

# Impact of non V-A contributions on the $T_2 \beta$ spectrum and their relevance for the KATRIN experiment

Diplomarbeit  
von  
Daniel Sevilla Sanchez

Institut für Physik  
Johannes Gutenberg-Universität Mainz  
April 2006



# Contents

<b>1</b>	<b>Motivation and Introduction</b>	<b>1</b>
1.1	History of neutrino physics . . . . .	1
1.2	Impact of non-zero neutrino mass on physics . . . . .	2
1.3	Neutrino mass measurement . . . . .	3
1.4	Expected result of the KATRIN experiment . . . . .	4
1.5	Objectives . . . . .	4
<b>2</b>	<b>Tritium <math>\beta</math> decay spectrum: Standard approach and corrections</b>	<b>7</b>
2.1	Within the Standard Model . . . . .	7
2.1.1	Fermi approach: Point-like interaction . . . . .	7
2.1.2	Parity violation, left-handed neutrinos and V-A structure . . . . .	9
2.1.3	Calculation of the decay probability . . . . .	11
2.2	Beyond the Standard Model . . . . .	12
2.2.1	Right-handed neutrinos and T, S currents . . . . .	12
2.2.2	Tachyonic neutrinos . . . . .	13
2.2.3	Violation of Lorentz invariance . . . . .	14
2.3	Energy considerations . . . . .	14
2.3.1	Radiative and electrostatic corrections to the spectrum . . . . .	14
2.3.2	Excitation of the daughter molecule . . . . .	15
<b>3</b>	<b>The KATRIN experiment</b>	<b>17</b>
3.1	MAC-E filter and the adiabatic motion of electrons . . . . .	20
3.2	Windowless gaseous tritium source and electron scattering . . . . .	24
3.3	Background events . . . . .	26
3.4	Measurement of the decay rate . . . . .	27
<b>4</b>	<b>Simulations of the KATRIN experiment</b>	<b>29</b>
4.1	Calculation procedure . . . . .	29
4.2	KATRIN simulations within the Standard Model . . . . .	33
4.3	KATRIN simulations beyond the Standard Model . . . . .	40
4.3.1	Non V-A structure . . . . .	40
4.3.2	Tachyonic neutrinos . . . . .	50
4.3.3	Violation of Lorentz invariance . . . . .	53
<b>5</b>	<b>Conclusion and Outlook</b>	<b>55</b>
<b>A</b>	<b>Parameters used in the simulations</b>	<b>57</b>

<b>B Simulation code</b>	<b>59</b>
B.1 Simulation package with MINUIT algorithms . . . . .	59
B.1.1 Tools.h . . . . .	59
B.1.2 ResponseFunction.h . . . . .	61
B.1.3 DifferentialSpectrum.h . . . . .	66
B.1.4 IntegratedSpectrum.h . . . . .	80
B.1.5 DataGenerator.h . . . . .	82
B.1.6 Function.h . . . . .	85
B.1.7 Fitting.h . . . . .	93
B.1.8 Spectrum.cpp . . . . .	104
B.2 Simulation of the Non V-A approximate description . . . . .	104
B.2.1 Levenberg.cpp . . . . .	104
 <b>List of Acronyms</b>	 <b>115</b>
 <b>Bibliography</b>	 <b>121</b>
 <b>Acknowledgements</b>	 <b>123</b>

# Chapter 1

## Motivation and Introduction

### 1.1 History of neutrino physics

Since Wolfgang Pauli postulated the existence of neutrinos in 1930 [1] to explain the continuous energy spectrum of beta decay by saving the principle of energy conservation, the verification of the physical properties of the neutrino has become a fundamental test for physics theories and even an important key for opening new physics branches.

The first successful attempt to describe processes in which neutrinos are involved was carried out by Enrico Fermi [2] few years after Pauli's postulate. Fermi found out that there were some analogies between the new so-called weak interaction and the very well-known electromagnetism and hence he built a weak Hamiltonian by comparison to the electromagnetic case. The result was a Hamiltonian formed by 4 different currents or particles whose main feature was, that they shared one point of the space-time structure (point-like interaction).

Due to the very low interaction probability between neutrinos and matter, the detection of neutrinos or the determination of any of their properties has always been an experimental challenge. During the 1950s, C.L. Cowan and F. Reines [3] measured for the first time processes that corroborated the existence of neutrinos in nature. At the same time, the first experiments that aimed to measure the electron neutrino mass were carried out [4]. All of them have been overcoming technical difficulties and improving the measurement sensitivity during years. However, they have concluded that the neutrino mass is either zero or smaller than an upper limit established by the experiment.

Besides the electron neutrino studies, it was discovered that there is one neutrino flavor for each element of the lepton family, that is in addition a muon neutrino [5] and a tau neutrino [6].

Fermi's theory was corrected by R. Feynman, M. Gell-Mann [7], E.C.G. Sudarshan and R.E. Marshak [8] after discovering the presence of parity violation [9] in nature as well as the left-handed helicity of neutrinos (right-handed for anti-neutrinos), leading to the Vector minus Axialvector (V-A) structure of the interaction.

The present and most complete description of the weak interaction was found by S.L. Glashow, S. Weinberg and A. Salam by assuming the existence of intermediate bosons which carry the force and break the point-like image of the interaction. Both, the weak force and the electromagnetism are mixed by the theory to demonstrate that they are the same force. This theory, as described by Glashow-Weinberg-Salam, is one of the pillars of the Standard Model (SM). Since only this part of the SM is of concern, we shall not

mention the strong interaction or quantum chromodynamics in this work.

Although the Glashow-Weinberg-Salam theory has been verified in all its predictions some corrections to the SM were suggested. For instance, if neutrinos were right-handed or tachyons, then electroweak physics should be corrected. At present, the possibility for physics to have features beyond the SM description is still open and therefore many experiments keep on looking for the Standard Model goodness. This is extremely important for physics and specifically for neutrino physics, since the weak interaction, aside the negligible gravitation, is the only one neutrinos are involved in.

The hypothesis of possible oscillations [10] between the three neutrino flavors  $\nu_e$ ,  $\nu_\mu$ ,  $\nu_\tau$  came up some years ago to explain the lack of electron neutrinos from the Sun [11]. Such a situation might only occur if neutrinos have non-vanishing rest masses in disagreement with the Standard Model.

So far, there have not been conclusions about the exact value of the neutrino mass, although recent experiments (SNO, Kamiokande, GNO, Gallex, Sage, Homestake, Soudan2, MACRO) have shown, that neutrino flavor oscillations are possible and therefore non-vanishing neutrino masses have been postulated [12, 13, 14, 15]. This conclusion has turned out to be the first proof of physics beyond the Standard Model and therefore it opens a new range of possibilities.

## 1.2 Impact of non-zero neutrino mass on physics

Knowing the neutrino mass has a deep impact on physics, either for answering a fundamental question or for ruling out some current models. The first reason needs no explanation, the second one has many implications on different fields. For instance, concerning astrophysics and cosmology, neutrinos are a good candidate for explaining hot dark matter ( $\nu$ HDM) and hence the evolution of large scale structures (LSS) [16]. The reason for which neutrinos could explain how LSS behave is that the ratio between neutrinos and baryons is at present around 1 billion. Therefore, these relic neutrinos could play an important role on the average density of the universe leading to the metric that will determine the evolution of LSS.

On the other hand, cosmological models of structure formation strongly depend on the ratio between cold dark matter (CDM) and HDM. CDM and HDM are differentiated by the relativistic properties of the particles that form matter. If they are highly relativistic then matter is HDM, while no relativistic particles are claimed to be CDM. As neutrinos have a very small mass, at the moment of decoupling, neutrinos moved with velocities near  $c$  and therefore they behaved strongly relativistic. Hence, a determination of the neutrino contribution to the total dark matter content of the universe is important for our understanding of structure formation. In terms of cosmology, it is known that neutrinos heavier than  $1 \text{ eV}/c^2$  would dominate the universe [17] and a neutrino mass of about  $10 \text{ eV}/c^2$  may close the universe.

Regarding particle physics and quantum field theory, it would also be very interesting to elucidate whether neutrinos are Dirac particles or Majorana particles. The experimental ability to distinguish between Dirac or Majorana states depends very strongly on the actual value of the neutrino rest mass [18]. This distinction is important basically to know whether neutrinos are their own antiparticles or not. Moreover, it could also help to understand theory mass models, i.e., why the neutrino mass is much smaller than the

other particle masses.

Measuring the exact neutrino mass or an upper limit could help us to solve the above mentioned problems on cosmology and particle physics, but in addition it would give us the answer to the fundamental question of knowing the physical properties of the smallest constituents of nature.

### 1.3 Neutrino mass measurement

KATRIN (**K**arlsruhe **T**ritium **N**eutrino **M**ass Experiment, see Fig. 1.1) is intended to measure the electron neutrino mass by investigating the  $T_2$   $\beta$ -decay spectrum near the endpoint. For that purpose, an integrating spectrometer is being built. The setup, which will be ready in 2008, is based on the expertise gained in the Mainz and Troitsk neutrino mass measurement experiments [19, 20, 21]. As we shall discuss in detail in Ch. 3, the method used to measure the spectrum near the endpoint consists on selecting only the most energetic electrons of the decay. By using the above mentioned integrating spectrometer, a retarding potential is placed between the decay source and the detector, such that only electrons with energy above the retarding potential are able to reach the detector. The measurement is therefore the integral of the decay rate from the retarding potential to the decay endpoint.

Experimentally, it is possible to achieve such a measurement by means of a MAC-E filter, that is **M**agnetic **A**diabatic **C**ollimation followed by an **E**lectrostatic filter [22]. This type of spectrometer not only filters electrons but also guides them from the source to the detector due to the magnetic field action. During this pathway their energy is transformed adiabatically in order to have only one motion component parallel to the magnetic field.

To perform such an extremely difficult measurement, many other experimental challenges must be solved for KATRIN. For instance, a windowless gaseous tritium source (WGTS) needs to be prepared. The column density inside this source has to amount to  $10^{17}$  atoms/cm<sup>2</sup> which must be kept within very small fluctuations of 0.1% [22] at most. This type of tritium source was ideally designed to deliver the most profitable relationship between the number of decays and the number of scattering processes for the  $\beta$ -particles. In addition, the retarding potential applied by the MAC-E filter has to be known with a precision of 1ppm [22] and the background sources must be recognized and shielded. For that purpose, an inner electrode is being constructed, since one of the biggest background sources are electrons from the surface of the spectrometer vessel. This inner electrode will repel those electrons [23, 24].

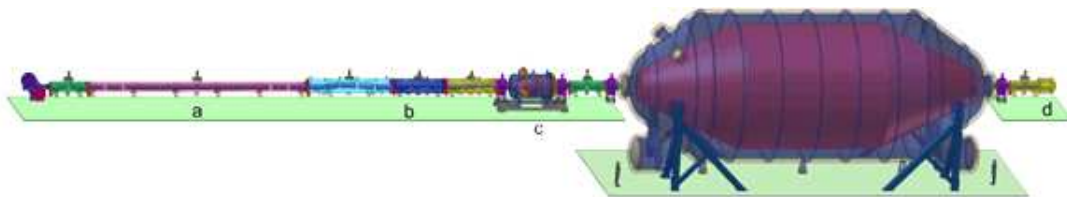


Figure 1.1: Overview of the experimental setup for the KATRIN experiment [22].

A general overview of the KATRIN experiment can be seen in Fig. 1.1. The long tube coming from the left represents the tritium source (WGTS) followed by the pumping

sections which pump tritium out of the WGTS. Tritium must be continuously injected and pumped out in order to keep the column density as constant as possible. After the WGTS and the pumping sections (thin cylinder in Fig. 1.1) and right before the main spectrometer (large cylinder in Fig. 1.1) is placed the so-called pre-spectrometer which is intended to filter most of the electrons but let the fastest ones go towards the main spectrometer. The functionality of the pre-spectrometer is to reduce the number of electrons to be delivered to the main spectrometer in order to avoid ionization of the residual gas inside it. The setup is completed with the installation of the detector right after the main spectrometer.

After building this experimental configuration with the technical features explained above and overcoming some other experimental difficulties, the KATRIN experiment is supposed to improve the current limits on the neutrino mass up to a factor of 10.

## 1.4 Expected result of the KATRIN experiment

The most precise experiments so far related to the neutrino mass measurement are the ones from Mainz and Troitsk [19, 20]. They established an upper limit for the neutrino mass of  $m(\nu_e) \leq 2.3 \text{ eV}/c^2$  (experiment sensitivity) (95% C.L.)[21].

If the neutrino mass is actually very close to a vanishing value, the KATRIN experiment is expected to set an upper limit for the neutrino mass of about  $m(\nu_e) \leq 0.2 \text{ eV}/c^2$  (90% C.L.). If not, the lowest value of  $m_\nu$  that can be established with 99.9999% C.L. is  $0.35 \text{ eV}/c^2$  (discovery potential).

At this point, it is interesting to comment that both, the experiment sensitivity and the discovery potential values rely on the theoretical functions used to fit the experimental data, that means that the experiment is to some extent model-dependent.

The numbers given above are calculated under the assumption that neutrinos obey physics within the Standard Model. But as will be demonstrated through this work, by calculating these quantities with Non V-A physics, the discovery potential is worse.

Nevertheless, devoted experiments like a high-precision measurement of the  $Q$  value of  ${}^3\text{T}(\beta^-){}^3\text{He}$  decay could help to solve this situation, since better experimental limits on the tritium endpoint and the Non V-A coupling constants would almost restore both the discovery potential and the experiment sensitivity to the optimal values of the Standard Model. These devoted experiments are the only way to *break* the model dependence of KATRIN.

## 1.5 Objectives

The principal objective of this work is to determine the new KATRIN sensitivity on the electron neutrino mass when Non Vector minus Axialvector (Non V-A) physics is considered in the analysis of the experimental data from tritium beta decay. This sort of study is accomplished by means of Monte Carlo (MC) simulations which reproduce the experiment many times by creating experimental data with statistical fluctuations. A secondary product of such calculation is the experiment sensitivity on other physical magnitudes, like the tritium beta decay endpoint, the Non V-A coupling constants or the background rate on which the experiment could in principle also improve the present experimental limits.



To reach such objectives a simulation package has been developed. This package was enlarged to also check the sensitivity of the experiment on the SM as well as to cover other Non Standard Model (Non SM) problems like tachyonic neutrinos or the theory of violation of Lorentz invariance.

This thesis is mainly split into five chapters, as follows:

- Chapter 2 is devoted to study the basics of tritium beta decay either within the Standard Model or within physics beyond the Standard Model. Also, energy corrections common to both cases are explained. It summarizes all approaches to the tritium beta decay we shall deal with.

- Chapter 3 is dedicated to talk about the grounds of the experiment. The main experimental features and their mathematical treatments will be justified.

- In chapter 4, it is explained in detail how the Monte Carlo simulations were done. Results that were obtained by means of this method are discussed.

- Chapter 5 summarizes the most important results of this work. It also insists on suggesting present possible improvements to be done on the same direction of this research with the aim of finding enhanced results.

There are several references related to the different mentioned topics through out the whole work. The reader may find useful to follow the KATRIN Design Report [22] as a general guidebook and specifically for the experimental details.



## Chapter 2

# Tritium $\beta$ decay spectrum: Standard approach and corrections

In this chapter the basics of the beta decay will be put forward. Specifically, here the beta decay on tritium is of interest, represented by:

$$T \rightarrow He^+ + e^- + \bar{\nu}_e \quad (2.1)$$

Actually, on a more fundamental level, the process (2.1) occurs by involving the following particles:

$$n \rightarrow p^+ + e^- + \bar{\nu}_e \quad (2.2)$$

It can be even described at the quark level:

$$d \rightarrow u + e^- + \bar{\nu}_e \quad (2.3)$$

which summarizes the change of flavor from the quark *down* to the quark *up* and the generation of two leptons.

### 2.1 Within the Standard Model

With the aim of obtaining the  $\beta$  spectrum of molecular tritium  $T_2$  one should follow Fermi's first theory. Afterwards corrections due to parity violation and chirality properties of neutrinos will be incorporated.

Although, the Standard Model approach should include intermediate bosons, we shall work with a reduced description that deals only with the point-like image of the interaction.

#### 2.1.1 Fermi approach: Point-like interaction

Enrico Fermi built a Hamiltonian for the weak interaction by taking the electromagnetic Hamiltonian as a guide [25].

The Hamiltonian density for the electromagnetic interaction can be expressed as:

$$\mathcal{H} = \sum_n e j^\mu(\mathbf{r}_n) A_\mu(\mathbf{r}_n) \quad (2.4)$$

where  $A_\mu$  is the vector potential,

$j^\mu$  is the current density of the particle  $n$  on the position  $\mathbf{r}_n$ ,

$n$  is the index which identifies each particle within the electromagnetic field,

$\mathbf{r}_n$  is the position vector of the particle  $n$ , and

$e$  is the charge unity and determines how strong the interaction is.

Finally, summation over the index  $\mu$  is assumed.

Thus, in analogy, the Hamiltonian density for the weak interaction is:

$$\mathcal{H} = \sum_n g J^\mu(\mathbf{r}_n) L_\mu(\mathbf{r}_n) \quad (2.5)$$

where  $L_\mu$  is the “vector potential” of the “field” emitted as a lepton,

$J^\mu$  is the “current density” of a neutron-proton transition,

$n$  is the index which identifies each particle on the process,

$\mathbf{r}_n$  is the position vector of the particle  $n$ , and

$g$  is the new coupling constant that will set the strength of the interaction.

While working in relativistic quantum mechanics, both Dirac’s equation and Klein-Gordon’s equation are valid. In this work, Dirac’s equation has been chosen and therefore fields like  $\Psi$  represent four-vectors of the Dirac image. In (2.4) currents may be built like  $j^\mu = \bar{\Psi}_f \gamma^\mu \Psi_i$ . Hence, the *weak current* must also have a structure like  $J^\mu = \bar{\Psi}_p \gamma^\mu \Psi_n$ . Moreover, it is assumed that the “vector potential”  $L_\mu$  associated to the lepton current is linear with respect to the lepton and neutrino field, in order to achieve a Hamiltonian density  $\mathcal{H}$  that is relativistic invariant. Consequently,  $L_\mu = \bar{\Psi}_e \gamma_\mu \Psi_\nu$ .

In quantum electrodynamics  $\Psi_p, \Psi_n, \Psi_e, \Psi_\nu$ , allude to the annihilation of respectively  $p, n, e, \nu$  and the creation of their antiparticles, such that an interaction like

$$\mathcal{H} \sim (\bar{\Psi}_p \gamma^\mu \Psi_n) (\bar{\Psi}_e \gamma_\mu \Psi_\nu) \quad (2.6)$$

may describe any of the following processes:

1.  $n \rightarrow p + e^- + \bar{\nu}_e$
  2.  $n + \nu_e \rightarrow p + e^-$
  3.  $n + e^+ \rightarrow p + \bar{\nu}_e$
  4.  $\bar{p} \rightarrow \bar{n} + e^- + \bar{\nu}_e$
- (2.7)

Since the Hamiltonian has to be hermitic, the hermitic conjugate of (2.6) must be added up to the Hamiltonian density in such a way, that the new part describes the opposite processes to those given in (2.7):

$$\begin{aligned}
1. \quad & p \rightarrow n + e^+ + \nu_e \\
2. \quad & p + \bar{\nu}_e \rightarrow n + e^+ \\
3. \quad & p + e^- \rightarrow n + \nu_e \\
4. \quad & \bar{n} \rightarrow \bar{p} + e^+ + \nu_e
\end{aligned} \tag{2.8}$$

To complete the analogy with the electromagnetism it is necessary to think about how particles without charge behave inside an electromagnetic field.

Equation (2.4) describes the behaviour of a current inside an electromagnetic field, but it may occur that a particle without charge but magnetic moment feels the presence of a magnetic field[25]. In such a case, the Hamiltonian density is written like  $\mathcal{H} = m(\sigma \mathbf{B})$  where  $\sigma$  is the spin of the particle,  $m$  is its magnetic moment and  $\mathbf{B}$  is the magnetic field.

In order to include the above mentioned effect into the electromagnetic Hamiltonian as well as to force it to be Lorentz invariant, we have to rewrite the Hamiltonian density of the electromagnetic situation as:

$$\mathcal{H} = m\bar{\Psi}(\gamma^\mu\gamma^\rho - \gamma^\rho\gamma^\mu)\Psi F_{\mu\rho} \tag{2.9}$$

where  $F_{\mu\rho}$  is the electromagnetic tensor.

If we extend the analogy to the weak interaction instead of having equation (2.6), we obtain:

$$\mathcal{H} = g(\bar{\Psi}_p\gamma_\mu\gamma_\rho\Psi_n)(\bar{\Psi}_e\gamma_\mu\gamma_\rho\Psi_\nu) + \text{h.c.} \tag{2.10}$$

The latter is the Hamiltonian we shall work with. A new coupling constant  $g$  has been introduced to describe how strong the interaction is.

### 2.1.2 Parity violation, left-handed neutrinos and V-A structure

When we constructed the weak Hamiltonian from the the electromagnetic case, we dealt with:

$$\mathcal{H} = \sum_n gJ^\mu(\mathbf{r}_n)L_\mu(\mathbf{r}_n)$$

where

$$J^\mu = \bar{\Psi}_p\gamma^\mu\Psi_n$$

and

$$L_\mu = \bar{\Psi}_e\gamma_\mu\Psi_\nu$$

but we could have used combinations of Dirac matrices instead of using only one operator.

Let us now write a new Hamiltonian which contains a more general operator called  $\mathcal{O}_i$ :

$$\mathcal{H}_i = g_i(\bar{\Psi}_p\mathcal{O}_i\Psi_n)(\bar{\Psi}_e\mathcal{O}_i\Psi_\nu) + \text{h.c.} \tag{2.11}$$

where the index  $i$  refers to the type of operator we are using.

It works for  $i = S, V, T, A, P$  which means scalar, vector, tensor, axial and pseudoscalar, respectively. Mathematically they are built like:

$$\begin{aligned}
 S &\longrightarrow \mathcal{O}_i = 1 \\
 V &\longrightarrow \mathcal{O}_i = \gamma_\mu \\
 T &\longrightarrow \mathcal{O}_i = \gamma_\mu \gamma_\nu \\
 A &\longrightarrow \mathcal{O}_i = \gamma_\mu \gamma_5 \\
 P &\longrightarrow \mathcal{O}_i = \gamma_5
 \end{aligned} \tag{2.12}$$

The name each operator received is due to the geometrical properties the object  $\bar{\Psi} \mathcal{O}_i \Psi$  has under Lorentz transformations.

The overall Hamiltonian is then calculated from adding up all the contributions we talked about above, such that:

$$\mathcal{H} = g \sum_i C_i \mathcal{H}_i \tag{2.13}$$

Here, the coupling constants have also been renamed to:  $g_i = g C_i$

The *sub-Hamiltonian*  $\mathcal{H}_i$  we just studied describes the so-called *even* interactions which are differentiated from another set of *odd* interactions that might occur in nature. These new interactions would be described by the following Hamiltonian:

$$\mathcal{H}_i^{odd} = g_i (\bar{\Psi}_p \mathcal{O}_i \Psi_n) (\bar{\Psi}_e \mathcal{O}_i \gamma_5 \Psi_\nu) + h.c. \tag{2.14}$$

This distinction was proposed after discovering the parity violation in weak interactions and hence a combination of both descriptions has to be mixed as follows:

$$\mathcal{H} = g \sum_i \left( C_i \mathcal{H}_i^{even} + C'_i \mathcal{H}_i^{odd} \right) + h.c. \tag{2.15}$$

where a relation of normalization is fulfilled among the 10 coupling constants:

$$\sum_i |C_i|^2 + |C'_i|^2 = 1$$

Now, if we want to take into account the possibility of non-conservation of the leptonic number, we must introduce a new set of 10 coupling constants represented by  $D_i$ ,  $D'_i$ , as well as differentiate both neutrino and antineutrino states. In that case, the Hamiltonian is rearranged:

$$\mathcal{H} = g \sum_i \bar{\Psi}_p \mathcal{O}_i \Psi_n \bar{\Psi}_e \mathcal{O}_i \left[ \left( C_i + C'_i \gamma_5 \right) \Psi_\nu + (D_i + D'_i \gamma_5) \gamma_5 \Psi_\nu^C \right] + h.c. \tag{2.16}$$

From this, which is the most general case, we may want to reduce the Hamiltonian to a more practical one.

Experiments tell us, that the V,A,S,T,P currents are not present in nature with the same frequency. As a matter of fact, the operators for S,T and P can be totally suppressed and operators for V and A are sufficient for explaining all known experimental results.

Thus, by including in Eq.(2.16) only the currents that are according to experiments not negligible (Vector and Axialvector), we finally get to an appropriate Hamiltonian density:

$$\mathcal{H} = g \frac{C_V}{\sqrt{2}} [\bar{\Psi}_p \gamma_\mu (1 - \lambda \gamma_5) \Psi_n] [\bar{\Psi}_e \gamma_\mu (1 - \gamma_5) \Psi_\nu] \quad (2.17)$$

with  $\lambda = C_A/C_V$  and  $C_A^2 + C_V^2 = 1$ .

### 2.1.3 Calculation of the decay probability

We just found the Hamiltonian of the interaction with which we can calculate physical observables. For us, the most useful observable is the decay probability of the transition (2.1).

By using perturbation theory it is possible to deduce Fermi's Golden rule, which relates the transition probability, the matrix element of the transition, and the density of final states:

$$\frac{d^2 N}{dE dt} = \frac{2\pi}{\hbar} |M|^2 \frac{dn}{dW} \quad (2.18)$$

Here, the quantity  $\frac{dn}{dW}$  represents the number of final states in the interval of energy  $dW$ . The transition matrix element  $|M|^2$  must be calculated by integrating the Hamiltonian of the interaction over the nuclear volume  $d\tau$  once per nucleon, that is:

$$|M|^2 = \sum_{k=1}^A \int \mathcal{H} d\tau_k \quad (2.19)$$

The number of possible final states is estimated by taking a differential volume in the phase space of the electron and neutrino:

$$dn = \frac{1}{\hbar^6} p_e^2 dp_e p_\nu^2 dp_\nu d\Omega_e d\Omega_\nu$$

Here,  $d\Omega_e$  and  $d\Omega_\nu$  denote the solid angles in which the electron and neutrino are emitted.

It only remains to compute the decay probability to solve Eq.(2.19). For that purpose, a Hamiltonian density has to be chosen. Let us check the solution when the Hamiltonian we already derived (2.17) is put into (2.19). A detailed discussion of this procedure can be found for instance in [25]:

$$\frac{d^2 N}{dE dt} = K \cdot p_e \cdot E_{tot} \cdot \epsilon \cdot \sqrt{\epsilon^2 - m_\nu^2} \cdot \theta(\epsilon - m_\nu) \quad (2.20)$$

where  $K$  is a set of fundamental constants  $K = G_F^2 \frac{\cos^2 \theta_C |M|^2}{2\pi^3 \hbar^7 c^5}$ . All the energies and masses are given in units of  $m_e c^2$  and momenta in units of  $m_e c$ .

$G_F$  is the Fermi constant,  $\theta_C$  is the Cabibbo angle,  $E_{tot}$  denotes the total energy of the electron whereas  $p_e$  denotes its momentum,  $\epsilon$  refers to the total energy of the neutrino and  $m_\nu$  to its mass.

## 2.2 Beyond the Standard Model

Three different theoretical approaches beyond the Standard Model have been checked with the purpose of investigating their influence on the result of the KATRIN experiment. The first one takes into account the possibility of having more than the Vector and Axialvector currents in the weak Hamiltonian. The second one deals with neutrinos as tachyons since some years ago this hypothesis came up to explain the fact that some experiments concluded a negative neutrino mass. Finally, we shall test the influence of the violation of Lorentz invariance since special relativity might only be right at low energies.

### 2.2.1 Right-handed neutrinos and T, S currents

Let us write the differential spectrum (decay rate) of the tritium beta decay by including more coupling constants than those that generate the Vector and Axialvector currents. This decay rate can be calculated as a correction to the Standard Model (see [26, 25, 27]). Thus, we must keep in mind equation (2.20):

$$\left[ \frac{d^2 N}{dt dE} \right]_{\text{NonV-A}} = \left[ \frac{d^2 N}{dt dE} \right]_{\text{SM}} \left( 1 + a \frac{p_e}{W_e} \cos \theta + b \left( \frac{m_e}{W_e} - d' \frac{m_\nu}{W_\nu} \right) - c' \frac{m_e}{W_e} \frac{m_\nu}{W_\nu} \right) \quad (2.21)$$

Here,  $W_e$  and  $W_\nu$  denote the electron and neutrino total energy, respectively,  $\theta$  is the angle between the outgoing neutrino and electron. The new constants  $a, b, d', c'$  resume the influence of the Tensor and Scalar currents.

As we want to know the total number of decays per unit of time and energy, we must integrate over the angle  $\theta$  between neutrino and electron from 0 to  $2\pi$ , but:

$$\int_0^{2\pi} \cos \theta d\theta = 0$$

Thus, the contribution of the term with the constant  $a$  is totally negligible. At this point it is interesting to point out that the constant  $b$  has a structure like  $A/B$  with a very small  $A$  whereas the constant  $d'$  is like  $C/A$  with the same  $A$ . This means that the term with only  $b$  is neglected, although combinations of  $bd'$  must be taken into account. Thus, equation (2.21) can be rewritten as:

$$\left[ \frac{d^2 N}{dt dE} \right]_{\text{NonV-A}} = \left[ \frac{d^2 N}{dt dE} \right]_{\text{SM}} \left( 1 - bd' \frac{m_\nu}{W_\nu} - c' \frac{m_e}{W_e} \frac{m_\nu}{W_\nu} \right) \quad (2.22)$$

With the purpose of having the minimum number of new parameters, we shall make one last approximation in order to gather all the new coupling constants into one. The approximation consists in considering the term  $\frac{m_e}{W_e}$  as a constant within the energy range we work, very near the endpoint where the electron energy is maximum.

Thus, let us check whether the approximation is reasonable in the energy range we are interested in, i.e. where the electron kinetic energy ranges from 18525 eV to 18575 eV:



$$\kappa \equiv \frac{m_e}{W_e} \cong [0.9642, 0.9643]$$

Hence, the differential spectrum of tritium beta decay with non V-A physics can be written as:

$$\left[ \frac{d^2 N}{dt dE} \right]_{\text{NonV-A}} = \left[ \frac{d^2 N}{dt dE} \right]_{\text{SM}} \left( 1 + \frac{m_\nu}{W_\nu} (-bd' - c' \kappa) \right) \quad (2.23)$$

Finally, for the sake of simplicity the constants set  $-bd' - c' \kappa$  is grouped into  $\delta \equiv -bd' - c' \kappa$  to obtain:

$$\left[ \frac{d^2 N}{dt dE} \right]_{\text{NonV-A}} = \left[ \frac{d^2 N}{dt dE} \right]_{\text{SM}} \left( 1 + \delta \frac{m_\nu}{W_\nu} \right) \quad (2.24)$$

Thus, we have introduced only one new parameter  $\delta$  into the interaction. This  $\delta$  contains the non V-A physics we want to check since it has combinations of Tensor-Axial and Scalar-Vector currents.

The physical boundaries of  $\delta$  go from  $[-1, +1]$  although the present experimental limits are set in  $[-0.28, +0.28]$ . By means of Monte Carlo simulations we shall elucidate whether the KATRIN experiment will or will not improve these limits.

### 2.2.2 Tachyonic neutrinos

The hypothesis of tachyonic neutrinos came up some years ago with the goal of explaining the unexpected decay rate at the very end of the spectrum that led to negative values of  $m_\nu^2$ . A tachyon is a particle that is claimed to move faster than light. Therefore, in order to deal with tachyons, the theory of relativity is not sufficient since causality would be violated. A new causal theory of tachyons has been recently proposed [28] and calculations for tritium beta decay have been done by considering neutrinos as tachyons.

Within this theory the energy-momentum relation changes to  $E^2 - \mathbf{p}^2 = -\kappa^2$  where  $\kappa$  is the so-called *tachyonic mass* [28] such that the value of  $\kappa$  is always positive. Depending on whether the *helicity* or *chirality* coupling is treated into the lagrangian of the interaction, two differential spectra are derived for tritium decay.

