiency on the VDC2 X-plane and a strange triangular structure in TD

5125 (11 enA), Q1 E425 (2.5 deg) Cd-116

5124 (9.5 enA), Q1 E425 (2.5 deg) Sn-122

5123 (9.5 enA), Q1 E425 (2.5 deg) 13CH2

5122 not good. SCFC is still inside.

5121 (10 enA), D1 E425 (0 deg) 13CH2

5120 (8.6 enA), D1 E425 (2.5 deg) Te-126

5119 (9 enA), D1 E425 (2.5 deg) Te-126

5118 (9 enA), D1 E425 (2.5 deg) Ge-74

5117 (8.5 enA), D1 E425 (2.5 deg) Te-126

5116 (8 enA), D1 E425 (2.5 deg) Te-126 (RF of AVF-cyclotron went down. So we aborted).

5115 (8 enA), D1 E425 (2.5 deg) Ge-74

5114 (10.6 enA), D1 E425 (2.5 deg) Ge-74

5113 (9.58 enA), D1 E425 (2.5 deg) Te-126

5112 (9.8 enA), D1 E425 (0 deg) Ge-74 small frame (Aluminium)

5111 (8.5 enA), D1 E425 (0 deg) Te-126

5110 (10 enA), D1 E425 (0 deg) Ge-74

5109 (10 enA), D1 E425 (0 deg) Ge-74 (Data is corrupted, only 0 kB)

5108 (10 enA), D1 E425 (0 deg) Ge-74

5107 (9 enA), D1 E425 (0 deg) Ge-74 (operators re-aligned the beam)

5106 (10 enA), D1 E425 (0 deg) Te-126 small frame

5105 (7.1 enA), D1 E425 (0 deg) Ge-74 (${}^3\text{He}, t$) NOTE: The DAQ comment here is Te-126, but that is wrong.

5104 (6.6 enA), D1 E425 (0 deg) Te-126 small frame

5103 (6.9 enA), D1 E425 (2.5 deg) Te-126

5100-5102: problem with vmedaq_builder. Hence these runs do not mean anything.

5099 (6.5 enA), D1 E425 (2.5 deg) Te-126 (${}^3\text{He}, t$) peak at -30 = IAS peak.

5098 (7.4 enA), D1 E425 (2.5 deg) Te-126

5097 (7 enA), Q1 E425 (2.5 deg) Sn-116

5096 (7 enA), Q1 E425 (2.5 deg) Te-126

5095 (7.5 enA), Q1 E425 (2.5 deg) Sn-116
 5094 (7 enA), Q1 E425 (2.5 deg) Te-126 small frame
 5093 (6 enA), Q1 E425 (2.5 deg) Te-126
 5092 (6 enA), Q1 E425 (2.5 deg) Te-126
 5091 (6 enA), D1 E425 (0 deg) Te-126 (something with 0-)
 -268: 13C ground state 2-
 -267: 1+ state
 5090 (6 enA), D1 E425 (0 deg) Te-126
 5089 (6 enA), D1 E425 (0 deg) Sn-116
 5088 (6 enA), D1 E425 (0 deg) Te-126
 5087 (6 enA), D1 E425 (0 deg) Sn-116
 -213: 3+ ground state (or 3+ between ground state)
 -212: 1+ state 948 keV width
 -202: 2+ state 550 keV width
 5086 (6 enA), D1 E425 (0 deg) Te-126
 5085 (5 enA), D1 E425 (0 deg) Te-126 small frame
 -269: 126I ground state 2-
 -267: 1+ 126-Te-->126-I
 -246: large peak
 ==> problem on VDC X2 efficiency ==> with TDC. (TDC shape an deficiency problem)
 5084 (5 enA), D1 E425 (0 deg) Te-126 (Run stopped when RF frequency went down.)
 5083 (5 enA), D1 E425 (0 deg) Te-126 beam current decreased to 3.76 enAin the end of the run.
 5082 (5 enA), D1 E425 (0 deg) Ge-74 small frame (3He,t)
 5081 (5 enA), D1 E425 (0 deg) C-natural small frame
 5078 (5 enA), D1 E425 (0 deg) C-natural small frame
 5077 (5 enA), D1 E425 (0 deg) Ge-74 (DAQ halted in the end)
 5076 (5 enA), D1 E425 (0 deg) C-natural
 5074 (5 enA), D1 E425 (0 deg) Ge-74
 -261: 2- state
 -256: 1+ state
 5073 (5 enA), D1 E425 (0 deg) C-natural small frame. At +50 12C ground state peak.
 At +135: 12N-peak 3.6 MeV.
 5072 (5 enA), D1 E425 (0 deg) Ge-74 small frame
 -280: 180 ground state. (0 deg)
 -263: 74Ge 2- ground state. (0 deg)
 -259: 74Ge 1+ state. (0 deg)
 -260: 180--> 0+ state.
 -270: 13C-peak (1 deg)
 5071 (5 enA), D1 E425 (0 deg) small empty frame
 5070 (5 enA), D1 E425 (0 deg) C-natural: +55: 2H ground state peak. 135: 12C-12N 3.6 MeV excited state
 5069 (9 enA), Q1 E425 (2.5 deg) Sn-122 beam tuning. Part of the run is also faint beam. After failure
 5068 (9 enA), Q1 E425 (2.5 deg) C-natural (power failure in progress)
 5067 (9.5 enA), Q1 E425 (2.5 deg) C-natural (power failure in progress)
 5066 (9.5 enA), Q1 E425 (2.5 deg) C-natural (power failure started, so run was aborted)
 5065 (10 enA), Q1 E425 (2.5 deg) Sn-122
 5064 (10 enA), Q1 E425 (2.5 deg) Sn-122
 5063 (8 enA), Q1 E425 (2.5 deg) Ge-74
 5062 (4.5 enA), Q1 E425 (2.5 deg) Ge-74 beam current was 4.5 between 12:00 and 12:20. then it became 6
 5061 (4.5 enA), Q1 E425 (2.5 deg) Ge-74
 5060 (7 enA), Q1 E425 (2.5 deg) C-natural
 5059 (7 enA), Q1 E425 (2.5 deg) Sn-122
 5058 (7 enA), Q1 E425 (2.5 deg) Ge-74 + Al-frame
 -270: 13C ground state
 -263: 2- ground state
 -259: 1+ state, 206 keV
 -243: 0.44 MeV
 -223: 1.88 MeV
 -115: IAS-peak, 6.7 MeV
 5057 (5 enA), Q1 E425 (2.5 deg) C-natural; x=+55: 2C 12N peak Q = 17.3
 5056 (5 enA), Q1 E425 (2.5 deg) C-natural
 5055 (6.5 enA), D1 E425 (0 deg) Sn-122

5054 (6.5 enA), D1 E425 (0 deg) No target
 5052 (7 enA), D1 E425 (0 deg) Sn-122 (3He,t)
 5051 (7 enA), D1 E425 (0 deg) Sn-122 (3He,t)
 5050 (7 enA), D1 E425 (0 deg) Sn-122 (3He,t)
 -287: 2- ground state
 -283: 1+ excited state
 -256: 1+ excited state (run 5050 or 5047)
 5049 (7.5 enA), D1 E425 (0 deg) V-51
 5048 (10 enA), SC E425 (0 deg) V-51
 sharp energy peak at GR_X (=Energy) around -230 and having width of 66 keV.
 5047 (7.5 enA), SC E425 (0 deg) Sn-122
 peaks at GR_X at -375 (1.65 MeV) and -410 (Double peak ==> 2- (left peak) and 1+ states (right peak) A
 5046 (9 enA), SC E425 (0 deg) Mg-natural, similar test at 2.5 deg and D1 is possible.
 -380: 11B 11C ground state
 -315: 11B 11C 2.000
 -310: 25Al ground state +26 Al 0.228
 -285: 26Al 1.058
 -255: 25Al 16.5 & 26Al 1.85
 -254: omega
 -245: 11B 11C 4.319
 -235: 11B 11C 4.808
 GR_X is in mm and 1mm = 34 keV approx.
 Ex is MeV gives us:
 -0.2: 18F ground state
 0.0: 1C ground state 3/2
 2.0: 11C 2.000
 4.3: 11C 4.319 5/2
 4.8: 11C 4.804 3/2
 5.0: 10C 3.35 2+
 7.0: 10C 5.22
 8.2: 11C 8.105 3/2
 8.5: 1C 8.420 5/2
 13.5: 16F ground state
 13.95: 16F 0.42
 15.4 12N ground state
 ==> Ex values from Y. Fujita et al.
 5045 (8 enA), D1 E425 (0 deg) Mg-natural (3He,t)
 5044 (5->8 enA), D1 E425 (0 deg) Mg-natural (3He,t)
 5043 (5 enA), D1 E425 (0 deg) No target
 5042 (faint enA), D1 E425 (0 deg) No target, small area.
 large peak at GR_X at 56 and smaller one at 60 (not understood. Penetrating the target frame?) 29 keV
 5041 (5 enA), D1 Beam Tuning (8 deg) no target, or 197-Au? (He-elastic)
 peak at GR_X = 60.
 5040 (19 enA), D1 E424 Cd-116 (3He,t)
 GR_X peaks at -460 (Cd-116), -430 (Cd-114) and -390 (CD-112) (run 5038)
 3He+ peak is at -340 (5036 and 5038)
 5039 (18 enA), D1 E424 (0 deg) Cd-116 (3He,t)
 5038 (18 enA), D1 E424 (0 deg) Cd-116
 5037 (18 enA), D1 E424 (0 deg) Cd-natural
 5036 (18 enA), D1 E424 (0 deg) Cd-natural
 5035 (5 enA), D1 E424 (0 deg) Si-30
 5034 (1 enA), D1 E424 (0 deg) Si-30
 5033 (1 enA), D1 E424 (0 deg) Si-30
 5032 (7.5 enA), D1 E424 (0 deg) Al-27 (3He,t) 0.98 mg/cm
 best resolution after dispersive matching is 25 keV
 5031 (9 enA), D1 Sieve Slit (0 deg) 13-CH2
 5030 (6 enA), D1 Sieve Slit (0 deg) 13-CH2
 GR_Y gives you phi-target. GR_Theta gives you
 theta-target. What about GR_X?
 5029 (2.2 enA), D1 Beam Tuning (0 deg) 13-CH2
 5028 (0? enA), D1 Background (0 deg) 13-CH2

```

5027 (3 enA), D1 Background(8 deg) 13-CH2
5026 (6 enA), D1 Sieve Slit (0 deg) 13-CH2
5025 (6 enA), D1 Beam Tuning (8 deg) 197-Au (He-elastic) 1.68 mg/cm
5023 (? enA), D1 Beam Tuning (8 deg) 197-Au (He-elastic)
5006 (? enA), D1 Beam Tuning (8 deg) 197-Au (He-elastic)
5005 (? enA), D1 Beam Tuning (8 deg) 197-Au (He-elastic)
5004 (? enA), D1 Beam Tuning (8 deg) 197-Au (He-elastic)
5003 (? enA), D1 Beam Tuning (8 deg) 197-Au (He-elastic)
5001 (? enA), D1 Beam Tuning (8 deg) 13-CH2 (He-elastic)

```

5.0.13 Identifying states

We take an ^{122}Sn target as an example to explain this procedure. Before the collision, the target is obviously in its ground state, which we know from literature that is a 0^+ -state. Then we shoot a 420 MeV ^3He -beam on this fixed target. Then first the ^3H -particle is absorbed by the ^{122}Sn target, forming a highly unstable ^{125}Te nucleus that is in a 420 MeV excited state.

Next, a ^3H -particle is emitted. This particle takes most of the 420 MeV energy with it, but not all of it. Then a ^{122}Sb -nucleus is left. Depending on the energy the ^3H -particle took, the ^{122}Sb -nucleus will be in its ground state, or in one of its excited states.

Now the ^3H -particle will be bend differently by the spectrometer, depending on its energy. Hence depending on its energy, the ^3H -particle will arrive at different x -positions on the focal plane, where its position is detected in the VDC-planes. Hence the x -position is determined by the ^3H -particle energy, which is determined by in which state the ^{122}Sb -nucleus is left.

The y -axis is upward and the z -axis is along the beam. Hence the x -axis has to point towards the scattering chamber of Grand Raiden. Hence if the ^{122}Sb -nucleus is left in its ground state, the ^3H -particle takes a lot of energy with it and is therefore weakly bend. It will therefore end up at negative x -values. If on the other hand the ^{122}Sb -nucleus is left in a highly excited state, the ^3H -particle takes little energy with it and is therefore strongly bend. It will therefore end up at positive x -values.

Now plot x versus counts. Then the peak at smallest x -values corresponds to the ground state. Then every next peak means a higher excited state. The peak position tells you the excitation energy (the ground state is defined at $E = 0$). The peak height tells you the cross section of producing such a state.

Now if you make plots of x versus counts for different bins in scattering θ -angle, the peak heights will be different for different values of θ . Now pick a single state and plot the peak height versus θ . Fit a Bessel Function though this data. The order of the Bessel function tells you Δl , the change in angular momentum with respect to the initial ground state of ^{122}Sn . $\Delta \pi = (-1)^{\Delta l}$, the change in Parity with respect to the ground state of ^{122}Sn . However, we know that due to nuclear effects all beta decay are Gamov-Teller, hence $\Delta s_{GT} = 1$ and $\Delta \pi_{GT} = +$. So From the fact that it is Gamov-Teller we know that the spin part is always 1^+ . The Orbital part you get from the fitting. Then you have to combine those things to get the full effect. Then knowing that the ground state of ^{122}Sn is a 0^+ -state will then tell you what kind of state the peak is that you are analyzing. See also your theory chapter of this log.

Also notice that these Bessel functions are only true in the so-called Plane-Wave Born Approximation. When you assume that the $^3\text{He}^{2+}$ wave function is a flat wave front before the collision. This is not always true due to distortions. In a distorted way, the Δl fits will look like figure 5.23. Hence from the spectra we see that the first peak is $\Delta l = 1$. Then $\Delta pi = (-1)^1 = -$. Hence $\Delta l = 1^-$. From Gamov-Teller we have that $\Delta s = 1^+$. Hence the total state is $\Delta j = 2^-$. That is the trick!!!

5.1 Running an experiment at RCNP

When chaining a run, the following steps have to be taken.

step 1 Stop the DAQ. Go to the DAQ controller and press **stop**. See figure 5.24.