-Helicity coupling:

$$\frac{d^2 N}{dE dt} = \frac{G_F}{2\pi^3} \left[ \kappa m_e (1 - 3g_A^2) + (1 + 3g_A^2) E_{\text{tot}} \sqrt{\epsilon^2 + \kappa^2} \right] p_e \sqrt{\epsilon^2 + \kappa^2} \quad (2.25)$$

-Chirality coupling:

$$\frac{d^2 N}{dE dt} = \frac{G_F}{4\pi^3} (1 + 3g_A^2) E_{\text{tot}} p_e \left[ \epsilon^2 + \epsilon \sqrt{\epsilon^2 + \kappa^2} + \kappa^2 \right] \quad (2.26)$$

In both equations all energies are dimensionless as they are written in units of  $m_e c^2$ .  $\epsilon$  can be extracted from the electron energy, by doing  $\epsilon = E_0 - E$ .  $E_0$  still represents the

endpoint energy and  $E$  the electron kinetic energy.  $E_{tot}$  is the total electron energy and  $p_e$  the electron momentum. A Heaviside function that does not allow any decay when  $E > E_0$  is understood. Not both of these equations have been studied in this work. The chirality coupling option was chosen to analyze a tritium beta decay spectrum ruled by tachyonic physics.

### 2.2.3 Violation of Lorentz invariance

Another explanation for the unexpected number of counts near the endpoint that leads to a negative value of  $m_\nu^2$  is the violation of Lorentz invariance. This hypothesis considers Lorentz invariance as a physical law that works properly only for low energies. Thus, the relativistic energy-momentum relation is not suitable and must be changed [29]. Indeed, concerning neutrinos, a new relation is found such that fulfills:

$$E_\nu^2 = p_\nu^2 + m_\nu^2 + 2\lambda|\mathbf{p}_\nu| \quad (2.27)$$

where  $\mathbf{p}_\nu$  is the neutrino three-momentum and  $\lambda$  is a new parameter that breaks the invariance and fulfills  $\lambda \geq 0$ . A new tritium beta decay spectrum is found by using (2.27):

$$\frac{d^2N}{dtdE} = K p_e E_{tot} \epsilon \left[ (\epsilon^2 + \lambda^2 - m_\nu^2)^{\frac{1}{4}} - \frac{\lambda}{(\epsilon^2 + \lambda^2 - m_\nu^2)^{\frac{1}{4}}} \right]^2 \theta(\epsilon - m_\nu) \quad (2.28)$$

## 2.3 Energy considerations

As we need to reproduce the tritium beta decay spectrum with high accuracy, every possible correction to the ideal case must be considered. The most important ones regard possible changes on the energy of the electron. For instance, through their pathway from the decay to the detector, electrons might lose energy due to *bremssstrahlung*. Thus, the electrodynamics of the electron must be computed in order to avoid systematic errors. Another necessary process to take into account which involves energy changes, is the electrostatic interaction between the outgoing  $\beta$ -particle and the daughter molecule. This interaction is not included in the Hamiltonian deduced before. Therefore, the decay rate must be somehow corrected by the connection between  $\beta$ -particle and molecule. In addition, it might happen that, when decaying, the daughter molecule takes away some energy from the electron to stay in an excited state. This extra energy can be stored in the molecule either inside the electronic configuration of the atoms of the molecule or inside the whole molecule in a rotational or vibrational state. Fortunately, for tritium, after beta decay, the probabilities of leaving the molecule on different states as well as their corresponding energies is already available in the scientific literature [30, 31]. There are some corrections that are not included in this work, for example, change of the electron energy due to the molecular recoil, or due to the thermal molecular motion. Consequently, it is assumed that the tritium molecule is totally stationary before and after the decay.

### 2.3.1 Radiative and electrostatic corrections to the spectrum

It is very important to distinguish between the actual energy of the electron at the moment of decaying and the one that will be measured. A proper theoretical approach to the

spectrum would consider at least the biggest sources of changes in the energy and therefore deviations from the ideal case.

For that purpose, the electrostatic interaction between the outgoing electron and the daughter molecule is calculated by means of the well-known Fermi function [32]. On the other hand, energy losses due to electrodynamic considerations are computed by means of the Sirlin function [33].

Both functions can be described with different approximations depending on the energy range of the spectrum we are interested in. A very suitable one which works very well in the last 50 eV of the spectrum is found in [32]. These two functions are gathered into only one correction:

$$F(\epsilon, y) = 2\pi y [1 - \exp(-2\pi y)]^{-1} \left(1 + \frac{2\alpha\beta}{3\pi}\right) \log(2\epsilon) \left(a - \frac{b}{\beta}\right) \quad (2.29)$$

where  $a$  and  $b$  are empirical coefficients (see Appendix A).  $\epsilon$  is the total neutrino energy in units of  $m_e c^2$ ,  $\alpha$  is the fine structure constant,  $y = \frac{Z\alpha}{\beta}$ ,  $Z$  is the number of protons in the daughter atom and  $\beta$  is the relativistic factor that considers the electron energy and velocity.

The corrected spectrum has to be one of the spectra we explained above multiplied by the function in (2.29). For instance, by taking the Standard Model spectrum:

$$\frac{d^2 N}{dE dt} = K \cdot F(\epsilon, y) p_e \cdot E_{tot} \cdot \epsilon \cdot \sqrt{\epsilon^2 - m_\nu^2} \cdot \theta(\epsilon - m_\nu) \quad (2.30)$$

### 2.3.2 Excitation of the daughter molecule

As tritium is a molecule formed by two atoms, it might happen that when one of the atoms decays some energy gets stored into the molecule as rotational or vibrational energy. Of course, it is also probable that the energy stays in the electronic configuration of one of the atoms, i.e., as excitation energy. If these corrections were not taken into account the systematic error on the energy of the outgoing electron would be huge.

$\beta$ -decay can be treated in sudden approximation, i.e., the change of the nucleus is fast compared to the time the electronic shell can rearrange itself. The transition from the wave function of the ground state of T to the wave function of  ${}^3\text{He}^+$  can be very precisely calculated. The same holds for the excitation of the rotational-vibrational states of the  $\text{T}_2$  molecule. Therefore, the possible final states are very well-known, both the energy and the probability of each state. This means that the systematic error is not present any longer as soon as the calculation for distribution of states is carried out with sufficient precision.

This calculation was done some years ago [30, 31] and got recently even more improved. Thus, let us put this last correction into (2.30):

$$\frac{d^2 N}{dE dt} = K \sum_i W_i \cdot F(\epsilon_i, y) p_e \cdot E_{tot} \cdot \epsilon_i \cdot \sqrt{\epsilon_i^2 - m_\nu^2} \cdot \theta(\epsilon_i - m_\nu) \quad (2.31)$$

Where  $W_i$  is the probability of decaying to the state  $i$  and now the total neutrino energy  $\epsilon$  becomes  $\epsilon_i = E_0 - E_e - E_i$  and  $E_i$  is the energy stored in the molecule.

The same energy corrections must be included in the extended models. By doing so, the following spectra are obtained:

Non V-A:

$$\frac{d^2 N}{dE dt} = K \sum_i W_i \cdot F(\epsilon_i, y) p_e \cdot E_{etot} \cdot \epsilon_i \cdot \sqrt{\epsilon_i^2 - m_\nu^2} \cdot \theta(\epsilon_i - m_\nu) \left[ 1 + \delta \frac{m_\nu}{\epsilon_i} \right] \quad (2.32)$$

here  $K$  is the usual  $K = G_F^2 \frac{\cos^2 \theta_C |M|^2}{2\pi^3 \hbar^7 c^5}$ .

Tachyonic neutrinos, helicity coupling:

$$\frac{d^2 N}{dE dt} = K \sum_i W_i \cdot F(\epsilon_i, y) p_e \left[ \kappa m_e (1 - 3g_A^2) + (1 + 3g_A^2) E_{etot} \sqrt{\epsilon_i^2 + \kappa^2} \right] p_e \sqrt{\epsilon_i^2 + \kappa^2} \quad (2.33)$$

here  $K = \frac{G_F}{2\pi^3}$ .

Tachyonic neutrinos, chirality coupling:

$$\frac{d^2 N}{dE dt} = K \sum_i W_i \cdot F(\epsilon_i, y) E_{etot} p_e \left[ \epsilon_i^2 + \epsilon_i \sqrt{\epsilon_i^2 + \kappa^2} + \kappa^2 \right] \quad (2.34)$$

here  $K = \frac{G_F}{4\pi^3} (1 + 3g_A^2)$ .

Violation of Lorentz invariance:

$$\frac{d^2 N}{dE dt} = K \sum_i W_i \cdot F(\epsilon_i, y) p_e E_{etot} \epsilon_i \left[ (\epsilon_i^2 + \lambda^2 - m_\nu^2)^{\frac{1}{4}} - \frac{\lambda}{(\epsilon_i^2 + \lambda^2 - m_\nu^2)^{\frac{1}{4}}} \right]^2 \theta(\epsilon_i - m_\nu) \quad (2.35)$$

here  $K$  is also the usual  $K = G_F^2 \frac{\cos^2 \theta_C |M|^2}{2\pi^3 \hbar^7 c^5}$ .

## Chapter 3

# The KATRIN experiment

KATRIN (**K**arlsruhe **T**ritium **N**eutrino mass experiment) is the next generation experiment on the measurement of the electron neutrino mass by means of a very accurate analysis of the tritium beta decay spectrum.

As commented in Ch. 1, KATRIN aims to improve the present experimental limit on the electron neutrino mass by a factor of 10. In case of having an almost vanishing neutrino mass the new upper limit will be 0.2 eV.

The long history of attempts to determine the electron neutrino mass from tritium beta decay is justified by the fact that tritium presents extraordinary features for this kind of measurement. Its endpoint  $E_0$  is around 18600 eV, which is a very low value in comparison with other nuclei. This means that the retarding potential we have to generate for filtering electrons is achievable with an uncertainty and stability of  $\leq 1$  eV. In addition, its half life is relatively short,  $t_{1/2} = 12.3\text{a}$ , so that it is not difficult to get good statistics.

Moreover, tritium is one of the simplest molecules in nature. Consequently, theoretical calculations of excited states of the molecule can be carried out with high accuracy.

Let us study how the electron neutrino mass influences the tritium beta decay spectrum. When working within the Standard Model, the differential decay follows the already mentioned formula in (2.31), which includes corrections to the ideal decay and is, of course, sensitive to changes of the neutrino mass. Despite of knowing this precise model, and for the sake of simplicity, let us examine the neutrino mass influence by *switching off* the role of the Fermi and Sirlin functions and the excited states of the daughter molecule. Thus, going back to:

$$\frac{d^2N}{dEdt} = K \cdot p_e \cdot E_{tot} \cdot \epsilon \cdot \sqrt{\epsilon^2 - m_\nu^2} \cdot \theta(\epsilon - m_\nu) \quad (3.1)$$

From this description, it is expected to find some non vanishing function in those points that fulfill  $\epsilon > m_\nu$ , i.e., when conservation of energy is fulfilled and the total energy of the neutrino is bigger than its mass.

Given a decay endpoint, the total neutrino energy could be indirectly computed from the endpoint and the electron kinetic energy,  $\epsilon = E_0 - E_e$ . Then, it is clear that the bigger the electron energy is the smaller the neutrino energy becomes.

The actual shape of the decay was calculated by following (3.1) with these conditions:  $E_0 = 18575$  eV and  $m_\nu^2 = 0$  eV<sup>2</sup>. It is shown for the whole energy range in Fig. 3.1. However, the experiment will examine only the fastest electrons, so that only the last 50 eV of this function will be analyzed.

The factor  $\epsilon \cdot \sqrt{\epsilon^2 - m_\nu^2}$  in (3.1), which alludes to the combination of the neutrino total energy and the neutrino momentum, is responsible for the decrease of the decay probability after the maximum (around 4 keV) in the spectrum.

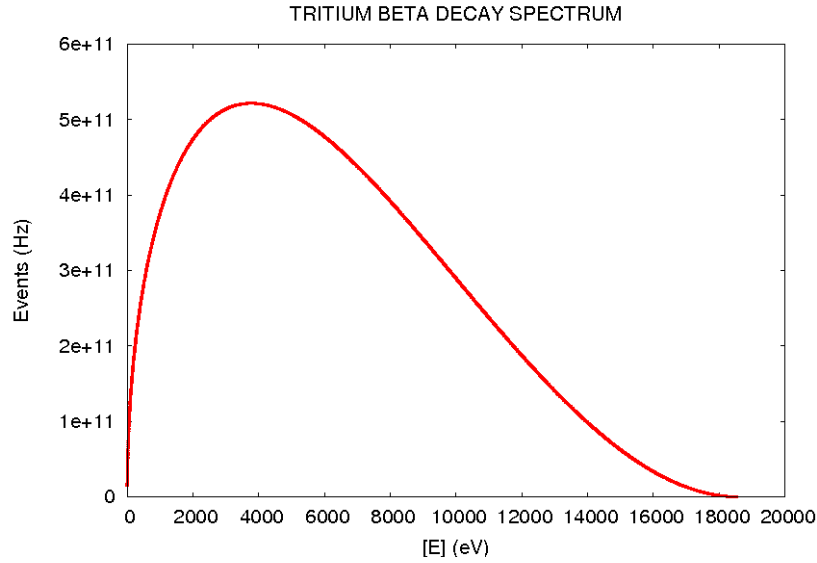


Figure 3.1: Spectral shape of the tritium  $\beta$  decay.

The influence of the electron neutrino mass on the spectrum is more evident in the vicinity of the endpoint. This influence can be seen in Fig. 3.2 where three different spectra are shown corresponding to  $m_\nu^2 = 0$  eV<sup>2</sup> (center line),  $m_\nu^2 = -3$  eV<sup>2</sup> (upper line) and  $m_\nu^2 = 3$  eV<sup>2</sup> (lower line). The endpoint is for all cases  $E_0 = 18575$  eV.

Let us remark that within the Standard Model, neutrinos with negative values of  $m_\nu^2$  do not have any physical meaning but a statistical one. The lack of physical frames for those neutrinos allows the violation of conservation of energy. To avoid this, the spectrum of  $m_\nu^2 = -3$  eV<sup>2</sup> (upper line) in Fig. 3.2 reaches the endpoint  $E_0 = 18575$  eV. How the spectrum is treated for those negative values will be explained in chapter 4.

From Fig. 3.2 it seems to be obvious that a direct measurement of the neutrino mass might be possible by detecting the point where the spectrum ends. Unfortunately, the real experimental situation turns out to be different due to the unavoidable presence of random background (BG) events. Details on the background are not subject of this work and the item will hence be treated very briefly. The remarks are based on the extensive studies of background carried out with the Mainz spectrometer [22, 24, 34]. The aim of these experiments was to locate and eliminate sources of BG. BG can originate from surfaces or come out of the spectrometer volume.

BG from surfaces:

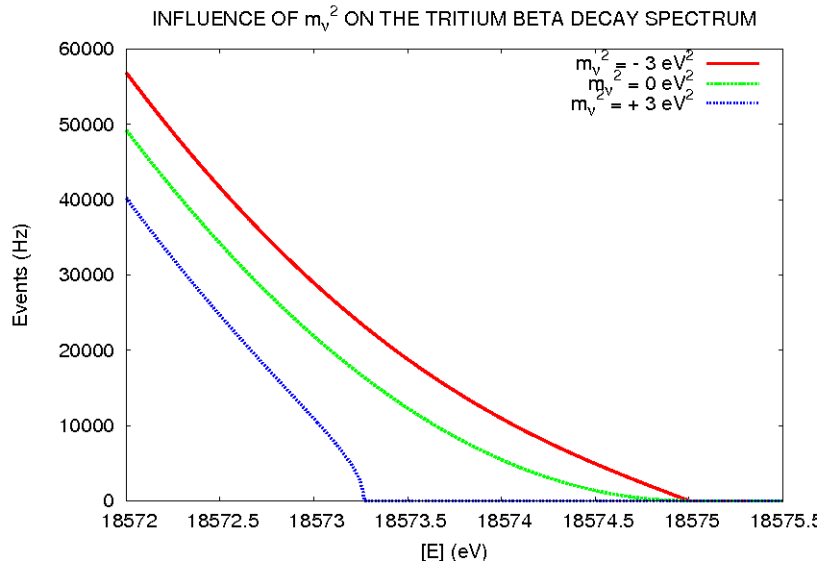


Figure 3.2: Influence of the neutrino mass on the shape of the tritium  $\beta$  decay spectrum.  $m_\nu^2 = -3 \text{ eV}^2$  (upper line),  $m_\nu^2 = 0 \text{ eV}^2$  (center line),  $m_\nu^2 = +3 \text{ eV}^2$  (lower line).

- Field emission. These processes should play a minor role in KATRIN as the spectrometer vessel is on high voltage reducing the field gradient very strongly compared to the experiment in Mainz.

- BG caused by radioactivity. Cosmic rays and environmental radioactivity induce the emission of low energy electrons from metallic surfaces. A low fraction of these electrons enters the volume of the spectrometer and may be focused onto the detector. To keep this part of the BG under control, radioactivity in the vicinity of the KATRIN spectrometer is kept low by proper selection of the materials (low activity concrete and low activity steel). The BG caused by electrons from the surface could be strongly reduced in the Mainz experiments by a grid electrode shielding the solid electrodes. Such a system will be used in the KATRIN spectrometer.

BG out of the volume:

- Decay of tritium. The partial pressure of tritium will be reduced by the differential pumping section of KATRIN to yield a background level below 1 mHz in the spectrometer volume.

- BG due to collisions with trapped electrons and residual gas. This type of BG was present in the Mainz setup but could be completely eliminated by removing the traps for electrons. The KATRIN spectrometer will be free of traps and the residual pressure will be in the XHV region of  $10^{-11}$  mbar.

We expect the BG to be in the order of 10 recorded events in 1000s. This value will be used in the simulations.

Let us introduce how the system works with real  $\beta$ -particles. Coming back to the idea of the fundamental grounds of a tritium-based experiment, it is useful to understand the general scheme of the Mainz experiment (see Fig. 3.3 [21]).

A tritium source (left part) decays producing electrons with kinetic energy from 0 eV to the endpoint value  $E_0 \approx 18600 \text{ eV}$ . Those electrons get out of the source without a

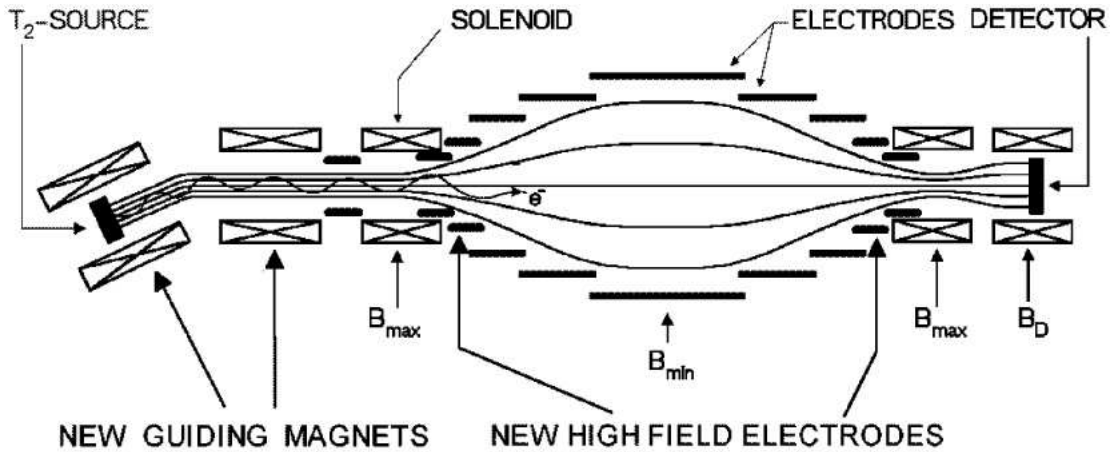


Figure 3.3: General scheme of the Mainz electron neutrino mass experiment [22].

preferred direction, that means, electrons might be detected in the whole solid angle  $4\pi$  with the same probability.

A set of superconducting magnets is responsible of selecting electrons that travel within a solid angle of  $2\pi$  (semisphere faced to the spectrometer). Then, the magnetic field guides these electrons across the spectrometer towards the detector. The natural pathway of the  $\beta$ -particles is changed by the magnetic field, but the energy remains unchanged

Only the most energetic electrons are useful for the determination of the neutrino mass. Consequently, the spectrometer must filter almost all electrons but let the fastest ones pass through. The potential barrier slows down electrons that do not have sufficient energy. This process occurs in the first half part of the spectrometer, from the decay up to the so-called *analyzing plane* (APL), where the magnetic field is minimum (see  $B_{min}$  in Fig. 3.3). Since the retarding potential is symmetrical around the analyzing plane, electrons that overcome the middle of the spectrometer are speed up again towards the detector (see Fig. 3.3, right part).

Unfortunately, the energy resolution of present detectors is not good enough for determining precisely the energy of those electrons that cross our system. Therefore, given a potential barrier, all detected events are added up as an integral.

### 3.1 MAC-E filter and the adiabatic motion of electrons

It has been already pointed out that the final *tail* of the tritium beta decay spectrum is *reconstructed* in the experiment by guiding and filtering electrons from the decay. Let us have a more detailed look at how this is experimentally done as well as at the physics underneath.

First of all, it is important to emphasize that the magnetic guiding of electrons does not change their energy. This guiding mechanism works under the adiabatic conditions, which ensure that electrons move in a cyclotron motion around the magnetic field lines meanwhile the magnetic moment  $\mu = \frac{E_{\perp}}{B}$  remains an invariant of the motion.

On every point of the electron trajectory,  $E_{\perp}$  is the energy associated to the perpendicular component of the electron momentum to the magnetic field.



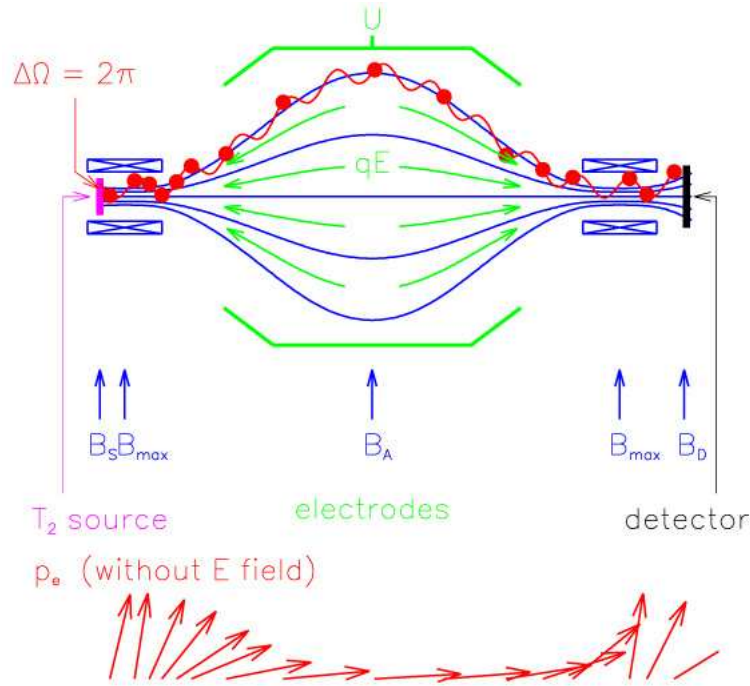


Figure 3.4: Basics of the MAC-E filter and transformation of the momentum of the electron along its pathway [22].

Around the analyzing plane one can easily visualize how the motion progress. Since  $\mu$  must be constant, and  $B$  is minimum at the analyzing plane,  $E_{\perp}$  almost vanishes in order to increase the parallel component  $E_{\parallel}$  up to its maximum value. This is a real experimental difficulty since filtration and guiding have to be done at the same time.

The motion of electrons inside the spectrometer is shown in Fig. 3.4. The vectorial representation at the bottom refers to the electron momentum which is forced to change by the magnetic field and becomes antiparallel to the electrostatic gradient at the analyzing plane. The original momentum is restored when arriving at the detector.

The value of the magnetic field in some points of the experimental setup plays a vital role for this motion to be achieved. The most determinant places and the magnetic field at these positions are:

- The tritium source,  $B_s = 3.6$  T
- The analyzing plane,  $B_A = 3 \cdot 10^{-4}$  T
- The detector,  $B_D = 3$  T
- The transport section,  $B_T = 5.6$  T
- The pinch,  $B_{max} = 6$  T

The last one is the largest magnetic field value in the experiment. The pinch can be placed anywhere between source and detector. It reflects electrons with high starting angles. In KATRIN it will be placed in front or behind the main spectrometer. The location behind has the advantage to reject part of the BG. A scheme of these values of the magnetic field is given in Fig. 3.5.

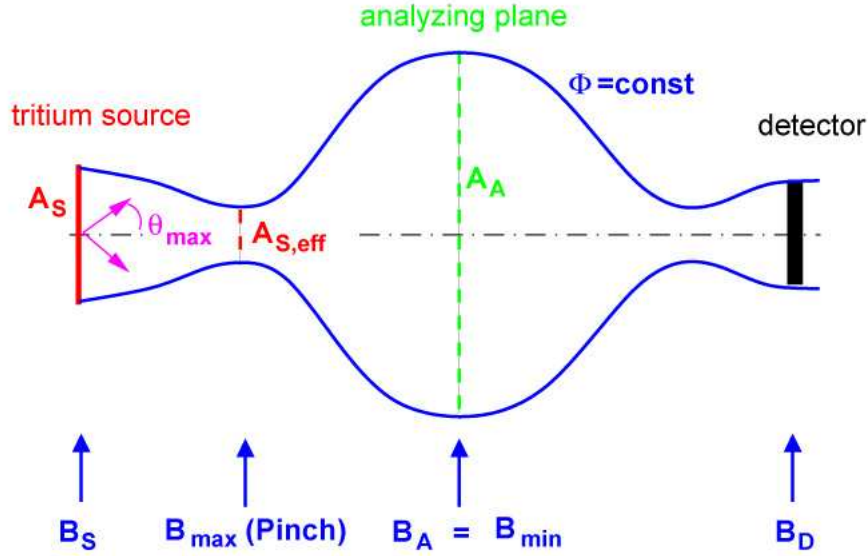


Figure 3.5: Magnetic field at relevant places of the experiment.

So, as stated before, the magnetic field is decisive for the transmission of electrons. Of course, and more intuitive, the energy of the electron and the electrostatic gradient are important as well. One can quantitatively calculate how much each of these factors contribute to the transmission. For a detailed explanation on how to proceed, see for instance [35]. The function in (3.2) is obtained for the KATRIN spectrometer. We shall refer to it as *transmission function* from now on.

$$T(E, qU_0) = \begin{cases} 0 & E - qU_0 < 0 \\ \frac{1 - \sqrt{1 - \frac{B_S(E - qU_0)}{EB_A}}}{1 - \sqrt{1 - \frac{\Delta E B_S}{EB_A}}} & E - qU_0 \geq 0 \quad \wedge \quad E - qU_0 \leq \Delta E \\ 1 & E - qU > \Delta E \end{cases} \quad (3.2)$$

Here,  $E$  is the electron energy,  $qU_0$  the retarding potential times the electron charge, the notation for magnetic fields has been explained above and  $\Delta E = \frac{E \cdot B_A}{B_{Max}}$  is the spectrometer resolution. This function gives the probability for an electron in the accepted solid angle with energy  $E$  to pass through the spectrometer. The transmission function of our spectrometer looks like Fig. 3.6.

It is important to note, that the function (3.2) is not a perfect step-function but it has some slope when going from a vanishing probability to 1. This deviation from an ideal Heaviside function yields the spectrometer resolution  $\Delta E$  which for the reference parameters of KATRIN amounts to 0.93 eV. Thus, the KATRIN setup is for sure able to detect electrons with energy  $\Delta E$  above the retarding potential.

For ease of discussion and calculation it was not mentioned that this transmission function makes sense as long as we change the retarding potential by varying the potential in the analyzing plane, otherwise, other effects must be mentioned.

Let us assume it will be done so. In such a case, function (3.2) is a good description for explaining the electron behaviour once it has reached the spectrometer. There are other substantial modifications to be considered, though. Mainly, when dealing with energy

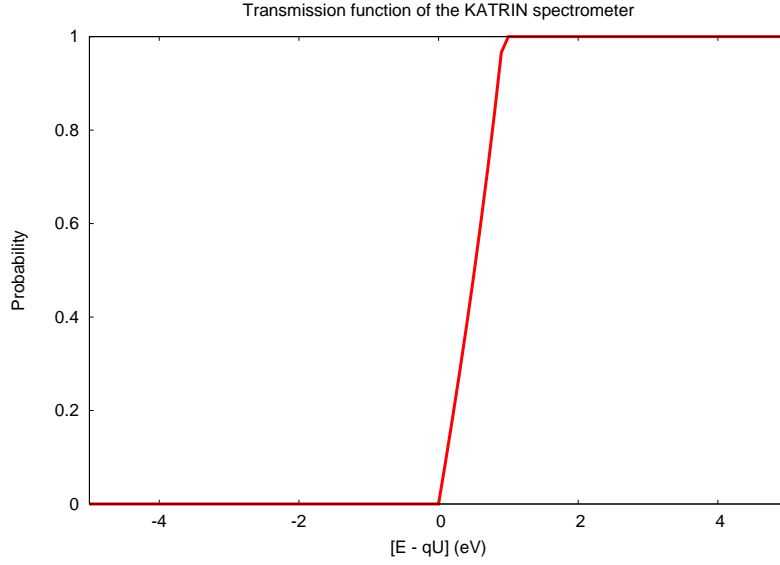


Figure 3.6: Transmission function of the KATRIN spectrometer.

losses on electrons due to scattering. These modifications will be discussed in the next section.

Now, let us come back to the interaction between the magnetic field and the electrons from the decay, since not all the necessary information is in the transmission function.

Not every decay in the source will contribute to the measurement. The reason is that only those magnetic lines that come out from the source and go through the analyzing plane onto the detector will be able to transport electrons appropriately. The source area is not fully imaged onto the detector to make sure, that electrons emitted from tritium decaying on the walls nearby do not reach the detector. The area imaged onto the detector is called the guided flux. For standard settings of KATRIN it is  $192 \text{ Tcm}^2$ , the total flux in the source is  $\sim 220 \text{ Tcm}^2$ . The cross section of the guided flux can be easily calculated by using magnetic flux conservation:

$$\int_S B_S dA_S = \int_A B_A dA_A$$

where the index  $S$  indicates source and  $A$  analyzing plane (APL).

$$B_S A_S = B_A A_A$$

$$A_S = \frac{B_A}{B_S} A_A$$

$$A_S = \frac{B_A}{B_S} \pi R_A^2 \quad (3.3)$$

Here,  $R_A = 4.5 \text{ m}$  is the radius of the analyzing plane used for the experiment. The total radius is close to  $4.9 \text{ m}$ . The grid electrode system covers about  $25 \text{ cm}$ . The remaining  $15 \text{ cm}$  to the guided flux make sure that the electric potential has reached the mean value and is only very little influenced by the details of the grid electrode.

Moreover, one should rule out those electrons that get out of the source with very large starting angles with the purpose of not counting electrons that suffer many scatterings. A maximum starting angle is set by the magnetic mirror effect:

$$\frac{\Delta\Omega}{2\pi} = 1 - \cos\theta_{max} \quad (3.4)$$

where

$$\theta_{max} = \arcsin\left(\sqrt{\frac{B_S}{B_{max}}}\right)$$

Once more, the configuration of the magnetic field selects the suitable particles. Finally, the solid angle of (3.4) refers to  $2\pi$  just to take into account the semisphere faced to the spectrometer.

In Ch. 2, we discussed different approaches to the tritium beta decay. The differential spectrum then calculated must now include the restrictions we just showed. We shall insert the information of (3.4) and (3.3) into the constant  $K$  to describe properly the number of measured events.

For example, by choosing the Standard Model approach given in (2.31), the decay rate will be artificially altered by rewriting  $K$ :

$$\frac{d^2N}{dEdt} = K \sum_i W_i \cdot F(\epsilon_i, y) p_e \cdot E_{tot} \cdot \epsilon_i \cdot \sqrt{\epsilon_i^2 - m_\nu^2} \cdot \theta(\epsilon_i - m_\nu) \quad (3.5)$$

with  $K = G_F^2 \frac{\cos^2\theta_C |M|^2}{2\pi^3 \hbar^7 c^5} \frac{1 - \cos\theta_{max}}{2} \frac{B_A}{B_S} \pi R_A^2$

### 3.2 Windowless gaseous tritium source and electron scattering

We are already able to characterize the interaction between electrons and the spectrometer. To avoid systematic uncertainties one should estimate as well the interaction between electrons and the tritium source.

The KATRIN tritium source, called Windowless Gaseous Tritium Source, WGTS in short, is expected to deliver a high luminosity and a low systematic uncertainty. Its design is based on the experience of the Troitsk experiment [36].