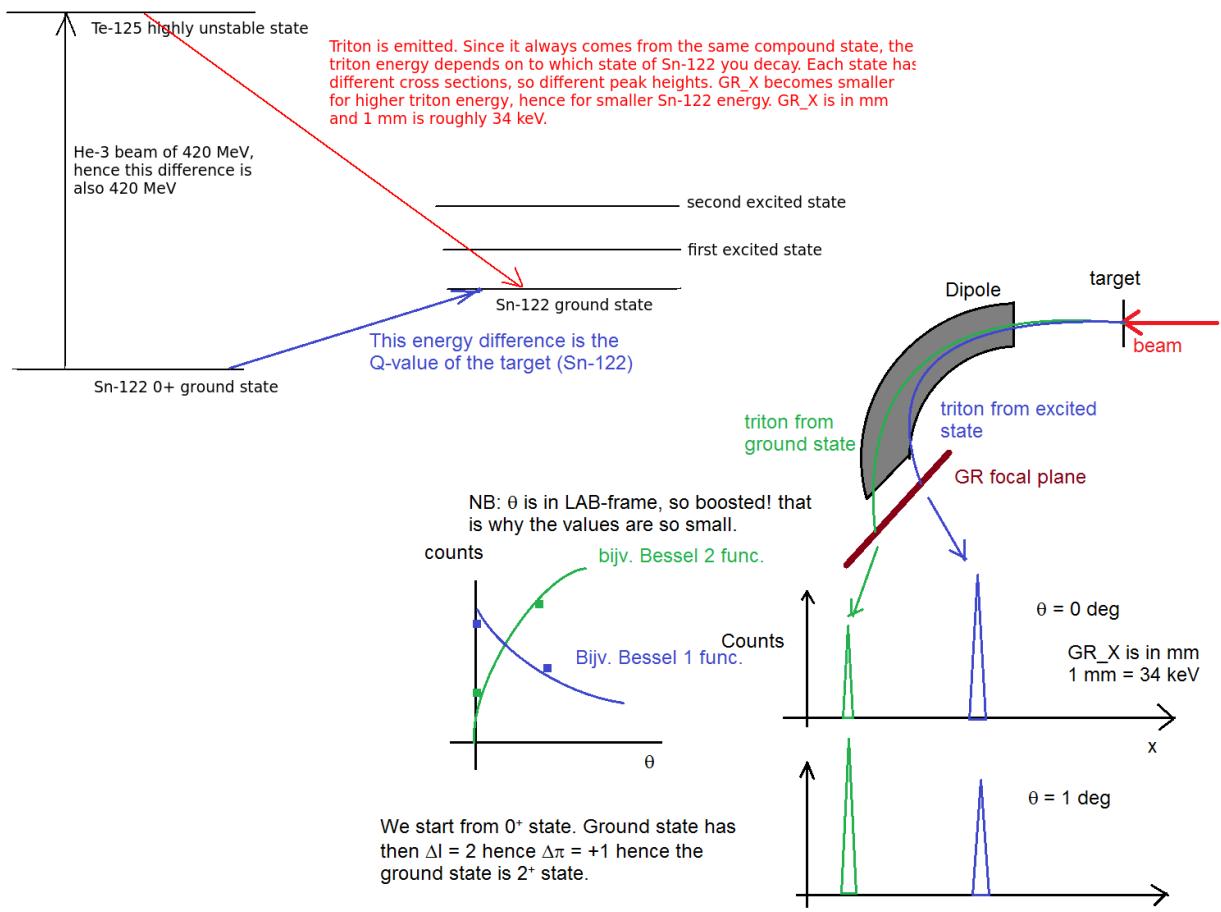


Figure 5.17: How to identify states.

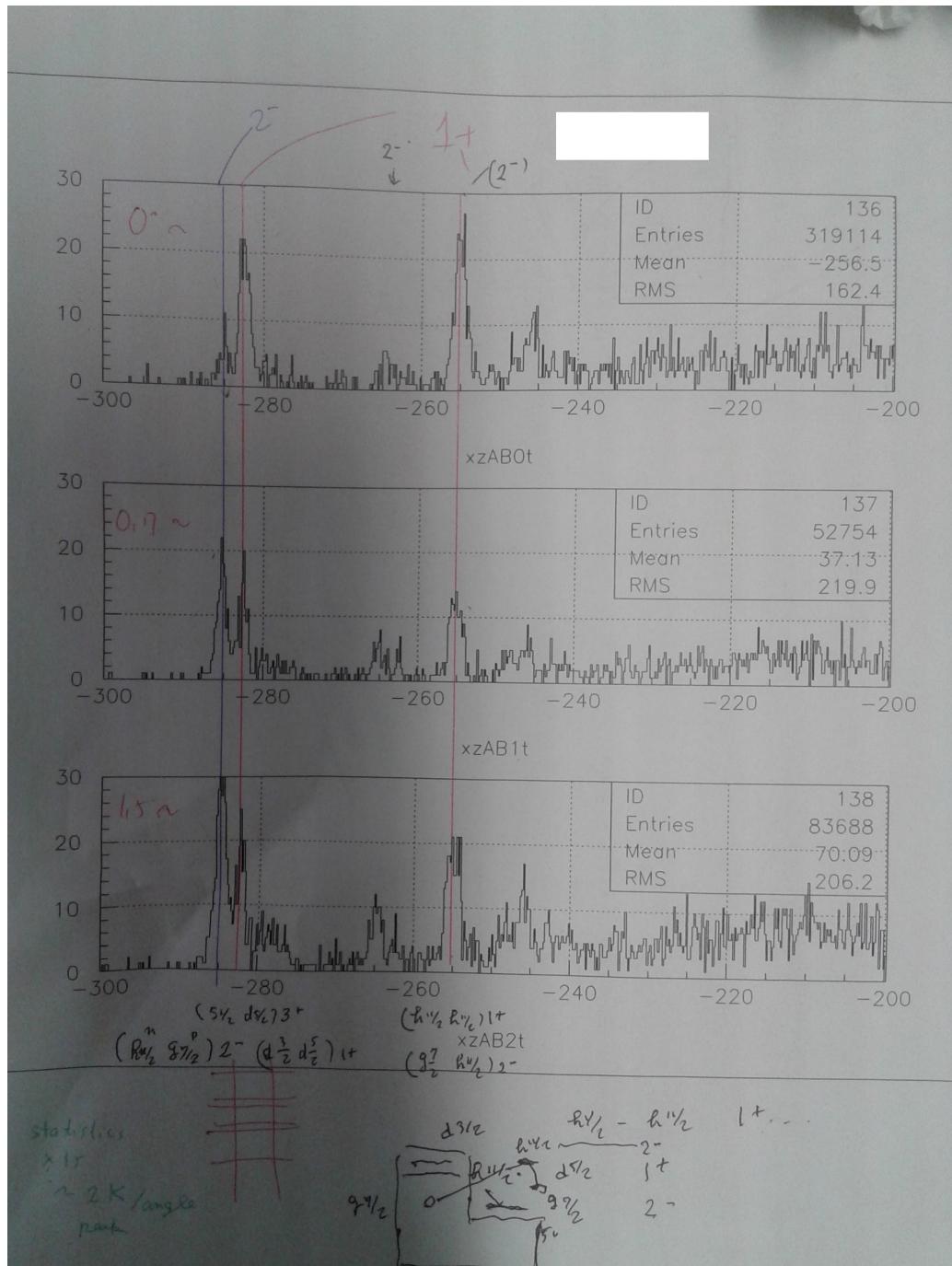


Figure 5.18: Example of how to identify states. Analysis performed by Akimune. Simulation data is run 5050.

x position at GR Focal Plane with cuts

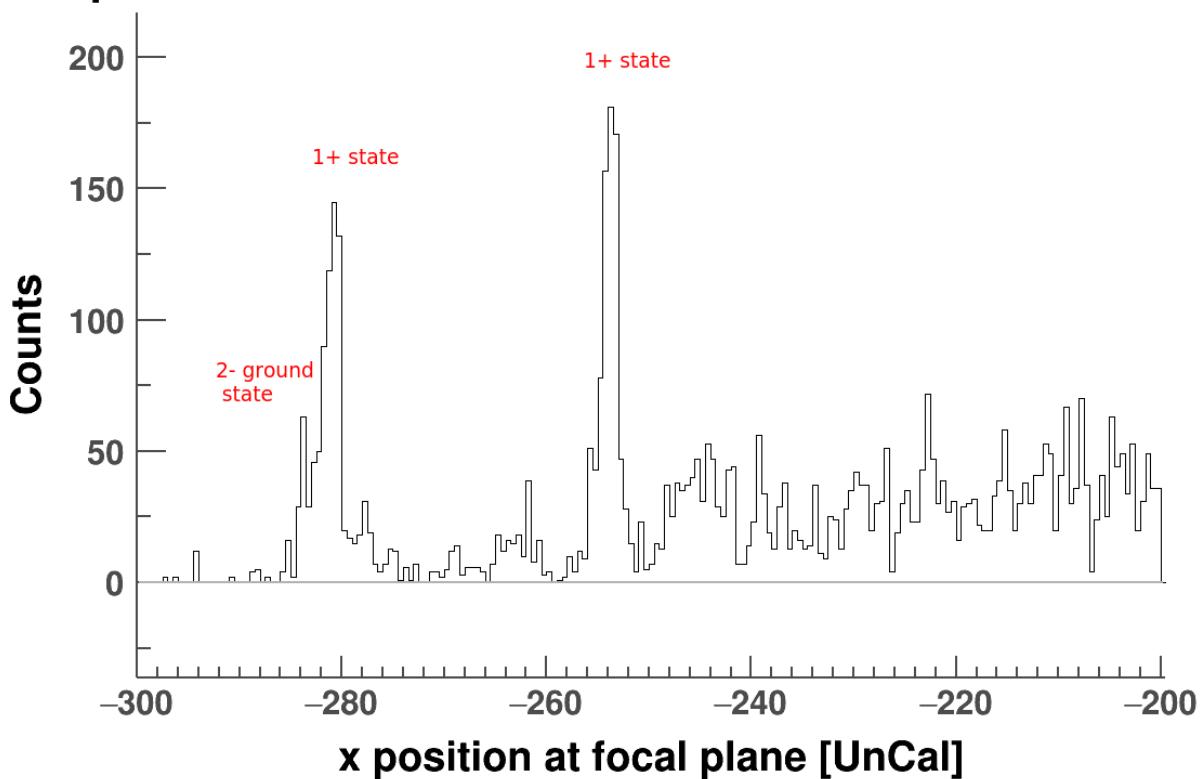


Figure 5.19: My own analysis code. $0 \leq \sqrt{\theta^2 + \phi^2} \leq 0.5$

x position at GR Focal Plane with cuts

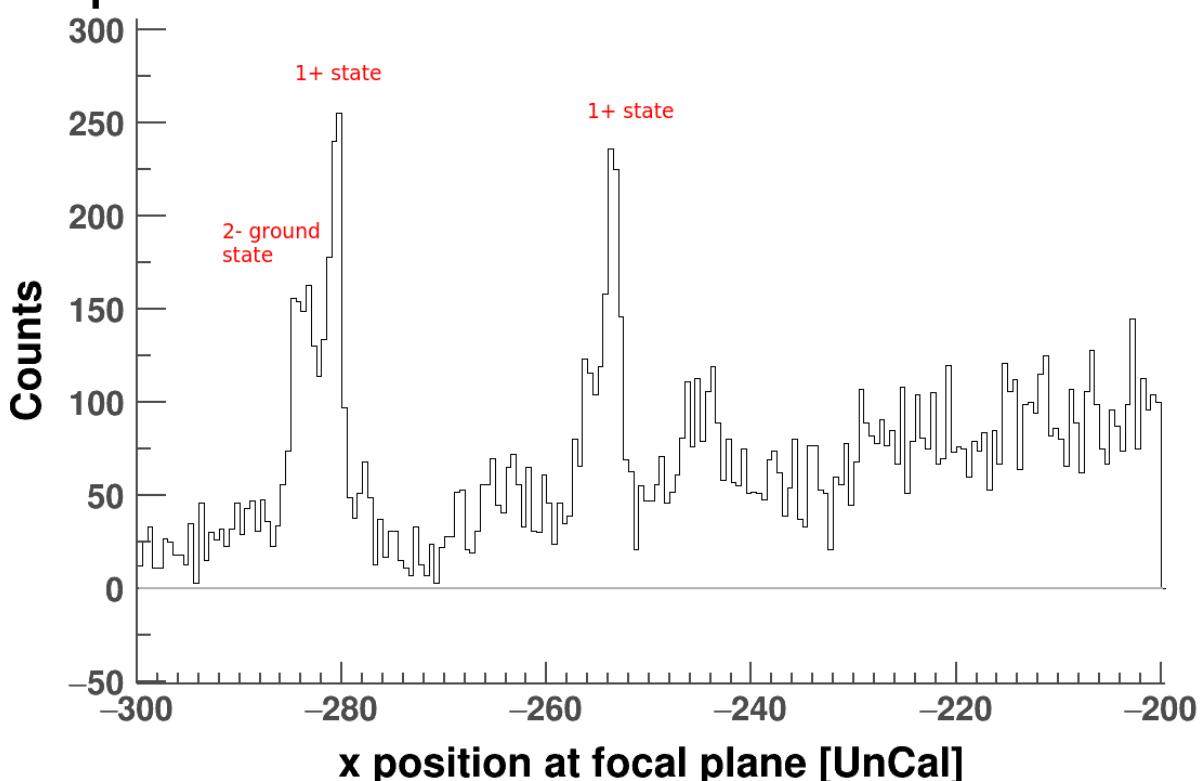


Figure 5.20: My own analysis code. $0.5 \leq \sqrt{\theta^2 + \phi^2} \leq 1.0$

x position at GR Focal Plane with cuts

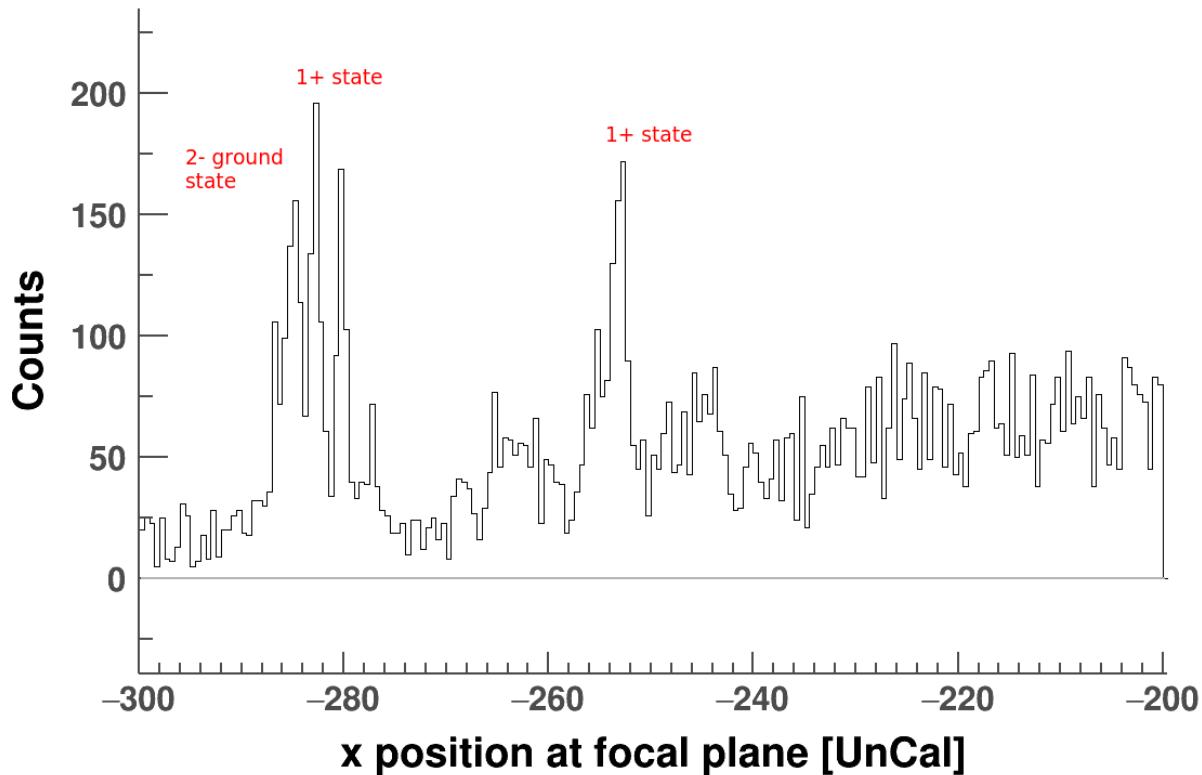


Figure 5.21: My own analysis code. $1.0 \leq \sqrt{\theta^2 + \phi^2} \leq 1.5$

step 2 Call the operator. Use the white wired phone on the desk and run number 08836 Then tell him **beamu o tomete kudasai** This means to stop the beam. **beamu o** means 'what to do with the beam', **tomete** means 'stop' and **kudasai** means 'please'.

step 3 Use the experimental controller for the scattering chamber to put in the SCFC: Scattering Chamber Faraday Cup. See figue 5.25.

step 4 Call the operator. Use the white wired phone on the desk and run number 08836 Then tell him **beamu o viewer no tsuyosa ni shite kudasai** This means to test the beam. You can see the beam profile on the Channel 1 camera on the computer screen. if everything is OK, ask the operator **shashin o tote kudasai** which means to make a picture of the beam profile. It will be send to your printer automatically. Put this picture in the log.