Basically, this source is a 10 m long cylindrical tube of 90 mm diameter (see Fig. 3.7) filled with gaseous molecular tritium, which has to be continuously reinjected and pumped out in order to achieve the optimum column density  $\rho d = 10^{17}$  atoms/cm<sup>2</sup> within an uncertainty of 0.1% [22], since no material window is allowed to stop the tritium gas flow. Such level of accuracy relies on several variables, primarily the tube temperature and the rate of injection.

The system is kept at a temperature of 27 K. The injection of tritium is carried out at a pressure of  $3.35 \cdot 10^{-3}$  mbar and it takes place at the center of the cylinder, such that tritium is spread to yield an inhomogeneous density along the longitudinal axis.

The injected molecular tritium is highly pure, although some other species coexist together inside the WGTS. Among them, H<sub>2</sub>, HD, HT, D<sub>2</sub> and DT are the most important

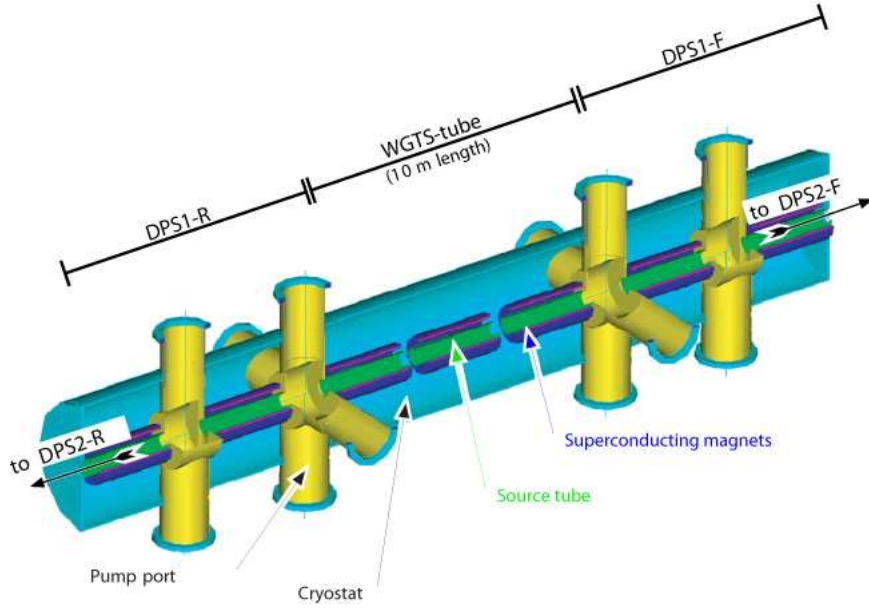


Figure 3.7: WGTS and pump stations DPS1-R, DPS1-F of KATRIN.

ones. The radiopurity of tritium and deuterium amounts to  $\epsilon_T = 0.95\%$  and  $0.05\%$ , respectively. These ratios will transform for molecular species into 90% for  $T_2$  and the missing 10 % for  $DT$  and  $D_2$ .

When tritium decays it generates  $\beta$ -particles that have to travel through some fragment of the WGTS system before getting into the spectrometer. During this flight, there is a non vanishing probability of inelastic scattering by which the electron loses some energy. This process could yield misleading results when measuring the spectrum by means of the method explained before, due to the fact that the spectrometer, i.e. the transmission function, would act over electrons with erroneous energy.

In order to solve this problem and to avoid the systematic uncertainty, one has to calculate the scattering probability as well as the energy loss [37]. The following function in (3.6) returns the probability for an electron to lose the amount of energy  $k$  in a dispersion process:

$$f(k) = \begin{cases} A_1 \exp\left(-\frac{2(k-k_1)^2}{\omega_1^2}\right) & k \leq k_c \\ A_2 \frac{\omega_2^2}{\omega_2^2 + 4(k-k_2)^2} & k \geq k_c \end{cases} \quad (3.6)$$

In the latter function,  $A_1, A_2, k_1, k_2, \omega_1, \omega_2$  are empirical values and  $k_c$  was explicitly calculated to guarantee the continuity of  $f(k)$ . Their actual values can be found in Appendix A.

Furthermore, one electron could suffer not only one scattering process but many of them, so that probabilities of several scatterings must be estimated. Electrons with kinetic energy within the region of interest, that is from 50 eV below the endpoint up to the endpoint, i.e. from 18550 eV to 18600 eV, could have four scatterings at most [37]. A

fifth dispersion would take from the electron sufficient energy as to be retarded by the spectrometer.

Let us name  $P_0, P_1, P_2, \dots$  the probability of zero-scattering, one-scattering, and so on (see Appendix A). It is important to emphasize that these probabilities were calculated without taking into account the position in which the decay occurs inside the WGTS.

To calculate the probability of losing an amount of energy  $k$  in, for instance, a two-scattering process, one needs to convolute (3.6):

$$(f \otimes f)(k) = \int_0^{E/2} f(k) f(k - k') dk'$$

Here, the symbol  $\otimes$  stands for the convolution operation and  $E$  is the energy of the electron at the moment of decaying.

The interaction between a  $\beta$ -particle with energy  $E$ , the WGTS system and the spectrometer must include the transmission function, probabilities of 0-scattering up to 4-scattering and their respective energy losses. This interaction can be summarized as follows:

$$f_{res}(E, qU) = \int_0^{E/2} T(E - k, qU) \{P_0 \delta(k) + P_1 f(k) + P_2 (f \otimes f)(k) + \dots\} dk \quad (3.7)$$

Function (3.7), called *response function* from now on, will describe mathematically the experimental setup. Since the summation over the probabilities of scattering in (3.7) goes on up to  $P_4$ , the terms with  $P_3$  and  $P_4$  are convoluted convolutions and so forth.

Given a retarding potential  $qU$ , the magnetic fields as in (3.2) and an electron which gets out of the decay with energy  $E$ , the response function gives us the probability that this electron has to reach the detector after having suffered some scattering processes in the source and having gone across the spectrometer.

The shape of function (3.7) is shown in Fig. 3.8 where it is possible to observe how the probability of transmission grows up as the energy of the electron increases. The plateau at the very beginning can be identified with those electrons that do not have sufficient energy to pass through the system after having one scattering. Thus, detected electrons with that energy will cross the setup without any scattering.

The functionality of this tool is twofold:

- On the one hand, it serves to characterize the interaction between electrons and the spectrometer
- and on the other, it solves the systematic uncertainty of the energy loss inside the WGTS, meanwhile reproduces the relation between electrons and the tritium source.

### 3.3 Background events

The origin of background has been extensively studied by the Mainz group [35, 24, 23, 38]. A very detailed summary is given in the KATRIN Design Report [22].

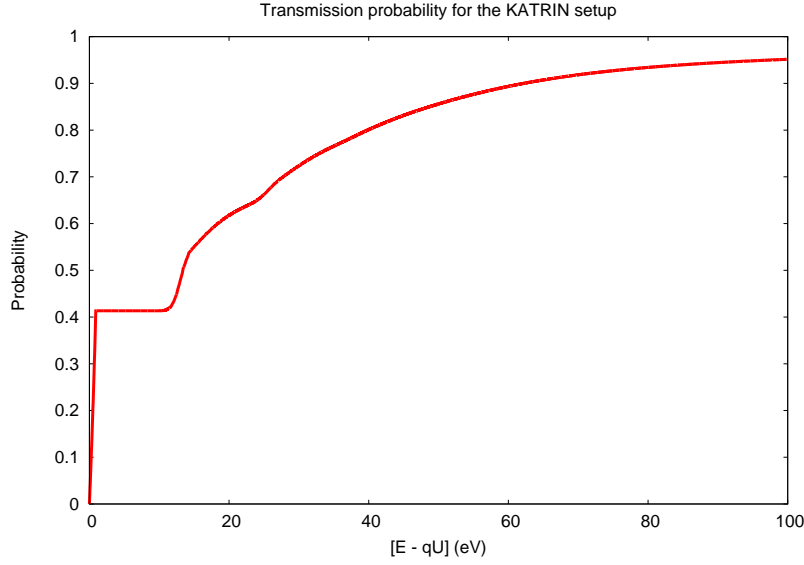


Figure 3.8: Response function for the KATRIN setup.

In summary, we can say that ionization of residual gas can be completely suppressed by working with ultrahigh vacuum  $p < 10^{-10}$  mbar if no traps for electrons are present in the spectrometer. The large number of electrons created at the surface of the electrodes (in KATRIN on the wall of the spectrometer vessel) are almost completely shielded from reaching the detector by the magnetic field.

A very important improvement in this shielding was achieved by introducing a set of electrodes formed by thin wires covering only 3% of the surface. By a guard potential these electrodes suppress the electrons from the massive electrodes and reduce the background. In the Mainz experiment this method led to an enhancement in the decrease of background of a factor 20 [24]. The background rate due to the environmental radioactivity amounts to  $< 2$  mHz. The design parameters for KATRIN establish an expected average background rate of:

$$\Gamma_b = 10 \text{ mHz}$$

which is in agreement with the experience from Mainz and Troitsk. Nevertheless, there exists no way to estimate analytically the exact quantity and therefore we have to rely on the experience with the Mainz and Troitsk experiments.

This rate seems to be independent of the applied retarding potential within the energy *window* we are interested in. In consequence, the spectrometer does not act as an integrating filter when dealing with background electrons.

### 3.4 Measurement of the decay rate

Let us gather all the experimental features previously explained in order to calculate the number of detected electrons. The experimental properties already inserted into the spectrum are:

- The measurement was restricted to those electrons which travel within a given solid angle.

- Only the decays *placed* in points of the source that can be potentially guided by magnetic lines which end in the detector were chosen.

Other experimental characteristics are:

- The number of atoms in the source. In section 3.2 it was shown that the column density of the tritium source is  $\rho d = 10^{17}$  atoms/cm<sup>2</sup>.

- The isotopic purity of the source. It was pointed out that the tritium source is 95% pure,  $\epsilon_T = 0.95$ <sup>1</sup>.

- The detector used in KATRIN has a 90% efficiency,  $\epsilon_d = 0.9$ .

All this information can be gathered into the constant  $K$  in the differential decay equation, since the corrections we just talked about are independent of the energy:

$$\frac{d^2 N}{dE dt} = K \sum_i W_i \cdot F(\epsilon_i, y) p_e \cdot E_{tot} \cdot \epsilon_i \cdot \sqrt{\epsilon_i^2 - m_\nu^2} \cdot \theta(\epsilon_i - m_\nu) \quad (3.8)$$

$$\text{with } K = G_F^2 \frac{\cos^2 \theta_C |M|^2}{2\pi^3 \hbar^7 c^5} \frac{1 - \cos \theta_{max}}{2} \frac{B_A}{B_S} \pi R_A^2 \rho d \epsilon_T \epsilon_d.$$

Thus, we have considered not only the fundamental decay but the real experiment by counting the number of molecules, the isotopic purity, the detector efficiency, the accepted solid angle, and the proper section of the source.

On the other hand, the interaction between electrons and the setup was given in (3.7). Now, let us simulate the measured rate by integrating the number of electrons that overcome the spectrometer and hit the detector as well as considering the rate of background. If one sets the retarding potential in  $qU$  during the time  $t_U$  and considers that the background is independent of the time, during the time  $t_U$  the measured rate would be:

$$S(qU, t_U) = t_U \left( \int_{qU}^{E_0} f_{res}(qU, E) \frac{d^2 N}{dt dE} dE + \Gamma_b \right) \quad (3.9)$$

This way of calculating the measured rate will be the starting point to perform Monte Carlo simulations of KATRIN as we shall explain in the following chapter.

---

<sup>1</sup>Mixed molecules containing tritium are of concern as they have slightly different rotational vibrational states. They may be included at a later stage when their fraction is determined by Raman laser spectroscopy.



## Chapter 4

# Simulations of the KATRIN experiment

A simulation package which reproduces the KATRIN experiment under the Monte Carlo (MC) method point of view has been developed. Such a tool enables us to create simulated experimental data by following a physical model. Later on, the experimental information can be fitted by choosing either the same theoretical model or a different one. When creating the experimental data with its statistical fluctuations, some hypothetical values for the free parameters of the experiment must be established. After fitting these data, the MC results should in principle look like Gaussian distributions for each free parameter, whose center will provide the value yielded by the experiment. Their standard deviations will serve to calculate the sensitivity of the experiment for each parameter. This method includes possible statistical fluctuations which are based on the fact that a real independent measurement would produce different results if repeated. The determined measurement would therefore be the mathematical weighted average over all measurements.

This simulation tool has been programmed (see Appendix B) with the C++ programming language. The final version interacts very often with a package already made by the GNU project. To reach the most efficient performance, some numerical computing techniques and external algorithms have been tested, mainly the ones included in the MINUIT package of CERN and the GSL (GNU scientific library). Despite of having several tools for integrating mathematical functions in the GSL library, the Bode's rule algorithm for integration was specifically implemented for convenience. Its precision has been checked to avoid any unpleasant behaviour. The package can be easily run on any LINUX operating system with the standard g++ compilers and the installation of the GSL library. Some input files for the KATRIN setup are needed, basically the data for the organization of the measurement points (retarding potentials  $qU$ , measurement times  $t_U$ ) and the information of the final state distribution (FSD) of the daughter molecule in the  ${}^3\text{T}(\beta^-){}^3\text{He}$  decay.

### 4.1 Calculation procedure

The information which will be explained in the following concerns how to construct the KATRIN MC simulation. The method is common for all problems mentioned previously, either SM or Non SM treatments of the tritium beta decay. For the sake of simplicity and generalization, the differential decay distribution in this section will be represented by  $\frac{d^2N}{dEdt}$  which could refer to any theoretical description of the spectrum.

As stated before, it is possible to create simulated experimental data by integrating the decay spectrum and considering the interaction between the  $\beta$ -particles, the WGTs source, and the spectrometer. Some background rate has to be taken into account as well. Coming back to that idea, by setting the retarding potential to the value  $U$  during the time  $t_U$  and some background rate  $\Gamma_b$  (usually 10 mHz), the expected number of detected events would be:

$$S(qU, t_U, p_1, \dots, p_q) = t_U \left( \int_{qU}^{E_0} f_{res}(qU, E) \frac{d^2 N}{dt dE} dE + \Gamma_b \right) \quad (4.1)$$

where the variables on which the quantity  $S$  depends are explicitly written. Some of them have to be understood as variables of the decay distribution  $\frac{d^2 N}{dt dE}$ , some others allude to experimental features like background. The periods are purposely emphasized to remark that  $S$  could be created by considering different models, since, the number of free parameters and the free parameters themselves are different for each model.

The simulated data of Eq. (4.1) is lacking the intrinsic statistical behaviour of a real measurement. It is known that in an experiment the repeated measurements of the same magnitude disagree with each other by the influence of intrinsic unavoidable factors. Experience tells us, that these measurements are almost always distributed around the real physical value in Gaussian distributions centered on it. Thus, in order to produce properly simulated experimental data, it sounds reasonable to spread randomly the quantity in (4.1) in a Gaussian distribution centered on itself. The standard deviation chosen to spread  $S$  is  $\sqrt{S}$ . Thus, a measurement in which statistical fluctuations are considered could for instance be:

$$M_{exp}(qU, t_U, p_1, \dots, p_q) = S(qU, t_U, p_1, \dots, p_q) + \text{Randomizer} \left( \sqrt{S(qU, t_U, p_1, \dots, p_q)} \right) \quad (4.2)$$

where  $qU$ ,  $t_U$  and  $p_1 \dots p_q$  are the variables on which  $S$  depends and Randomizer is an algorithm that spreads  $S$  with a standard deviation  $\sqrt{S(qU, t_U, p_1, \dots, p_q)}$ . It is interesting to remark that all algorithms like this need an initializer number (seed) after which the algorithm generates the same sequence of so-called quasi-random numbers. Therefore, these algorithms rely very strongly on the seed. To avoid any bias on the MC simulation, it would be very appropriate to break this dependence. A very useful trick in this sense is to use the “time.h” library with which the seed is started up with the number of seconds gone by since 1st January 1970. Therefore, by running the simulation many times, the result of the independent runnings is never repeated and the bias gets broken. Thus, it does not make much sense to emphasize the variables  $M_{exp}$  depends on since the relation between the function and the variables is not univocal due to the presence of the randomizer function.

The KATRIN protocol establishes that several measurements in different values of the retarding potential have to be done in the studied energy region to fit the data and to conclude a neutrino mass. For example, when analyzing within the energy range from 20 eV below the endpoint to the endpoint, the protocol has been set to have 31 measurement points distributed in several values of  $qU$  with different times  $t_U$ . Actually, some of them are carried out above  $E_0$  to study more precisely the background component. That set of data can be fitted to deduce the actual values of the free parameters that generated the mentioned 31 measurements. This kind of fitting procedure is accomplished by means of

the well-known Chi-Square method using:

$$\chi^2(r_1, \dots, r_l) = \sum_{i=1}^n \left[ \frac{M_{exp_i} - M_{theo_i}(r_1, \dots, r_l)}{\sqrt{M_{theo_i}(r_1, \dots, r_l)}} \right]^2 \quad (4.3)$$

where both the theoretical function and the Chi-Square function depend on  $l$  free parameters called  $r$ . The  $\chi^2$  function is calculated by adding up the data of  $n$  measurements in different retarding potentials  $U$ . This fitting method assumes that the set of  $r_1, \dots, r_l$ , which produces the minimum value of  $\chi^2$  summarizes the real value of the parameters. With the aim of making the method clear, it is important to explain the dependence of functions on variables. Note that  $\chi^2$  and  $M_{theo}$  depend on the same variables. The dependence of the experimental data on their inner variables has been removed due to the randomizer function. Therefore, from now on, when talking about dependences of  $M_{exp}$  it is understood that we are referring to the model chosen to create it. Hence,  $M_{theo}$  and  $M_{exp}$  can have either different dependences or the same ones. This has the enormous advantage of offering the possibility of mixing models, i.e., creating data with one model and analyzing with another model, which is very useful to check the potential systematic uncertainties related to mixing up different theories.

In order not to get confused with the procedure, it is better to forget how the experimental information was generated and simply to assume that a set of experimental data is available to be fitted. Nothing is known about the actual value of the free parameters  $p_1, \dots, p_q$  that originated these measurements. Only the values of  $qU$  and  $t_U$  used to measure  $M_{exp}$  are known.

The Chi-Square method gives back fitted data by trying many  $M_{theo}$  and looking for the one that leads to the minimum value of the Chi-Square function. The  $M_{theo}$  values are the expected decay rates on the  $qU$  retarding potentials during the times  $t_U$ . In general, they are computed like:

$$M_{theo}(qU, t_U, R_s, R_b, p_1, \dots, p_q) = t_U \left( R_s \int_{qU}^{E_0} f_{res}(qU, E) \frac{d^2 N}{dt dE} dE + R_b \Gamma_b \right) \quad (4.4)$$

where two new free parameters  $R_s$  and  $R_b$  have been introduced to let both the decay rate and the background fluctuate.

Let us discuss it with an example. With this kind of simulation it is possible to create decays which follow physics of any of the theoretical models explained in Ch. 2. For instance, by choosing the SM and the last 20 eV of the spectrum as interesting region, a set of 31 integrated rates is produced with:

$$\frac{d^2 N}{dE dt} = K \sum_i W_i \cdot F(\epsilon_i, y) p_e \cdot E_{tot} \cdot \epsilon_i \cdot \sqrt{\epsilon_i^2 - m_\nu^2} \cdot \theta(\epsilon_i - m_\nu) \quad (4.5)$$

This type of spectrum leads to the following rates:

$$S(qU, t_U, E_0, m_\nu^2) = t_U \left( \int_{qU}^{E_0} f_{res}(qU, E) \frac{d^2 N}{dt dE} dE + \Gamma_b \right) \quad (4.6)$$

which must be randomly spread to yield the simulated experimental data:

$$M_{exp} = S(qU, t_U, E_0, m_\nu^2) + \text{Randomizer} \left( \sqrt{S(qU, t_U, E_0, m_\nu^2)} \right) \quad (4.7)$$

Note again that  $M_{exp}$  has no explicit dependences. These measurements could now be analyzed with several models, for instance, the SM choice would produce the following Chi-Square:

$$\chi^2(E_0, m_\nu^2, R_s, R_b) = \sum_{i=1}^n \left[ \frac{M_{exp_i} - M_{theo_i}(E_0, m_\nu^2, R_s, R_b)}{\sqrt{M_{theo_i}(E_0, m_\nu^2, R_s, R_b)}} \right]^2 \quad (4.8)$$

which is a four dimensional function of which one has to find the global minimum. To obtain a conclusion on the MC simulation, the calculation in (4.8) has to be repeated many times with the same fixed input parameters. In every independent run of (4.8), the action of the randomizer in the experimental data will produce different global minima in Chi-Square. When repeating the randomization in  $M_{exp}$  with the same input in  $S$ , a different measurement of the same quantity is being carried out, leading to little changes on  $\chi^2$ . Therefore, the global minimum is located in different values of  $E_0$ ,  $m_\nu^2$ ,  $R_s$ ,  $R_b$  for each  $\chi^2$  surface. This process leads for the free parameters to Gaussian distributions centered on the input, i.e., centered on the *actual* value of the parameters. As an illustrative example, Fig. 4.1 shows the result for the endpoint fit after running 5000 times the simulation. This number of experiments has been set as a standard for the MC simulations in this work. The experimental data of Fig. 4.1 was produced by setting  $E_0$  to 18575 eV. The measurements are spread around the actual value of the endpoint leading to a Gaussian distribution whose standard deviation will give the sensitivity of the KATRIN experiment on this parameter by considering SM physics.

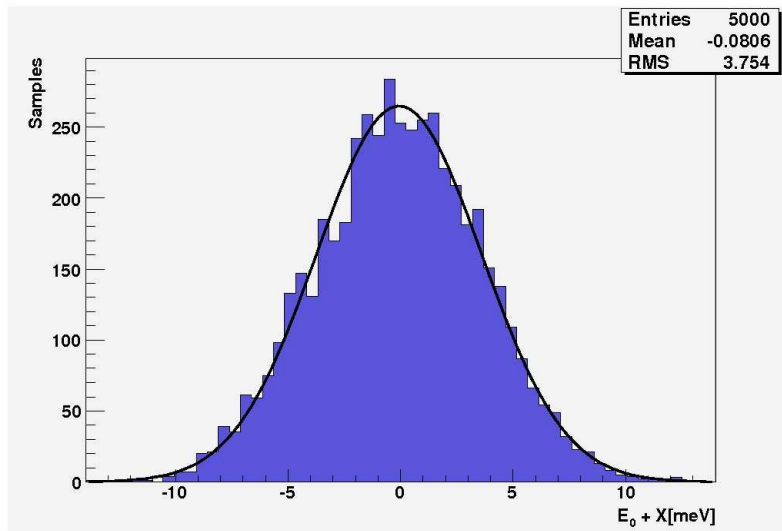


Figure 4.1: Typical output of the endpoint measurement in a Monte Carlo simulation of KATRIN.

For the other parameters similar functions were obtained. Note, that the fitting is done for all free parameters at the same time. Since the MC output is centered on the input, the package is considered to be working properly. Another test for the fitting process is the value of the Chi-Square function. Its 5000 different values are sorted not in a Gaussian distribution but in a Chi-Square one. Furthermore, its mean value must be near the number of measurements minus the number of free parameters [39], although the

presence of correlations between parameters could break this statement. In the example shown in Fig. 4.2 the Chi-Square mean value should be 27 (31 measurements minus 4 free parameters) whereas the found value is 27.18. Thus, there is a good agreement between the expected Chi-Square and the obtained one.

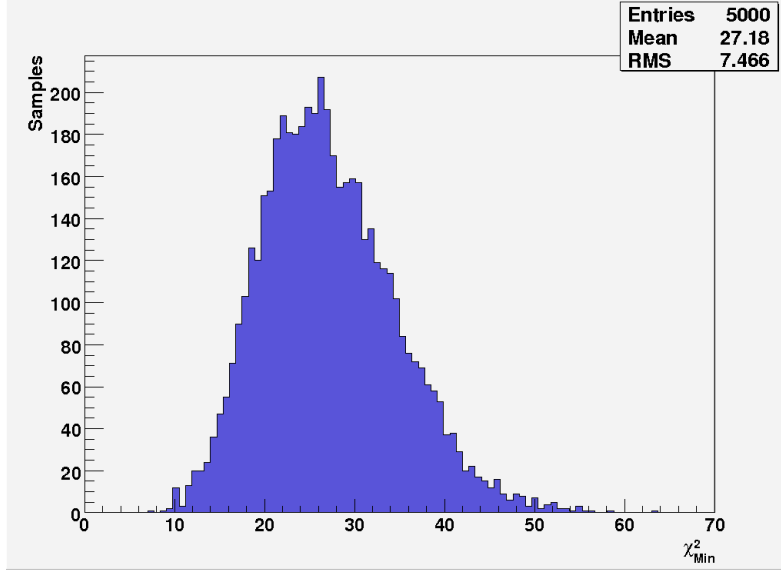


Figure 4.2: Typical output of the Chi-Square result in a Monte Carlo simulation of KATRIN when measuring in the last 20 eV.

## 4.2 KATRIN simulations within the Standard Model

After building the necessary tool, the first calculation has to be related to the sensitivity of the experiment when the data is analyzed with physics of the SM. The reason why this is a good starting point is that the SM decay is the simplest description since it deals with the smallest number of free parameters. Furthermore, there are other older simulations that already computed these sensitivities with which one can compare the MC result. Therefore, it serves also as a test to check whether the simulation code has been written properly. We worked with the following decay distribution, both to create the experimental and the theoretical data:

$$\frac{d^2 N}{dE dt}(E_0, m_\nu^2) = K \sum_i W_i \cdot F(\epsilon_i, y) p_e \cdot E_{tot} \cdot \epsilon_i \cdot \sqrt{\epsilon_i^2 - \frac{m_\nu^2}{m_e^2}} \cdot \theta(\epsilon_i - \frac{m_\nu}{m_e}) \quad (4.9)$$

In the SM, the Chi-Square function depends on four variables,  $E_0$ ,  $m_\nu^2$ ,  $R_s$  and  $R_b$ . One has to look for the best combination of these parameters which leads to the minimum of Chi-Square. At this point, the unsolved problem is to find that minimum. There are many ways to look for the global minimum. In this case, within the SM, a very practical and simple routine was chosen, even though it was not a very efficient algorithm. It consists on building a 4-dimensional grid with many fixed points in which Chi-Square is computed. Later on, all values are compared and the minimum one is claimed to be the global minimum of Chi-Square. Thus, by varying the numerical values of the four

free parameters of  $M_{theo}$  within given intervals, a grid is constructed to produce many computations of Chi-Square.

An example of the performance of this method is shown in Fig. 4.3 where the sensitivity of the experiment to the four free parameters can be deduced. The input was created with  $E_0 = 18575$  eV,  $m_\nu^2 = 0.1225$  eV<sup>2</sup>,  $R_s = 1$  and  $R_b = 1$  which is around the KATRIN discovery potential on  $m_\nu$ .

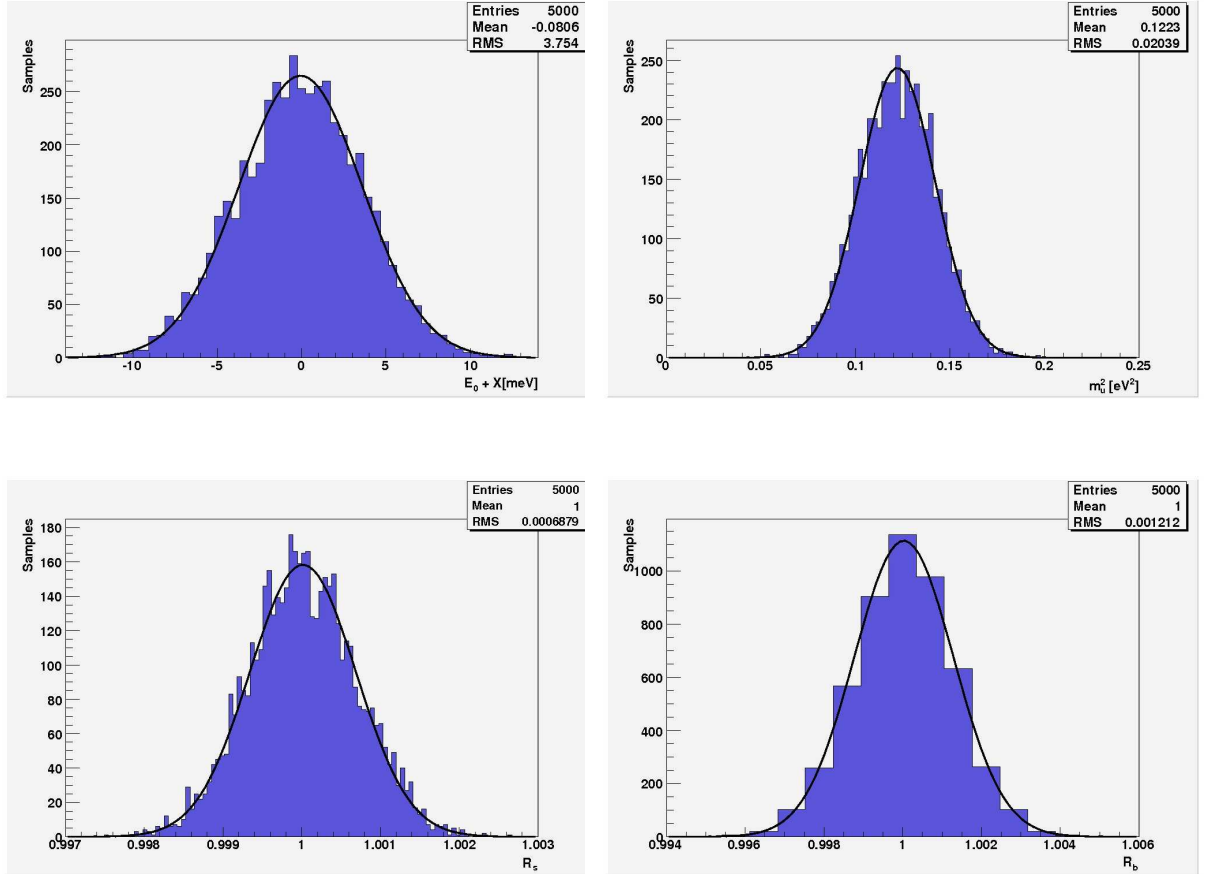


Figure 4.3: Sensitivity of the experiment on  $E_0$  (top left),  $m_\nu^2$  (top right),  $R_s$  (bottom left) and  $R_b$  (bottom right) when creating and analyzing data with physics of the SM. The input was  $E_0 = 18575$  eV,  $m_\nu^2 = 0.1225$  eV<sup>2</sup>,  $R_s = 1$  and  $R_b = 1$ .

The good behaviour over all variables predicts a good performance of the package. Note, the binning in the background variable is bigger compared to the others. This is because a smaller number of points in this variable was set up when the 4-dimensional grid was constructed. The reason why this variable was treated differently is, that the grid method is quite inefficient from the computational point of view, i.e., it is very time consuming. Since, a priori,  $R_b$  is the least correlated variable, one expects to fit this parameter with only a few points whereas the other variables need a finer treatment. The correlations between parameters quantify how much variables depend on each other, which is very important to know whether external experiments can improve the sensitivity of KATRIN, since if a fit parameter can be deduced by an independent experiment one can in principle use this information to restrict the range in which the parameter is allowed to vary in the fit. As can be seen in Fig. 4.4,  $E_0$ ,  $m_\nu^2$  and  $R_s$  are quite correlated with each

other. The background, as expected, is not correlated at all with any other variable.