step 5 Call the operator again. Use the white wired phone on the desk and run number 08836 Then tell him **beamu o tomete kudasai** This means to stop the beam. When you move the target the beam MUST be stopped!!!

step 6 Use the experimental controller for the scattering chamber to deactivate the SCFC again (IMPORTANT!!!): Scattering Chamber Faraday Cup. See figue 5.25.

step 6 Select the proper new target. Use the experimental controller. See figue 5.26. Notice that it will take some time to move the target.

step 7 Call the operator. Use the white wired phone on the desk and run number 08836 Then tell him **beamu o dashite kudasai** This means to restart the beam.

step 8 Restart the DAQ. Go to the DAQ controller and first enter the appropriate comment (about which target you selected!). This is of VITAL importance!!! Then press **start**. See figue 5.24.

step 9 Use the online monitor to plot x and maybe x versus θ . Save a screenshot with printscreen directly onto

x position at GR Focal Plane

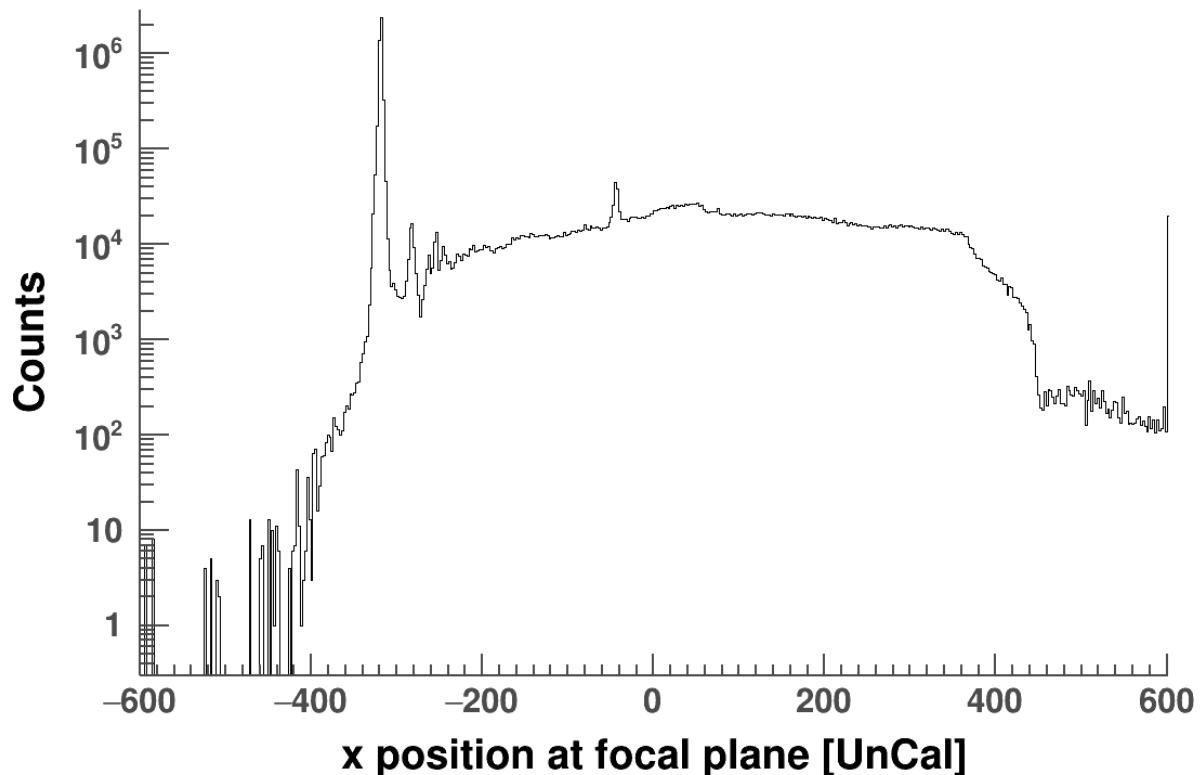


Figure 5.22: Full spectrum without cut. The large peak can be ignored. It is ${}^3\text{He}^{1+}$. This is basically your beam that did not produce an event, but just picked up an electron. This peak we use to focus the dispersive mode. Hence this is always your zero point. Hence you should analyze anything that is NOT inside this peak. Also notice that Tamii and Gey his analyzer have different GR_X definitions : $x_{\text{Gey}} = \sqrt{2} * x_{\text{Tamii}}$ since Gey uses VDC1 as focal plane, but Tamii uses a plane perpendicular to the beam as focal plane. Spectra above are all with x_{Tamii} .

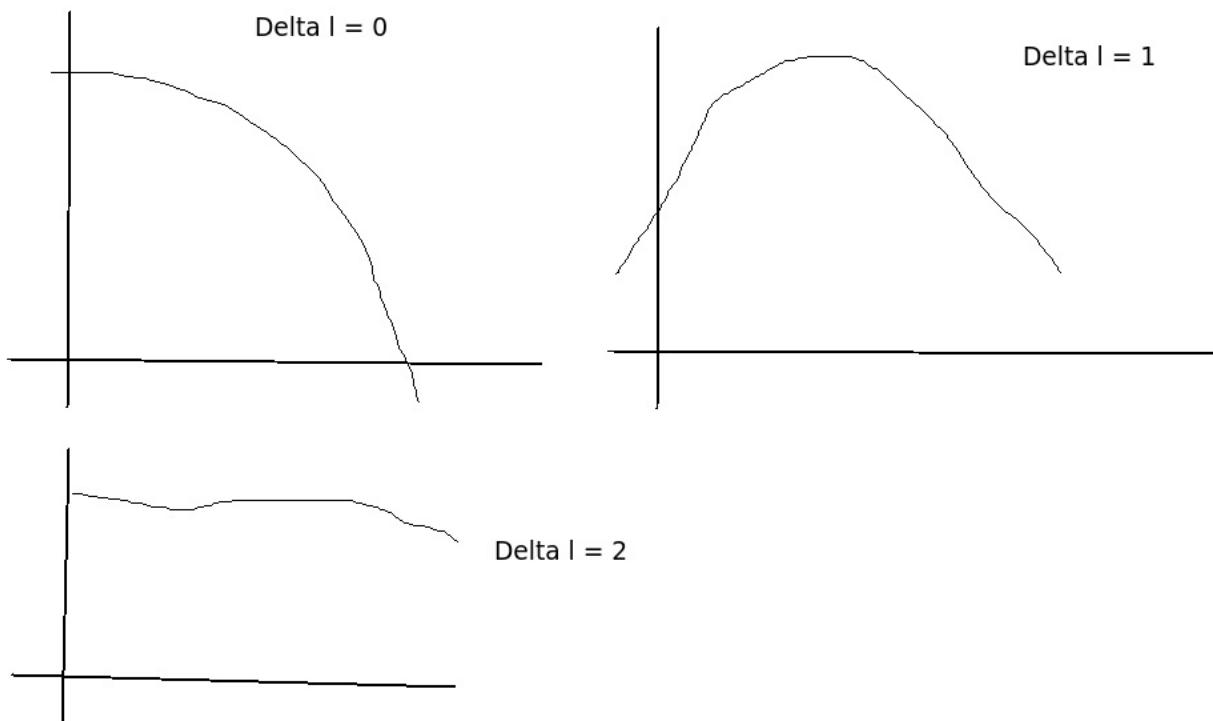


Figure 5.23: Distorted angular distributions.

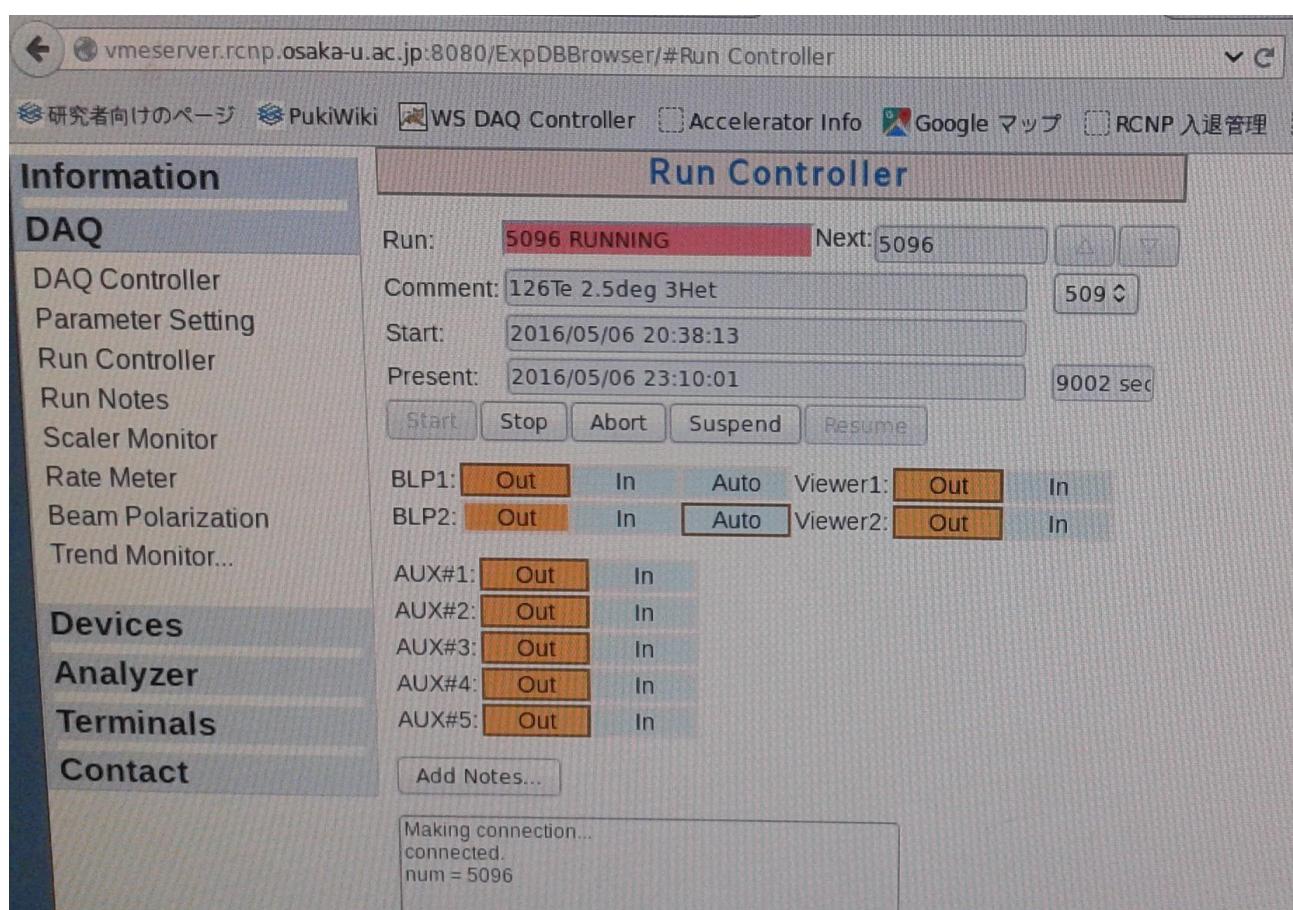


Figure 5.24: DAQ control screen.

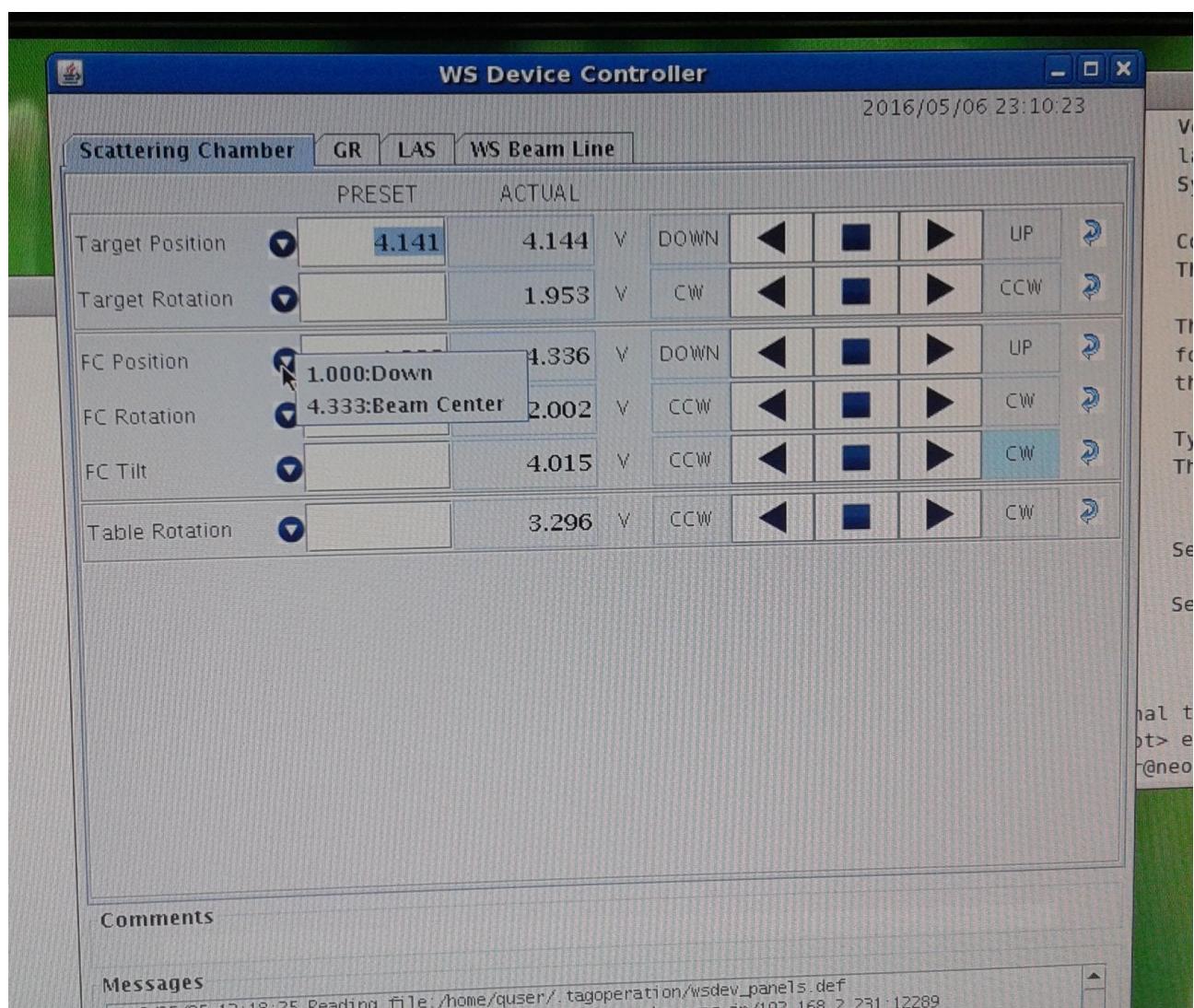


Figure 5.25: Experimental controller.

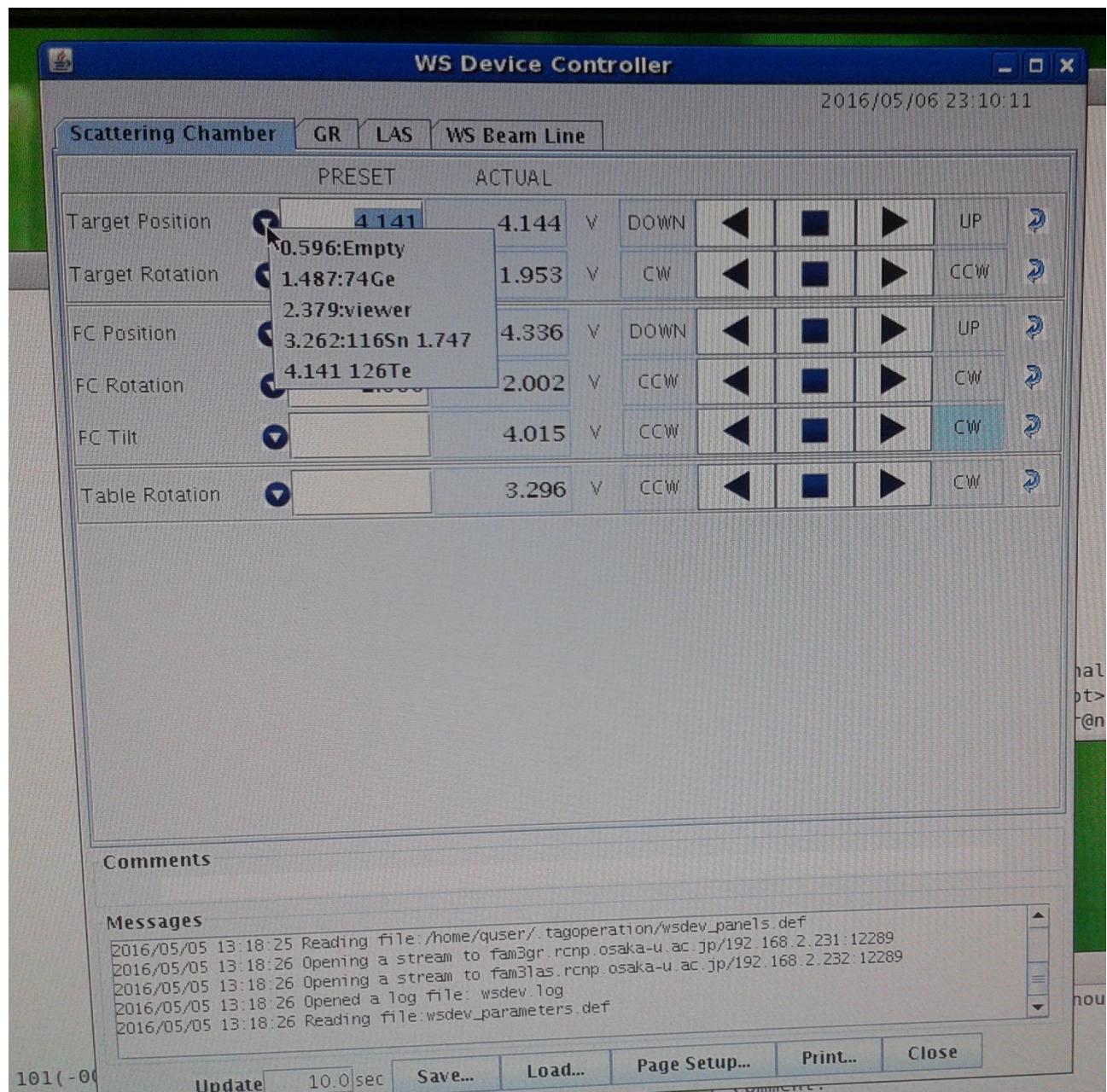


Figure 5.26: Experimental controller.

the Desktop and then print this picture and put it in the log. See figure 5.27 for how to use the online monitor.

The screenshot shows a terminal window titled '163 ab_t' displaying a list of steps. The steps are categorized into two main sections: 'PAW > zone 1 2; hi/pl' and 'PAW > hi/li'. The 'PAW > zone 1 2; hi/pl' section contains numerous entries of '107(-00.:80.)' followed by 'hi/pl 101(-00.:80.)'. The 'PAW > hi/li' section lists various parameters and variables:

Step Number	Description
82	DUMMY
83	GRADC12
84	RF1
85	RF2
86	X1
87	X3
88	X5
89	X7
90	Y1
91	Y3
92	Y3_t
93	Y5
94	Y7
95	X5Y5
96	X3A
97	a
98	b
99	ax
100	a0
101	a0x
102	a0x_t
103	a0_t

Figure 5.27: RCNP Paw Online Monitor

These steps are represented by figure 5.28.

5.2 Experimental setup of FAIR and NUSTAR

5.3 Own Analysis steps

stop, DAQ [...]
 stop, [08836]
 beamu o/ tomete kudasai;
 ~ ZnS
 ~ Scatterer chamber FC in
 (08836) beamu o viewer no tsuyosa ni shite kudasai
 CHECK CH01 | ~~1-1~~ ¹⁻¹ 08836 shashin o futekudasai
 stop : [08836] beamu o domete kudasai;
 ~ Scatter chamber FC out
 ~ move to target
 start : [08836] beamu o dashite kudasai;
 start DAQ
 note: change comment at DAQ
 → print a picture of GR-X

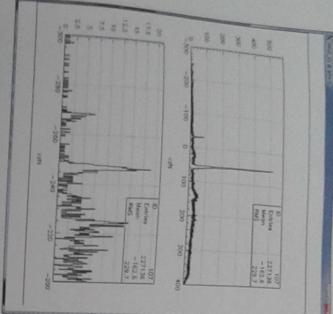


Figure 5.28: List of actions during shifts.

Chapter 6

Results

Chapter 7

Conclusion

Chapter 8

Operation of the Fold package

The theoretical computation of the cross sections of the (${}^3\text{He}, t$) charge-exchange reaction is summarized in figure 8.1.

Computation of cross sections in nuclear physics

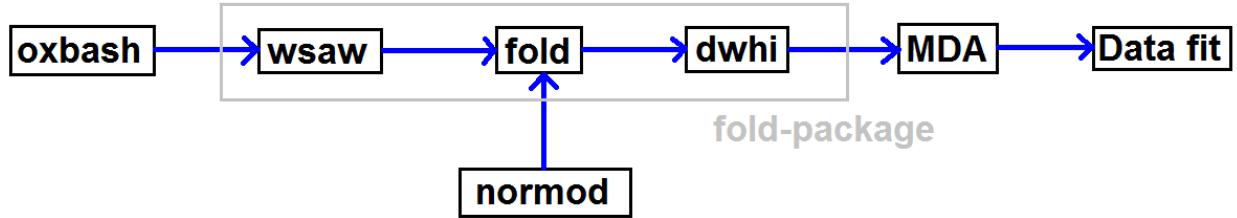


Figure 8.1: Schematic overview of how to compute cross sections in nuclear physics for data interpretation with the fold-package.

8.1 oxbash

The first things that have to be calculated are the binding energies of the single-particle states in the nuclear shell model. For this calculation we assume a pure shell model. We employ the code **oxbash** [16]. This package comes as one large folder called **oxbash**. To run this package, duplicate this folder on a windows computer under the location **C:\oxbash**.

Go to the directory **C:\oxbash\login** and open the file **Oxbash-command-prompt**. This opens a windows-terminal with a VMS command structure. Use the command **cd** to go to the directory **C:\oxbash\rsh**. Use **mkdir** to make a new sub-directory for the nucleus you want to compute and navigate to it. Once there, run the command **dens** to start the program.

Give the command **cp** to change the potential. give **sk20** as an input, to use a Skyrme-interaction between the nucleons. This skyrme-interaction is based on [17]. Then, give the command **az** to specify the Z and N of the target nucleus. For ${}^{116}\text{Sn}$, for example, the input would read like **116,50**

Then give the command **gd** to start the calculation. Let the program run for a few seconds and then stop the calculation with **st**. The calculation itself takes about half a second, but the program enters a stupid loop that should be interrupted like this. Then, to view the output, use the command **more dens.dao** and scroll to the very bottom. Then, scroll up until you find the single-particle states for protons and neutrons. Quantum numbers n , l and j are specified (n starts at 1, not 0!) along with the binding energies in MeV. Look for the neutron/proton bound state results.

These binding energies become the inputs for the **wsaw**-package. Run **oxbash** once for each nucleus you want

to know the binding energies for.

8.2 wsaw

`wsaw` is part of the fold-package **INSERT REFERENCE!!!** and is written in fortran. It should be used under Linux. It is located in the sub-folder `wsaw` of the fold-package folder. You can place it anywhere you like on your system. Then, go to this folder and compile the package. One possibility to do it is to use a Fortran makefile that calls `gfortran` as compiler. Next, you can run the program with the command `./wsaw <file.inp> file.out`. Where the `.inp`-file is the input file and the outer one is the output file.

Notice that this is written in Fortran!!! Hence, all numbers and inputs should be exactly in the right place in the file. The number of white spaces should be exactly right! This is because Fortran uses a predefined number of characters per number to read. In this case, I5 and F10: 5 characters per integer and 10 per floating point number. Hence, a working input file as example is your best method of supplying inputs. Also, be wary of the positions of minus-signs and whether a floating point is `.` or `.0`. Stick to the convention!

The purpose of `wsaw` is to compute the spatial distribution of the shell model wave functions. Hence, the input has to tell the program which wave functions to compute. The first line of the input file contains the parameters `RMESH`, `RMAX`, `NPUNCH`, `NPUNCH`, `IDBUG`

`RMESH` is the mesh size to compute the wave functions in femtometers. 0.1 is a good choice. `RMAX` is the maximum radius until where the wave functions are computed (also in femtometers). 20 is a good choice generally. `NPUNCH` is the number of mesh points that will be used. If $RMESH * NPUNCH < RMAX$, then the smaller value will be the true maximum radius that truncates our calculation. 150 does generally fine. This means we go to 15 fm. Since a good estimate for the nuclear radius is $1.25 \cdot A^{1/3}$, this will be more than enough. `IDBUG` is an output flag for debugging. Set it to zero.

The second line of the input should be the output filename. This output is different than the `.out`-file and should be used as input for `fold` (see next stage). Hence, there will be two output files produced by `wsaw`.

Then we have line 3 and 4 of the input file. These lines are repeated for each of the single particle wave functions that you want to calculate. Line 3 contains: `TMC`, `TIZC`, `V0`, `A`, `R0`, `RC`, `VSO` where `TMC` is the core mass, equal to $A - 1$. Since calculate wave functions of 1 nucleon trapped in the potential of the other $A - 1$ nucleons. `TIZC` is the core charge. Hence, if we calculate neutron wave functions, this equals Z . For proton wave functions, it equals $Z - 1$. `V0` is the depth of the Wood-Saxon potential in MeV (mean field approximation). It is assumed that the other $A - 1$ nuclei generate this potential together and then we compute the single-particle wavefunctions of the other nucleon in this field. This value will be fitted by `wsaw` so here we only have to give a good initial guess. 60 will do nicely. `A` is the diffuseness of the Wood-Saxon potential. A typical nuclear textbook value is 0.65. We will use this too. `R0` is the nuclear radius in femtometer and `RC` is the coulomb radius in femtometer. The actual radii will be these inputs times $A^{1/3}$ so we can supply a textbook value of 1.25 to them. `VSO` is the strength of the spin-orbit coupling. we use a textbook value of 7.0. For the textbook values, see `oxbash` **INSERT REFERENCE!!!**

The 4th line contains `EBIND`, `TMP`, `TL`, `TNODE`, `TIZP`, `XJ`, `XS` `XS` is the nucleon spin, hence 0.5 always. `TL`, `XJ`, `TNODE` are the quantum numbers of the specific nucleon wave function that you want to use. `TNODE` is n , starting from 0. `TL` is $s = 0$, $p = 1$, etc. and `XJ` is the resulting j from spin-orbit coupling, hence $l - 1/2$ or $l + 1/2$. `TIZP` is the nucleon charge: 0 for neutrons and 1 for protons. `TMP` is the nucleon mass, hence it is always 1. Finally, `EBIND` is the binding energy obtained with `oxbash` in the previous section.

However, there are a few remarks with regard to the binding energies. At first, `oxbash` return negative numbers, while we need positive numbers here. Secondly, if a state has a binding energy below 2, it is probably unbound. `oxbash` cannot compute that and `wsaw` always needs to have a bound state, since otherwise the wavefunction is not localized near the origin, meaning we cannot compute it. Hence, we must make these states bound artificially by setting their binding energies equal to 2. For charge-exchange reactions, this manipulation will have little effect on the cross-section `oxbash` **Do not use a reference, but do the calculation also with 1.0 and 3.0, etc. This is your justification. Just try out a few values.**

Finally, one must notice that the output of `wsaw` is input for `fold` which means that the output should contain all information necessary for the reaction. hence, we will not get 1 input-file for `wsaw` per nucleus, but one per reaction. Hence, for, say, a $^{122}\text{Sn}(^3\text{He}, t)^{122}\text{Sb}$ charge-exchange reaction, we remove a neutron and we put back

a proton. Hence, we will need all neutron wavefunctions of ^{122}Sn and all proton wavefunctions of ^{122}Sb in the same file together! We could also put in the proton wavefunctions of ^{122}Sn and the neutron wavefunctions of ^{122}Sb but they will not be needed in our calculation. Important in that the wavefunctions of both the target and the recoil are in the same input file for `wsaw`. Technically, one would also need the wavefunctions of the projectile and ejectile, but the fold-package has ab-initio calculations for ^3He and ^3H build-in into the package, so we can omit them from the input-file IF we are interested in a $(^3\text{He}, t)$ charge-exchange reaction. **One very important remark:** never enter more than 50 wavefunctions in the `wsaw` input. If you do, `fold` can no longer process the output. Hence, one must make a sensible choice on which wavefunctions are necessary. for $(^3\text{He}, t)$, only non-fully occupied proton shells and non-empty neutron shells are needed.

The list of lines 3 and 4 should be terminated by a `-1`.-character below the last wave-function. This introduces an additional problem. It means we have to truncate our list of wave functions somewhere, since we cannot enter or calculate infinitely many orbitals. We do need unbound states too in our calculation because we produce nuclei in excited states. We must include those resonances in our models. Hence, the bound/unbound boundary is NOT the right place to truncate our list of wavefunctions. We must make sure that we include enough of them to include all relevant physics, but not so much that the CPU-time becomes too large. This is a choice the user must make. We will just include all orbitals that are in the oxbash output. This will be enough for our purposes.

8.3 normod

Up to now we have just been calculating single-particle nuclear wavefunctions. We have not concerned ourselves with the question which of these wavefunctions participate in a certain transition and which remain inert. A so-called normal modes calculation will tell us which wavefunctions participate in which transitions. Hence, we must now specify the nuclear reaction we are interested in. For us, this is a $(^3\text{He}, t)$ charge-exchange reaction. This means that a neutron is taken out and a proton is put back in. The proton can only be put in in a shell that is NOT filled completely. Hence, from all nuclear wave functions of the recoil nucleus that we have calculated, we should only give the proton shells as an input that can accept an additional proton. The so-called hole-shells. Similarly, for the target nucleus, we should only give the neutron shells that are actually occupied by neutrons. The so-called particle-shells.

The principles of `normod` are discussed in [18]. The code itself is dicussed in [19]. However, this reference is untrackable nowadays, so refer to [20] and [21] for the procedure, since they also discuss the principles and have [19] in their reference list.

The input-structure of the package `normod` that performs the normal modes calculation is as follows (Fortan again, like `wsaw`). The first line is an output-flag that should be put to 1. The second line contains a single number: the number of particle-shells. Then, the third line is repeated for each of the particle-shells. It contains: FNP FLP F2JP U2 NPTYPE NPFLAG ISHELP FNP is the quantum number n , starting from zero. FNP is the quantum number l of this shell. F2JP is the quantum number $2 * j$. NPTYPE is 0 for a neutron shell and 1 for a proton shell. NPFLAG equals $2 * n + l$. ISHELP is a key-number identifying this specific shell. It can be anything, as long as you do not use the same key-number more then once in the entire input file. Finally, U2 is the fullness of the particle-shell. In a perfect shell-model, this is always 1.0 All particle-shells are occupied by definition. Except for the last shell, which does not necessarily contain all particles that it possibly can contain. In that case the fullness is the number of particles divided by the max. number of particles. However, the nuclear shell-model is not perfect. In reality, the nucleus also contains correlations, which fragment the wavefunctions over different states. In these case, the fullness decreases and becomes a number between 0 and 1. For Sn122 and Sn116, you can obtain some of these numbers from [22]. For Sn122, we had to extrapolate the fullness of the $g_{7/2}$ and the $h_{11/2}$ from the table.