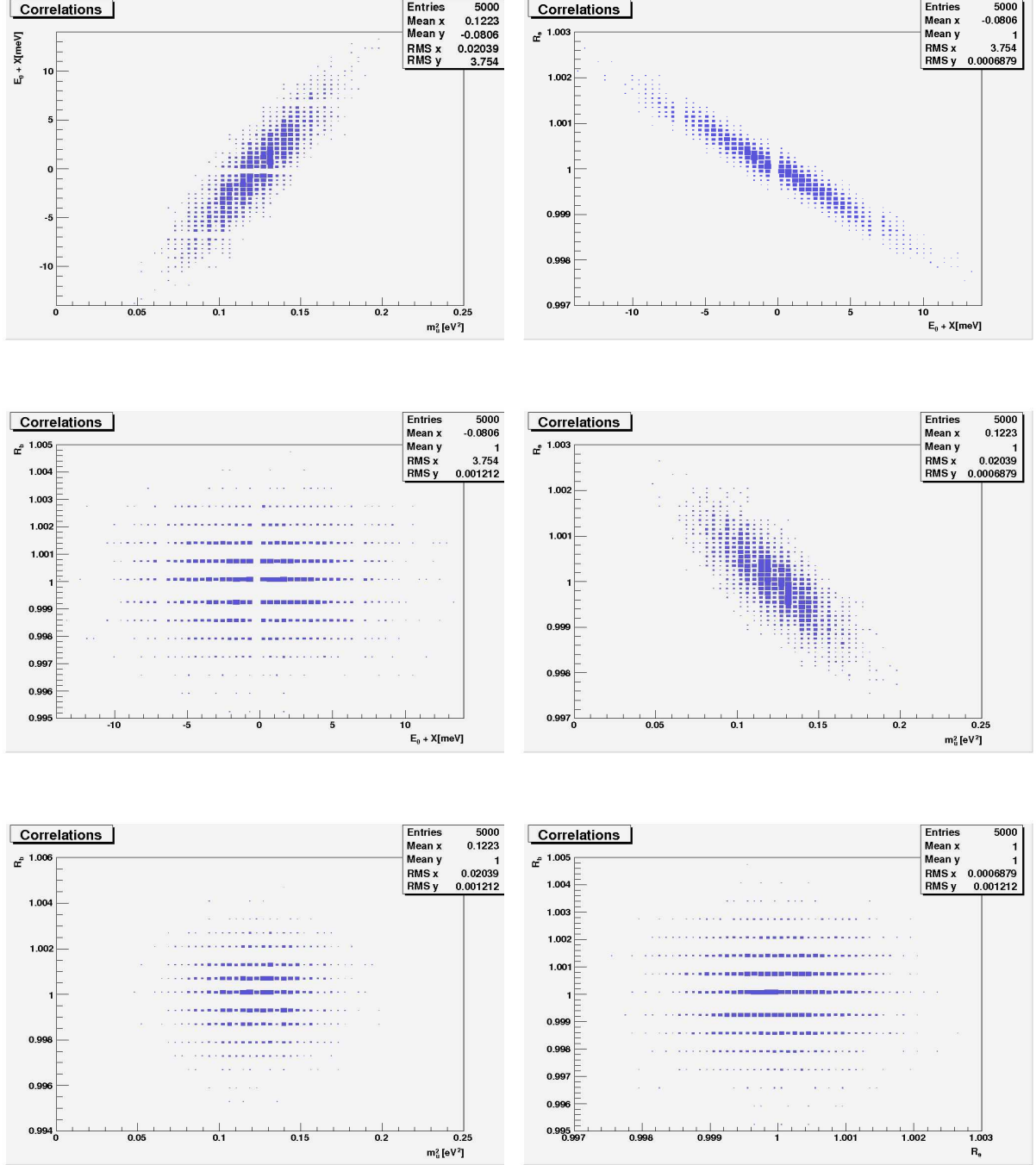


Figure 4.4: Correlations between the variables of the analysis of the experiment.  $E_0$  vs.  $m_\nu^2$  (first row, left),  $E_0$  vs.  $R_s$  (first row, right),  $E_0$  vs.  $R_b$  (second row, left),  $m_\nu^2$  vs.  $R_s$  (second row, right),  $m_\nu^2$  vs.  $R_b$  (third row, left) and  $R_s$  vs.  $R_b$  (third row, right).

It becomes obvious, that the significant correlations occur always between the combinations of  $E_0$ ,  $m_\nu^2$  and  $R_s$ . As stated before, the background term is the least correlated component of the analysis, so by choosing a few points in its description we did not make any systematic error.

The next step is to calculate the experimental sensitivity on the neutrino mass when the input is a vanishing neutrino mass. This will give the potential new upper limit of the neutrino mass. When working with such an input, it is expected to find a Gaussian distribution on the neutrino mass centered on 0 eV<sup>2</sup>. This means that half of the distribution falls in the negative region of  $m_\nu^2$  which is not in contradiction with the already found experimental results [20, 21], although no physical meaning is accepted within the SM. Some hints about the treatment of this problem beyond the SM have already been given. Now, the important point is to create for that region some decay rate physically meaningless but statistically useful to calculate the sensitivity. The following definition of the spectrum has been followed with success:

$$\frac{d^2N}{dEdt} = \begin{cases} K \sum_i W_i \cdot F(\epsilon_i, y) p_e \cdot E_{tot} \cdot \epsilon_i \cdot \sqrt{\epsilon_i^2 - m_\nu^2} \cdot \theta(\epsilon_i - m_\nu) & m_\nu^2 \geq 0 \\ K \sum_i W_i \cdot F(\epsilon_i, y) p_e \cdot E_{tot} \cdot \epsilon_i \cdot \sqrt{\epsilon_i^2 - m_\nu^2} \cdot \theta(\epsilon_i) & m_\nu^2 < 0 \end{cases} \quad (4.10)$$

where for the negative part the Heaviside function does not take care of the neutrino mass value. Note that a negative  $m_\nu^2$  would lead to an imaginary  $m_\nu$ . This definition makes Chi-Square be continuous in the transition for the positive to the negative part of  $m_\nu^2$ . Under these conditions, the simulation has been run many times to estimate the sensitivities when measuring during 3 years in the last 20 eV, 25 eV, 30 eV, 40 eV and 50 eV. The results for the endpoint and the neutrino mass are given in Fig. 4.5-4.9, the correlations are similar to the ones already shown.  $\chi^2$ ,  $R_s$  and  $R_b$  behaved properly for all these simulations.

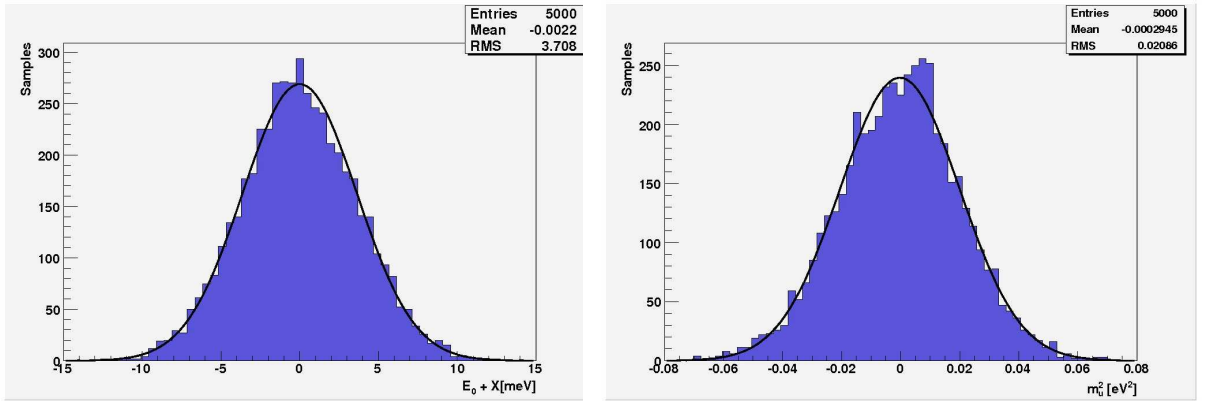


Figure 4.5: Sensitivity of the experiment on  $E_0$  (left) and  $m_\nu^2$  (right) when creating and analyzing data with physics of the SM. The measurement time is around 3 years. The lowest retarding potential was placed in 20 eV below the endpoint. The input was  $E_0 = 18575$  eV,  $m_\nu^2 = 0$  eV<sup>2</sup>,  $R_s = 1$  and  $R_b = 1$ .



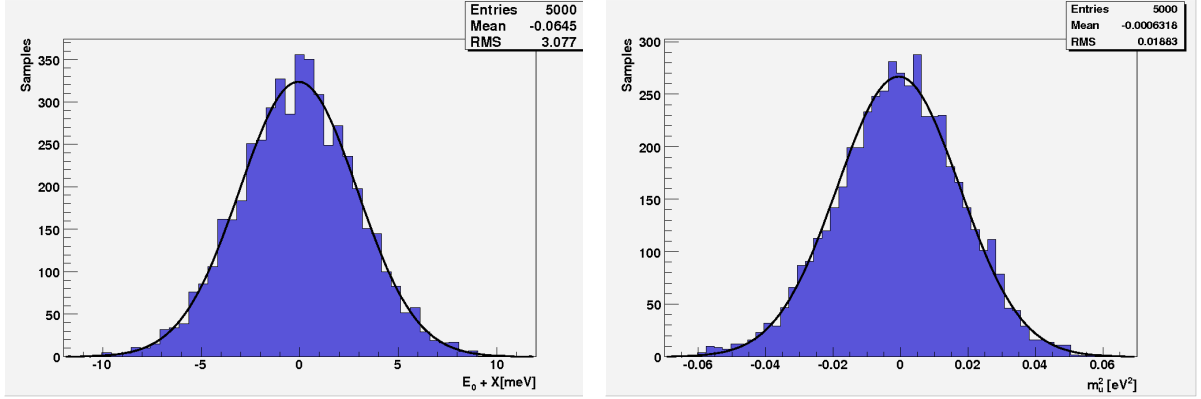


Figure 4.6: Sensitivity of the experiment on  $E_0$  (left) and  $m_\nu^2$  (right) when creating and analyzing data with physics of the SM. The measurement time is around 3 years. The lowest retarding potential was placed in 25 eV below the endpoint. The input was  $E_0 = 18575$  eV,  $m_\nu^2 = 0$  eV<sup>2</sup>,  $R_s = 1$  and  $R_b = 1$ .

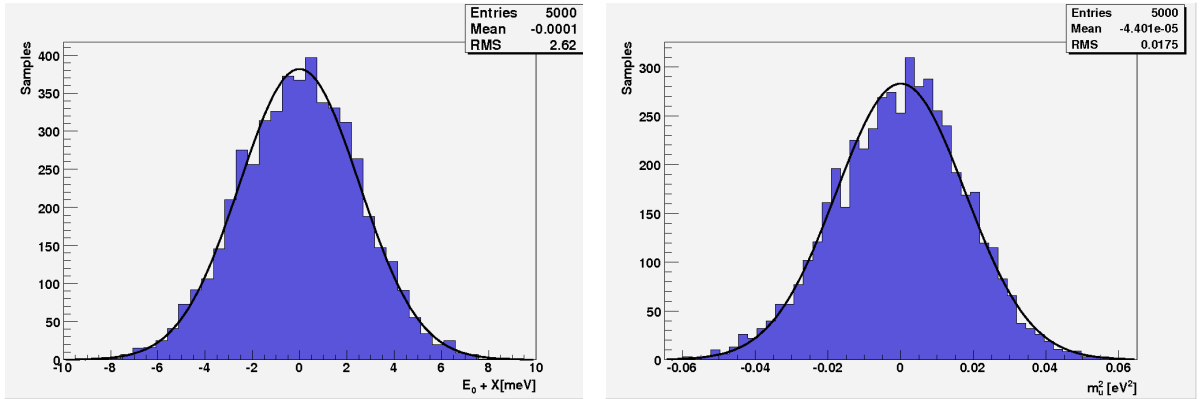


Figure 4.7: Sensitivity of the experiment on  $E_0$  (left) and  $m_\nu^2$  (right) when creating and analyzing data with physics of the SM. The measurement time is around 3 years. The lowest retarding potential was placed in 30 eV below the endpoint. The input was  $E_0 = 18575$  eV,  $m_\nu^2 = 0$  eV<sup>2</sup>,  $R_s = 1$  and  $R_b = 1$ .

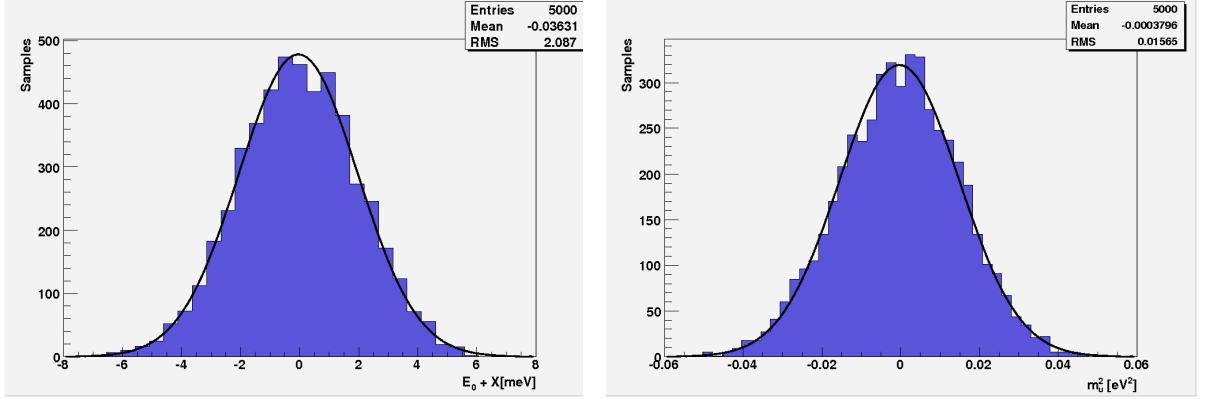


Figure 4.8: Sensitivity of the experiment on  $E_0$  (left) and  $m_\nu^2$  (right) when creating and analyzing data with physics of the SM. The measurement time is around 3 years. The lowest retarding potential was placed in 40 eV below the endpoint. The input was  $E_0 = 18575$  eV,  $m_\nu^2 = 0$  eV<sup>2</sup>,  $R_s = 1$  and  $R_b = 1$ .

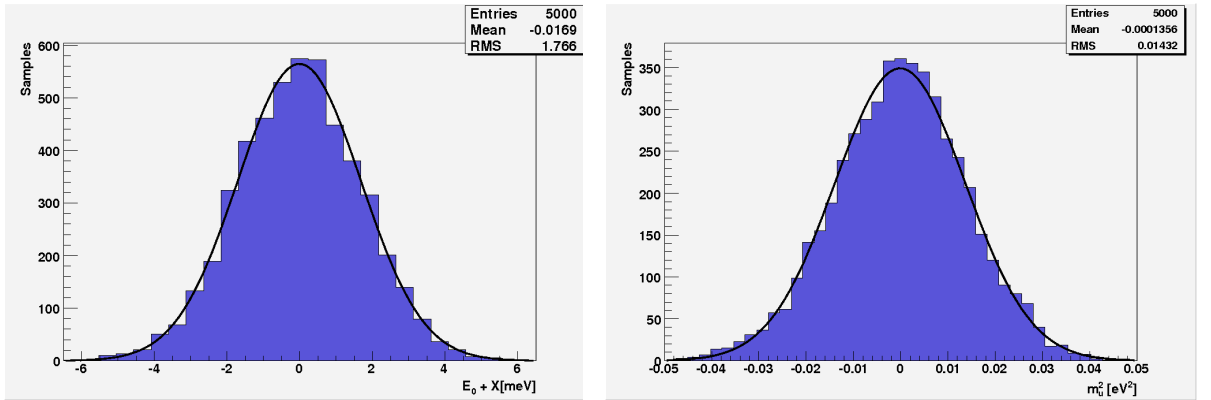


Figure 4.9: Sensitivity of the experiment on  $E_0$  (left) and  $m_\nu^2$  (right) when creating and analyzing data with physics of the SM. The measurement time is around 3 years. The lowest retarding potential was placed in 50 eV below the endpoint. The input was  $E_0 = 18575$  eV,  $m_\nu^2 = 0$  eV<sup>2</sup>,  $R_s = 1$  and  $R_b = 1$ .

It is interesting to note that the deeper the analysis is done the better is the accuracy found in both the endpoint and the neutrino mass. The experimental sensitivity ( $1.64485 \sigma$ ) on the neutrino mass and the discovery potential ( $5 \sigma$ ) of the experiment are summarized in Table 4.1, where the standards 90% C.L. ( $1.64485 \sigma$ ) and 99.9999% C.L. ( $5 \sigma$ ) have been followed.

$E(\text{eV})$	$\sigma_{m_\nu^2} (\text{eV}^2)$	$1.64485\sigma_{m_\nu^2} (\text{eV}^2)$	$5\sigma_{m_\nu^2} (\text{eV}^2)$	$1.64485\sigma_{m_\nu} (\text{eV})$	$5\sigma_{m_\nu} (\text{eV})$
20	0.0209	0.0343	0.104	0.185	0.323
25	0.0188	0.0310	0.0941	0.176	0.307
30	0.0175	0.0288	0.0875	0.170	0.296
40	0.0156	0.0257	0.0782	0.160	0.280
50	0.0143	0.0235	0.0716	0.153	0.268

Table 4.1: KATRIN sensitivity (column 5) and discovery potential (column 6) on the neutrino mass for different energy ranges (only statistical errors).

Table 4.1 contains information for the neutrino mass square and the neutrino mass. Some of the data of Table 4.1 are summarized in Fig. 4.10 where the change of sensitivity and discovery potential are shown as functions of the studied energy range. Both quantities get improved when the analysis is carried out over a bigger energy range in the spectrum. The reason for improving the limit on  $m_\nu$  for larger energy ranges is the high statistics of the points further away from the endpoint. This advantage, however, is overcompensated by the increasing systematic uncertainties of these data, which are not taken into account in this simulation. We can introduce systematic uncertainties by assuming that after 3 years of measurement systematic and statistics contribute equally to the error for the 30 eV interval. To consider also the systematic uncertainties, the statistical  $5\sigma_{m_\nu^2}$  value is multiplied by  $\sqrt{2}$  to yield a new  $5\sigma_{m_\nu^2}$  which takes into account both error sources, leading to a discovery potential limit of 0.352 eV as against 0.354 eV for the value extracted from the KATRIN Design Report [22]. The insignificant third digit should be omitted leading both cases to 0.35 eV. It can also be seen that the sensitivities obtained by means of this

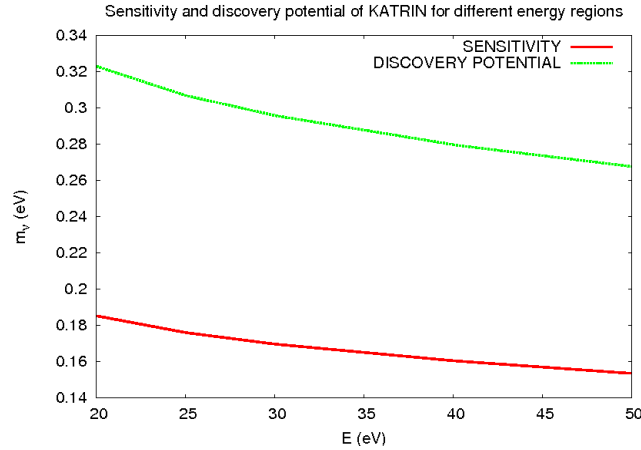


Figure 4.10: Contribution of the statistical uncertainty to the sensitivity and discovery potential as a function of the energy region.

MC simulation are even better than those published in the KATRIN Design Report [22]. This is due to the fact that some systematic uncertainties have not been considered in this simulation, like fluctuations of the retarding potential, fluctuations in the column density, or other energy changes in the  $\beta$ -particles like the ones produced by the recoil energy of the daughter molecule, etc.. Among them, the most important one is the fluctuation and uncertainty of the retarding potential. The present experimental limit on the endpoint of tritium beta decay establishes an uncertainty of  $\pm 1.7$  eV [40]. The MC results shown

above reduce this sensitivity to about  $\pm 10$  meV, i.e.,  $\Delta E_0$  gets improved by two orders of magnitude. However, no claims regarding these improvements were done because no considerations on the uncertainty of the retarding potential were included. Once this is taken into account, the endpoint uncertainty will be increased again up to the uncertainty of the retarding potential. This retarding potential uncertainty has been set around  $\pm 300$  meV [41] which is far from the MC result. Nevertheless, a very improved high-precision Penning trap measurement of the  $Q$  value of  ${}^3\text{T}(\beta^-){}^3\text{He}$  decay [42] is planned to be carried out during the coming years, which could help to reduce systematic errors in KATRIN if the measurement of  $Q$  is within the uncertainty of the retarding potential.

### 4.3 KATRIN simulations beyond the Standard Model

Results obtained by means of this MC package concerning physics beyond the SM are explained in this section. The main attention has been led to the Non V-A problem whereas the problem with tachyonic neutrinos has also been solved. Regarding the violation of Lorentz invariance there are no results to talk about since the problem was only implemented into the package without finding any result due to the lack of time. Nevertheless, we want to remark that everything is ready to run and to find interesting approaches to this problem.

#### 4.3.1 Non V-A structure

To deal with the Non V-A problem the calculation procedure of section 4.1 has been followed. The simulation concept is exactly the same which was explained in the previous section, except for the decay rate which had to be changed to include Non V-A physics.

The new spectrum was already obtained in Ch. 2, although no description was found for the negative region of the neutrino mass square. As it was explained, this region has to be well-defined, because of the statistical fluctuations of the data. No physical meaning is attributed to it neither within the SM nor within the Non V-A problem, but it is necessary to have such a description in order to find a proper Gaussian distribution when the input is a vanishing  $m_\nu^2$ .

The new decay rate is:

$$\frac{d^2 N}{dE dt} = \begin{cases} K \sum_i W_i \cdot F(\epsilon_i, y) p_e \cdot E_{\text{tot}} \cdot \epsilon_i \cdot \sqrt{\epsilon_i^2 - m_\nu^2} \cdot \theta(\epsilon_i - m_\nu) \left[ 1 + \delta \frac{m_\nu}{\epsilon_i} \right] & m_\nu^2 \geq 0 \\ K \sum_i W_i \cdot F(\epsilon_i, y) p_e \cdot E_{\text{tot}} \cdot \epsilon_i \cdot \sqrt{\epsilon_i^2 - m_\nu^2} \cdot \theta(\epsilon_i) \left[ 1 + \delta \frac{m_\nu}{\epsilon_i} \right] & m_\nu^2 < 0 \end{cases} \quad (4.11)$$

where the relationship between the new quantity  $m_\nu$  and the old parameter  $m_\nu^2$  has been always treated like  $m_\nu = \sqrt{|m_\nu^2|}$ , so that  $m_\nu^2$  keeps being the free parameter. Note again, a new free parameter  $\delta$  has been introduced to handle the Non V-A physics, since this  $\delta$  contains combinations of coupling constants of the Tensor and Scalar currents among others.

The steps to follow should in principle be the ones followed in the SM case, with the final objective of finding the new KATRIN sensitivity and discovery potential. The same mathematical treatment (Chi-Square) was chosen and a five-dimensional grid (one more free parameter than in the SM case) was built to look for the minimum of Chi-Square. However, the actual situation turned out to be very different, since many problems came

up right after introducing the new physics. These problems can be summarized in the fact that no Gaussian distributions were found at all, and instead *chaotic* distributions not even centered on the input were obtained. The same behaviour in the outputs was found independently of the input. This problem was at the beginning interpreted as a bug in the MC package and a lot of efforts were directed to find and fix it, although later on it was demonstrated that the problem came from a deeper reason, i.e., correlations between parameters. When having this kind of problem it seems to be very reasonable to reduce the number of possible error sources. For that purpose, the Fermi and Sirlin function were deactivated as well as the interaction with the spectrometer (response function) and the final states distribution (FSD) of the daughter molecule. In addition, the studied energy range was restricted to the last 20 eV of the spectrum once it was checked that the MC results did not depend on the studied energy region. Therefore, the spectra were directly integrated with analytical integrals which reduced very much the computing time although did not improve the MC outputs. At this point, the study on the influence of Non V-A currents on tritium beta decay became more a qualitative study than a quantitative one. The integrated decay necessary to the Chi-Square computation was analytically handled from here on. From (4.11) a new integrated spectrum was derived, both for the positive and the negative regions of  $m_\nu^2$ . By integrating (4.11) and applying Barrow's rule taking special care of the Heaviside function, from  $qU$  (the retarding potential) to  $E_0$  in both regions:

$$N(E_0, m_\nu^2, \delta, qU) = \begin{cases} K \left[ \frac{1}{3} \Omega^{\frac{3}{2}} + \delta m_\nu \left( \Lambda \sqrt{\Omega} - m_\nu^2 \ln \left( \frac{\Lambda + \sqrt{\Omega}}{2} \right) \right) \right] & m_\nu^2 \geq 0 \\ K \left[ \frac{1}{3} \left( \Omega^{\frac{3}{2}} - (-m_\nu^2)^{\frac{3}{2}} \right) + \delta m_\nu \left[ \Lambda \sqrt{\Omega} + m_\nu^2 \ln \left( \frac{2m_\nu}{\Lambda + \sqrt{\Omega}} \right) \right] \right] & m_\nu^2 < 0 \end{cases} \quad (4.12)$$

here  $\Lambda = E_0 - qU$  and  $\Omega = \Lambda^2 - m_\nu^2$ . The electron energy and momentum were taken out of the integral and treated as constants without making a big error, so the factor  $K p_e E_{tot}$  must be understood in  $K$ . With the computation of the integrated spectrum, taking into account the measurement time  $t_U$  and the background rate, Chi-Square is calculated as explained before. All the simplifications in the spectrum did not lead to any success when finding the sensitivity of the experiment. However, in order to know how dangerous it is not to take into account the right currents problem, it was checked how much a Non V-A spectrum influences on a SM analysis, i.e., what happens if the experimental information is fitted with the SM whereas the actual spectrum follows Non V-A physics. This mixture of theoretical approaches can be done by creating decays with Non V-A and analyzing them with SM, i.e, creating with a non vanishing  $\delta$  and analyzing with  $\delta = 0$ . When doing this, it was found out that the KATRIN measurement would always be shifted with respect to the real value of the parameters. Fortunately, the shift is mainly carried by the endpoint and not so strongly by the neutrino mass square. This makes sense, since  $E_0$  and  $\delta$  seemed to be very correlated and therefore, misleadings in any of these two parameters would produce misleadings in the other one as well. This analysis of the influence of a  $\delta \neq 0$  in a SM fit was carried out in the last 20 eV of the spectrum. Several cases for different values of  $\delta$  within its actual experimental limits  $[-0.28, +0.28]$  were studied. Beside the shift of the parameters  $E_0$  and  $m_\nu^2$ , the KATRIN sensitivities  $\sigma_{E_0}$  and  $\sigma_{m_\nu^2}$  were computed. This calculation was done by running the MC package many times for the following inputs.

$$E_0 = 18575 \text{ eV};$$

$$m_\nu^2 = 0 \text{ eV}^2, m_\nu^2 = 0.1 \text{ eV}^2, m_\nu^2 = 0.2 \text{ eV}^2, m_\nu^2 = 0.3 \text{ eV}^2, m_\nu^2 = 0.4 \text{ eV}^2, m_\nu^2 = 0.5 \text{ eV}^2, \\ m_\nu^2 = 0.6 \text{ eV}^2, m_\nu^2 = 0.7 \text{ eV}^2, m_\nu^2 = 0.8 \text{ eV}^2, m_\nu^2 = 0.9 \text{ eV}^2, m_\nu^2 = 1 \text{ eV}^2, m_\nu^2 = 1.5 \text{ eV}^2, \\ m_\nu^2 = 2 \text{ eV}^2, m_\nu^2 = 2.5 \text{ eV}^2, m_\nu^2 = 3 \text{ eV}^2, m_\nu^2 = 3.5 \text{ eV}^2, m_\nu^2 = 4 \text{ eV}^2;$$

$$\delta = -0.3, \delta = -0.2, \delta = -0.1, \delta = 0 \text{ (SM)}, \delta = +0.1, \delta = +0.2, \delta = +0.3,$$

and all possible combinations between them.

The obtained results concerning the shift on neutrino mass square, endpoint and their respective standard deviations are shown in Fig. 4.11-4.14 .

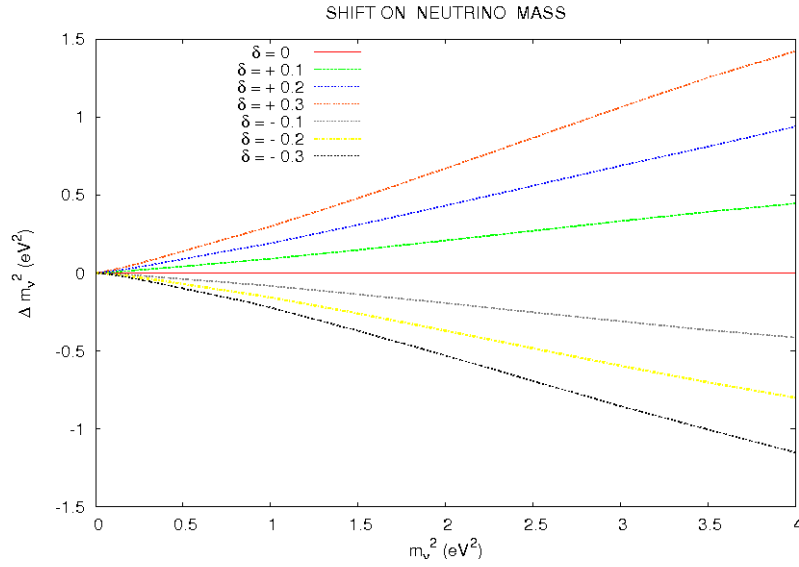


Figure 4.11: Shift on the SM fit of  $m_\nu^2$  as a function of the real  $m_\nu^2$  and  $\delta$ . Each line corresponds to individual values of  $\delta$ . From top to bottom:  $\delta = +0.3$ ,  $\delta = +0.2$ ,  $\delta = +0.1$ ,  $\delta = 0$  (SM),  $\delta = -0.1$ ,  $\delta = -0.2$ ,  $\delta = -0.3$ .

The positive conclusion from these plots is that both the sensitivity on the neutrino mass square  $\sigma_{m_\nu^2}$  and the endpoint  $\sigma_{E_0}$  do not change too much either with  $\delta$  or the actual  $m_\nu^2$ . The negative conclusion is, that unfortunately, if physics actually follows the Non V-A theory and the experiment is fitted with the SM, an unpleasant shift will be found for  $m_\nu^2$ . Let us examine the worst cases when  $\delta \sim +0.3$  or  $\delta \sim -0.3$ , near the KATRIN discovery potential ( $m_\nu = 0.35 \text{ eV}$ ). Close to this value is one of the chosen values of the study,  $m_\nu = 0.33 \text{ eV}$ , i.e.,  $m_\nu^2 = 0.1 \text{ eV}^2$ . For  $\delta = +0.3$  the measured value on  $m_\nu^2$  would be shifted around 20%, whereas for  $\delta = -0.3$  the shift is around 12%. Therefore, an exact fit which includes Non V-A physics is needed, although the correlation between  $E_0$  and  $\delta$  seems not to allow to fit it.

To solve the problem when analyzing with Non V-A physics, some investigations on the Chi-Square shape led to the conclusion that the problem came from a wrong method when looking for the global minimum of the function. Therefore, the employment of the grid method was given up, due to the fact that it is not only inefficient at finding the global minima but also very slow. These global minimum finding problems occur very often when the correlations between parameters are very strong. It produces on the object surface either a lot of local minima or very flat zones, which makes the solution be hidden.

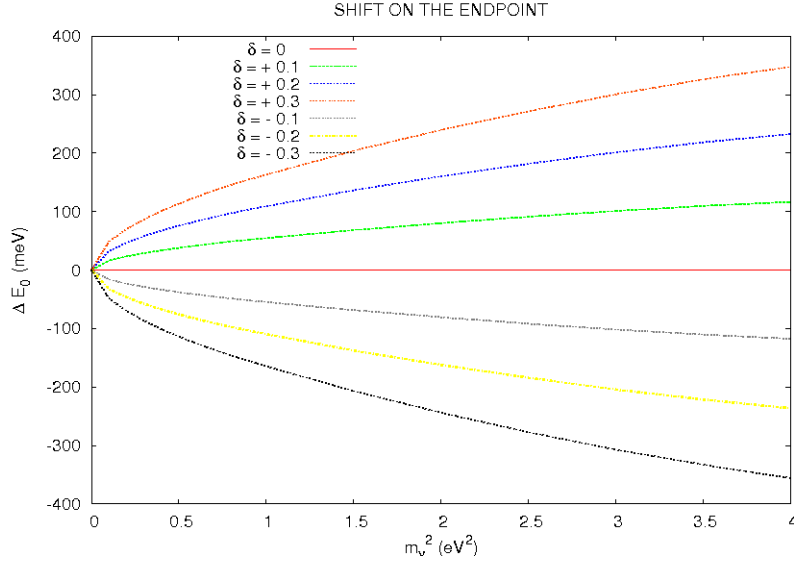


Figure 4.12: Shift on the SM fit of  $E_0$  as a function of the real  $m_\nu^2$  and  $\delta$ . Each line corresponds to individual values of  $\delta$ :  $\delta = +0.3$ ,  $\delta = +0.2$ ,  $\delta = +0.1$ ,  $\delta = 0$  (SM),  $\delta = -0.1$ ,  $\delta = -0.2$ ,  $\delta = -0.3$ .