Then, the next line contains the total number of hole-shells that you will specify. Then, another set of lines comes with a structure of FNH FLH F2JH V2 NHTYPE NHFLAG ISHELP. This structure is completely analogous to the particle-shells, except that the input V2 is now the emptiness of the hole-shell, not fullness. Emptiness is 1.0 for an unoccupied shell. Again, a perfect shell-model would put 1.0 everywhere here (except for the first one, which may be partially occupied, in which case the emptiness is the number of holes the shell dibided by the maximum number of particles in the shell). However, [22] does only have emptiness of the neutron shells, not of the proton shells. So we are forced to use a pure shell-model there. Finally, the input-file is terminated by one last line of inputs.

The last line of inputs is the most important one: M JTRAN PARITY AMASS ISPIN REACTION IDIFL IDIFH. M is the λ , the power of r in the transition operator for which you do the normal modes calculation. For a Gamow-Teller or Fermi-transition, this is 0. JTRAN is the total number of angular momentum transferred in the transition (in units of quantum numbers). This is 0 for Fermi and 1 for Gamow-Teller. PARITY is the parity of the transition, being 1 or -1 . For both Fermi and Gamow-Teller, this is 1. AMASS is the mass number of the target nucleus. ISPIN is the isospin transfer of the transition. For our charge-exchange reaction, this is obviously 1. REACTION is the isospin projection change. For our charge-exchange reaction, this is -1 . IDIFL and IDIFH are the minimum and maximum differences between principal shell number $2 * n + l$ that are allowed. For Fermi, the minimum and maximum are zero.

One can suffice with a single input-file per nuclear reaction. However, one needs to add a last line for each of the transition types that one whishes to study.

The normal modes calculation will automatically identify which of the particles-shells you specified can perform a cross-over to one of the hole-shells you specified. Such a cross-over is called a 1p1h-transition (1-particle-1-hole transition). For example, if you specified a Gamow-Teller transition, $\Delta L = 0$, $\Delta S = 1$, Hence $\Delta J = \Delta L + \Delta S = 0, 1$ (notice the vectorial adding!) Further, $\Delta n = 0$ and $\Delta T = 1$. So a neutron from $1d_{3/2}$ can cross to a proton $1d_{5/2}$ by Gamow-Teller, but that same neutron cannot go to a $2s_{1/2}$ by Gamow-Teller. The normal modes calculation will identify ALL allowed combinations for you (within the set of particle and hole functions you entered) and respond with a set of OBTD's (One Body Transition Densities): one number for each transition. Sometimes, a transition is not in the output and sometimes, it is, with $OBTD = 0$. In both cases, that 1p1h-transition does not contribute. Only non-zero OBTD's will contribute.

The idea of a normal modes calculation is that you assume that all strength of the transition type you specified is assumed to be in a single state in the excitation-energy spectrum. For experimental analysis, this is fine, because we will fit the normalizations to the data anyway. Then we can use the same OBTD's for each of the states. For testing the nuclear shell model, the normal modes calculation is obviously insufficient, but for extracting observables from the data, it is.

For Fermi and Gamow-Teller transitions, `normod` calculates $B(F)/4\pi$ and $B(GT)/4\pi$ automatically. According to the sum-rules, they should be equal to $B(F) = |N - Z|$ and $B(GT) = 3|N - Z|$. Because of the definition of the normal-modes calculation, the output should always (more or less) match the sum-rules. However, for Gamow-Teller decay the sum rule literally states $S_+ - S_- = 3|N - Z|$ Where S_{\pm} are the polarized decay rates of beta-decay. With your normal modes calculation, you compute $(^3\text{He}, t)$, which mimics only S_+ . Hence, the `normod` output will be somewhat larger than the sum rule for Gamow-Teller decay.

8.4 fold

Now the next step is to calculate the form-factor of the nuclear reaction. This form-factor is given by equation (3.24) in [20]. The idea is that you compute:

$$F(\vec{r}) = \sum_{i,j,k,l} \langle \Psi_i \Psi_j | V | \Psi_k \Psi_l \rangle$$

Ψ_k are the single-nucleon wavefunctions of the target. Ψ_i of the recoil. Hence the sum over i and k is from 1 to A . Ψ_l are the single-nucleon wavefunctions of the projectile (^3He for us) and Ψ_j of the ejectile. (^3H for us). Hence, for us the sum over j and l runs from 1 to 3. V is the effective nucleon-nucleon interaction potential. `fold` uses the Love-Franey interaction potential from [23, 24]. The second paper is an update to the first one and we use the updated version. This determines the effective nucleon-nucleon potential from single-nucleon scattering data.

The form-factor integrates out all spatial coordinates (and other coordinates), except the distance between the target and projectile (or recoil and ejectile after the collision). This is the \vec{r} -dependence of the form-factor. This procedure is called double-folding, because you perform it both over the projectile/ejectile and over the target/recoil.

With this form-factor, one can correct the DWBA-calculation for the internal structure effects of the nuclei and regard the nuclei as pure particles further on. The Love and Franey potential contains only a radial dependence, so the inner product splits out in a radial part and a spherical part. The spherical part only contains creation and annihilation operators because the potential has no spherical dependence.

The **fold**-package needs the computed wavefunctions in **wsaw** as an input-file and it needs an input-file **.inp** of its own. In this file, one must manually enter the OBTD's from the niormal modes calculation.

The input-file for **fold** has the following structure. Again, we use F10 and I5 Fortan input. The first line is **KEXCHG**, **KPUNCH**, **FILNAM** If **KEXCHG= 0** we will only use a direct interaction potential and if we use **KEXCHG= 1** we will use the zero-range exchange approximation. See the [23, 24] Love and Franey papers for more details on this. We basically assume that the exchange-etrm in the potential is a delta-function around the average momentum transfer, because anything else is too hard to compute. The program has also an option **KEXCHG= 2** that should not be used. You should use **KEXCHG= 1**. **KPUNCH** should equal 1 and is a flag that determines how much output is generated. **FILNAM** is the name of the output of **fold**.

The second line contains **NR**, **H**, **ELAB**, **APROJ**, **IPRTR**, **IPRTQ**, **IPRTF** where **NR** is the number of mesh points to compute the form factor. 600 is a good choice. **H** is the mesh size in femtometers. 0.03 is a good choice. **ELAB** is the total beam energy in MeV. In your case, this is 420. **APROJ** is the projectile mass number, which is 3 in your case. The other three are output flags, that should all be 1.

Next, we move to the target-projectile system. The first line in this system is **FJF**, **PARF**, **FJI**, **PARI**. **FJF** is the spin of the ejectile. For a triton this is 0.5. **FJI** is the spin of the projectile. For ${}^3\text{He}$, this is again 0.5. **PARF** is the parity of the ejectile and **PARI** is the parity of the projectile. For $({}^3\text{He}, t)$, both are +. Hence, the input reads 0.5 + 0.5+. This is because we assume that the projectile and the ejectile internal structure do not matter too much. We assume that everything here stays in the $0s_{1/2}$ -shell and that the projectile-ejectile always goes from 0.5+-state to a 0.5+-state. **INSERT REFERENCE!!!**

The next line is the isospin line. **TF**, **TFM**, **TI**, **TIM**. **TF** is the final isospin length and **TFM** is the final isospin projection. **TI** and **TIM** are the same, but now for the projectile (initial state). $TM = N - Z$ and T_{length} is assumed to be equal to this in magntitude. Hence, one can compute all isospins for all nuclei. For the $({}^3\text{He}, t)$, this line looks like 0.5 0.5 0.5 – 0.5. However: technically, **TF** can become **TI ± 1** for $({}^3\text{He}, t)$. However, the states associated with anything else then $TF = TI - 1$ are usually to weak and/or at too high excitation energy to measure. Except for the IAS, which is defined as $TF = TI$.

The third line equals **NTYPF**, **KOPTN**, **ALPHA**. **NTYPF** can be 1 for static computations, 2 for inelastic reactions and 3 for charge-exchange reactions. **KOPTN** is a flag to define the input formalism. Use 3 for the isospin formalism. **ALPHA** is a flag that should be set to zero.

The fourth line defines the 1-particle-1-hole transitions that contribute to the form factor. It reads **IDI**, **IDF**, **JX**, **Z1**, **Z2**. **IDF** is the final (hole) shell and **IDI** the initial (particle) shell of the trasnition. They are characterized by integers, see figure 8.2. **JX** is the total change in angular momentum. 0 for Fermi, 1 For Gamow-Teller (for both the projectile-ejectile transitions and for the target-recoil transitions). **Z1** is a flag that should be zero. **Z2** are the OBTD's obtained from the **normod**-calculation. This line should be repeated for all transitions that are output of the **normod**-calculation. For $({}^3\text{He}, t)$, this line will always read 1 1 0 0 $1/\sqrt{2}$ for Fermi and 1 1 1 0 $1/\sqrt{2}$ for Gamow-Teller. **one important remark:** The initial (particle) shell comes first and the final (hole) shell comes last. Even if other documentation states otherwise.

Finally, the fifth line is the filename fo the single-particle wavefunctions. For $({}^3\text{He}, t)$, this file is given by the package and obtained from ab-initio calculations. For the target and recoil, this is the **wsaw** output file that you generated. Hence, this file should be located inside the directory of **fold** in order to run the program.

Now we repeat everything that we did for the projectile-ejectile system once more but now for the target-recoil system. Notice that this can vbe quite lengthy and annoying!

The next line defines the interaction potential **V** for the form-factor. It reads **FNRM1**, **FNRM2**, **FNRM3**, **FRCEFILE**. The first 3 inputs are normalization parameters. The fourth should be **love_140** to use the Love and Franey potential. Based on the equations of the Love and Franey papers [23, 24], we can calculate them to be for Sn116 and Sn122 (both the same): 1 = 0.922, 2 = 1.97 and 3 = 0.0.

The next line is the number of form-factors you want to calculate. Just 1 number on the line. In case this is more then 1, all input below this number has to be repeated for each form-factor. Best is to just compute 1 form factor, since different transitions cannot be dealth with at the same time anyway. We return to this issue in the discussion of the MDA (Multipole decomposition anaylsis).

The next line is **JR**, **JP**, **JT**, **KFORCE**. **JP** is the total angular momentum change in the projectile-ejectile and **JT** is the same for the target-recoil. Both are 0 for Fermi and 1 for Gamow-Teller. **KFORCE** determines which

1 = 0s1/2	16 = 2p1/2	31 = 2f5/2
2 = 0p1/2	17 = 2p3/2	32 = 2f7/2
3 = 0p3/2	18 = 1f5/2	33 = 1h9/2
4 = 1s1/2	19 = 1f7/2	34 = 1h11/2
5 = 0d3/2	20 = 0h9/2	35 = 0j13/2
6 = 0d5/2	21 = 0h11/2	36 = 0j15/2
7 = 1p1/2	22 = 3s1/2	37 = 4s1/2
8 = 1p3/2	23 = 2d3/2	38 = 3d3/2
9 = 0f5/2	24 = 2d5/2	39 = 3d5/2
10 = 0f7/2	25 = 1g7/2	40 = 2g7/2
11 = 2s1/2	26 = 1g9/2	41 = 2g9/2
12 = 1d3/2	27 = 0i11/2	42 = 1i11/2
13 = 1d5/2	28 = 0i13/2	43 = 1i13/2
14 = 0g7/2	29 = 3p1/2	44 = 0k15/2
15 = 0g9/2	30 = 3p3/2	45 = 0k17/2

Figure 8.2: fold-package shell model states ID numbers

forces are taken along. -1 means both central and tensor and 0 means only central. Finally, JR is the relative angular momentum transfer. This is a direct sum of JT and JP . Hence, it is zero for Fermi and can be 0 or 2 for Gamow-Teller (plus minus contributions). The actual Gamow-Teller has $JR = 0$, but because this is not pure beta-decay, but charge-exchange, we face a little competition from $JR = 2$. This is the reason why one can compute 2 form-factors within the same input-file. It si also the very reason why one needs to do MDA, which we will discuss in more detail in another section.

Finally, one needs two lines of 7 time 1. These are nuclear deformation parameters, but since our nuclei are not deformed, they should all be equal to unity.

And that finishes our discussion of `fold`. Run the program by first putting the `wsaw`-output in the `fold`-directory and put this input file there too. Then run `./fold <file.inp> file.out` just like with `wsaw` (compiling works similar too). Gamow-Teller and Fermi sum-rules can be checked in the `.out`-files. However, notice that they should mathe the output of `normod` exactly, up to the 4π -factor. But the Gamow-Teller matrix element will be a little larger then the sum rule (see section 8.3).

Notice that each type of transition needs a separate `fold`-calculation. Hence, we must prepare 1 input-file for a Fermi transition, 1 for a Gamow-Teller transition, 1 for a Spin-dipole transition, etc. But we can use the same input and output for, say, two Gamow-Teller transitions ate different excitation energy.

8.5 dwhi

The package `dwhi` performs the actual DWBA-calculation and gives a `.plot`-file as an output that contains the angular distribution of the differential cross section. The computation is conceptually very easy. See redPhD thesis of Remco Zegers chapter 3 for the details. We compute:

$$\frac{d\sigma}{d\Omega} = \left(\frac{\mu}{2\pi i \hbar^2} \right)^2 \cdot \frac{k_f}{k_i} \cdot |T_{fi}|^2$$

Here, μ is the reduced mass of the projectile and the target. $k_i = \|\vec{p}_{beam}\|/\hbar$ and $k_i = \|\vec{p}_{ejectile}\|/\hbar$ and T_{fi} is the transition amplitude. This amplitude is defined as

$$T_{fi} \langle f | F(\vec{r}) | i \rangle$$

Here, $F(\vec{r})$ is the form factor computed by `fold` and this is where the nuclear structure enters the DWBA. The $\langle \cdot \rangle$ -functions are distorted waves, depending only on the same \vec{r} as the form factor, which is intergrated out in the inner product, leaving only a pure matrix element.

Asymptotically, the $\langle \cdot \rangle$ -functions should be plane waves. However, they are distorded by the nuclear potential. the package `dwhi` takes care of computing this distortion, but it needs a good optical potential to do this. Normally, the only method of obtaining this potential is by doing a scattering experiment. The real part of the optical potential corresponds to elastic scattering and the imaginary part to inelastic scattering. For ^{122}Sn and ^{116}Sn , this scattering data is not measured, so we will use the scattering data of other nuclei to interpolate the optical potential. The measured data used for this optical potential comes from [?, 25], but the analysis of this data and the interpolation model of the potential was done conform [26].

The input-file for `dwhi` should have the following structure. It is again Fortran, but now with F7 and I3. the first line should always be 1210000041000000 which are just flags that do not have anything to do with physics. Only the 10-th character should be 1 to use relativistic kinematics. 0 uses Newtonian kinematics.

the next line is the filename of the `fold`-output that contains the form factor. The third line contains 3 parameters: Number of mesh points for the scattering angle, start of the scattering angle and stop of the scattering angle. Hence, they define the range and resolution at which the angular cross section has to be computed.

Then, the line after that has the following structure: L, NFF, ISA, ISB, JA, JB where L is the number of partial waves used for the DWBA. This is like a Taylor-expansion and enough terms should be taken along. 160 does generally fine. NFF is the number of form factors you computed in `fold`. This corresponds to the number of lines JR, JP, JT, KFORCE in the `fold`-input file you entered. ISA and ISB are the target initial and final spin. For $(^3\text{He}, t)$, these numbers should both be 1, since everything is assumed to happen only in the $0s_{1/2}$ -shell, so all transitions for the projectile-ejectile system are $0.5^+ \rightarrow 0.5^+$ transitions, so both spins are 0.5, so we enter twice this number. Hence, both inputs are 1. JA and JB are the smae but now for the target-recoil system.

The next line contains the mesh size and number of mesh points used to compute the form factor. They should correspond to the numbers on the line NR, H, ELAB, APROJ, IPRTR, IPRTQ, IPRTF in the `fold`-input file.

Then we move to the incoming channel. The first line reads E, FM, Z, FMA, ZA, RY, FS, QCD Here, E is the total beam energy in MeV. FM is the projectile mass number and Z is the projectile charge number. FMA is the target mass number A and ZA is the target charge number Z. RY is the nucear radius. This should be set to 1.25 fm since the true radius used in the calculation is this number multiplied by $A^{1/3}$. FS is twice the projectile spin, which is 1 for $(^3\text{He}, t)$. QCD is a Q-value that should be put to zero. It is NOT the reaction Q-value.