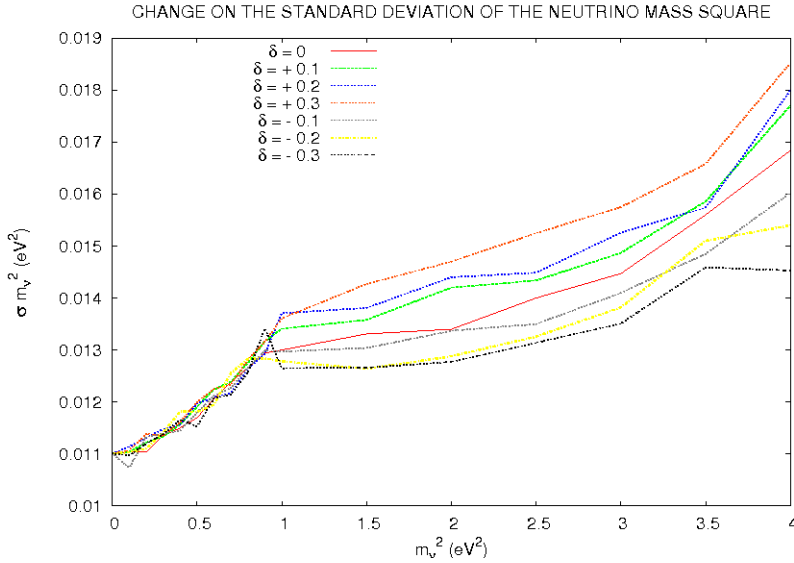


Figure 4.13: Shift on the SM fit of  $\sigma_{m_\nu^2}$  as a function of the real  $m_\nu^2$  and  $\delta$ . Each line corresponds to individual values of  $\delta$ :  $\delta = +0.3$ ,  $\delta = +0.2$ ,  $\delta = +0.1$ ,  $\delta = 0$  (SM),  $\delta = -0.1$ ,  $\delta = -0.2$ ,  $\delta = -0.3$ .

From this point, after realizing that there was a very difficult technical problem, it was decided that the solution, i.e. the global minimum of the  $\chi^2$ , has to be found by using another kind of algorithm when looking for the minima. Many different algorithms have been tested causing a lot of rewriting of the code for each of them. In the following, all the algorithms that were tested are summarized. They are sorted as they were checked. Most of them belong to the GSL (GNU Scientific library) [43] and MINUIT (CERN) [44] packages, and some of them delivered some improvement which was interpreted as a proof of being well-led to the solution. These algorithms are briefly explained. In some cases the original sources of [43, 44] are quoted:

- The grid method: It has already been explained and delivered no improvement at

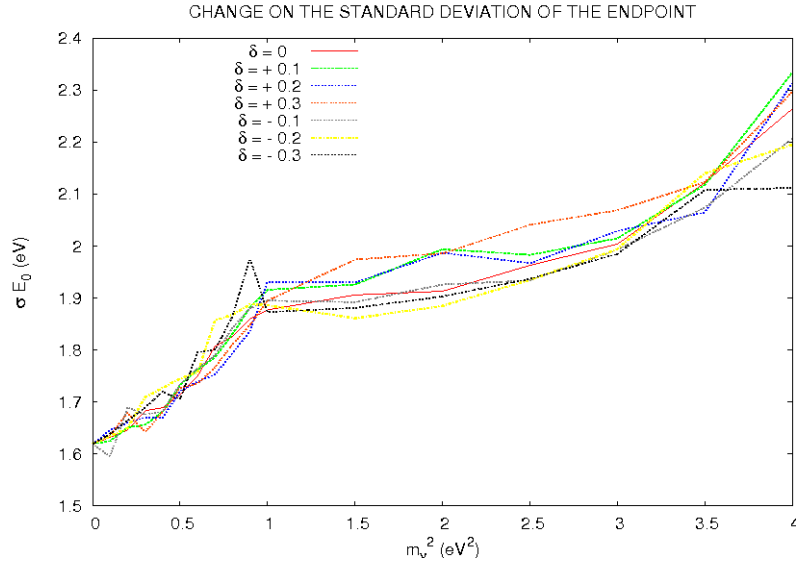


Figure 4.14: Shift on the SM fit of  $\sigma_{E_0}$  as a function of the real  $m_\nu^2$  and  $\delta$ . Each line corresponds to individual values of  $\delta$ :  $\delta = +0.3$ ,  $\delta = +0.2$ ,  $\delta = +0.1$ ,  $\delta = 0$  (SM),  $\delta = -0.1$ ,  $\delta = -0.2$ ,  $\delta = -0.3$ .

all.

- The simplex algorithm of Nelder and Mead: It constructs over the surface a hyper-triangle with  $n+1$  vertices in which the global minimum is looked for.  $n$  is the number of dimensions of the problem. This algorithm already gave some improvement [43].

- The Fletcher-Reeves conjugate gradient algorithm: It proceeds as a succession of line minimizations. The sequence of search directions is used to build up an approximation to the curvature of the function in the neighborhood of the minimum [43]. No improvements were found in comparison to the simplex treatment.

- The Polak-Ribiere gradient algorithm: It is similar to the Fletcher-Reeves method, differing only in the choice of the coefficient  $\beta$  which serves to update the search direction [43]. No improvements were found in comparison to the simplex treatment.

- The vector Broyden-Fletcher-Goldfarb-Shanno conjugate gradient algorithm: It is a quasi-Newton method which builds up an approximation to the second derivatives of the function using the difference between successive gradient vectors [43]. No improvements were found in comparison to the simplex treatment.

- The steepest descent algorithm. This algorithm follows the downhill gradient of the function at each step. When a downhill step is successful the step-size is increased by a factor of two. If the downhill step leads to a higher function value then the algorithm backtracks and the step size is decreased using the parameter *tolerance* [43]. No improvements were found in comparison to the simplex treatment.

- The Levenberg-Marquardt algorithm. It is a special method for Non-linear least-squares fitting. The algorithm uses a generalized trust region to keep each step under control. In order to be accepted, a proposed new position  $x'$  must satisfy the condition  $|D(x' - x)| < \lambda$ , where  $D$  is a diagonal scaling matrix and  $\lambda$  is the size of the trust region. The components of  $D$  are computed internally, using the column norms of the Jacobian to estimate the sensitivity of the residual to each component of  $x$ . This improves the behavior



of the algorithm for badly scaled functions. On each iteration the algorithm attempts to minimize the linear system  $|F + Jp|$  subject to the constraint  $|Dp| < \Delta$ . The solution to this constrained linear system is found using the Levenberg-Marquardt method. The proposed step is now tested by evaluating the function at the resulting point,  $x'$ . If the step reduces the norm of the function sufficiently, and follows the predicted behavior of the function within the trust region, then it is accepted and the size of the trust region is increased. If the proposed step fails to improve the solution, or differs significantly from the expected behavior within the trust region, then the size of the trust region is decreased and another trial step is computed [43]. This algorithm led to a huge improvement and allowed to fit the experimental data. At the beginning, only inputs bigger than  $m_\nu^2 = 0.2 \text{ eV}^2$  could be fitted. Therefore, it served for guaranteeing that the correlations between some parameters are extremely strong, specially between the endpoint and the Non V-A factor.

- MIGRAD: This is the MINUIT main algorithm recommended for nearly all functions. It is a variable-metric method with inexact line search, a stable metric updating scheme, and checks for positive-definiteness [44]. No improvements were found in comparison to the Levenberg-Marquardt case.

- MnMinimize and CombinedMinimizer: These are the MINUIT tools which use MIGRAD and call Simplex if MIGRAD fails [44]. No improvements were found in comparison to the Levenberg-Marquardt case.

- The Fumili algorithm: This is a specialized minimizer for Chi-Square and likelihood fits, that uses the property of these functions to estimate the Hessian matrix and in general it converges faster (with less iterations) to the minimum [44]. No improvements were found in comparison to the Levenberg-Marquardt case.

As will be shown later in this chapter, the correlation between  $\delta$  and  $E_0$  is so strong that some of the above algorithms could not find the global minimum of Chi-Square even when  $E_0$  was fixed, i.e., not even when assuming during the fitting that the endpoint is perfectly known.

Fortunately, the Levenberg-Marquardt algorithm is almost able to solve perfectly the problem. Therefore it has been chosen as the proper routine to deal with the Non V-A problem. When working with it, in the case that the input  $m_\nu^2$  is far enough away from a vanishing value, the results are very good. If the input is very close to 0, then it is still possible to find some bias in the solution, although further improvements can be done in this direction.

The solution to this problem came up first for big inputs  $m_\nu^2 \sim 4 \text{ eV}^2$ . The present experimental limit on  $m_\nu^2$  was established by the Mainz experiment [21] around  $5.3 \text{ eV}^2$ , so that a neutrino mass square slightly smaller than this should be measured by KATRIN. It is then interesting to check the reaction of KATRIN from a vanishing neutrino mass to the present experimental limit. Let us have a look to some cases (see Fig. 4.15-4.17).

Plots for Chi-Square,  $R_s$  and  $R_b$  are not shown since they are less relevant and their distributions were as expected. As it was supposed, the correlation between  $E_0$  and  $\delta$  is very strong since their relationship is almost a straight line, which causes the search for the global minimum to be a very difficult technical problem to solve. The correlation between  $E_0$  and  $m_\nu^2$  has also become stronger than the one found within the SM. The standard deviation for  $E_0$  and for  $m_\nu^2$  has grown dramatically. Fortunately, the standard

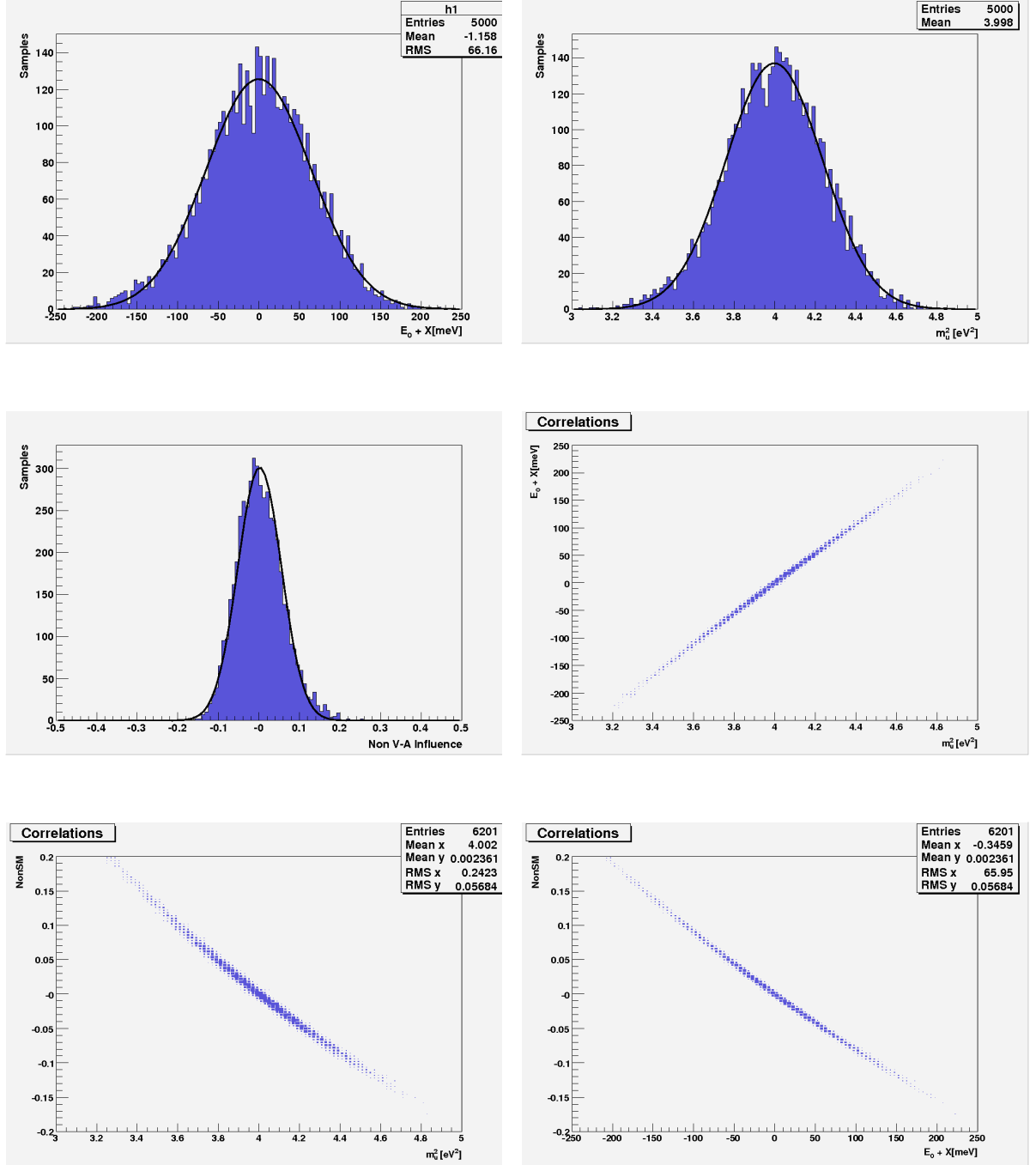


Figure 4.15: MC results in a KATRIN Non V-A analysis for the input  $E_0 = 18575$  eV,  $m_\nu^2 = 4$  eV<sup>2</sup>,  $\delta = 0$ . Distribution for the endpoint (first row, left), the neutrino mass square (first row, right), and the Non V-A parameter (second row, left). Correlation between  $E_0$  and  $m_\nu^2$  (second row, right), between  $m_\nu^2$  and  $\delta$  (third row, left), and between  $E_0$  and  $\delta$  (third row, right).

deviation of  $m_\nu^2$  depends on the actual value of  $m_\nu^2$  as can be seen in Fig. 4.16 and 4.17, where the same results are shown for inputs  $m_\nu^2 = 1 \text{ eV}^2$  and  $m_\nu^2 = 0 \text{ eV}^2$ .

The same behaviour as the one found for the input  $m_\nu^2 = 4 \text{ eV}^2$  occurs for the input  $m_\nu^2 = 1 \text{ eV}^2$ , but with a smaller standard deviation in  $m_\nu^2$ , which means that  $\sigma_{m_\nu^2}$  actually depends on the real value of  $m_\nu^2$ . In the latter case, some bias can be noticed on the  $E_0$  and  $\delta$  distributions, since some points are separated from the ideal Gaussian distribution (black line in Fig. 4.16). The smaller the input in  $m_\nu^2$  is, the more difficult is the search of the global minimum. Thus, for a vanishing neutrino mass one can expect a bigger bias than the one found for  $m_\nu^2 = 1 \text{ eV}^2$ . Results for this case are shown in Fig. 4.17.

It is surprising that the correlations between the three principal variables  $E_0$ ,  $m_\nu^2$  and  $\delta$  have almost been broken when the input corresponds to  $m_\nu^2 = 0 \text{ eV}^2$ , whereas the correlations for other inputs were always very strong. Thus the problem is now solved although more qualitatively than quantitatively and therefore no conclusive numbers on the change of the KATRIN sensitivity can be given. As a guide, the SM case was also solved under the same conditions, the same approximations, by using the analytical integral in order to compare the change in the standard deviation of  $m_\nu^2$ . The Non V-A case led to  $\sigma_{m_\nu^2} = 0.1113$  whereas the SM led to  $\sigma_{m_\nu^2} = 0.1101$ . This means that the KATRIN sensitivity on the neutrino mass is not dramatically changed if the Non V-A physics is considered. For the discovery potential we can not assert the same.

When calculating the discovery potential within the SM,  $\sigma_{m_\nu^2}$  was taken out for an input  $m_\nu^2 = 0 \text{ eV}^2$  and multiplied by a factor of five to get the 99.9999 % C.L. demanded for a discovery potential. We did that because it was known that  $\sigma_{m_\nu^2}$  hardly changes as a function of  $m_\nu^2$  in the SM. Within the Non V-A physics, the discovery potential has to be computed differently since we already know from the MC simulations that  $\sigma_{m_\nu^2}$  grows quite a lot when increasing  $m_\nu^2$ . Thus, to know the exact value of the discovery potential, the package has to be run many times with the purpose of finding inputs which lead to a distribution separated by  $5\sigma$  from  $0 \text{ eV}^2$ . It makes more sense to do that when everything works as desired and approximations can be abandoned. What we can already assert is that the sensitivity of KATRIN does barely change with Non V-A physics, although the KATRIN discovery potential may be significantly worse.

A short study about the little remaining bias in the Levenberg-Marquardt algorithm has been carried out. The point is that this algorithm relies very strongly upon the accuracy of the derivatives of the function to minimize (Chi-Square in our case) with respect to the parameters it depends on. Thus, these derivatives play a vital role on the performance of the algorithm. The more accurate they are, the more reliable the algorithm is and the faster it works. The results given above were obtained by bringing the derivatives (calculated numerically) to the maximum allowed precision of regular compilers and regular computers. This happens because inside computers, numbers can not be stored with infinite precision. The best data type of regular compilers is the double type (long double is not supported in most external libraries) which can store numbers up to  $\pm 10^{308}$  but with limited precision. The maximum number of digits of a cipher in the double data type is around 16, which limits severely the calculation of any numerical derivative since they are calculated by increasing and decreasing the function to derive. The good news in this sense is, that further improvements can be done by working with the so-called Multiple Precision Arithmetic Libraries [45], which allow to calculate the derivatives with even better precision. After including this type of library the little bias in the Levenberg-

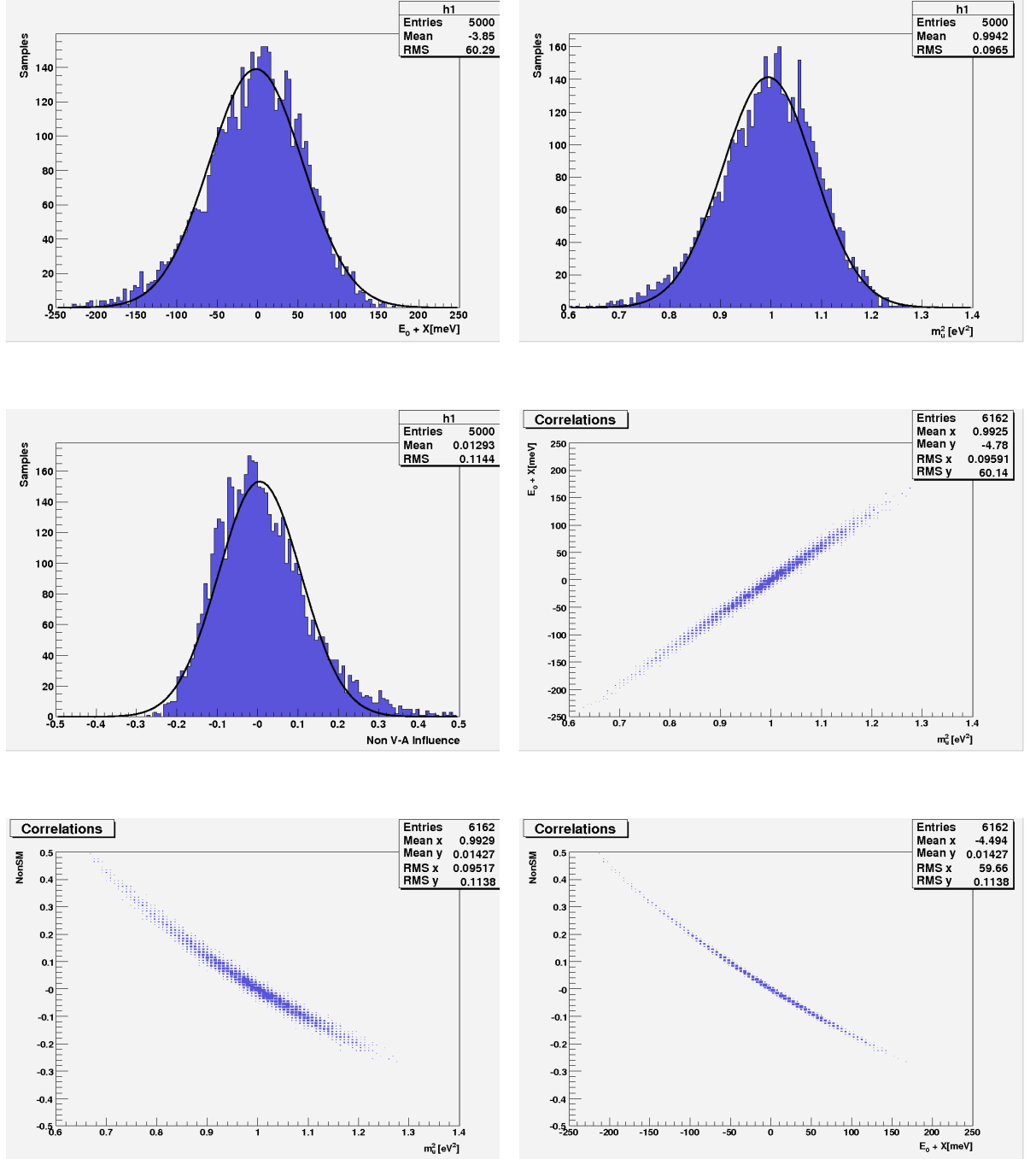


Figure 4.16: MC results in a KATRIN Non V-A analysis for the input  $E_0 = 18575$  eV,  $m_\nu^2 = 1$  eV<sup>2</sup>,  $\delta = 0$ . Distribution for the endpoint (first row, left), the neutrino mass square (first row, right), and the Non V-A parameter (second row, left). Correlation between  $E_0$  and  $m_\nu^2$  (second row, right), between  $m_\nu^2$  and  $\delta$  (third row, left), and between  $E_0$  and  $\delta$  (third row, right).

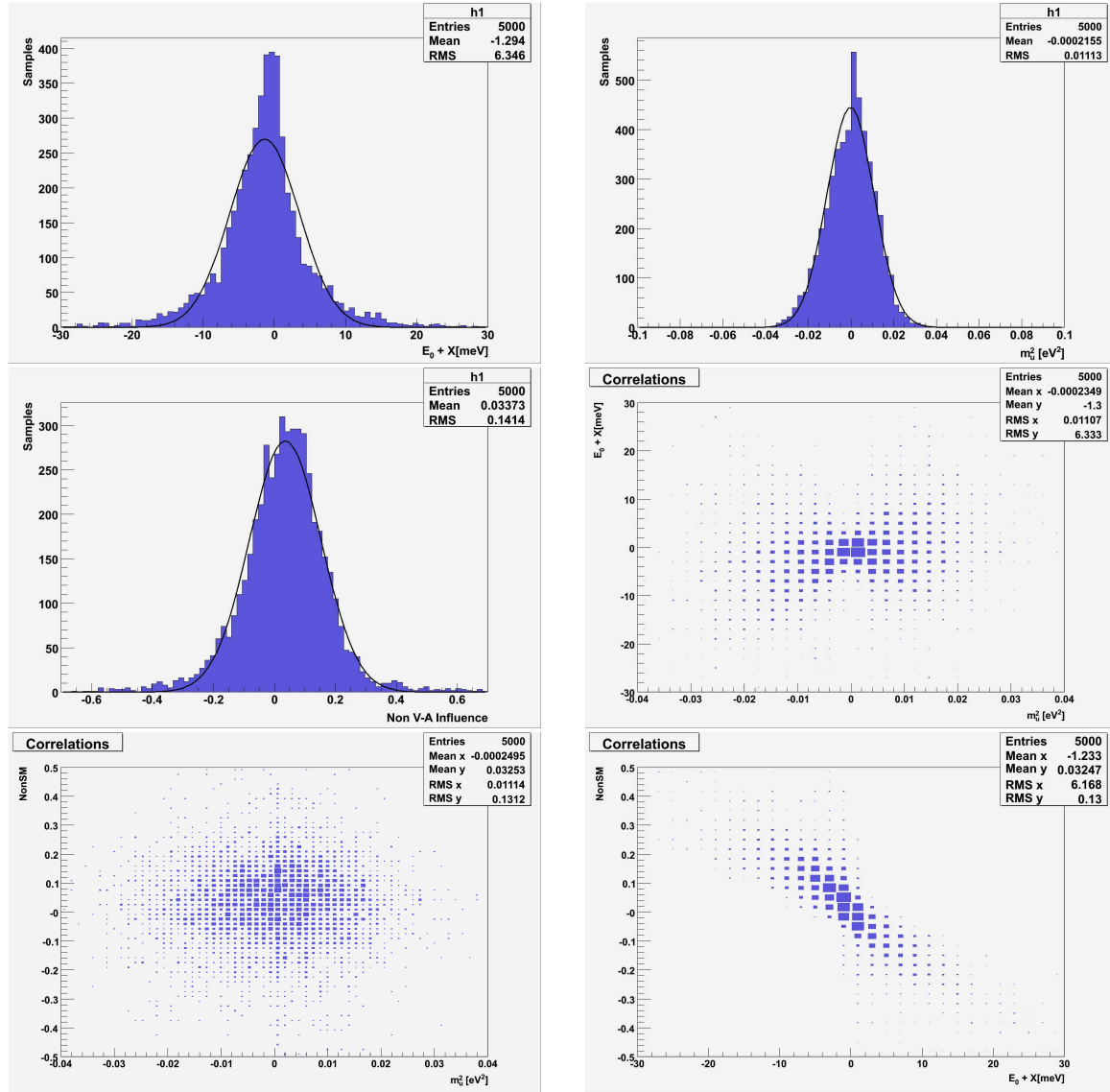


Figure 4.17: MC results in a KATRIN Non V-A analysis for the input  $E_0 = 18575$  eV,  $m_\nu^2 = 0$  eV<sup>2</sup>,  $\delta = 0$ . Distribution for the endpoint (first row, left), the neutrino mass square (first row, right), and the Non V-A parameter (second row, left). Correlation between  $E_0$  and  $m_\nu^2$  (second row, right), between  $m_\nu^2$  and  $\delta$  (third row, left), and between  $E_0$  and  $\delta$  (third row, right).

Marquardt algorithm is expected to disappear and therefore the approximations in the spectrum can be removed by including again the FSD of the daughter molecule, the Fermi and Sirlin functions, and the response function of the experimental setup with which the real KATRIN sensitivity on  $m_\nu^2$  with Non V-A physics could be calculated.

### 4.3.2 Tachyonic neutrinos

Some studies have been done to find out the KATRIN sensitivity in the case neutrinos are tachyons. Also a mixed situation between tachyons and the SM was tried out to know whether it is possible to describe the positive region of  $m_\nu^2$  with the SM and the negative region with tachyons. Finally, the possibility of mixing up both models in order to look for potential systematic errors was analyzed.

All these calculations were done only in the last 20 eV of the spectrum. A systematic study, if necessary, can be still done in deeper ranges. To handle this problem we followed formula (2.34) in which all corrections are included. Therefore, this case was quantitatively solved without approximations. Once more, a suitable description for the negative part of  $m_\nu^2$  must be found. Let us remark that the tachyonic formula for the tritium beta decay spectrum deals with  $\kappa$  instead of  $m_\nu^2$ . This formula makes sense only when  $\kappa^2 > 0$ , which corresponds to  $m_\nu^2 < 0$ , since this model was specifically chosen to describe that range. Thus, one has to invent some description for the  $\kappa^2 < 0$ , which corresponds to  $m_\nu^2 > 0$ , and make it behave continuously with the other part of the function. This was done by following the experience gained on the SM problem, thus by doing it the other way around:

$$\frac{d^2 N}{dE dt} = \begin{cases} K \sum_i W_i \cdot F(\epsilon_i, y) E_{tot} p_e [\epsilon_i^2 + \epsilon_i \sqrt{\epsilon_i^2 + \kappa^2} + \kappa^2] \theta(\epsilon_i) & \kappa^2 \geq 0 \\ K \sum_i W_i \cdot F(\epsilon_i, y) E_{tot} p_e [\epsilon_i^2 + \epsilon_i \sqrt{\epsilon_i^2 + \kappa^2} - \kappa^2] \theta(\epsilon_i - \sqrt{|\kappa^2|}) & \kappa^2 < 0 \end{cases} \quad (4.13)$$

with which the MC method can be built by following once more the above explanation about the Chi-Square method.

Now, the point is to determine the KATRIN sensitivity when creating experimental data with tachyons and analyzing that data with the tachyonic model, i.e., what would happen if neutrinos are actually tachyons. This problem did not show any minimization difficulty and therefore it was solved with the grid algorithm, the simplest one when looking for global minima. Here, the free parameters are  $E_0$ ,  $\kappa^2$ ,  $R_s$ ,  $R_b$ . Their Gaussian distributions which look very much like the SM ones, are shown in Fig. 4.18, which were obtained by using the already typical input  $E_0 = 18575$  eV,  $\kappa^2 = 0$  eV<sup>2</sup>:

The distribution for Chi-Square is of course the expected one, centered in the number of measurements minus the number of free parameters which is a proof of the good performance of the method, the package and the physics involved. Surprisingly, the sensitivity on the tachyonic neutrino mass is better than the sensitivity on the regular neutrino mass. In the tachyonic case,  $\sigma_{\kappa^2} = 0.01108 \text{ eV}^2$ , which would lead for a 90% C.L. to a sensitivity on  $\kappa$  of 0.135 eV, whereas the one found in the SM is 0.185 eV. Hence, the discovery potential would also be improved from 0.32295 eV in the SM to 0.235 eV when handling tachyons. The correlations between parameters look also very much like the ones found in the SM. They are shown in Fig. 4.19.

Concerning correlations between parameters, the only slight difference in comparison to the SM is that the dependence between  $E_0$  and  $\kappa^2$  is a bit stronger than the dependence between  $E_0$  and  $m_\nu^2$ .

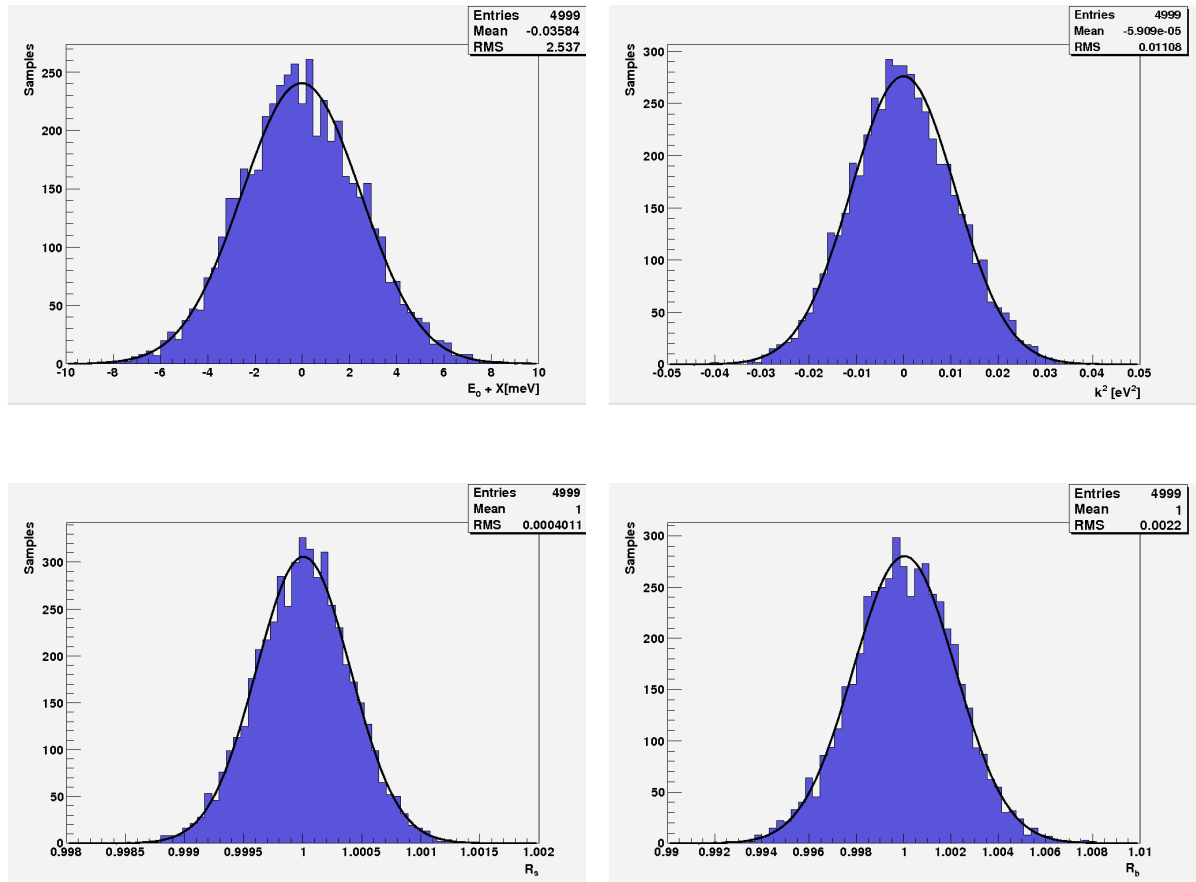


Figure 4.18: MC results in a KATRIN tachyonic analysis for the input  $E_0 = 18575$  eV,  $\kappa^2 = 0$  eV<sup>2</sup>. Distribution for the endpoint (top, left), the tachyonic neutrino mass square (top, right), for  $R_s$  (bottom, left), and for  $R_b$  (bottom, right).

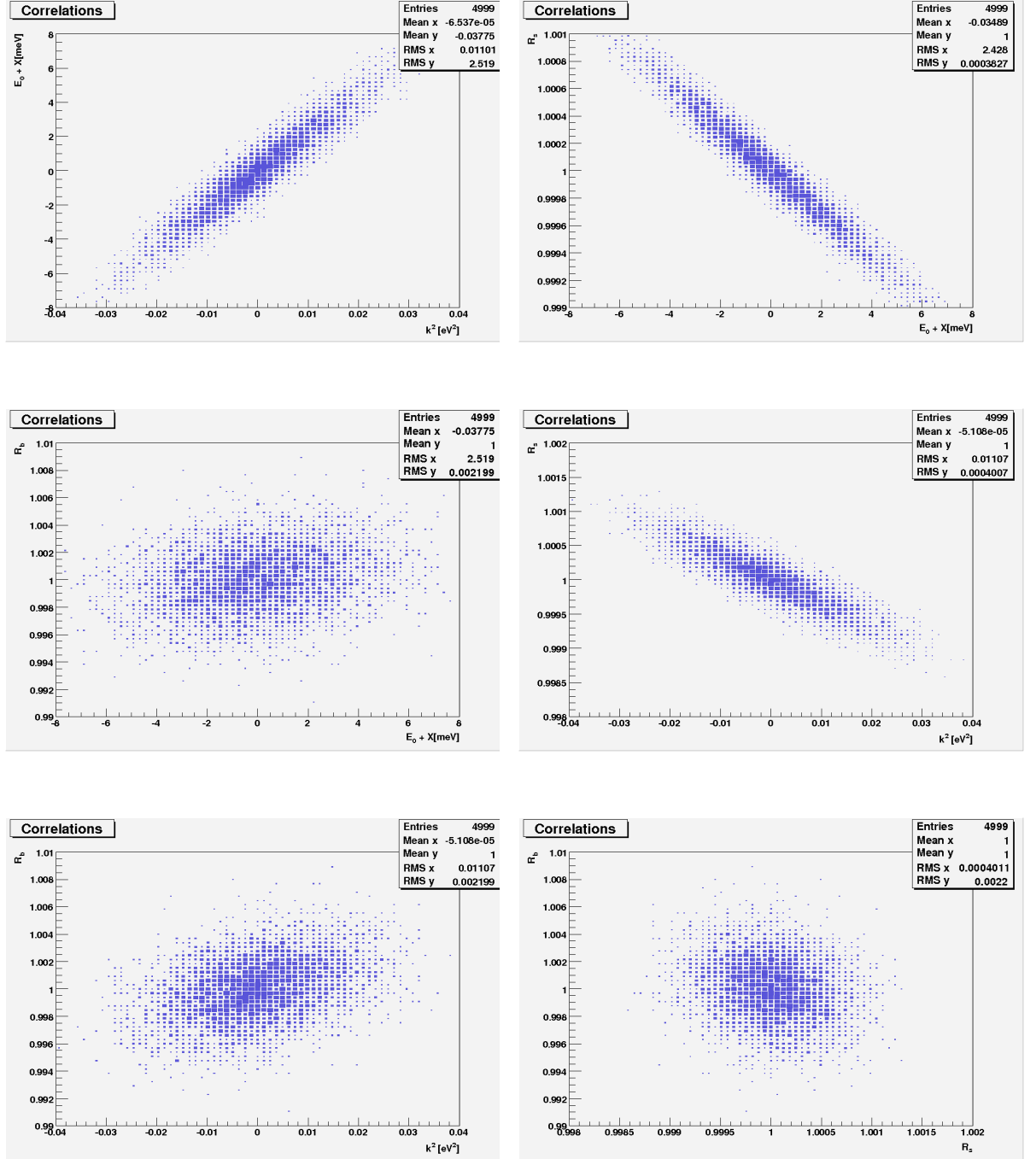


Figure 4.19: MC results in a KATRIN tachyonic analysis for the input  $E_0 = 18575$  eV,  $\kappa^2 = 0$  eV<sup>2</sup>. Correlation between  $E_0$  and  $\kappa^2$  (first row, left), between  $E_0$  and  $R_s$  (first row, right), between  $E_0$  and  $R_b$  (second row, left), between  $\kappa^2$  and  $R_s$  (second row, right), between  $\kappa^2$  and  $R_b$  (third row, left), and between  $R_s$  and  $R_b$  (third row, right).



A further step in the study of the tachyonic tritium beta decay spectrum was the search of possible systematic errors. This was done by creating experimental data with tachyons and analyzing with the SM, i.e., what would happen if neutrinos were actually tachyons and the fitting procedure of KATRIN was done with physics of the SM. In this sense, the solution found by means of the MC simulation is very simple. The result is that by mixing these two models the obtained values of Chi-Square are very large which serves to rule out the theoretical model that is being used to fit the experimental data. This means, that in case neutrinos are actually tachyons and the KATRIN data is fitted with the SM, we would immediatly realize it because of the large Chi-Square. This behaviour in Chi-Square was not found when mixing the SM and the Non V-A model. The tachyonic case offers this kind of advantage. How large must be Chi-Square to be ruled out depends basically on the standards of confidence levels. An expected mean Chi-Square of 27 like our case, reaches the 90% C.L. around 37. The values obtained by mixing the SM and the tachyonic model were much larger and therefore they can be totally ruled out and serve to realize that the theoretical model used to fit the experimental data is wrong.

Tachyonic neutrinos got into the tritium beta decay topic to explain the  $m_\nu^2 < 0$  region. Thus, a final test was needed for tachyons. It made sense to try to mix both SM and tachyons now in the theoretical function used to fit the data. The experimental data is created with the SM and the theoretical functions to fit the data are now split into two parts. For  $m_\nu^2 \geq 0$  the SM is used and for  $m_\nu^2 < 0$  the tachyonic model is used. This combination should work if tachyons are a good explanation for the negative region. Here again huge values of Chi-Square were found. Sufficiently big to be ruled out too, which means that something wrong is being done. Consequently, this kind of trick can not be used.

### 4.3.3 Violation of Lorentz invariance

The violation of Lorentz invariance theory has been very briefly treated in this study. It was implemented into the package but no MC simulations have been run. Hence, the code is already prepared to carry out some study about the changes of the KATRIN sensitivity when taking into account this model. Probably some study about the correlations between its parameters is necessary. Therefore, another algorithm rather than Levenberg-Marquardt routine might be suitable to solve it. To implement this model into the code, a description for the negative region of  $m_\nu^2$  has to be chosen. Note that  $\lambda$  must fulfill  $\lambda \geq 0$ . To obtain the proper Gaussian distribution when the input is  $\lambda = 0$ , a meaningless physical description for the negative region  $\lambda < 0$  has to be invented as well:

If  $\lambda \geq 0$ :

$$\frac{d^2 N}{dE dt} = \begin{cases} K \sum_i W_i \cdot F(\epsilon_i, y) p_e E_{etot} \epsilon_i \left[ (\epsilon_i^2 + \lambda^2 - m_\nu^2)^{\frac{1}{4}} - \frac{\lambda}{(\epsilon_i^2 + \lambda^2 - m_\nu^2)^{\frac{1}{4}}} \right]^2 \theta(\epsilon_i - m_\nu) & m_\nu^2 \geq 0 \\ K \sum_i W_i \cdot F(\epsilon_i, y) p_e E_{etot} \epsilon_i \left[ (\epsilon_i^2 + \lambda^2 + m_\nu^2)^{\frac{1}{4}} - \frac{\lambda}{(\epsilon_i^2 + \lambda^2 + m_\nu^2)^{\frac{1}{4}}} \right]^2 \theta(\epsilon_i) & m_\nu^2 < 0 \end{cases} \quad (4.14)$$

If  $\lambda < 0$ :

$$\frac{d^2 N}{dE dt} = \begin{cases} K \sum_i W_i \cdot F(\epsilon_i, y) p_e E_{etot} \epsilon_i \left[ (\epsilon_i^2 + \lambda^2 - m_\nu^2)^{\frac{1}{4}} + \frac{\lambda}{(\epsilon_i^2 + \lambda^2 - m_\nu^2)^{\frac{1}{4}}} \right]^2 \theta(\epsilon_i - m_\nu) & m_\nu^2 \geq 0 \\ K \sum_i W_i \cdot F(\epsilon_i, y) p_e E_{etot} \epsilon_i \left[ (\epsilon_i^2 + \lambda^2 + m_\nu^2)^{\frac{1}{4}} + \frac{\lambda}{(\epsilon_i^2 + \lambda^2 + m_\nu^2)^{\frac{1}{4}}} \right]^2 \theta(\epsilon_i) & m_\nu^2 < 0 \end{cases} \quad (4.15)$$

## Chapter 5

# Conclusion and Outlook

This work summarizes the influence of some Non SM theoretical approaches of the weak interaction on the tritium beta decay spectrum and specifically on the KATRIN experimental result. A first study was done by involving only SM physics, which led to values near the already known KATRIN sensitivity and discovery potential on  $m_\nu^2$ . The values extracted from the MC simulation developed to reach such objectives among others, presented a little dependence on the studied energy range of the spectrum.

A further step was implemented by analyzing the Non V-A case of Non SM physics. A first study on how a Non V-A tritium beta decay spectrum influences on a SM analysis was carried out. The worst cases, i.e.,  $\delta = +0.3$  and  $\delta = -0.3$ , showed a shift in the neutrino mass measurement of about 20% and 12% respectively, both for an input of  $m_\nu = 0.33$  eV near the KATRIN discovery potential ( $m_\nu = 0.35$  eV). The shift in the  $m_\nu^2$  measurement mentioned above forced us to take into account Non V-A physics. When analyzing this case, the  $\chi^2$  global minimum necessary to fit the data turned out to be very hidden. A special search about the best algorithm to deal with this technical problem was accomplished. The Levenberg-Marquardt routine was claimed to be the proper tool, with which approximate results could be obtained. It was demonstrated that extremely strong correlations between some of the parameters of the fit underlay the problem of searching the global minima. A comparison between the SM case and the Non V-A problem was realized by using the same approximations and routines. Under these considerations, the change on  $\sigma_{m_\nu^2}$  was computed to yield  $\sigma_{m_\nu^2} = 0.1101$  eV<sup>2</sup> in the SM, and  $\sigma_{m_\nu^2} = 0.1113$  eV<sup>2</sup> in the Non V-A model. Therefore, the KATRIN sensitivity is not too affected when considering the Non V-A description of the weak interaction. Concerning the KATRIN discovery potential, it was found that the 0.35 eV limit of the SM might be increased within the Non V-A frame. If so, better experimental limits on the Non V-A coupling constants or the tritium beta decay endpoint could help to bring down again the discovery potential.

Other extensions of the SM were tested. Primarily, the case when neutrinos are tachyons after a  $T_2\beta$  decay. Briefly, the same decay under the assumption of the violation of Lorentz invariance.

The tachyonic model yielded a surprising result, since the sensitivity of the KATRIN experiment on  $\kappa$  ( the neutrino tachyonic mass) is better than the sensitivity on the regular neutrino mass  $m_\nu$ . Regarding the potential systematic error which might occur if neutrinos were really tachyons and the experimental data was fitted with SM physics, the  $\chi^2$  value

of the fits would allow to rule out the theoretical model used to fit the data. A mixture of theoretical models formed by the SM and the tachyonic model was also tested. In this sense, experimental data generated with SM was analyzed with tachyons for  $m_\nu^2 < 0 \text{ eV}^2$ , and with the SM for  $m_\nu^2 \geq 0 \text{ eV}^2$ . For this combination, the  $\chi^2$  value served again to rule out the physics models involved in the theoretical functions.

Finally, the violation of Lorentz invariance description of the  $T_2\beta$  decay was understood as to be implemented into the simulations. Although no MC results have been run, the necessary environment has been developed to allow future researches in this area.

The simulation package written to obtain the results explained in this work can be enlarged to treat other descriptions of the  $T_2\beta$  decay. Among them, a very interesting one concerns the so-called Kaluza-Klein towers [46] which deals with neutrinos in extra dimensions.

More KATRIN experimental features which are independent of the descriptions of the beta decay can also be introduced to quantify with higher accuracy the sensitivity on  $m_\nu$ . Among others, the most appropriate would be the fluctuations of the retarding potential, the fluctuations of the column density as well as the recoil energy of the daughter molecule.

As mentioned before, a quantitative result for the Non V-A theory is still necessary. Presently, this can be easily done by including again all the deactivated corrections in the spectrum. Also, the interaction with the setup must be restored. This calculation should be carried out by means of the Levenberg-Marquardt routine which strongly relies on the precision of the derivatives of  $\chi^2$  with respect to its inner variables (free parameters of the fit). To reach a more enhanced performance in this calculation, it is highly recommended to implement an interaction between this simulation package and a Multiple Precision Arithmetic Library [45].

Finally, the violation of Lorentz invariance treatment of the tritium beta decay is already available to be checked. Although, a preliminary study on the correlations between parameters and another minimization routine could be necessary.

## Appendix A

# Parameters used in the simulations

Parameter	Symbol	Value
Column density	$\rho_c$	$5 \cdot 10^{17}$ molecules/cm <sup>2</sup>
Magnetic field at the source	$B_S$	3.6 T
Magnetic field at the APL	$B_A$	$3 \cdot 10^{-4}$ T
Magnetic field at the pinch	$B_{max}$	6 T
Radius of the fluxtube in WGTS	$R_S$	4.11 cm
Radius of the fluxtube at the APL	$R_A$	450 cm
Maximum accepted angle for outgoing electrons	$\theta_{Max}$	51°
Efficiency of the detector	$\epsilon_{det}$	0.9
T <sub>2</sub> purity	$\epsilon_{T_2}$	0.95
Fermi-Sirlin function coefficient	$a$	1.018983
Fermi-Sirlin function coefficient	$b$	0.0015582
Probability of 0-scattering inside the WGTS	$P_0$	0.413339
Probability of 1-scattering inside the WGTS	$P_1$	0.292658
Probability of 2-scattering inside the WGTS	$P_2$	0.167331
Probability of 3-scattering inside the WGTS	$P_3$	0.079129
Probability of 4-scattering inside the WGTS	$P_4$	0.031776
Energy loss function coefficient	$A_1$	0.204
Energy loss function coefficient	$A_2$	0.0556
Energy loss function coefficient	$\omega_1$	1.85
Energy loss function coefficient	$\omega_2$	12.5
Energy loss function coefficient	$\epsilon_1$	12.6
Energy loss function coefficient	$\epsilon_2$	14.3
Energy loss function coefficient	$\epsilon_c$	14.092115609
Background rate	$\Gamma_b$	0.01 Hz
Tritium beta decay endpoint	$E_0$	18575 eV



# Appendix B

## Simulation code

### B.1 Simulation package with MINUIT algorithms

An implementation of the exact problem by using the MINUIT algorithms of minimization is shown in the following code.

#### B.1.1 Tools.h

```
#include<cstdio>
#include<cstdlib>
#include<iostream>
#include<fstream>
#include<math.h>
#include<iomanip>
#include<time.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
#include <gsl/gsl_math.h>
#include<gsl/gsl_vector.h>
#include<gsl/gsl_multimin.h>

#define sp(x) setprecision(x)

using namespace std;

/*An instance to this class generates the necessary environment to start
up the gaussian randomization. By calling the function "Randomize" a
random number is given back.*/
class Random{
public:
gsl_rng *r;
Random()
{
    const gsl_rng_type * T;
    gsl_rng_env_setup();
```

```

    T = gsl_rng_default;
    r = gsl_rng_alloc(T);
    gsl_rng_set(r,time(NULL));
}
double Randomize(double sigma)
{
return gsl_ran_gaussian(r,sigma);
}
~Random()
{
gsl_rng_free (r);
}
};

/*An instance to this class solves the integral of a function with
variable in array1 and function value in array2. The number of elements
p of both arrays is necessary for the algorithm to work.
The result is stored in the variable "Integral".*/
class Integration{
public:
double Integral;
Integration(double array1[],double array2[],int p)
{
//This algorithm integrates by using Bode's rule
double x[8],y[8];
double h;
int j;
Integral = 0.;
for(j=1;j<=5;j++)
{
x[j] = array1[j];
y[j] = array2[j];
}
y[6] = y[5];
x[6] = x[5];
x[2] = x[2] - x[1];
x[3] = x[3] - x[1];
x[4] = x[4] - x[1];
x[5] = x[5] - x[1];
x[1] = 0.;
h = x[2] - x[1];
Integral += (2./45.)*h*((7.*y[1]) + (32.*y[2]) + (12.*y[3]) +
(32.*y[4]) + (7.*y[5]));

while(j <= p)
{

```



```

x[2] = array1[j+1];
y[2] = array2[j+1];
x[3] = array1[j+2];
y[3] = array2[j+2];
x[4] = array1[j+3];
y[4] = array2[j+3];
x[7] = array1[j+4];
y[7] = array2[j+4];

x[1] = 0.;
y[1] = y[6];
x[2] = x[2] - x[6];
x[3] = x[3] - x[6];
x[4] = x[4] - x[6];
x[5] = x[7] - x[6];
x[6] = x[7];
y[6] = y[7];
y[5] = y[7];
if (x[5] > x[4] && x[4] > x[3] && x[3] > x[2] && x[2] > x[1])
Integral += (2./45.)*h*((7.*y[1]) + (32.*y[2]) + (12.*y[3]) +
(32.*y[4]) + (7.*y[5]));
else
{
if ((x[2] > x[1]) && (x[3] < x[2]))
Integral += h*((y[1]+y[2])/2.);
if ((x[3] > x[2]) && (x[2] > x[1]) && (x[4] < x[3]))
Integral += h*((y[1]+y[2])/2.) + ((y[2]+y[3])/2.);
if ((x[4] > x[3]) && (x[3] > x[2]) && (x[2] > x[1]))
Integral += h*((y[1]+y[2])/2.) + ((y[2]+y[3])/2.) +
((y[3]+y[4])/2.);
}
j +=4;
}
}

~Integration()
{
}
};

```

### B.1.2 ResponseFunction.h

```

#include<cstdio>
#include<cstdlib>
#include<iostream>
#include<fstream>
#include<math.h>

```

```

#include<iomanip>

#define sp(x) setprecision(x)
#define In Integration
#define Il Integral

using namespace std;

/*An instance to this class generates the necessary empirical coefficients
for the calculation of the energy loss function and the response function
of the KATRIN setup*/
class EmpiricalCoefficients{
public:
double P0,P1,P2,P3,P4;
double A1,A2,w1,w2,e1,e2,ec;

EmpiricalCoefficients()
{
//Scattering probabilities:
P0 = 0.413339;
P1 = 0.292658;
P2 = 0.167331;
P3 = 0.079129;
P4 = 0.031776;
//Energy loss function coefficients:
A1 = 0.204;
A2 = 0.0556;
    w1 = 1.85;
w2 = 12.5;
e1 = 12.6;
e2 = 14.3;
ec = 14.092115609;
}
~EmpiricalCoefficients()
{

}
};

/*An instance to this class generates the response function of the KATRIN
setup*/
class ResponseFunction{
public:
EmpiricalCoefficients coe;
TechnicalParameters tec;
double* E;

```

```
double* E2;
double* E3;
double* E4;
double* R;
double* R2;
double* R2_1;
double* R2_2;
double* R3;
double* R3_1;
double* R4;
double* Int;

ResponseFunction()
{
double E_main;
double e,e2,e3,e4;
double I_Step = 250.;
int i,de,c,c2,c3,c4;

E    = new double[300];
E2   = new double[300];
E3   = new double[300];
E4   = new double[300];
R    = new double[300];
R2   = new double[300];
R2_1 = new double[300];
R2_2 = new double[300];
R3   = new double[300];
R3_1 = new double[300];
R4   = new double[300];
Int  = new double[300];

e = 0.;
c = 0;
i = 0;

cout << "Calculating convolution integrals...\n";

while(e <= tec.E_High - tec.E_Low)
{
E[c] = e;
e2 = 0.;
c2 = 0;
while(e2 <= tec.E_High - tec.E_Low)
{
E2[c2] = e2;
```

```

R2[c2] = EL(e-e2)*EL(e2);
e3 = 0.;
c3 = 0;
while(e3 <= tec.E_High - tec.E_Low)
{
E3[c3] = e3;
R3[c3] = EL(e - e3 - e2)*EL(e3)*EL(e2);
e4 = 0.;
c4 = 0;
while(e4 <= tec.E_High - tec.E_Low)
{
E4[c4] = e4;
R4[c4] = EL(e-e4-e3-e2)*EL(e4)*EL(e3)*EL(e2);
e4 += (tec.E_High - tec.E_Low)/I_Step;
c4++;
i++;
}
In Cv4(E4,R4,c4 -1);
R3_1[c3] = Cv4.I1;
c3++;
e3 += (tec.E_High - tec.E_Low)/I_Step;
}
In Cv3_1(E3,R3_1,c3 -1);
In Cv3(E3,R3,c3 -1);
R2_2[c2] = Cv3_1.I1;
R2_1[c2] = Cv3.I1;
c2++;
e2 += (tec.E_High - tec.E_Low)/I_Step;
}
In Cv2_2(E2,R2_2,c2 -1);
In Cv2_1(E2,R2_1,c2 -1);
In Cv2(E2,R2,c2 -1);
R[c] = (coe.P1*EL(e)) + (coe.P2*Cv2.I1) + (coe.P3*Cv2_1.I1)
+ (coe.P4*Cv2_2.I1);
e += (tec.E_High - tec.E_Low)/I_Step;
c++;
cout << sp(5) << i*100./pow(I_Step + 1.,4.) << "% Done"
<< endl;
}
cout << "Convolutions calculation -----> Done\n";

c--;
e = 0.;
ofstream Response("ResponseFunction.dat");

for(E_main=tec.E_Low;E_main<=tec.E_High;E_main+=tec.Accuracy)

```

```

{
if(E_main == tec.E_Low)
{
(Response) << endl << "0." << " " << "0.";
}
else
{
de = 0;
while(de <= c)
{
Int[de] = T(E_main-E[de],tec.E_Low)*R[de];
if(E[de] > E_main)
break;
de++;
}
In I(E,Int,de);
(Response) << endl << sp(5) << E_main - tec.E_Low
<< " " << sp(10)
<< ((coe.P0*T(E_main,tec.E_Low)) + I.I1);
}
cout << sp(10) << E_main << endl;
}

Response.close();
}

double T(double x, double y)
{
//Transmission function
double TFunction;
double Delta_E;

Delta_E = x*tec.B_A/tec.B_Max;
if(x - y < 0.)
TFunction = 0.;
if((x - y >= 0. ) && (x - y <= Delta_E))
TFunction = (1. - sqrt(1. - ((x -y)*tec.B_S/(x*tec.B_A))))/
(1. - sqrt(1. - (Delta_E*tec.B_S/(x*tec.B_A))));
if(x - y > Delta_E)
TFunction = 1.;
return TFunction;
}

double EL(double x)
{
//EnergyLoss function

```

```

double Function;

if(x<=0.)
Function = 0.;
if(x>=0. && x < coe.ec)
Function = coe.A1*exp(-2.*pow(x - coe.e1,2.)/pow(coe.w1,2.));
if(x >= coe.ec)
Function = coe.A2*pow(coe.w2,2.)/(pow(coe.w2,2.)+
        4.*pow(x - coe.e2,2.));
return Function;
}
~ResponseFunction()
{
delete Int;
Int = NULL;
delete R4;
R4 = NULL;
delete R3_1;
R3_1 = NULL;
delete R3;
R3 = NULL;
delete R2_2;
R2_2 = NULL;
delete R2_1;
R2_1 = NULL;
delete R2;
R2 = NULL;
delete R;
R = NULL;
delete E4;
E4 = NULL;
delete E3;
E3 = NULL;
delete E2;
E2 = NULL;
delete E;
E = NULL;
}
};

```

### B.1.3 DifferentialSpectrum.h

```

#include<cstdio>
#include<cstdlib>
#include<iostream>
#include<fstream>

```

```
#include<math.h>
#include<iomanip>

#define sp(x) setprecision(x)

using namespace std;

/*An instance to this class generates experimental features, the
accuracy with which the spectrum will be calculated as well as the
starting points for the minimization .*/

class TechnicalParameters{
public:

    int Experiments;
    double B_S;
    double B_Max;
    double B_A;
    double rho;
    double r_A;
    double e_det;
    double e_T2;
    double E_Low;
    double E_High;
    double EMax;
    double EMin;
    double Accuracy;
    double Gamma_b;
    double m_nu2_Starting_Minuit;
    double k2_Starting_Minuit;
    double E_0_Starting_Minuit;
    double R_s_Starting_Minuit;
    double R_b_Starting_Minuit;
    double NonVA_Starting_Minuit;
    double Lambda_Starting_Minuit;

    TechnicalParameters()
    {
        Experiments = 5000;
        B_S = 3.6;
        B_Max = 6.;
        B_A = 3.e-4;
        rho = 10.e17;
        r_A = 450.;
        e_det = 0.9;
        e_T2 = 0.95;
```

```

E_Low = 18490.;
E_High = 18590.9;
EMax = 18585.;
EMin = 18525.;
Accuracy = 0.01;
Gamma_b = 0.01;
E_0_Starting_Minuit = 18577.;
m_nu2_Starting_Minuit = -5.;
k2_Starting_Minuit = 5.;
R_s_Starting_Minuit = 1.5;
R_b_Starting_Minuit = 1.5;
NonVA_Starting_Minuit = 0.;
Lambda_Starting_Minuit = 0.;
}
~TechnicalParameters()
{
}
};

/*An instance to this class creates the set of free parameters
independently of the theoretical approach.*/
class FreeParameters{
public:
double m_nu2;
double k2;
double E_0;
double R_s;
double R_b;
double NonVA;
double Lambda;

FreeParameters()
{

}
~FreeParameters()
{

}
};

/*An instance to this class creates the necessary fundamental constants
which will be necessary in the package*/
class PhysicalConstants{
public:
double e;

```



```

double G_f;
double Cos_Cab;
double M;
double c;
double m_e;
double h_bar;
double Alpha;
double g_A;
int Z;

PhysicalConstants()
{ e = 1.60217653e-19;
  G_f = e*89.620e-45;
  Cos_Cab = 0.9750;
  M = 2.348;
  c = 299792458.;
  m_e = 9.1093826e-31;
h_bar = 1.05457168e-34;
  Alpha = 7.297352568e-3;
g_A = 1.2673;
  Z = 2;
}
~PhysicalConstants()
{
}
};

/*This class generates the constant K for the spectrum.
Two constants K and KTachyons are differentiated*/
class K{
public:
double Constant;
PhysicalConstants a;
TechnicalParameters b;

K()
{
double Theta_Max;
Theta_Max = asin(sqrt((b.B_S)/(b.B_Max)));
Constant = b.rho*M_PI*pow(b.r_A,2.)*(b.B_A/(b.B_S))*
(b.e_det)*(b.e_T2)*((1.-(cos(Theta_Max)))/2.)*
pow(a.G_f,2.)*pow(a.m_e,5.)*pow(a.c,4.)*pow(a.Cos_Cab,2.)
*pow(a.M,2.)/(2.*pow(M_PI,3.)*pow(a.h_bar,7.));
}

void KTachyons(double TACHYONS)

```

```

{
double Theta_Max;
Theta_Max = asin(sqrt((b.B_S)/(b.B_Max)));
Constant = b.rho*M_PI*pow(b.r_A,2.)*(b.B_A/(b.B_S))*
(b.e_det)*(b.e_T2)*((1.-(cos(Theta_Max)))/2.)*
pow(a.G_f,2.)*pow(a.m_e,5.)*pow(a.c,4.)*(1. + (3.*pow(a.g_A,2.)))
/(4.*pow(M_PI,3.)*pow(a.h_bar,7.));
}

~K()
{

}

};

/*An instance to this class generates the differential spectrum for a set of
free parameters. All the studied theoretical approaches are here gathered.
The spectrum is stored in three arrays DS, DS2, DS3.*/
class DifferentialSpectrum{
public:
K k;
K kT;
double* E_energy;
double* P_energy;
double* DS;
double* DS2;
double* DS3;
double* PARTIAL;
int max,energylevels;

DifferentialSpectrum()
{
int y;
double EVolts,Energy;
PARTIAL = new double[11000];
DS = new double[11000];
DS2 = new double[11000];
DS3 = new double[11000];
E_energy = new double[300];
P_energy = new double[300];

kT.KTachyons(1.);

ifstream rEnergy("EnergyLevels.dat");
energylevels = 0;
while(!rEnergy.eof())

```

```

{
(rEnergy) >> E_energy[energylevels] >> P_energy[energylevels];
energylevels++;
}
energylevels--;
rEnergy.close();
for(y=0;y<=(k.b.E_High - k.b.E_Low)/k.b.Accuracy;y++)
{
EVolts = k.b.E_Low + (y*k.b.Accuracy);
if(EVolts > k.b.E_High)
break;
else
{
DS[y] = EVolts;
DS2[y] = (EVolts*k.a.e)/((k.a.m_e)*pow(k.a.c,2.));
Energy = (EVolts*k.a.e);
PARTIAL[y] = KTerms(Energy);
}
}
y--;
max = y;
}

double KTerms(double E_e)
{
double E_T,p;
E_T = ((k.a.m_e*pow(k.a.c,2.))+(E_e));
p = sqrt(pow(E_T,2.)-(pow(k.a.m_e,2.)*pow(k.a.c,4.)))/k.a.c;
E_T = E_T/(k.a.m_e*pow(k.a.c,2.));
p = p/((k.a.m_e)*(k.a.c));
return p*E_T;
}

double Fermi(double E_e,double E_nu,double mnu,FreeParameters f)
{
//Fermi-Sirlin function from paper:
//"Direct Measurement of the Neutrino Mass, KÄ½ndig et al."

double Gamma,v_e,Beta,y;
Gamma = 1. + ((E_e)/((k.a.m_e)*pow(k.a.c,2.)));
v_e = sqrt(pow(k.a.c,2.)*(pow(Gamma,2.)-1.)/pow(Gamma,2.));
Beta = v_e/(k.a.c);
y = ((k.a.Z)*(k.a.Alpha))/Beta;

if(f.m_nu2 >= 0.)
{

```

```

if(E_nu - mnu <= 0.)
{
    return 0.;
}
else
{
    return 2.*M_PI*y*(pow((1. - exp(-2.*M_PI*y)), -1.))*
        (1. + (2.*k.a.Alpha*pow(Beta,2.)/(3.*M_PI))*
        log(2.*E_nu))*(1.018983 - (0.0015582/Beta));
}
}

if(f.m_nu2 < 0.)
{
    if(E_nu <= 0.)
    {
        return 0.;
    }
    else
    {
        return 2.*M_PI*y*(pow((1. - exp(-2.*M_PI*y)), -1.))*
            (1. + (2.*k.a.Alpha*pow(Beta,2.)/(3.*M_PI))*
            log(2.*fabs(E_nu)))*(1.018983 - (0.0015582/Beta));
    }
}
}

void EventsCalculator(double SM, FreeParameters f)
{
    int m;
    for(m=0; m<=max; m++)
    {
        DS3[m] = PARTIAL[m]*PTerms((DS[m]*k.a.e), E_energy, P_energy,
        energylevels, f);
    }
}

void EventsCalculator(FreeParameters f, int NON_VA)
{
    int m;
    for(m=0; m<=max; m++)
    {
        DS3[m] = PARTIAL[m]*PTerms((DS[m]*k.a.e), E_energy, P_energy,
        energylevels, f, NON_VA);
    }
}