The second line specifies the optical potential of the incoming channel. References are discussed above. Its structure reads FZ, VR, RY, AR, VSOR, VI, RZ, AI, VSOI, PWR Here, FZ = 1 specifies that the real and imaginary part of the optical potential are Wood-Saxon, which corresponds to [26] VR, RY, AR, VSOR specify the real Wood-Saxons depth, radius, diffusiveness and spin-orbit coupling (this is zero, in accordance with [26]). VI, RZ, AI, VSOI specify the parameters of the imaginary Wood-Saxon. Finally, PMR is a flag that should be put to zero for physical calculations. Then, we terminate the incoming channel by a line with a single 0.

Next is the outgoing channel. These two lines have exactly the same structure as the incoming channel. Howvere, E is now not the beam energy, but the Q-value of the ground state plus the excitation energy of the specific state you want to compute the angular cross section for. **note that the totakl should be entered as a negative number!** Hence, a new calculation with `dwhi` is necessary for each individual state in the excitation energy spectrum. the optical potential of the outgong channel should be identical to the incoming channel, excpet that both depths should be multiplied by 0.85 [27].

The next line has a structure of LTRT, ISTRT, JTTR, which should be identical to the first three parameters in the line JR, JP, JT, KFORCE of `fold`. Howvere, ISTRT and JTTR should be twice the j while JP and JT should be j itslef.

next comes a line with nuclear deformation parameters. Since our nuclei are not deformed, this line should read 0 0 0 1. Finally, the inpt-file is terminated by the filename of the .plot-file that contains the output of the

DWBA-calculation.

8.6 MDA

This is an important analysis technique. It involves disentangling the various ΔL -states from the experimental data. This is necessary, because for (${}^3\text{He}, t$), the angular distribution will be polluted by a $\Delta L = 2$ -component.

Gamow-Teller is defined through beta-decay and has the characteristics $\Delta L = 0$, $\Delta S = 1$, $\Delta T = 1$. Then, $\vec{\Delta J} = \vec{\Delta L} + \vec{\Delta S}$ so $\Delta J = 1$. However, if you do a nuclear reaction like (${}^3\text{He}, t$), one technically cannot speak about a $L = 0$ transition due to the spin-orbit couplings in the reaction. One can only identify transitions like $0^+ \rightarrow 1^+$ (Gamow-Teller). Such transitions can be accomplished by $\Delta J = 1$ and a positive parity change. This means that due to the vectorial adding of L and S into J , we can have both a $\Delta L = 0, \Delta S = 1$ contribution and a $\Delta L = 2, \Delta S = 1$ contribution, which is a certain type of a quadrupole transition.

In order to disentangle them, we make a DWBA-computation of the angular cross section for both transitions, and then we fit two normalization parameters α and β to the data like:

$$\text{data} = \alpha \cdot \{\Delta L = 0, \Delta S = 1\} + \beta \cdot \{\Delta L = 2, \Delta S = 1\}$$

and then we determine the Gamow-Teller strength from only the $L = 0$ -component. This is called multipole decomposition analysis (MDA).

If one can resolve the distinct nuclear states one-by-one, the above is enough for Gamow-Teller transitions. For Fermi, it is even easier, since $\Delta L = 0$, $\Delta S = 0$, $\Delta T = 1$ can only and only correspond to $0^+ \rightarrow 0^+$. Hence, for Fermi, the MDA only has a single component and a single normalization parameter to fit, namely the $L = 0$ -component.

For Gamow-Teller transitions, the MDA will cause one important problem. The nuclear tensor force causes interference between the $L = 0$ -component and the $L = 2$ -component. Hence, what we identify as the $L = 0$ -component, is actually the 'true' $L = 0$ -component plus an interference contribution. This contribution cannot be resolved from experimental data and introduces a so-called systematic error to $B(GT)$ -values. Two studies have been done to this effect in [28] for ${}^{26}\text{Mg}$ and [29] for ${}^{64}\text{Zn}$. The systematic error can be quantified as $\Delta B_{sys} = 0.03 - 0.033 \ln(B(GT))$ and this estimate is nearly the same for both nuclei, hence we can take the estimate across different A to be valid. So we can apply it. We cannot know the true $B(GT)$ -value without the interference from $\Delta L = 2$, but with this we can get a handle on how precise we know $B(GT)$.

However, if the nuclear states can not be resolved individually, one can also perform a full MDA-analysis. For this analysis, one will cut the excitation-energy spectrum into different bins of arbitrary size and compute an angular distribution of the cross-section for each bin. Since each bin can contain multiple states at once, the angular distribution will NOT have a nice shape. Then, we will fit to the data

$$\text{data} = \sum_{l=0}^{L_{max}} \alpha_l \cdot \text{DWBA}_{\Delta L=l}$$

A $\Delta L = 0$ -transition is called a monopole transition, a $\Delta L = 1$ -transition is called a dipole transition, a $\Delta L = 2$ -transition is called a quadrupole transition, and so on. Hence, the name MDA. It is, however, imperative to know that there exist multiple types of monopole, dipole, quadrupole etc. transitions. They all depend on how ΔS and ΔJ are chosen in addition to ΔL .

For example, Fermi is $\Delta L = 0$, $\Delta S = 0$, $\Delta J = 0$, $\Delta T = 1$ and Gamow-Teller is $\Delta L = 0$, $\Delta S = 1$, $\Delta J = 0$, $\Delta T = 1$. Hence, both of them are monopole transitions. $\Delta L = 1$, $\Delta S = 1$, $\Delta J = 2$, $\Delta T = 1$ is called a spin-dipole transition (usually the strongest of the dipoles), and so on. However, different transition with the same ΔL are usually too similar in their angular distribution to resolve them individually through a fitting. Therefore, one can only take 1 type of monopole transition, 1 type of dipole transition, etc. along in the fit.

For a bin containing the IAS, the type of monopole obviously has to be Fermi. Since this IAS exhausts the full sum-rule, all other monopole transitions in the excitation-energy spectrum have to be Gamow-Teller. redwhat about IVGMR, etc? ask Remco!

one can then perform this fit for each bin of excitation-energy, extract the $L = 0$ -component and compute $B(GT)$. However, since we used arbitrary bins in excitation-energy, we technically do not compute absolute

$B(GT)$ -values, but a spectrum of the $B(GT)$ -distribution over excitation-energy. The advantage of this method is that one does not need to rely on the resolving of individual states AND that one will not need to perform a background subtraction. This background goes into the other components of the MDA. The disadvantage is that one can no longer assign absolute $B(GT)$ -values on specific transitions, but one has to be satisfied with an overall spectrum.

Remco suggested to do this analysis AND do my own state-by-state analysis. A state-by-state analysis does need background subtraction and heavily depends on the resolution in the excitation energy to resolve the different states. But one can assign absolute $B(GT)$ -states here. Then we can do both analysis procedures and compare. The idea of both procedures is outlined in figure ??fig:MDA.

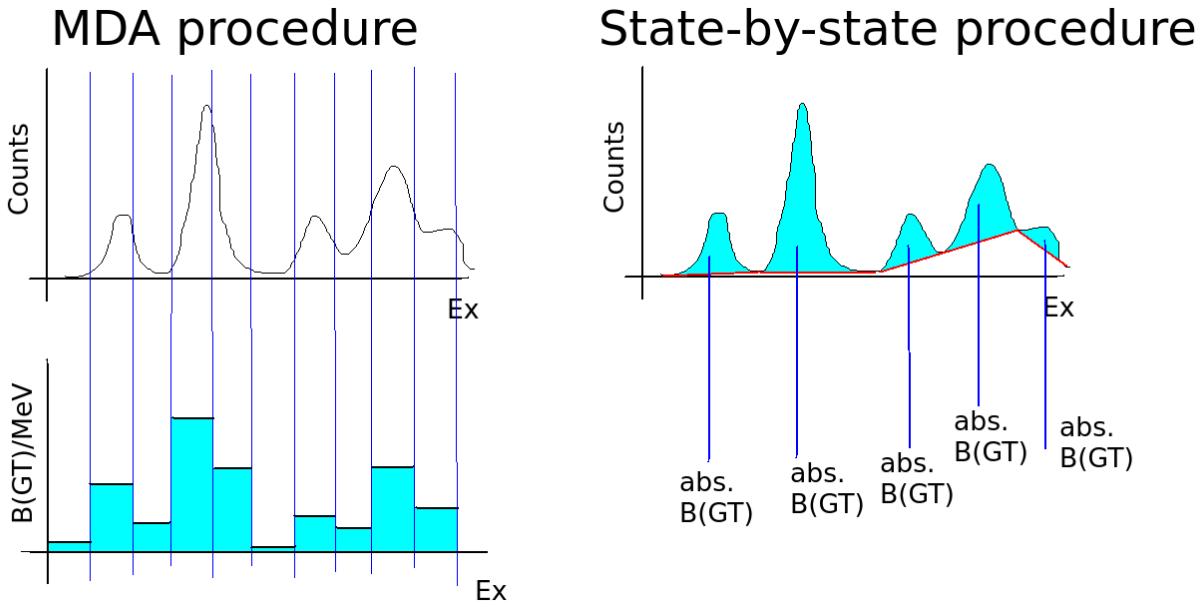


Figure 8.3: Outline of an MDA analysis versus a state-by-state analysis.

When comparing both procedures, one has to be extremely careful that the unit on the vertical axis of the $B(GT)$ -spectrum is **$B(GT)$ per MeV** while the state-by-state offers absolute $B(GT)$ -values.

8.7 Data fit

In order to obtain the $B(GT)$ or $B(F)$ -values, one takes the $\Delta L = 0$ -component of the MDA and evaluates this at scattering angle = 0. This is measured from the data fit and is denoted by

$$\frac{d\sigma}{d\Omega} \Big|_{\alpha=0, measured}$$

Then, we do 2 DWBA-calculations. One physical (as described above) and one with zero momentum transfer. This is done by putting the E-parameter in the DWBA-input-file equal to zero. This parameter is the Q-value of the ground state plus the excitation energy. Hence, putting it to zero means that we evaluate at zero momentum transfer. And then we evaluate both of these computed cross sections at $\alpha = 0$. then we compute:

$$\frac{d\sigma}{d\Omega} \Big|_{q=0} = \frac{\frac{d\sigma}{d\Omega} \Big|_{DWBA, physical}}{\frac{d\sigma}{d\Omega} \Big|_{DWBA, q=0}} \cdot \frac{d\sigma}{d\Omega} \Big|_{\alpha=0, measured}$$

This procedure corresponds to the extrapolation method in [30].

Then, we apply the eikonal approximation method from [30]:

$$\frac{d\sigma}{d\Omega} \Big|_{q=0} = \hat{\sigma} \cdot B(F/GT)$$

to obtain our observable of interest.

The best method is to use the phenomenological models from [26] for the unit cross section. This is the best method because this will employ experimental data and therefore it is model-independent, while a theoretical calculation is not. The recoil nuclei ^{116}Sb and ^{133}Sb do not have a ground state that can decay through Gamow-Teller decay, so we cannot measure the unit cross section directly. Therefore, the model from [26] is the best choice.

8.8 Final notes

The Faraday cups might have an efficiency of their own. Probably bets to use 79% for the 0 deg Faraday Cup and 100% for other settings. This efficiency also has an intrinsic error that should be taken along. We still need to find something for that. This causes that cross sections are usually never known better than 10%. See [31]: Comparison of data between Zegers and Fujiwara (Factor 0.88). A number of 80% was reported in [32] for the 0 deg setting. It is understandable that the Faraday cup in the dipole magnet will have an imperfect efficiency due to the dispersion matching technique. However, this should not pose a problem at other angular settings, so there we assume a perfect efficiency of the Faraday cup.

What do we do with the fact that multiple poles of the same order sometimes have their max at different places? This is not easy. The normal modes calculation that you do is actually for Giant Resonances, so your OBTD's are not always perfect. You can do a **fold** calculation for each of the 1-particle-1-hole transitions separately and see how it varies. The best pole that you can make for a certain state at specific E^* is to identify the excitation energy for each of the 1-particle-1-hole transitions in the normal modes calculation and then only take those that match the E^* of your specific state. Then do this for each pole in the MDA and then you get the best MDA possible. This is of course a lot of work, but it is possible.

The second bump in the IAS (between 3 and 4 degrees) depends on the spin-orbit interaction of the Love and Franey potential. So if you do not specify this correctly, your theoretical and experimental distributions might not match so well in this region. To cure this, one might take the **love_140**-file in the **fold**-package and modify the spin-orbit interaction. Check the papers [23, 24] learn how to do this. But it is probably not very easy to figure out which number to alter to turn off the spin-orbit interaction or modify it.

The mis-normalizations in your data at ^{13}C for example could also be caused by the use of different targets. we simply do not know that...

To smear a theoretical distribution with the experimental angular resolution. Take the DWBA-result, multiply that with $\sin \alpha$ (α is the scattering angle), to convert to count rates. Then convolute it with a Gaussian of the resolution you wish to obtain. Then divide through $\sin \alpha$ again. Now it is smeared.

The statistical error in integrated beam charge is the square root of the number of particles, not of the **sca**-file, according to Remco. This is very small, but the systematic error is much larger, which is contained in the error in detection efficiency of the Faraday Cup.

For the remarks on the SieveSlit analysis, see figure 8.4.

And remember that if you scale your angle, you have to transform the cross section too! This is done like:

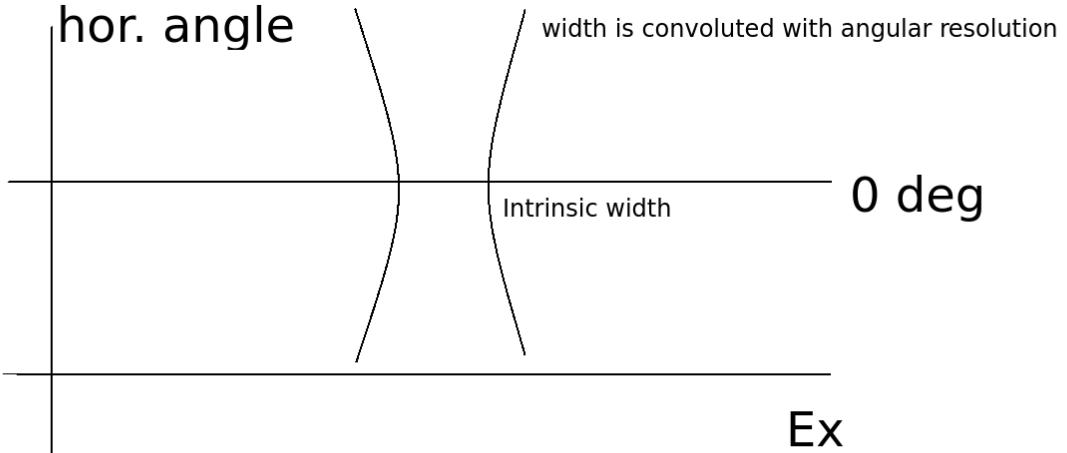
$$\alpha' = f(\alpha) \quad \frac{d\sigma}{d\Omega'} = \frac{d\sigma}{d\Omega} \cdot \frac{\sin(\alpha)}{\sin(f(\alpha)) \cdot f'(\alpha)}$$

for a linear transformation $\alpha' = c \cdot \alpha$ like suggested in figure 8.4, this reduces to

$$\frac{d\sigma}{d\Omega'} = \frac{d\sigma}{d\Omega} \cdot \frac{1}{c^2}$$

For ^{120}Sn and ^{118}Sn , the optical potentials of ^{90}Zr were used without any modifications. so why should you need any? **discuss with Nasser and Muhsin!** refer to [26] for the optical potential used.

Remco suggested to use the $\Delta L = 1$, $\Delta S = 1$, $\Delta J = 2$, $0^+ \rightarrow 2^-$ Dipole transition (This is SRD). He also suggested to use the $\Delta L = 2$, $\Delta S = 1$, $\Delta J = 2$, $0^+ \rightarrow 2^+$ Quadrupole transition and the $\Delta L = 3$, $\Delta S = 1$, $\Delta J = 3$, $0^+ \rightarrow 3^-$ Octupole transition for the MDA. If you want to do it better, then do it like described above.



$E(\text{recoil}) = c \cdot \text{angle}^2$ hence width is prop. to the angle.

If your SieveSlit is OK, then for different isotopes, one should have straight data for all of them. Else, your definition of the horizontal angle might be wrong! This might cause that you have to scale theta by a constant in order to get out the real angle. This might be why the minimum of the data and of DWBA are shifted.

Figure 8.4: Remarks on SieveSlit analysis.

Still need to look into detection efficiency error and into angular dependency of the detection efficiency. I already know that the latter has very poor resolution.

Right now, you only have a measurement error in your $B(F/GT)$ -values: the error from the fit parameter. However, the theoretical calculation clearly also has errors. The only way to know this is to vary the input parameters. This should be done in a sensible way, since there are clearly too many to do them all. The formula is the following:

$$B(GT) = \frac{1}{R \cdot \hat{\sigma}} \cdot \frac{d\sigma}{d\Omega} |_{\alpha=0}$$

Now the value $\frac{d\sigma}{d\Omega}|_{\alpha=0}$ only depends on your measured data. Hence, this error is determined only from the error in the normalization parameter, which comes from the errors in your datapoints. To get a handle on the error in R , the ratio between the $q = 0$ cross section and the measured one, the best way is to vary the optical potential. However, this error is probably not very big. The error in the Unit cross section is 5%, which is explained in [26]. then, finally, you have the systematic error in the $B(GT)$ (but not for $B(F)$), which comes from the $\Delta L = 2$ interference. This is how you build up the total error in the final answer.

EXTREMELY IMPORTANT!!! It looks like you swapped $\theta \rightarrow -\theta$ (θ is horizontal angle here) in your analysis! The 2.5 deg data for ^{12}C ground state has a larger cross section at larger angles! this should not be the case. Hence, try to flip this in your analysis and see what happens. It might solve the mismatch between the different runs for this state! It might also give a better shape of the Sn IAS-states and produce a better fit. Also, re-evaluate the use of your optical potentials again once you made this correction. It is important that, for making this angle flip, you do it in the LAB-frame and BEFORE you correct for the beam position. You have to flip around the LAB-origin! so simply do $\theta \rightarrow -\theta$, but do it all the way in the begin and make then sure that all your beam positions and acceptance cuts, etc. will work again. Hence, little programming time and lots and lots of CPU-time... But definitely try it!

Bibliography

- [1] Alexandros Apostolou, “Feasibility studies for heavy-light spectroscopy with PANDA and development of the reconstruction and analysis tools for the Straw Tube Tracker,” tech. rep., University of Groningen, September 2014.
- [2] Mohammad Ali Najafi, *Quasi-free Proton and Neutron knockout Reactions in ^{20}O* . PhD thesis, University of Groningen, January 2013.
- [3] Richard Liboff, *Introductory Quantum Mechanics*. Addison-Wesley, 4 ed., 2002.
- [4] R. Nave, “Some Nuclear Units.” <http://hyperphysics.phy-astr.gsu.edu/hbase/nuclear/nucuni.html>
- [5] , “Natural Units.” http://www.phys.ufl.edu/~korytov/phz4390/note_01_NaturalUnits_SMsummary.pdf
- [6] R. Eisberg, R. Resnick, *Quantum Physics of Atoms, Molecules, Solids, Nuclei and Particles*. John Wiley & Sons Inc, 2 ed., 1985.
- [7] , “Two-Electron Wave Functions.” <http://uw.physics.wisc.edu/~himpel/449exch.pdf>
- [8] M. G. Mayer, “Nuclear Configurations in the Spin-Orbit Coupling Model. I. Empirical Evidence,” *Physical Review*, vol. 78, April 1950.
- [9] , “Topic 15: Maximum Likelihood Estimation.” <http://math.arizona.edu/~jwatkins/o-mle.pdf>
- [10] The SAMURAI collaboration, “invariant mass spectroscopy experiment of the doubly magic nucleus ^{28}O ,” tech. rep., RI Beam Factory, 2014. Proposal for a Nuclear-Physics Experiment at the RI Beam Factory.
- [11] RCNP Collaboration, “High energy-resolution study of the ($^3\text{He}, \text{t}$) reactions on Ge, Sn and Te isotopes at 420 MeV and SD 2^- neutrino responses,” tech. rep., Research Center for Nuclear Physics, Osaka University, Februari 2014.
- [12] PEN Collaboration, “Technical Information Manual,” V1190 A/B, July 2012. NPO:00104/03:V1X90.MUTx/13
- [13] A. Tamii, Y. Fujita, H. Matsubara, T. Adachi, J. Carter, M. Dozono, H. Fujita, K. Fujita, H. Hashimoto, K. Hatanaka, T. Itahashi, M. Itoh, T. Kawabata, K. Nakanishi, S. Ninomiya, A. B. Perez-Cerdan, L. Popescu, B. Rubio, T. Saito, H. Sakaguchi, Y. Sakemi, Y. Sasamoto, Y. Shimbara, Y. Shimizu, F. D. Smit, Y. Tameshige, M. Yosoi, J. Zenhiro, “Measurement of high energy resolution inelastic proton scattering at and close to zero degrees,” *Nuclear Instruments and Methods in Physics Research A*, vol. 605, pp. 326–338, July 2009.
- [14] Atsushi Tamii, *Polarization transfer observables for proton inelastic scattering from ^{12}C at zero degrees*. PhD thesis, Kyoto University, September 1999.
- [15] H. Ejiri, N. Soukouti, J. Suhonen, “Spin-dipole nuclear matrix elements for double beta decays and astro-neutrinos,” *Physics Letters B*, vol. 729, pp. 27 – 32, 2014.
- [16] B. A. Brown, *Oxbash for Windows PC*, vol. MSUCL-1289 of *NSCL Reports*. Michigan: Michigan State University, 2004.
- [17] M. A. Hofstee, S. Y. van der Werf, A. M. van den Berg, N. Blasi, J. A. Borderwijk, W. T. A. Borghols,

- R. de Leo, G. T. Emery, S. Fortier, S. Gales, M. N. Harakeh, P. den Heijer, C. W. de Jager, H. Langevin-Joliot, S. Micheletti, M. Morlet, M. Pignanelli, J. M. Schippers, H. de Vries, A. Willis and A. van der Woude, “New Skyrme interaction for normal and exotic nuclei,” *Nuclear Physics A*, vol. 588, pp. 729–766, February 1995.
- [18] B. A. Brown, “Localized $1\hbar\omega$ particle-hole strength in nuclei,” *Physical Review C*, vol. 58, July 1998.
- [19] S. Y. van der Werf, “Program NORMOD.”.
- [20] R. G. T. Zegers, *Search for isovector giant monopole resonances*. PhD thesis, University of Groningen, November 1999.
- [21] J. Guillot, C. Bäumer, D. Beaumel, A. M. van den Berg, S. Brandenburg, G. Colò, B. Davids, S. Fortier, D. Frekers, E. W. Grewe, M. Fujiwara, S. Galès, P. Haefner, M.N. Harakeh, M. Hunyadi, M. de Huu, B.C. Junk, E. Rich, N. Van Giai, S.Y. van der Werf and H.J. Wörtche, “Investigation of Isovector Excitations via the $(t,3He)$ reaction at $E_t=43\text{MeV/u}$ on ^{58}Ni and ^{48}Ca targets: microscopic interpretation,” *Nuclear Physics A*, vol. 752, pp. 349 – 352, 2005. Proceedings of the 22nd International Nuclear Physics Conference (Part 2).
- [22] E. J. Schneid, A. Prakash and B. L. Cohen, “ (d,p) and (d,t) Reactions on the Isotopes of Tin,” *Physical Review*, vol. 156, April 1967.
- [23] W. G. Love and M. A. Franey, “Effective nucleon-nucleon interaction for scattering at intermediate energies,” *Physical Review C*, vol. 24, pp. 1073–1094, September 1981.
- [24] W. G. Love and M. A. Franey, “Nucleon-nucleon t-matrix interaction for scattering at intermediate energies,” *Physical Review C*, vol. 31, pp. 488–498, Februari 1985.
- [25] J. Kamiya, K. Hatanaka, T. Adachi, K. Fujita, K. Hara, T. Kawabata, T. Noro, H. Sakaguchi, N. Sakamoto, Y. Sakemi, Y. Shimbara, Y. Shimizu, S. Terashima, M. Uchida, T. Wakasa, Y. Yasuda, H. P. Yoshida and M. Yosoi, “Cross section and induced polarization in ^3He elastic scattering at 443,” *Physical Review C*, vol. 67, p. 064612, June 2003.
- [26] R. G. T. Zegers, T. Adachi, H. Akimune, S. M. Austin, A. M. van den Berg, B. A. Brown, Y. Fujita, M. Fujiwara, S. Galès, C. J. Guess, M. N. Harakeh, H. Hashimoto, K. Hatanaka, R. Hayami, G. W. Hitt, M. E. Howard, M. Itoh, T. Kawabata, K. Kawase, M. Kinoshita, M. Matsubara, K. Nakanishi, S. Nakayama, S. Okumura, T. Ohta, Y. Sakemi, Y. Shimbara, Y. Shimizu, C. Scholl, C. Simenel, Y. Tameshige, A. Tamii, M. Uchida, T. Yamagata, M. Yosoi, “Extraction of Weak Transition Strengths via the $(^3\text{He},t)$ Reaction at 420,” *Physical Review Letters*, vol. 99, p. 202501, November 2007.
- [27] S. Y. van der Werf, S. Brandenburg, P. Grasdijk and W. A. Sterrenburg, “The Effective ^3He -Nucleon Force in a Microscopic DWBA Approach to the $(^3\text{He},t)$ Charge-Exchange Reaction,” *Nuclear Physics A*, vol. 496, pp. 305–332, 1989.
- [28] R. G. T. Zegers, H. Akimune, S. M. Austin, D. Bazin, A. M. van den Berg, G. P. A. Berg, B. A. Brown, J. Brown, A. L. Cole, I. Daito, Y. Fujita, M. Fujiwara, S. Galès, M. N. Harakeh, H. Hashimoto, R. Hayami, G. W. Hitt, M. E. Howard, M. Itoh, J. Jänecke, T. Kawabata, K. Kawase, M. Kinoshita, T. Nakamura, K. Nakanishi, S. Nakayama, S. Okumura, W. A. Richter, D. A. Roberts, B. M. Sherrill, Y. Shimbara, M. Steiner, M. Uchida, H. Ueno, T. Yamagata and M. Yosoi, “The $(t,^3\text{He})$ and $(^3\text{He},t)$ reactions as probes of Gamow-Teller strength,” *Physical Review C*, vol. 74, p. 024309, August 2006.
- [29] G. W. Hitt, R. G. T. Zegers, S. M. Austin, D. Bazin, A. Gade, D. Galaviz, C. J. Guess, M. Horoi, M. E. Howard, W. D. M. Rae, Y. Shimbara, E. E. Smith and C. Tur, “Gamow-Teller transitions to ^{64}Cu measured with the $^{64}\text{Zn}(t,^3\text{He})$ reaction,” *Physical Review C*, vol. 80, p. 014313, July 2009.
- [30] G. Perdikakis, R. G. T. Zegers, S. M. Austin, D. Bazin, C. Caesar, J. M. Deaven, A. Gade, D. Galaviz, G. F. Grinyer, C. J. Guess, C. Herlitzius, G. W. Hitt, M. E. Howard, R. Meharchand, S. Noji, H. Sakai, Y. Shimbara, E. E. Smith, C. Tur, “Gamow-Teller unit cross sections for $(t,^3\text{He})$ and $(^3\text{He},t)$ reactions,” *Physical Review C*, vol. 83, p. 054614, May 2011.
- [31] R. G. T. Zegers, E. F. Brown, H. Akimune, S. M. Austin, A. M. van den Berg, B. A. Brown, D. A. Chamulak, Y. Fujita, M. Fujiwara, S. Galès, M. N. Harakeh, H. Hashimoto, R. Hayami, G. W. Hitt, M. Itoh, T. Kawabata, K. Kawase, M. Kinoshita, K. Nakanishi, S. Nakayama, S. Okumura, Y. Shimbara,

- M. Uchida, H. Ueno, T. Yamagata and M. Yosoi, “Gamow-Teller strength for the analog transitions to the first $T = 1/2, J^\pi = 3/2^-$ states in ^{13}C and ^{13}N and the implications for type Ia supernovae,” *Physical Review C*, vol. 77, p. 024307, Februari 2008.
- [32] R. G. T. Zegers, R. Meharchand, T. Adachi, S. M. Austin, B. A. Brown, Y. Fujita, M. Fujiwara, C. J. Guess, H. Hashimoto, K. Hatanaka, M. E. Howard, H. Matsubara, K. Nakanishi, T. Ohta, H. Okamura, Y. Sakemi, Y. Shimbara, Y. Shimizu, C. Scholl, A. Signoracci, Y. Tameshige, A. Tamii, M. Yosoi, “Spectroscopy of ^{24}Al and extraction of Gamow-Teller strengths with the $^{24}\text{Mg}(^3\text{He}, t)$ reaction at 420,” *Physical Review C*, vol. 78, p. 014314, July 2008.
- [33] , “Antisymmetrizer.” <http://en.wikipedia.org/wiki/Antisymmetrizer>
- [34] , “FAIR - An International Facility for Antiproton and Ion Research.” <http://www.fair-center.eu/>
- [35] , “FairRootGroup/FairSoft.” <https://github.com/FairRootGroup/FairSoft>
- [36] , “Lecture 10-11: Multi-electron atoms.” http://www.tcd.ie/Physics/people/Peter.Gallagher/lectures/js_atoms.html
- [37] , “Pauli Matrices.” http://en.wikipedia.org/wiki/Pauli_matrices
- [38] , “Spinorbit interaction.” http://en.wikipedia.org/wiki/Spin%E2%80%93orbit_interaction
- [39] Amos De Shalit, Herman Feshbach, *Theoretical Nuclear Physics: Nuclear Structure v 1*, vol. 1. John Wiley & Sons Inc, First Edition ed., January 1974. ISBN-10: 0471203858 & ISBN-13: 978-0471203858.
- [40] B. M. Sherrill, C. A. Bertulani, “Proton-tetraneutron elastic scattering,” *Physical Review C*, vol. 69, February 2004.
- [41] B. Srtoustrup, *The C++ Programming Language*. Addison-Wesley, 1985.
- [42] BESIII Collaboration, “Precision measurement of the D^{*0} decay branching fractions at BESIII.” PACS numbers: 12.38.Qk, 13.25.Ft, 13.20.Fc.
- [43] BESIII Collaboration, “Evidence for $e^+e^- \rightarrow \gamma\chi_{c1,2}$ at center-of-mass energies from 4.009 to 4.360 GeV.” PACS numbers: 14.40.Pq, 13.25.Gv, 13.66.Bc.
- [44] Brandenburg S, Ostendorf R, Hofstee M, Kiewiet H and Beijers H, “The irradiation facility at the AGOR cyclotron,” *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 261, pp. 82 – 85, 2007.
- [45] C. A. Douma, J. Gellanki, J. Mulder, “Simulation Verification.” Internal progress report at KVI-CART, Nuclear and Hadron Physics.
- [46] COMSOL, Inc., “Comsol: Modeling and Simulation for Everyone.” <http://www.comsol.com/>
- [47] D. Bertini, “R³Broot , simulation and analysis framework for the R³B-experiment at FAIR,” *Journal of Physics: Conference Series*, 2011.
- [48] D. Kresan, “R3Broot.” <http://r3broot.gsi.de/>
- [49] D. Kresan, M. Al-Turany, D. Bertini, R. Karabowicz, A. Manafov, A. Rybalchenko and F. Uhlig, “Event Reconstruction and Analysis in the R3BRoot Framework,” *Journal of Physics: Conference Series*, vol. 523, 2014.
- [50] D. V. Aleksandrov, E. Yu. Nikol'skii, B. G. Novatskii, S. B. Sakuta, D. N. Stepanov, “Search for Resonances in the Three- and Four-Neutron Systems in the $^7\text{Li}(^7\text{Li}, ^{11}\text{C})^3\text{n}$ and $^7\text{Li}(^7\text{Li}, ^{10}\text{C})^4\text{n}$ Reactions,” *Journal of Experimental and Theoretical Physics Letters*, vol. 18, pp. 43–46, January 2005.
- [51] D., M. Al-Turany, I. Koenig, F. Uhlig, “The FAIR simulation and analysis framework,” *Journal of Physics: Conference Series*, vol. 119, 2008.
- [52] E. Kozlova, H. Weick, B. Achenbach, K. H. Behr, G. Fehrenbacher, H. Geissel, M. Gleim, C. Karagiannis, A. Kelic, A. Kratz, T. Radon, K. Sammerer and M. Winkler, “Layout of the Super-FRS target hall,” *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and*

Atoms, vol. 266, no. 1920, pp. 4275 – 4279, 2008. Proceedings of the {XVth} International Conference on Electromagnetic Isotope Separators and Techniques Related to their Applications.