```

```

void EventsCalculator(FreeParameters f,double TACHYONS)
{
  int m;
  for(m=0;m<=max;m++)
  {
    DS3[m] = PARTIAL[m]*PTerms((DS[m]*k.a.e),E_energy,P_energy,
    energylevels,f,TACHYONS);
  }
}

void EventsCalculator(FreeParameters f,double TACHYONS,int ANTISIMETRY)
{
  int m;
  for(m=0;m<=max;m++)
  {
    DS3[m] = PARTIAL[m]*PTerms((DS[m]*k.a.e),E_energy,P_energy,
    energylevels,f,TACHYONS,ANTISIMETRY);
  }
}

void EventsCalculator(FreeParameters f,double VIOLATION,double LORENTZ)
{
  int m;
  for(m=0;m<=max;m++)
  {
    DS3[m] = PARTIAL[m]*PTerms((DS[m]*k.a.e),E_energy,P_energy,
    energylevels,f,VIOLATION,LORENTZ);
  }
}

double PTerms(double E_e,double E_i[],double P_i[],int levels,
FreeParameters f)
{
  double PSum,ptot;
  double Enu;
  double mass;
  double root;

  PSum = 0.;
  ptot = 0.;

  for(int i=0;i<=levels;i++)
  {
    PSum += P_i[i];
    Enu = (k.a.e*(f.E_0 - E_i[i])) - E_e;

```

```

mass = sqrt(fabs(f.m_nu2));
mass = mass*k.a.e/(k.a.m_e*pow(k.a.c,2.));
Enu = Enu/(k.a.m_e*pow(k.a.c,2.));

if(f.m_nu2 >= 0.)
{
if(Enu - mass <= 0.)
{
ptot += 0.;
}
else
{
root = sqrt(fabs(pow(Enu,2.) - pow(mass,2.)));
ptot += k.Constant*Fermi(E_e,Enu,mass,f)*
P_i[i]*Enu*root;
}
}
if(f.m_nu2 < 0.)
{
if(Enu <= 0.)
{
ptot += 0.;
}
else
{
root = sqrt(fabs(pow(Enu,2.) + pow(mass,2.)));
ptot += k.Constant*Fermi(E_e,Enu,mass,f)*
P_i[i]*Enu*root;
}
}
}
return ptot/PSum;
}

double PTerms(double E_e,double E_i[],double P_i[],int levels,
FreeParameters f,int NON_VA)
{
double PSum,ptot;
double Enu;
double correction,mass;
double root;

PSum = 0.;
ptot = 0.;

for(int i=0;i<=levels;i++)

```

```

{
PSum += P_i[i];
Enu = (k.a.e*(f.E_0 - E_i[i])) - E_e;
mass = sqrt(fabs(f.m_nu2));

mass = mass*k.a.e/(k.a.m_e*pow(k.a.c,2.));
Enu = Enu/(k.a.m_e*pow(k.a.c,2.));

correction = 1. + (f.NonVA*mass/Enu);

if(f.m_nu2 >= 0.)
{
if(Enu - mass <= 0.)
{
ptot += 0.;
}
else
{
root = sqrt(fabs(pow(Enu,2.) - pow(mass,2.)));
ptot += k.Constant*Fermi(E_e,Enu,mass,f)*
P_i[i]*Enu*root*correction;
}
}
if(f.m_nu2 < 0.)
{
if(Enu <= 0.)
{
ptot += 0.;
}
else
{
root = sqrt(fabs(pow(Enu,2.) + pow(mass,2.)));
ptot += k.Constant*Fermi(E_e,Enu,mass,f)*
P_i[i]*Enu*root*correction;
}
}
}
return ptot/PSum;
}

double PTerms(double E_e,double E_i[],double P_i[],int levels,
FreeParameters f,double TACHYONS)
{
double PSum,ptot;
double Enu;
double mass;

```

```

double root;

PSum = 0.;
ptot = 0.;

for(int i=0;i<=levels;i++)
{
PSum += P_i[i];
Enu = (k.a.e*(f.E_0 - E_i[i])) - E_e;
mass = sqrt(fabs(f.k2));

mass = mass*k.a.e/(k.a.m_e*pow(k.a.c,2.));
Enu = Enu/(k.a.m_e*pow(k.a.c,2.));

if(f.k2 >= 0.)
{
if(Enu <= 0.)
{
ptot += 0.;
}
else
{
root = sqrt(fabs(pow(Enu,2.) + pow(mass,2.)));
ptot += kT.Constant*Fermi(E_e,Enu,mass,f)*
P_i[i]*(pow(Enu,2.) + (Enu*root) + pow(mass,2.));
}
}
if(f.k2 < 0.)
{
if(Enu - mass <= 0.)
{
ptot += 0.;
}
else
{
root = sqrt(fabs(pow(Enu,2.) - pow(mass,2.)));
ptot += kT.Constant*Fermi(E_e,Enu,mass,f)*
P_i[i]*(pow(Enu,2.) + (Enu*root) - pow(mass,2.));
}
}
}
return ptot/PSum;
}

double PTerms(double E_e,double E_i[],double P_i[],int levels,
FreeParameters f,double TACHYONS,int ANTISIMETRY)

```



```

{
double PSum,ptot;
double Enu;
double mass;
double root;

PSum = 0.;
ptot = 0.;

for(int i=0;i<=levels;i++)
{
PSum += P_i[i];
Enu = (k.a.e*(f.E_0 - E_i[i])) - E_e;
mass = sqrt(fabs(f.k2));

mass = mass*k.a.e/(k.a.m_e*pow(k.a.c,2.));
Enu = Enu/(k.a.m_e*pow(k.a.c,2.));

if(f.k2 >= 0.)
{
if(Enu <= 0.)
{
ptot += 0.;
}
else
{
root = sqrt(fabs(pow(Enu,2.) + pow(mass,2.)));
ptot += kT.Constant*Fermi(E_e,Enu,mass,f)*P_i[i]*
(pow(Enu,2.) + (Enu*root) + pow(mass,2.));
}
}
if(f.k2 < 0.)
{
if(Enu - mass <= 0.)
{
ptot += 0.;
}
else
{
root = sqrt(fabs(pow(Enu,2.) - pow(mass,2.)));
ptot += k.Constant*Fermi(E_e,Enu,mass,f)*P_i[i]*Enu*root;
}
}
}
return ptot/PSum;
}

```

```

double PTerms(double E_e,double E_i[],double P_i[],int levels,
FreeParameters f,double VIOLATION,double LORENTZ)
{
double PSum,ptot;
double Enu;
double mass;
double FourthRoot;

PSum = 0.;
ptot = 0.;

for(int i=0;i<=levels;i++)
{
PSum += P_i[i];
Enu = (k.a.e*(f.E_0 - E_i[i])) - E_e;
mass = sqrt(fabs(f.m_nu2));
mass = mass*k.a.e/(k.a.m_e*pow(k.a.c,2.));
Enu = Enu/(k.a.m_e*pow(k.a.c,2.));

if(f.Lambda >= 0.)
{
if(f.m_nu2 >= 0.)
{
if(Enu - mass <= 0.)
{
ptot += 0.;
}
else
{
FourthRoot = pow(fabs(pow(Enu,2.) +
pow(f.Lambda,2.) - f.m_nu2),0.25);
ptot += k.Constant*Fermi(E_e,Enu,mass,f)*P_i[i]*
Enu*pow(FourthRoot - (f.Lambda/FourthRoot),2.);
}
}
if(f.m_nu2 < 0.)
{
if(Enu <= 0.)
{
ptot += 0.;
}
else
{
FourthRoot = pow(fabs(pow(Enu,2.) +
pow(f.Lambda,2.) + f.m_nu2),0.25);

```

```

ptot += k.Constant*Fermi(E_e,Enu,mass,f)*P_i[i]*
Enu*pow(FourthRoot - (f.Lambda/FourthRoot),2.);
}
}
}
else
{
if(f.m_nu2 >= 0.)
{
if(Enu - mass <= 0.)
{
ptot += 0.;
}
else
{
FourthRoot = pow(fabs(pow(Enu,2.) +
pow(f.Lambda,2.) - f.m_nu2),0.25);
ptot += k.Constant*Fermi(E_e,Enu,mass,f)*P_i[i]*
Enu*pow(FourthRoot + (f.Lambda/FourthRoot),2.);
}
}
if(f.m_nu2 < 0.)
{
if(Enu <= 0.)
{
ptot += 0.;
}
}
else
{
FourthRoot = pow(fabs(pow(Enu,2.) +
pow(f.Lambda,2.) + f.m_nu2),0.25);
ptot += k.Constant*Fermi(E_e,Enu,mass,f)*P_i[i]
*Enu*pow(FourthRoot + (f.Lambda/FourthRoot),2.);
}
}
}
}
return ptot/PSum;
}

```

```

~DifferentialSpectrum()
{
delete PARTIAL;
PARTIAL = NULL;

```

```

delete P_energy;
P_energy = NULL;
delete E_energy;
E_energy = NULL;
delete DS3;
DS3 = NULL;
delete DS2;
DS2 = NULL;
delete DS;
DS = NULL;
}
};

```

#### B.1.4 IntegratedSpectrum.h

```

#include<cstdio>
#include<cstdlib>
#include<iostream>
#include<fstream>
#include<math.h>
#include<iomanip>

#define sp(x) setprecision(x)

using namespace std;

/*An instance to this class prepares some differential spectrum to be
integrated by using also the response function of the setup.
Afterwards, the functions "LoadDifSpectra" can be instantiated to
load a specific theoretial approach of the spectrum. Finally, the
function "Integrate" integrates the spectrum from the retarding
potential to the endpoint.*/

class IntegratedSpectrum{
public:
DifferentialSpectrum dif;
double Integrated;
double* RF1;
double* RF2;
double* A;
double* B;
int j;

IntegratedSpectrum()
{

```

```
RF1 = new double[11000];
RF2 = new double[11000];
A = new double[11000];
B = new double[11000];
ifstream rRes("ResponseFunction.dat");
if(!rRes.is_open())
{
    cout << "Error while opening file\n";
}
else
{
    {
        j = 0;
        while(!rRes.eof())
        {
            (rRes) >> RF1[j] >> RF2[j];
            j++;
        }
        rRes.close();
    }
    j--;
}

void LoadDifSpectra(double SM,FreeParameters free)
{
    dif.EventsCalculator(1.,free);
}

void LoadDifSpectra(FreeParameters free,int NON_VA)
{
    dif.EventsCalculator(free,NON_VA);
}

void LoadDifSpectra(FreeParameters free,double TACHYONS)
{
    dif.EventsCalculator(free,TACHYONS);
}

void LoadDifSpectra(FreeParameters free,double TACHYONS,int ANTISIMETRY)
{
    dif.EventsCalculator(free,TACHYONS,ANTISIMETRY);
}

void LoadDifSpectra(FreeParameters free,double VIOLATION,double LORENTZ)
{
    dif.EventsCalculator(free,VIOLATION,LORENTZ);
}
```

```

void Integrate(double Wall)
{
    int k,l,m;
    k = 0;
    l = 0;
    m = 1;
    for(;;)
    {
        if(dif.DS[k] >= Wall)
        {
            A[m] = dif.DS2[k];
            B[m] = dif.DS3[k]*RF2[l];
            l++;
            m++;
        }
        if(k == j)
            break;
        k++;
    }
    Integration I(A,B,m-1);
    Integrated = I.Integral;
}

~IntegratedSpectrum()
{
    delete B;
    B = NULL;
    delete A;
    A = NULL;
    delete RF2;
    RF2 = NULL;
    delete RF1;
    RF1 = NULL;
}
};

```

### B.1.5 DataGenerator.h

```

#include<cstdio>
#include<cstdlib>
#include<iostream>
#include<fstream>
#include<time.h>
#include<vector>

```

```

using namespace std;

/*An instance to this class creates experimental data. The data is spread
in a Gaussian way by means of the function "Calculator". The retarding
potential an measuring time with which the measurements are carried out
are stored in the arrays "thePotentials" and "theTimes", respectively.
The experimantal measurement is stored in "theMeasurements".*/

class DataGenerator{
private:
std::vector<double> thePotentials;
    std::vector<double> theMeasurements;
    std::vector<double> theTimes;

public:
double trash[2];
Random rnd;
TechnicalParameters T;
IntegratedSpectrum Int;
FreeParameters F;
std::vector<double> theIntegrals;

DataGenerator(double E_0,double m_nu2,double R_s,double R_b,double NONVA,
double LAMBDA,char *E_Range,int OPTION)
{
F.E_0 = E_0;
F.m_nu2 = m_nu2;
F.k2 = -m_nu2;
F.NonVA = NONVA;
F.Lambda = LAMBDA;
F.R_s = R_s;
F.R_b = R_b;

switch(OPTION)
{
case 0:
//SM
Int.LoadDifSpectra(1.,F);
break;
case 1:
//NON SM, NON V-A
Int.LoadDifSpectra(F,1);
break;
case 2:
//NON SM, TACHYONS, SIMETRIC AROUND 0eV
Int.LoadDifSpectra(F,1.);

```

```

break;
case 3:
//NON SM, TACHYONS, ANTISIMETRIC AROUND 0eV
Int.LoadDifSpectra(F,1.,1);
break;
case 4:
//NON SM, VIOLATION OF LORENTZ INVARIANCE
Int.LoadDifSpectra(F,1.,1.);
break;
}
ifstream rTime(E_Range);
while(!rTime.eof())
{
(rTime) >> trash[0] >> trash[1];
if(trash [1] != 0.)
{
thePotentials.push_back(trash[0]);
theTimes.push_back(trash[1]);
Int.Integrate(trash[0]);
theIntegrals.push_back(trash[1]*((R_s*Int.Integrated) +
(R_b*T.Gamma_b)));
}
}
rTime.close();
}

void Calculator()
{

theMeasurements.clear();
for(int i = 0;i < theIntegrals.size();i++)
{
theMeasurements.push_back(theIntegrals[i] +
rnd.Randomize(sqrt(theIntegrals[i])));
}
}

std::vector<double> potentials() const
{
return thePotentials;
}

std::vector<double> measurements() const
{
return theMeasurements;
}

std::vector<double> timedis() const

```



```
{
return theTimes;
}
```

```
~DataGenerator()
{

}
};
```

### B.1.6 Function.h

```
#include<cstdlib>
#include<cstdio>
#include<iostream>
#include<math.h>
#include<cassert>
#include "Minuit/FCNBase.h"
#include<vector>
#include<fstream>
#include<iomanip>

#define sp(x) setprecision(x)

using namespace std;

/*This class only plays a bit with the free parameters to prepare the
necessary environment of MINUIT*/
class myFunction{
public:
TechnicalParameters TFunction;
FreeParameters fFunction;
IntegratedSpectrum IFunction;

myFunction(double E_0,double m_nu2,double Rs,double Rb)
{
fFunction.E_0 = E_0;
fFunction.m_nu2 = m_nu2;
fFunction.R_s = Rs;
fFunction.R_b = Rb;
IFunction.LoadDifSpectra(1.,fFunction);
}

myFunction(double E_0,double m_nu2,double Rs,double Rb,double NONVA)
{
fFunction.E_0 = E_0;
fFunction.m_nu2 = m_nu2;
```

```

fFunction.R_s = Rs;
fFunction.R_b = Rb;
fFunction.NonVA = NONVA;
IFunction.LoadDifSpectra(fFunction,1);
}

myFunction(double E_0,double k2,double Rs,double Rb,int TACHYONS)
{
fFunction.E_0 = E_0;
fFunction.k2 = k2;
fFunction.R_s = Rs;
fFunction.R_b = Rb;
IFunction.LoadDifSpectra(fFunction,1.);
}

myFunction(double E_0,double k2,double Rs,double Rb,int TACHYONS,
int ANTISIMETRY)
{
fFunction.E_0 = E_0;
fFunction.k2 = k2;
fFunction.R_s = Rs;
fFunction.R_b = Rb;
IFunction.LoadDifSpectra(fFunction,1.,1);
}

myFunction(double E_0,double m_nu2,double Rs,double Rb,double LAMBDA,
int VIOLATION_LORENTZ)
{
fFunction.E_0 = E_0;
fFunction.m_nu2 = m_nu2;
fFunction.R_s = Rs;
fFunction.R_b = Rb;
fFunction.Lambda = LAMBDA;
IFunction.LoadDifSpectra(fFunction,1.,1.);
}

~myFunction()
{

}

double operator()(double Wall)
{
IFunction.Integrate(Wall);
return (fFunction.R_s*IFunction.Integrated)+
(fFunction.R_b*TFunction.Gamma_b);
}

```

```
};

/*This class calculates Chi-Square as required by MINUIT. The same
implementation is repeated some times to allow the analysis of the
same experimental data be done by means of different theoretical approaches*/
class TheFunctionSM : public FCNBase
{
public:

TheFunctionSM(const std::vector<double>& meas,
const std::vector<double>& pot, const std::vector<double>& times):
theMeasurements(meas),thePotentials(pot),theTimes(times),theErrorDef(1.)
{

}
~TheFunctionSM()
{

}

virtual double up() const
{
return theErrorDef;
}
virtual double operator()(const std::vector<double>&) const;

std::vector<double> measurements() const
{
return theMeasurements;
}
std::vector<double> potentials() const
{
return thePotentials;
}
std::vector<double> timedis() const
{
return theTimes;
}
void setErrorDef(double def)
{
theErrorDef = def;
}

private:

std::vector<double> theMeasurements;
```

```

std::vector<double> thePotentials;
std::vector<double> theTimes;
double theErrorDef;
};

double TheFunctionSM::operator()(const std::vector<double>& par) const
{
    double chi2;
    assert(par.size() == 4);
    myFunction f(par[0],par[1],par[2],par[3]);

    chi2 = 0.;
    for(unsigned int n = 0;n < theMeasurements.size();n++)
    {
        chi2 += pow(((theTimes[n]*f(thePotentials[n])) - theMeasurements[n])/
sqrt(theTimes[n]*f(thePotentials[n])),2.);
    }
    return chi2;
}

class TheFunctionNONVA : public FCNBase
{
public:

    TheFunctionNONVA(const std::vector<double>& meas,
const std::vector<double>& pot, const std::vector<double>& times):
theMeasurements(meas),thePotentials(pot),theTimes(times),
theErrorDef(1.)
    {

    }

    ~TheFunctionNONVA()
    {

    }

    virtual double up() const
    {
        return theErrorDef;
    }

    virtual double operator()(const std::vector<double>&) const;

    std::vector<double> measurements() const
    {
        return theMeasurements;
    }
}

```

```

std::vector<double> potentials() const
{
    return thePotentials;
}
std::vector<double> timedis() const
{
    return theTimes;
}
void setErrorDef(double def)
{
    theErrorDef = def;
}

private:

std::vector<double> theMeasurements;
std::vector<double> thePotentials;
std::vector<double> theTimes;
double theErrorDef;
};

double TheFunctionNONVA::operator()(const std::vector<double>&par) const
{
    double chi2;
    assert(par.size() == 5);
    myFunction f(par[0],par[1],par[2],par[3],par[4]);

    chi2 = 0.;
    for(unsigned int n = 0;n < theMeasurements.size();n++)
    {
        chi2 += pow(((theTimes[n]*f(thePotentials[n])) - theMeasurements[n])/
sqrt(theTimes[n]*f(thePotentials[n])),2.);
    }
    return chi2;
}

class TheFunctionTACHYONS : public FCNBase
{
public:

TheFunctionTACHYONS(const std::vector<double>& meas,
const std::vector<double>& pot, const std::vector<double>& times):
theMeasurements(meas),thePotentials(pot),theTimes(times),theErrorDef(1.)
{

}

```

---

```

~TheFunctionTACHYONS()
{

}

virtual double up() const
{
return theErrorDef;
}

virtual double operator()(const std::vector<double>&) const;

std::vector<double> measurements() const
{
return theMeasurements;
}

std::vector<double> potentials() const
{
return thePotentials;
}

std::vector<double> timedis() const
{
return theTimes;
}

void setErrorDef(double def)
{
theErrorDef = def;
}

private:

std::vector<double> theMeasurements;
std::vector<double> thePotentials;
std::vector<double> theTimes;
double theErrorDef;
};

double TheFunctionTACHYONS::operator()(const std::vector<double>&par) const
{
double chi2;
assert(par.size() == 4);
myFunction f(par[0],par[1],par[2],par[3],1);

chi2 = 0.;
for(unsigned int n = 0;n < theMeasurements.size();n++)
{
chi2 += pow(((theTimes[n]*f(thePotentials[n])) - theMeasurements[n])/

```

```
sqrt(theTimes[n]*f(thePotentials[n])),2.);
}
return chi2;
}

class TheFunctionTACHYONS_ANTISIMETRIC : public FCNBase
{
public:

TheFunctionTACHYONS_ANTISIMETRIC(const std::vector<double>& meas,
const std::vector<double>& pot, const std::vector<double>& times):
theMeasurements(meas),thePotentials(pot),theTimes(times),theErrorDef(1.)
{

}
~TheFunctionTACHYONS_ANTISIMETRIC()
{

}

virtual double up() const
{
return theErrorDef;
}
virtual double operator()(const std::vector<double>&) const;

std::vector<double> measurements() const
{
return theMeasurements;
}
std::vector<double> potentials() const
{
return thePotentials;
}
std::vector<double> timedis() const
{
return theTimes;
}
void setErrorDef(double def)
{
theErrorDef = def;
}

private:

std::vector<double> theMeasurements;
```

```

std::vector<double> thePotentials;
std::vector<double> theTimes;
double theErrorDef;
};

double TheFunctionTACHYONS_ANTISIMETRIC::
operator()(const std::vector<double>& par) const
{
double chi2;
assert(par.size() == 4);
myFunction f(par[0],par[1],par[2],par[3],1,1);

chi2 = 0.;
for(unsigned int n = 0;n < theMeasurements.size();n++)
{
chi2 += pow(((theTimes[n]*f(thePotentials[n])) - theMeasurements[n])/
sqrt(theTimes[n]*f(thePotentials[n])),2.);
}
return chi2;
}

class TheFunctionVIOLATION_LORENTZ : public FCNBase
{
public:

TheFunctionVIOLATION_LORENTZ(const std::vector<double>& meas,
const std::vector<double>& pot, const std::vector<double>& times):
theMeasurements(meas),thePotentials(pot),theTimes(times),theErrorDef(1.)
{

}
~TheFunctionVIOLATION_LORENTZ()
{

}

virtual double up() const
{
return theErrorDef;
}
virtual double operator()(const std::vector<double>&) const;

std::vector<double> measurements() const
{
return theMeasurements;
}

```



```

std::vector<double> potentials() const
{
    return thePotentials;
}
std::vector<double> timedis() const
{
    return theTimes;
}
void setErrorDef(double def)
{
    theErrorDef = def;
}

private:

std::vector<double> theMeasurements;
std::vector<double> thePotentials;
std::vector<double> theTimes;
double theErrorDef;
};

double TheFunctionVIOLATION_LORENTZ::operator()(const std::
vector<double>&par) const
{
    double chi2;
    assert(par.size() == 5);
    myFunction f(par[0],par[1],par[2],par[3],par[4],1);

    chi2 = 0.;
    for(unsigned int n = 0;n < theMeasurements.size();n++)
    {
        chi2 += pow(((theTimes[n]*f(thePotentials[n])) - theMeasurements[n])/
sqrt(theTimes[n]*f(thePotentials[n])),2.);
    }
    return chi2;
}

```

### B.1.7 Fitting.h

```

#include<cstdio>
#include<cstdlib>
#include<iostream>
#include<fstream>
#include<math.h>
#include<iomanip>
#include<strings.h>

```

```

#include "Minuit/FunctionMinimum.h"
#include "Minuit/MnUserParameterState.h"
#include "Minuit/MnPrint.h"
#include "Minuit/MnMigrad.h"
#include "Minuit/MnMinimize.h"
#include "Minuit/MnMinos.h"
#include "Minuit/MnContours.h"
#include "Minuit/MnPlot.h"
#include "Minuit/MnStrategy.h"
#include "Minuit/FumiliMinimizer.h"
#include "Minuit/MnMachinePrecision.h"
#include "Minuit/CombinedMinimizer.h"

#define sp(x) setprecision(x)

using namespace std;

/*By means of this class the Monte Carlo simulation is carried out.
The same experiment is repeated many times (see Tfitting.Experiments)
and the result is recorded in the "wChi" file. The fitted data is obtained
by finding the global minimum of the Chi-Square function. The global
minimum search is done by the MINUIT package, which needs to interact with
another programs in a very strict way. Many of the lines of this file are
just needed by MINUIT. The class allows to fit with any theoretical
approach, although some combinations of experimental data and theoretical
description could lead to unpleasant behaviours.*/
class Fitting{

public:
    TechnicalParameters Tfitting;
    MnUserParameters upar;

    Fitting(double E_0,double m_nu2,double R_s,double R_b,double NONVA,
    double LAMBDA,char *E_Range,int OPTION_1,int OPTION_2)
    {
        DataGenerator gdg(E_0,m_nu2,R_s,R_b,NONVA,LAMBDA,E_Range,OPTION_1);

        char sChi[128] = "ChiMin";
        strcat(sChi,E_Range);
        ofstream wChi(sChi);

        std::vector<double> pos = gdg.potentials();
        std::vector<double> var = gdg.timedis();

        MnStrategy st(2);

```

```

if(OPTION_2 == 0)
{
upar.add("E_0",TFitting.E_0_Starting_Minuit,1.e-10);
upar.add("m_nu2",TFitting.m_nu2_Starting_Minuit,1.e-10);
upar.add("Rs",TFitting.R_s_Starting_Minuit,1.e-10);
upar.add("Rb",TFitting.R_b_Starting_Minuit,1.e-10);

upar.setLowerLimit("E_0",18570.);

for(int j=1;j<=TFitting.Experiments;j++)
{
gdg.Calculator();
std::vector<double> meas = gdg.measurements();

TheFunctionSM theFCN(meas,pos,var);
MnMinimize minimize(theFCN,upar,st);

FunctionMinimum min1 = minimize();
minimize.fix("E_0");
FunctionMinimum min4 = minimize();
minimize.release("E_0");
FunctionMinimum min5 = minimize();
minimize.fix("E_0");
FunctionMinimum min6 = minimize();
minimize.release("E_0");
minimize.fix("m_nu2");
FunctionMinimum min15 = minimize();
minimize.release("m_nu2");
FunctionMinimum min16 = minimize();
minimize.fix("m_nu2");
FunctionMinimum min17 = minimize();
minimize.release("m_nu2");
FunctionMinimum min22 = minimize();
minimize.fix("E_0");
FunctionMinimum min23 = minimize();
minimize.release("E_0");
FunctionMinimum min24 = minimize();
minimize.fix("E_0");
FunctionMinimum min25 = minimize();
minimize.release("E_0");
FunctionMinimum min = minimize();

if(min.isValid())
{
(wChi) << endl << sp(10) << min.fval() << "  "
<< sp(10) << min.userState().value("E_0") << "  "

```

```

<< sp(10) << min.userState().value("m_nu2") << " "
<< sp(10) << sqrt(fabs(min.userState().value("m_nu2")))
<< " " << sp(10) << min.userState().value("Rs")
<< " " << sp(10) << min.userState().value("Rb");
}
}
}
if(OPTION_2 == 1)
{
upar.add("E_0",TFitting.E_0_Starting_Minuit,1.e-10);
upar.add("m_nu2",TFitting.m_nu2_Starting_Minuit,1.e-10);
upar.add("Rs",TFitting.R_s_Starting_Minuit,1.e-10);
upar.add("Rb",TFitting.R_b_Starting_Minuit,1.e-10);
upar.add("NONVA",TFitting.NonVA_Starting_Minuit,1.e-10);

upar.setLowerLimit("E_0",18570.);

for(int j=1;j<=TFitting.Experiments;j++)
{
gdg.Calculator();
std::vector<double> meas = gdg.measurements();

TheFunctionNONVA theFCN(meas,pos,var);
MnMinimize minimize(theFCN,upar,st);
minimize.fix("NONVA");
FunctionMinimum min1 = minimize();
minimize.release("NONVA");
minimize.fix("E_0");
FunctionMinimum min2 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
FunctionMinimum min3 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min4 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
minimize.fix("m_nu2");
FunctionMinimum min5 = minimize();
minimize.release("m_nu2");
FunctionMinimum min6 = minimize();
minimize.fix("m_nu2");
FunctionMinimum min7 = minimize();
minimize.release("m_nu2");
FunctionMinimum min8 = minimize();
minimize.fix("E_0");

```

```
minimize.release("NONVA");
FunctionMinimum min9 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
FunctionMinimum min10 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min11 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
FunctionMinimum min12 = minimize();
minimize.release("NONVA");
minimize.fix("E_0");
FunctionMinimum min13 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
FunctionMinimum min14 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min15 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
minimize.fix("m_nu2");
FunctionMinimum min16 = minimize();
minimize.release("m_nu2");
FunctionMinimum min17 = minimize();
minimize.fix("m_nu2");
FunctionMinimum min18 = minimize();
minimize.release("m_nu2");
FunctionMinimum min19 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min20 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
FunctionMinimum min21 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min22 = minimize();
minimize.release("E_0");

minimize.fix("NONVA");
FunctionMinimum min23 = minimize();
minimize.release("NONVA");
minimize.fix("E_0");
FunctionMinimum min24 = minimize();
```

```
minimize.release("E_0");
minimize.fix("NONVA");
FunctionMinimum min25 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min26 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
minimize.fix("m_nu2");
FunctionMinimum min27 = minimize();
minimize.release("m_nu2");
FunctionMinimum min28 = minimize();
minimize.fix("m_nu2");
FunctionMinimum min29 = minimize();
minimize.release("m_nu2");
FunctionMinimum min30 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min31 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
FunctionMinimum min32 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min33 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
FunctionMinimum min34 = minimize();
minimize.release("NONVA");
minimize.fix("E_0");
FunctionMinimum min35 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
FunctionMinimum min36 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min37 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
minimize.fix("m_nu2");
FunctionMinimum min38 = minimize();
minimize.release("m_nu2");
FunctionMinimum min39 = minimize();
minimize.fix("m_nu2");
FunctionMinimum min40 = minimize();
minimize.release("m_nu2");
```

```

FunctionMinimum min41 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min42 = minimize();
minimize.release("E_0");
minimize.fix("NONVA");
FunctionMinimum min43 = minimize();
minimize.fix("E_0");
minimize.release("NONVA");
FunctionMinimum min44 = minimize();
minimize.release("E_0");
FunctionMinimum min = minimize();

if(min.isValid())
{
(wChi) << endl << sp(10) << min.fval() << "  "
<< sp(10) << min.userState().value("E_0") << "  "
<< sp(10) << min.userState().value("m_nu2") << "  "
<< sp(10) << sqrt(fabs(min.userState().value("m_nu2")))
<< "  " << sp(10) << min.userState().value("Rs")
<< "  " << sp(10) << min.userState().value("Rb")
<< "  " << sp(10) << min.userState().value("NONVA");
}
}
}
if(OPTION_2 == 2)
{
upar.add("E_0",TFitting.E_0_Starting_Minuit,1.e-10);
upar.add("k2",TFitting.k2_Starting_Minuit,1.e-10);
upar.add("Rs",TFitting.R_s_Starting_Minuit,1.e-10);
upar.add("Rb",TFitting.R_b_Starting_Minuit,1.e-10);

upar.setLowerLimit("E_0",18570.);

for(int j=1;j<=TFitting.Experiments;j++)
{
gdg.Calculator();
std::vector<double> meas = gdg.measurements();

TheFunctionTACHYONS theFCN(meas,pos,var);
MnMinimize minimize(theFCN,upar,st);

FunctionMinimum min1 = minimize();
minimize.fix("E_0");
FunctionMinimum min4 = minimize();
minimize.release("E_0");

```

```

FunctionMinimum min5 = minimize();
minimize.fix("E_0");
FunctionMinimum min6 = minimize();
minimize.release("E_0");
minimize.fix("k2");
FunctionMinimum min15 = minimize();
minimize.release("k2");
FunctionMinimum min16 = minimize();
minimize.fix("k2");
FunctionMinimum min17 = minimize();
minimize.release("k2");
FunctionMinimum min22 = minimize();
minimize.fix("E_0");
FunctionMinimum min23 = minimize();
minimize.release("E_0");
FunctionMinimum min24 = minimize();
minimize.fix("E_0");
FunctionMinimum min25 = minimize();
minimize.release("E_0");
FunctionMinimum min = minimize();

if(min.isValid())
{
    (wChi) << endl << sp(10) << min.fval() << " "
    << sp(10) << min.userState().value("E_0") << " "
    << sp(10) << min.userState().value("k2") << " "
    << sp(10) << sqrt(fabs(min.userState().value("k2")))
    << " " << sp(10) << min.userState().value("Rs")
    << " " << sp(10) << min.userState().value("Rb");
}
}
}
if(OPTION_2 == 3)
{
    upar.add("E_0",TFitting.E_0_Starting_Minuit,1.e-10);
    upar.add("k2",TFitting.k2_Starting_Minuit,1.e-10);
    upar.add("Rs",TFitting.R_s_Starting_Minuit,1.e-10);
    upar.add("Rb",TFitting.R_b_Starting_Minuit,1.e-10);

    upar.setLowerLimit("E_0",18570.);

    for(int j=1;j<=TFitting.Experiments;j++)
    {
        gdg.Calculator();
        std::vector<double> meas = gdg.measurements();
    }
}