- [53] F. M. Marques, M. Labiche, N. A. Orr, J. C. Anglique, L. Axelsson, B. Benoit, U. C. Bergmann, M. J. G. Borge, W. N. Catford, S. P. G. Chappell, N. M. Clarke, G. Costa, N. Curtis, A. D'Arrigo, E. de Ges Brennand, F. de Oliveira Santos, O. Dorvaux, G. Fazio, M. Freer, B. R. Fulton, G. Giardina, S. Grvy, D. Guillemaud-Mueller, F. Hanappe, B. Heusch, B. Jonson, C. Le Brun, S. Leenhardt, M. Lewitowicz, M. J. Lpez, K. Markenroth, A. C. Mueller, T. Nilsson, A. Ninane, G. Nyman, I. Piqueras, K. Riisager, M. G. Saint Laurent, F. Sarazin, S. M. Singer, O. Sorlin, and L. Stuttge, “Detection of neutron clusters,” *Physical Review C*, vol. 65, April 2002.
- [54] F.M. Marques, N.A. Orr, H. Al Falou, G. Normand, N.M. Clarke, “On the possible detection of ${}^4\text{n}$ events in the breakup of ${}^{14}\text{Be}$,” *Physique Laboratoire Corpusculaire*, January 2005.
- [55] Gellanki J *et al*, *Super-FRS slit system and possible passive cooling techniques*, vol. 2015-1 of *GSI Report*, p. 498 p. Darmstadt: GSI Helmholtzzentrum für Schwerionenforschung, 2015. psp 2.4.6.2.3 Super-FRS.
- [56] Gutbrod H H, Augustin I, Eickhoff H, Gro K D, Henning WF, Kramer D, Walter G, “FAIR baseline technical report,” tech. rep., GSI, 2006.
- [57] H. Ejiri, “Double Beta Decays and Neutrino Masses,” *Journal of the Physical Society of Japan*, vol. 74, p. 21012127, August 2005.
- [58] H. Ejiri, “Nuclear spin isospin responses for low-energy Neutrinos,” *Physics Reports*, vol. 338, pp. 265–351, 2000.
- [59] H. Geissel, P. Armbruster, K. Behr *et al.*, “The GSI projectile fragment separator (FRS); a versatile magnetic system for relativistic heavy ions.,” *Nuclear Instruments and Methods in Physics Research Section B*, vol. 70, pp. 286–297, 1992.
- [60] H. Geissel, Th. Schwab, P. Armbruster, J.P. Dufour, E. Hanelt, K. H. Schmidt, B. Sherrill and G. Manzenberg, “Ions penetrating through ion-optical systems and matter: a non-liouvillian phase-space modelling,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 282, no. 1, pp. 247 – 260, 1989.
- [61] H. Moeini, *Feasibility Experiment and Simulations for EXL*. PhD thesis, University of Groningen, The Netherlands, 2010.
- [62] H. Simon, “Exploring the continuum, proton and neutron rich.” <http://slideplayer.com/slide/3235754/>
- [63] H. Weick, “The GSI Fragment Separator Website.” <http://www-wnt.gsi.de/frs/index.asp>
- [64] H. Weick, “Super-FRS Status,” in *NuSTAR Collaboration Meeting*, 2015.
- [65] Hctor Alvarez Pol, Jos Benlliure Anaya, Enrique Casarejos Ruiz, Dolores Cortina Gil, Ignacio Durn Escribano, Martn Gascn Vzquez, Noelia Montes Alonso, David Prez Loureir, “The web of the ”Reactions with Radiative Relativistic Beams” activities on USC.” <http://www.usc.es/genp/r3b/>
- [66] I. Tews, T. Krger, K. Hebeler, A. Schwenk, “Neutron Matter at Next-to-Next-to-Next-to-Leading Order in Chiral Effective Field Theory,” *Physical Review Letters*, vol. 110, January 2013.
- [67] J. Gellanki, “A New Cooling Technique for the Super-FRS Slit System,” in *NNV Annual meeting*, 2014.
- [68] J. Gerl, “NUSTAR Status,” in *NuSTAR Collaboration Meeting*, 2015.
- [69] Luis Miguel Orona, Helmut Weick, Jouni Mattila, Faraz Amjad, Ekaterina Kozlova, Christos Karagianannis, Karl-Heinz Behr, Martin Winkler, “Super-FRS Target Area RemoteHandling: Scenario and Development,” *International Journal of Advanced Robotic Systems*, vol. 10, p. 9, September 2013. DOI: 10.5772/57073.
- [70] M. N. Harakeh, A. van der Woude, *Giant Resonances*. Oxford University Press, July 2001. ISBN: 9780198517337.

- [71] M. Verhart, “General Introduction,” in *SWG: Spring Workshop on Geant4*, 2015.
- [72] M. Winkler, H. Geissel, H. Weick, B. Achenbach, K. H. Behr, D. Boutin, A. Branle, M. Gleim, W. Haller, C. Karagiannis, A. Kelic, B. Kindler, E. Kozlova, H. Leibrock, B. Lommel, G. Moritz, C. Mahle, G. Manzenberg, C. Nociforo, W. Plass, C. Scheidenberger, H. Simon, K. Sammerer, N. A. Tahir, A. Tauschwitz, M. Tomut, J. S. Winfield, M. Yavor, “The status of the Super-FRS in-flight facility at FAIR,” *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 266, no. 1920, pp. 4183 – 4187, 2008. Proceedings of the {XVth} International Conference on Electromagnetic Isotope Separators and Techniques Related to their Applications.
- [73] Mathworks, “Matlab: The Language of Technical Computing.” <http://nl.mathworks.com/products/matlab/>
- [74] N. Hall, “FACILITY FOR ANTIPIRON AND ION RESEARCH: An international science centre in Europe for studying the building blocks of matter and the evolution of the Universe.” http://www.fair-center.eu/fileadmin/fair/publications_FAIR/FAIR_Broschuere_autumn2013_V3_72dpi.pdf
- [75] N. Kalantar-Nayestanaki, “NUSTAR activities at FAIR,” in *International Conference on Sciene and Technology for FAIR in Europe 2014*, 2014.
- [76] NuSTAR Collaboration Super-FRS working group: H. Geissel *et al.*, “Technical design Report on the Super-FRS,” tech. rep., GSI and Collaborators, 2009.
- [77] RCNP Collaboration, “High-resolution study of Gamow-Teller transitions in odd- A high S_p nuclei,” tech. rep., Research Center for Nuclear Physics, Osaka University, March 2014.
- [78] RCNP Collaboration, “Using high resolution ($^3\text{He}, t$ reactions on ^{116}Cd for nuclear and neutrino physics,” tech. rep., Research Center for Nuclear Physics, Osaka University, Februari 2014.
- [79] ROOT Development Team, “ROOT Data Analysis Framework.” <https://root.cern.ch/drupal/>
- [80] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, F. Behner, L. Bellagamba, J. Boudreau, L. Broglia, A. Brunengo, H. Burkhardt, S. Chauvie, J. Chuma, R. Chytracek, G. Cooperman, G. Cosmo, P. Degtyarenko, A. Dell’Acqua, G. Depaola, D. Dietrich, R. Enami, A. Feliciello, C. Ferguson, H. Fesefeldt, G. Folger, F. Foppiano, A. Forti, S. Garelli, S. Giani, R. Giannitrapani, D. Gibin, J.J. Gmez Cadenas, I. Gonzlez, G. Gracia Abril, G. Greeniaus, W. Greiner, V. Grichine, A. Grossheim, S. Guatelli, P. Gumplinger, R. Hamatsu, K. Hashimoto, H. Hasui, A. Heikkinen, A. Howard, V. Ivanchenko, A. Johnson, F.W. Jones, J. Kallenbach, N. Kanaya, M. Kawabata, Y. Kawabata, M. Kawaguti, S. Kelner, P. Kent, A. Kimura, T. Kodama, R. Kokoulin, M. Kossov, H. Kurashige, E. Lamanna, T. Lampn, V. Lara, V. Lefebure, F. Lei, M. Liendl, W. Lockman, F. Longo, S. Magni, M. Maire, E. Medernach, K. Minamimoto, P. Mora de Freitas, Y. Morita, K. Murakami, M. Nagamatu, R. Nartallo, P. Nieminen, T. Nishimura, K. Ohtsubo, M. Okamura, S. O’Neale, Y. Oohata, K. Paech, J. Perl, A. Pfeiffer, M.G. Pia, F. Ranjard, A. Rybin, S. Sadilov, E. Di Salvo, G. Santin, T. Sasaki, N. Savvas, Y. Sawada, S. Scherer, S. Sei, V. Sirotenko, D. Smith, N. Starkov, H. Stoecker, J. Sulikimo, M. Takahata, S. Tanaka, E. Tcherniaev, E. Safai Tehrani, M. Tropeano, P. Truscott, H. Uno, L. Urban, P. Urban, M. Verderi, A. Walkden, W. Wander, H. Weber, J.P. Wellisch, T. Wenaus, D.C. Williams, D. Wright, T. Yamada, H. Yoshida, D. Zschiesche, “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.
- [81] S. Bacchi, *Study of Compression Modes in 56Ni using an Active Target*. PhD thesis, University of Groningen, March 2015.
- [82] S. C. Pieper, “Can Modern Nuclear Hamiltonians Tolerate a Bound Tetraneutron?,” *Physical Review Letters*, vol. 90, June 2003.
- [83] S. Cierjacks, G. Markus, W. Michaelis, W. Pnitz, “Further Evidence for the Nonexistence of Particle-Stable Tetraneutrons,” *Physical Review*, vol. 137, January 1965.
- [84] S. Quaglioni P. Navrati, “Ab Initio Many-Body Calculations of $n - ^3\text{H}$, $n - ^4\text{He}$, $p - ^{3,4}\text{He}$, and $n - ^{10}\text{Be}$ Scattering,” *arXiv: 0804:560v1*, 2008. PACS numbers: 21.60.De, 25.10.+s, 27.10.+h, 27.20.+n.
- [85] Siemens PLM Software, “NX CAE.” http://www.plm.automation.siemens.com/en_us/products/nx/

for-simulation/cae/

- [86] The R³B collaboration , “HEAT, Laboratory of High Energy Physics, Astrophysics, Advanced Technologies.” <http://www2.spacescience.ro/heat/collaborations/r3b.php>
- [87] The R³B collaboration , “Reactions with Relativistic Radioactive Beams.” <http://www.fair-center.eu/for-users/experiments/nustar/experiments/r3b.html>
- [88] The R³B collaboration , “A next generation experimental setup for studies of Reactions with Relativistic Radioactive Beams.” <http://www.fair-center.eu/public/experiment-program/nustar-physics/r3b.html>
- [89] The R³B collaboration , “A next generation experimental setup for studies of Reactions with Relativistic Radioactive Beams.” https://www.gsi.de/work/forschung/nustarennna/nustarennna_divisions/kernreaktionen/activities/r3b.html
- [90] The R³B collaboration , “Technical Report for the Design, Construction and Commissioning of NeuLAND: The High-Resolution Neutron Time-of-Flight Spectrometer for R³B ,” tech. rep., GSI and Collaborators, 2011.
- [91] The NuSTAR collaboration, “NUSTAR-NUclear STructure, Astrophysics and Reactions.” <http://www.fair-center.eu/for-users/experiments/nustar/experiments.html>
- [92] The SAMURAI collaboration, “Investigation of the ⁴n system at SAMURAI by measuring p, p α quasi-free scattering at large momentum transfer in complete kinematics,” tech. rep., RI Beam Factory, May 2014. Proposal for a Nuclear-Physics Experiment at the RI Beam Factory.
- [93] The SAMURAI collaboration, “Measurement of the neutron-decay lifetime of the ²⁶O ground state at the SAMURAI setup at RIBF,” tech. rep., RI Beam Factory, May 2014. Proposal for a Nuclear-Physics Experiment at the RI Beam Factory.
- [94] W. G. Love, M. A. Franey, “Effective nucleon-nucleon interaction for scattering at intermediate energies,” *Physics Review C*, vol. 24, pp. 1073–1094, September 1981.
- [95] Weast R C and Chemical Rubber Company (Cleveland, Ohio), *CRC Handbook of Chemistry and Physics: A Ready-reference Book of Chemical and Physical Data*. CRC Press, 1975.
- [96] Y. Sakuragi and M. Katsuma, “Spin-orbit interactions for 3He-nucleus scattering at intermediate energies,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 402, no. 2, pp. 347–353, 1998. Proceedings of the 7th RCNP International Workshop on Polarized He Beams and Gas Targets and Their Application.
- [97] D. Y. Pang, P. Roussel-Chomaz, H. Savajols, R. L. Varner and R. Wolski, “Global optical model potential for $A = 3$ projectiles,” *Physical Review C*, vol. 79, p. 024615, Februari 2009.
- [98] J. Kamiya, K. Hatanaka, T. Adachi, K. Fujita, K. Hara, T. Kawabata, T. Noro, H. Sakaguchi, N. Sakamoto, Y. Sakemi, Y. Shimbara, Y. Shimizu, S. Terashima, M. Uchida, T. Wakasa, Y. Yasuda, H. P. Yoshida and M. Yosoi, “Cross section and induced polarization in ³He elastic scattering at 443,” *Physical Review C*, vol. 67, p. 064612, June 2003.
- [99] K. Pham, J. Jänecke, D. A. Roberts, M. N. Harakeh, G. P. A. Berg, S. Chang, J. Liu, E. J. Stephenson, B. F. Davis, H. Akimune and M. Fujiwara, “Fragmentation and splitting of Gamow-Teller resonances in Sn(³He, t)Sb charge-exchange reactions, $A = 112 - 124$,” *Physical Review C*, vol. 51, pp. 526–540, Februari 1995.
- [100] J. Jänecke, K. Pham, D. A. Roberts, D. Stewart, M. N. Harakeh, G. P. A. Berg, C. C. Foster, J. E. Lisantti, R. Sawafta, E. J. Stephenson, A. M. van den Berg, S. Y. van der Werf, S. E. Muraviev, M. N. Urin, “Fragmentation of Gamow-Teller strength observed in ^{117,120}Sn(³He, t)^{117,120}Sb charge-exchange reactions,” *Physical Review C*, vol. 48, pp. 2828–2839, December 1993.
- [101] W. T. Chou, E. K. Warburton and B. A. Brown, “Gamow-Teller beta-decay rates for $A \leq 18$ nuclei,” *Physical Review C*, vol. 47, pp. 163–177, January 1993.
- [102] D. Frekers, P. Puppe, J. H. Thies and H. Ejiri, “Gamow-Teller strength extraction from (³He, t) reactions,”

Appendix A

Dead Ends