```



```

TheFunctionTACHYONS_ANTISIMETRIC theFCN(meas,pos,var);
MnMinimize minimize(theFCN,upar,st);

FunctionMinimum min1 = minimize();
minimize.fix("E_0");
FunctionMinimum min4 = minimize();
minimize.release("E_0");
FunctionMinimum min5 = minimize();
minimize.fix("E_0");
FunctionMinimum min6 = minimize();
minimize.release("E_0");
minimize.fix("k2");
FunctionMinimum min15 = minimize();
minimize.release("k2");
FunctionMinimum min16 = minimize();
minimize.fix("k2");
FunctionMinimum min17 = minimize();
minimize.release("k2");
FunctionMinimum min22 = minimize();
minimize.fix("E_0");
FunctionMinimum min23 = minimize();
minimize.release("E_0");
FunctionMinimum min24 = minimize();
minimize.fix("E_0");
FunctionMinimum min25 = minimize();
minimize.release("E_0");
FunctionMinimum min = minimize();

if(min.isValid())
{
(wChi) << endl << sp(10) << min.fval() << " "
<< sp(10) << min.userState().value("E_0") << " "
<< sp(10) << min.userState().value("k2") << " "
<< sp(10) << sqrt(fabs(min.userState().value("k2")))
<< " " << sp(10) << min.userState().value("Rs")
<< " " << sp(10) << min.userState().value("Rb");
}
}
}
if(OPTION_2 == 4)
{
upar.add("E_0",TFitting.E_0_Starting_Minuit,1.e-10);
upar.add("m_nu2",TFitting.m_nu2_Starting_Minuit,1.e-10);
upar.add("Rs",TFitting.R_s_Starting_Minuit,1.e-10);
upar.add("Rb",TFitting.R_b_Starting_Minuit,1.e-10);
upar.add("LAMBDA",TFitting.Lambda_Starting_Minuit,1.e-10);

```

```

upar.setLowerLimit("E_0",18570.);

for(int j=1;j<=TFitting.Experiments;j++)
{
gdg.Calculator();
std::vector<double> meas = gdg.measurements();

TheFunctionVIOLATION_LORENTZ theFCN(meas,pos,var);
MnMinimize minimize(theFCN,upar,st);
minimize.fix("LAMBDA");
FunctionMinimum min1 = minimize();
minimize.release("LAMBDA");
minimize.fix("E_0");
FunctionMinimum min4 = minimize();
minimize.release("E_0");
minimize.fix("LAMBDA");
FunctionMinimum min5 = minimize();
minimize.fix("E_0");
minimize.release("LAMBDA");
FunctionMinimum min6 = minimize();
minimize.release("E_0");
minimize.fix("LAMBDA");
minimize.fix("m_nu2");
FunctionMinimum min15 = minimize();
minimize.release("m_nu2");
FunctionMinimum min16 = minimize();
minimize.fix("m_nu2");
FunctionMinimum min17 = minimize();
minimize.release("m_nu2");
FunctionMinimum min22 = minimize();
minimize.fix("E_0");
minimize.release("LAMBDA");
FunctionMinimum min23 = minimize();
minimize.release("E_0");
minimize.fix("LAMBDA");
FunctionMinimum min24 = minimize();
minimize.fix("E_0");
minimize.release("LAMBDA");
FunctionMinimum min25 = minimize();
minimize.release("E_0");
FunctionMinimum min = minimize();

if(min.isValid())
{
(wChi) << endl << sp(10) << min.fval() << "  "

```

```

<< sp(10) << min.userState().value("E_0") << "  "
<< sp(10) << min.userState().value("m_nu2") << "  "
<< sp(10) << sqrt(fabs(min.userState().value("m_nu2")))
<< "  " << sp(10) << min.userState().value("Rs") << "  "
<< sp(10) << min.userState().value("Rb")
<< "  " << sp(10) << min.userState().value("LAMBDA");
}
}
}

```

```

wChi.close();
}
~Fitting()
{

}
};

```

/\*The following class simply creates the necessary environment to allow the above analysis in different energy ranges of the spectrum. From 20 eV to 50 eV.\*/

```

class FitAllEnergyRanges{
public:

FitAllEnergyRanges(double E_0,double m_nu2,double R_s,double R_b,
double NONVA,double LAMBDA,int OPTION_1,int OPTION_2)
{
if(OPTION_1 <= 4 && OPTION_2 <= 4)
{
Fitting fit1(E_0,m_nu2,R_s,R_b,NONVA,LAMBDA,"Time20eV.dat",
OPTION_1,OPTION_2);
Fitting fit2(E_0,m_nu2,R_s,R_b,NONVA,LAMBDA,"Time25eV.dat",
OPTION_1,OPTION_2);
Fitting fit3(E_0,m_nu2,R_s,R_b,NONVA,LAMBDA,"Time30eV.dat",
OPTION_1,OPTION_2);
Fitting fit4(E_0,m_nu2,R_s,R_b,NONVA,LAMBDA,"Time40eV.dat",
OPTION_1,OPTION_2);
Fitting fit5(E_0,m_nu2,R_s,R_b,NONVA,LAMBDA,"Time50eV.dat",
OPTION_1,OPTION_2);
}
else
{
cout << endl << "Error from -FitAllEnergyRanges class-
\nEither OPTION_1 or OPTION_2 are not allowed!\n\n";
}
}
}

```

```

~FitAllEnergyRanges()
{

}

};

```

### B.1.8 Spectrum.cpp

```

#include<cstdio>
#include<cstdlib>
#include<iostream>
#include<fstream>
#include<math.h>
#include<iomanip>
#include<DifferentialSpectrum.h>
#include<Tools.h>
#include<ResponseFunction.h>
#include<IntegratedSpectrum.h>
#include<DataGenerator.h>
#include<Function.h>
#include<Fitting.h>

using namespace std;

/*Everything can be handled from here. By changing the numbers
in "FitAllEnergyRanges" different analysis are ordered.*/
int main()
{
//ResponseFunction r;
FitAllEnergyRanges fit(18575.,2.,1.,1.,0.,0.,1,1);
return 0;
}

```

## B.2 Simulation of the Non V-A approximate description

In the following, it is shown the code for the implementation of the Non V-A with approximations. The Levenberg-Marquardt algorithm as written in the GSL library is used.

### B.2.1 Levenberg.cpp

```

#include<cstdio>
#include<cstdlib>
#include<iostream>
#include<fstream>
#include<math.h>

```

```

#include<cassert>
#include<vector>
#include<iomanip>
#include<gsl/gsl_rng.h>
#include<gsl/gsl_randist.h>
#include<gsl/gsl_vector.h>
#include<gsl/gsl_math.h>
#include<gsl/gsl_deriv.h>
#include<gsl/gsl_blas.h>
#include<gsl/gsl_multifit_nlin.h>

#define sp(x) setprecision(x)

using namespace std;

double IntegrateGSL(double Wall,double E_0,double m_nu2,double NonSM)
{
//Calculation of the analytical integral
static double B_S = 3.6;
static double B_Max = 6.;
static double B_A = 3.e-4;
static double rho = 10.e17;
static double r_A = 450.;
static double e_det = 0.9;
static double e_T2 = 0.95;
static double e = 1.60217653e-19;
static double G_f = 89.620e-45;
static double Cos_Cab = 0.9750;
static double M = 2.348;
static double c = 299792458.;
static double m_e = 0.501e6;
static double h_bar = 6.58211913765e-16;
static int Z = 2;
double static Constant;
double static Theta_Max;
double emax;
double emin;
double nmass;
double result;
double E_T,momentum;
double root;
nmass = sqrt(fabs(m_nu2));

Theta_Max = asin(sqrt((B_S)/(B_Max)));
Constant = rho*M_PI*pow(r_A,2.)*(B_A/(B_S))*
(e_det)*(e_T2)*((1.-(cos(Theta_Max)))/2.)*

```

```

pow(G_f,2.)*pow(Cos_Cab,2.)
*pow(M,2.)/(2.*pow(M_PI,3.)*pow(c,6.)*pow(h_bar,7.));

E_T = m_e + 18560.;
momentum = sqrt(pow(E_T,2.)- pow(m_e,2.));
emax = E_0 - Wall;
if(m_nu2 > 0.)
{
emin = sqrt(m_nu2);
if(E_0 - nmass <= Wall)
{
result = 0.;
}

else
{
root = sqrt(pow(emax,2.) - m_nu2);
result = Constant*E_T*momentum*((pow(root,3.)/3.) +
(NonSM*nmass*((emax*root) - (m_nu2*log(emax + root)) +
(m_nu2*log(nmass)))/2.));
}
}
if(m_nu2 < 0.)
{
emin = sqrt(fabs(m_nu2));
if(E_0 <= Wall)
{
result = 0.;
}

else
{
root = sqrt(pow(emax,2.) - m_nu2);
result = Constant*E_T*momentum*((pow(root,3.) -
pow(-m_nu2,3./2.))/3.) + (NonSM*nmass*((emax*root) -
(m_nu2*log(emax + root)) + (m_nu2*log(nmass)))/2.));
if(result < 0.)
result = 0.;
}
}
if(m_nu2 == 0.)
{
emin = sqrt(m_nu2);
if(E_0 - nmass <= Wall)
{
result = 0.;
}
}

```

```

}

else
{
result = Constant*E_T*momentum*(((pow(pow(emax,2.),3./2.)/3.)));
}
}
return result;
}

class Random{
//Randomization
public:
gsl_rng *r;
Random()
{
    const gsl_rng_type * T;
    gsl_rng_env_setup();
    T = gsl_rng_default;
    r = gsl_rng_alloc(T);
    gsl_rng_set(r,time(NULL));
}
double Randomize(double sigma)
{
return gsl_ran_gaussian(r,sigma);
}
~Random()
{
gsl_rng_free (r);
}
};

class DataGenerator{
//Creation of the simulated experimental data
public:
std::vector<double> thePotentials;
    std::vector<double> theMeasurements;
    std::vector<double> theTimes;
Random rnd;

DataGenerator()
{

}

void Calculator(double E0,double mnu2,double Rs,double Rb,double bprimed)

```

```

{
double trash[2];
double I,I2;
static double Gamma_b = 0.01;
thePotentials.clear();
theMeasurements.clear();
theTimes.clear();
ifstream rTime("Time20eV.dat");
while(!rTime.eof())
{
(rTime) >> trash[0] >> trash[1];
if(trash [1] != 0.)
{
thePotentials.push_back(trash[0]);
I2 = (trash[1]*((Rs*IntegrateGSL(trash[0],E0,mnu2,bprimed))
+ (Rb*Gamma_b)));
I = I2 + rnd.Randomize(sqrt(I2));
theMeasurements.push_back(I);
theTimes.push_back(trash[1]);
}
}
rTime.close();
}
~DataGenerator()
{

}
};

struct data{
//Structure needed by the GSL library
size_t NumberMeas;
double *Wall;
double *Times;
double *ExpData;
};

int expb_f(const gsl_vector *x,void *params,gsl_vector *f)
{
//Setting the retarding potentials, times, measurements.
double E_0 = gsl_vector_get(x,0);
double m_nu2 = gsl_vector_get(x,1);
double Rs = gsl_vector_get(x,2);
double Rb = gsl_vector_get(x,3);
double bprime = gsl_vector_get(x,4);
static double Gamma_b = 0.01;

```



```

double I_theo;

size_t i;
size_t NumberMeas = ((struct data*)params)->NumberMeas;
double *Wall = ((struct data*)params)->Wall;
double *Times = ((struct data*)params)->Times;
double *ExpData = ((struct data*)params)->ExpData;

for(i=0;i < NumberMeas;i++)
{
I_theo = Times[i]*((Rs*IntegrateGSL(Wall[i],E_0,m_nu2,bprime))
+ (Rb*Gamma_b));
gsl_vector_set(f,i,(ExpData[i] - I_theo)/sqrt(I_theo));
}
return GSL_SUCCESS;
}

double my_f0(double x,void *params)
{
//Function necessary to calculate the Jacobian of Chi-Square
double *points = (double*)params;
double E_0 = x;
double m_nu2 = points[1];
double Rs = points[2];
double Rb = points[3];
double bprime = points[4];
double Wall = points[5];
double Times = points[6];
double ExpData = points[7];

static double Gamma_b = 0.01;
double I_theo;

I_theo = Times*((Rs*IntegrateGSL(Wall,E_0,m_nu2,bprime)) + (Rb*Gamma_b));
return (ExpData - I_theo)/sqrt(I_theo);
}

double my_f1(double x,void *params)
{
//Function necessary to calculate the Jacobian of Chi-Square
double *points = (double*)params;
double E_0 = points[0];
double m_nu2 = x;
double Rs = points[2];
double Rb = points[3];
double bprime = points[4];

```

```
double Wall = points[5];
double Times = points[6];
double ExpData = points[7];

static double Gamma_b = 0.01;
double I_theo;

I_theo = Times*((Rs*IntegrateGSL(Wall,E_0,m_nu2,bprime)) + (Rb*Gamma_b));
return (ExpData - I_theo)/sqrt(I_theo);
}

double my_f2(double x,void *params)
{
//Function necessary to calculate the Jacobian of Chi-Square
double *points = (double*)params;
double E_0 = points[0];
double m_nu2 = points[1];
double Rs = x;
double Rb = points[3];
double bprime = points[4];
double Wall = points[5];
double Times = points[6];
double ExpData = points[7];

static double Gamma_b = 0.01;
double I_theo;

I_theo = Times*((Rs*IntegrateGSL(Wall,E_0,m_nu2,bprime)) + (Rb*Gamma_b));
return (ExpData - I_theo)/sqrt(I_theo);
}

double my_f3(double x,void *params)
{
//Function necessary to calculate the Jacobian of Chi-Square
double *points = (double*)params;
double E_0 = points[0];
double m_nu2 = points[1];
double Rs = points[2];
double Rb = x;
double bprime = points[4];
double Wall = points[5];
double Times = points[6];
double ExpData = points[7];

static double Gamma_b = 0.01;
double I_theo;
```

```

I_theo = Times*((Rs*IntegrateGSL(Wall,E_0,m_nu2,bprime)) + (Rb*Gamma_b));
return (ExpData - I_theo)/sqrt(I_theo);
}

```

```

double my_f4(double x,void *params)
{
//Function necessary to calculate the Jacobian of Chi-Square
double *points = (double*)params;
double E_0 = points[0];
double m_nu2 = points[1];
double Rs = points[2];
double Rb = points[3];
double bprime = x;
double Wall = points[5];
double Times = points[6];
double ExpData = points[7];

```

```

static double Gamma_b = 0.01;
double I_theo;

```

```

I_theo = Times*((Rs*IntegrateGSL(Wall,E_0,m_nu2,bprime)) + (Rb*Gamma_b));
return (ExpData - I_theo)/sqrt(I_theo);
}

```

```

int expb_df(const gsl_vector *x,void *params,gsl_matrix *J)
{
//Calculation of the Jacobian of the Chi-Square function
size_t i;
size_t NumberMeas = ((struct data*)params)->NumberMeas;
double *Wall = ((struct data*)params)->Wall;
double *Times = ((struct data*)params)->Times;
double *ExpData = ((struct data*)params)->ExpData;
double result_0,err_0;
double result_1,err_1;
double result_2,err_2;
double result_3,err_3;
double result_4,err_4;
double p[8];
gsl_function F0,F1,F2,F3,F4;

```

```

p[0] = gsl_vector_get(x,0);
p[1] = gsl_vector_get(x,1);
p[2] = gsl_vector_get(x,2);
p[3] = gsl_vector_get(x,3);
p[4] = gsl_vector_get(x,4);

```

```
for(i=0;i < NumberMeas;i++)
{
p[5] = Wall[i];
p[6] = Times[i];
p[7] = ExpData[i];

F0.function = &my_f0;
F0.params = p;

F1.function = &my_f1;
F1.params = p;

F2.function = &my_f2;
F2.params = p;

F3.function = &my_f3;
F3.params = p;

F4.function = &my_f4;
F4.params = p;

gsl_deriv_central(&F0,gsl_vector_get(x,0),1.e-10,&result_0,&err_0);
gsl_deriv_central(&F1,gsl_vector_get(x,1),1.e-10,&result_1,&err_1);
gsl_deriv_central(&F2,gsl_vector_get(x,2),1.e-10,&result_2,&err_2);
gsl_deriv_central(&F3,gsl_vector_get(x,3),1.e-10,&result_3,&err_3);
gsl_deriv_central(&F4,gsl_vector_get(x,4),1.e-10,&result_4,&err_4);

gsl_matrix_set(J,i,0,result_0);
gsl_matrix_set(J,i,1,result_1);
gsl_matrix_set(J,i,2,result_2);
gsl_matrix_set(J,i,3,result_3);
gsl_matrix_set(J,i,4,result_4);
}
return GSL_SUCCESS;
}

int expb_fdf(const gsl_vector *x,void *params,gsl_vector *f,gsl_matrix *J)
{
//Environment for GSL
expb_f(x,params,f);
expb_df(x,params,J);

return GSL_SUCCESS;
}
```

```

void print_state(size_t iter,gsl_multifit_fdfsolver *s)
{
//Output the present calculation
cout << endl << iter << " " << sp(12) << gsl_vector_get(s->x,0) << " "
<< gsl_vector_get(s->x,1) << " " << gsl_vector_get(s->x,2) << " "
<< gsl_vector_get(s->x,3) << " " << gsl_vector_get(s->x,4) << " "
<< gsl_blas_dnrm2(s->f);
}

void Minimizer(double E0,double mnu2,double Rs,double Rb,double bprimed,char *name)
{
//Find the minimum by using the Levenberg-Marquardt algorithm
char sChi[128] = "ChiMin";
strcat(sChi,name);
ofstream wChi(sChi);
DataGenerator gdg;
const gsl_multifit_fdfsolver_type *T;
gsl_multifit_fdfsolver *s;
int status;
size_t i,iter;
for(int j=1;j<=5000;j++)
{
gdg.Calculator(E0,mnu2,Rs,Rb,bprimed);
const size_t NumberMeas = gdg.theMeasurements.size();
const size_t p = 5;
double Wall[NumberMeas];
double Times[NumberMeas];
double ExpData[NumberMeas];

for(i=0;i<NumberMeas;i++)
{
Wall[i] = gdg.thePotentials[i];
Times[i] = gdg.theTimes[i];
ExpData[i] = gdg.theMeasurements[i];
}

struct data d = {NumberMeas,Wall,Times,ExpData};
gsl_multifit_function_fdf f;
double x_init[5] = {18575.3,5.,1.5,1.5,1.};
gsl_vector_view x = gsl_vector_view_array(x_init,p);

f.f = &expb_f;
f.df = &expb_df;
f.fdf = &expb_fdf;
f.n = NumberMeas;
f.p = p;
f.params = &d;

```

```

T = gsl_multifit_fdfsolver_lmder;
s = gsl_multifit_fdfsolver_alloc(T, NumberMeas, p);
gsl_multifit_fdfsolver_set(s, &f, &x.vector);

print_state(iter, s);
iter = 0;
do
{
iter++;
status = gsl_multifit_fdfsolver_iterate(s);
print_state(iter, s);

if(status)
break;

status = gsl_multifit_test_delta(s->dx, s->x, 1.e-24, 1.e-24);
}
while(status == GSL_CONTINUE && iter < 500);

(wChi) << endl << pow(gsl_blas_dnorm2(s->f), 2.) << " "
<< sp(12) << gsl_vector_get(s->x, 0) << " "
<< gsl_vector_get(s->x, 1) << " "
<< sqrt(fabs(gsl_vector_get(s->x, 1))) << " "
<< gsl_vector_get(s->x, 2) << " "
<< gsl_vector_get(s->x, 3) << " " << gsl_vector_get(s->x, 4);
gsl_multifit_fdfsolver_free(s);
}
wChi.close();
}

int main()
{
//Everything can be run from this line:
Minimizer(18575., 4., 1., 1., 0., "4eV.dat");
return 0;
}

```

# List of Acronyms

<b>APL</b>	<b>A</b> nalyzing <b>P</b> lane
<b>BG</b>	<b>B</b> ack <b>G</b> round
<b>CDM</b>	<b>C</b> old <b>D</b> ark <b>M</b> atter
<b>C.L.</b>	<b>C</b> onfidence <b>L</b> evel
<b>FSD</b>	<b>F</b> inal <b>S</b> tate <b>D</b> istribution
<b>GSL</b>	<b>G</b> NU <b>S</b> cientific <b>L</b> ibrary
<b>HDM</b>	<b>H</b> ot <b>D</b> ark <b>M</b> atter
<b>KATRIN</b>	<b>K</b> arlsruhe <b>T</b> ritium <b>N</b> eutrino Mass Experiment
<b>LSS</b>	<b>L</b> arge <b>S</b> cale <b>S</b> tructures
<b>MAC-E filter</b>	<b>M</b> agnetic <b>A</b> diabatic <b>C</b> ollimation <b>E</b> lectrostatic <b>f</b> ilter
<b>MC</b>	<b>M</b> onte <b>C</b> arlo
<b>Non SM</b>	<b>N</b> on <b>S</b> tandard <b>M</b> odel
<b>Non V-A</b>	<b>N</b> on <b>V</b> ector <b>minus</b> <b>A</b> xialvector
<b>SM</b>	<b>S</b> tandard <b>M</b> odel
<b>V-A</b>	<b>V</b> ector <b>minus</b> <b>A</b> xialvector
<b>WGTS</b>	<b>W</b> indowless <b>G</b> aseous <b>T</b> ritium <b>S</b> ource





# Bibliography

- [1] W.E. Pauli (1930).  
Letter to the Physical Society of Tübingen,, unpublished; The letter is reproduced in  
Brown, L.M., *Physics Today*, **31**, No. 9, 23 (1978).
- [2] E. Fermi.  
**Versuch einer Theorie der  $\beta$ -Strahlen.**  
*Z. Physik*, **88**:161 (1934).
- [3] F. Reines and C.L. Cowan Jr.  
**Detection of the free neutrino.**  
*Phys. Rev. Lett.*, **92**:830 (1953).
- [4] E.J. Konopinski.  
 **$H^3$  and the mass of the neutrino.**  
*Phys. Rev. Lett.*, **72**:518 (1947).
- [5] G. Danby, J.-M. Gaillard, K. Goulianos, L.M. Lederman, N. Mistry, M. Schwartz,  
and J. Steinberger.  
**Observation of high-energy neutrino reactions and the existence of two  
kinds of neutrinos.**  
*Phys. Rev. Lett.*, **9**:36 (1962).
- [6] M.L. Perl, G.S. Abrams, A.M. Boyarski, M. Breidenbach, D.D. Briggs, F. Bulos,  
W. Chinowsky, J.T. Dakin, G.J. Feldman, C.E. Friedberg, D. Fryberger, G. Gold-  
haber, G. Hanson, B. Heile, B. Jean-Marie, J.A. Kadyk, R.R. Larsen, A.M. Litke, D.  
Lücke, B.A. Lulu, V. Lüth, D. Lyon, C.C. Morehouse, J.M. Paterson, F.M. Pierre,  
T.P. Pun, P.A. Rapidis, B. Richter, B. Sadoulet, R.F. Schwitters, W. Tanenbaum,  
G.H. Trilling, F. Vannucci, J.S. Withaker, F.C. Winkelmann, and J.E. Wiss.  
**Evidence for anomalous lepton production in  $e^+ - e^-$  annihilation.**  
*Phys. Rev. Lett.*, **35**:1489 (1975).
- [7] R.P. Feynman and M. Gell-Mann.  
**Theory of the Fermi interaction.**  
*Phys. Rev. Lett.*, **109**:193 (1958).
- [8] E.C.G. Sudarshan and R.E. Marshak.  
**Chirality invariance and the universal Fermi interaction.**  
*Phys. Rev. Lett.*, **109**:1860 (1958).

- 
- [9] C.S. Wu, E. Ambler, R.W. Hayward, D.D. Hoppes, and R.P. Hudson.  
**Experimental test of parity conservation in beta decay.**  
*Phys. Rev. Lett.*, **105**:1413 (1957).
- [10] V. Gribov and B. Pontecorvo.  
**Neutrino astronomy and lepton charge.**  
*Phys. Lett.*, **B28**:493 (1969).
- [11] R. Davis Jr.  
**Solar neutrinos. II. Experimental.**  
*Phys. Rev. Lett.*, **12**:303 (1964).
- [12] Y. Fukuda et al.  
**Evidence for oscillation of atmospheric neutrinos.**  
*Phys. Rev. Lett.*, **81**:1562 (1998).
- [13] S. Fukuda et al.  
**Constraints on neutrino oscillations using 1258 days of Super-Kamiokande solar neutrino data.**  
*Phys. Rev. Lett.*, **86**:5656 (2001).
- [14] Q.R. Ahmad et al.  
**Direct evidence for neutrino flavor transformation from neutral-current interactions in the Sudbury neutrino observatory.**  
*Phys. Rev. Lett.*, **89**:011301 (2002).
- [15] S. Eidelman et al.  
**Review of particle physics.**  
*Phys. Lett.*, **B592**:1 (2004).
- [16] S. Hannestad.  
**Cosmological neutrinos.**  
*New Journal of Physics*, **6**:108 (2004).
- [17] K. Winter.  
**Neutrino physics.**  
Cambridge University Press (1991).
- [18] F. Boehm and P. Vogel.  
**Physics of massive neutrinos.**  
Cambridge University Press (1987).
- [19] Ch. Weinheimer, B. Degen, A. Bleile, J. Bonn, L. Bornschein, O. Kazachenko, A. Kovalik, and E. Otten.  
**High precision measurement of the tritium  $\beta$  spectrum near its endpoint and upper limit on the neutrino mass.**  
*Phys. Lett.*, **B460**:219 (1999).
- [20] V.M. Lobashev, V.N. Aseev, A.I. Belesev, A.I. Berlev, E.V. Geraskin, A.A. Golubev, O.V. Kazachenko, Yu.E. Kuznetsov, R.P. Ostroumov, L.A. Rivkis, B.E. Stern, N.A. Titov, S.V. Zadorohny, and Yu.I. Zakharov.

- Direct search for mass of neutrino and anomaly in the tritium beta-spectrum.**  
*Phys. Lett.*, **B460**:227 (1999).
- [21] Ch. Kraus, B. Bornschein, L. Bornschein, J. Bonn, B. Flatt, A. Kovalik, B. Ostrick, E.W. Otten, J.P. Schall, Th. Thümmel, and Ch. Weinheimer.  
**Final Result from phase II of the Mainz Neutrino Mass Search in Tritium  $\beta$  Decay.**  
*Eur. Phys. J.*, **C40**:447 (2005).
- [22] The KATRIN Collaboration.  
**KATRIN Design Report.**  
FZKA Scientific Report 7090 (2004).
- [23] B. Müller.  
**Umbau des Mainzer Neutrinomassenexperiments und Untergrunduntersuchungen im Hinblick auf KATRIN.**  
Dipl. Thesis, Johannes Gutenberg-Universität Mainz (2002).
- [24] B. Flatt.  
**Voruntersuchungen zu den Spektrometern des KATRIN-Experiments.**  
PhD Thesis, Johannes Gutenberg-Universität Mainz (2004).
- [25] H.F. Schopper.  
**Weak interactions and nuclear beta decay.**  
North-Holland Pub. Co. (1966).
- [26] F. Glück.  
**Beta decay beyond the Standard Model.**  
Public talk, Johannes Gutenberg-Universität Mainz (2000).
- [27] C.P. Enz.  
**Fermi interaction with non-conservation of *lepton charge* and of parity.**  
*Il nuovo cimento*, **VI**:1 (1957).
- [28] J. Ciborowski and J. Rembielinski.  
**Tritium decay and the hypothesis of tachyonic neutrinos.**  
*Eur. Phys.*, **C8**:157 (1999).
- [29] J.M. Carmona and Cortés J.L.  
**Testing Lorentz invariance violations in the tritium beta-decay anomaly.**  
*Phys. Lett.*, **B494**:75 (2000).
- [30] A. Saenz, S. Jonsell, and P. Froelich.  
**Improved molecular final-state distribution of  $\text{HeT}^+$  for the beta-decay process of  $\text{T}_2$ .**  
*Phys. Rev. Lett.*, **84**:242 (2000).
- [31] N. Doss.  
Public talk, BSCW 95-TRP-1712-Doss.ppt; KATRIN BSCW server.

- [32] W. Kündig, E. Holzschuh, M. Fritschi, and H. Stüssi.  
**Direct measurement of the neutrino mass.**  
In Winter, K. (ed.): Neutrino physics (1992), p.144.
- [33] A. Sirlin.  
**General properties of the electromagnetic corrections to the beta decay of a physical nucleon.**  
*Phys. Rev.*, **164**:1767 (1967).
- [34] F. Schwamb.  
**Untergrunduntersuchungen für das KATRIN-Experiment.**  
PhD Thesis, Universität Karlsruhe (2004).
- [35] S. Sanchez.  
**Study of background suppression in MAC-E filters.**  
Dipl. Thesis, Johannes Gutenberg-Universität Mainz (2003).
- [36] A.I. Belesev et al.  
**Results of the Troitsk experiment on the search for the electron anti-neutrino rest mass in tritium beta decay.**  
*Phys. Lett.*, **B350**:263 (1995).
- [37] J. Kaspar.  
**Influence of energy scale imperfections on neutrino mass sensitivity in the KATRIN experiment.**  
Dipl. Thesis, Institute of Nuclear Physics, Rez/Prague (2003).
- [38] J.P. Schall.  
**Untersuchungen zu Untergrundprozessen am Mainzer Neutrinomassenexperiment.**  
Dipl. Thesis, Johannes Gutenberg-Universität Mainz (2001).
- [39] L. Lyons.  
**Statistics for nuclear and particle physicist.**  
Cambridge University Press (1986).
- [40] G. Audi, A.H. Wapstra, and C. Thibault.  
**The AME 2003 atomic mass evaluation.**  
*Nucl. Phys.*, **A729**:337 (2003).
- [41] J. Bonn.  
**Private communication.**  
*Johannes Gutenberg-Universität Mainz.*
- [42] K. Blaum.  
**High accuracy mass spectrometry with stored ions.**  
*Phys. Rep.*, **425**:1 (2006).
- [43] **GSL Manual.**  
<http://www.gnu.org/software/gsl/manual>.

- 
- [44] **MINUIT Manual.**  
<http://seal.web.cern.ch/seal/snapshot/work-packages/mathlibs/minuit/>.
- [45] **GMP Manual.**  
<http://www.swox.com/gmp>.
- [46] G.C. McLaughlin and J.N. Ng.  
**The use of nuclear  $\beta$ -decay as a test of bulk neutrinos in extra dimensions.**  
*Physical Review*, **D63**:053002 (2001).



# Acknowledgements

I would like to start by expressing my gratitude specially to **Dr. Jochen Bonn**. For being so patient with me and letting me face the work my way. For teaching me physics and for so many interesting and intelligent conversations. And of course, for giving me the opportunity to join his group and to get to know this wonderful country.

I also want to thank **Dr. Klaus Blaum** for being very supportive with me, for his corrections of the thesis as well as for being also responsible of my chance to join the Johannes Gutenberg-Universität Mainz.

To **Dr. Ferenc Glück** and **Dr. Klaus Eitel**, for many valuable discussions and teaching, also for checking my work and giving me many good advices.

To **Prof. Dr. Ernst Otten**, for guiding me with his wise insight into the tachyonic problem.

I would like to say thanks to the **Spanish community of Mainz**, who made me feel a bit closer to home. Mainly, to **Ana** and **Cris** who cheered me up just by smiling. Also, to **Raquel** and **Fidel** who witnessed the development of this work day by day. To **Jou** from Mainz and **Juan Angel** and **David** from Spain who helped me out everytime my computer crashed.

Además, me gustaría decir gracias a **Pablo, Germán, Ricardo** y de nuevo a **David**, porque nunca hubo cinco personas tan peculiarmente entrelazadas. Soy consciente de que tengo mucho que agradecerlos.

Por último, el agradecimiento más profundo y sentido es a **mi familia**, porque nunca nadie ha tenido un mejor ejemplo de abnegación y buena voluntad. Especialmente a **mis padres**, por ser los mejores padres que se puede imaginar. Os quiero.