Spring 5-16-2014

The Analysis of Experimental Error in Parity Violation Experiments with Polarized Neutrons

Jonathan Serpico
Western Kentucky University, jonathan.serpico929@topper.wku.edu

THE ANALYSIS OF EXPERIMENTAL ERROR IN PARITY VIOLATING EXPERIMENTS WITH POLARIZED NEUTRONS

A Capstone Experience/ Thesis Project

Presented in Partial Fulfilment of the Requirements for

the Degree of Bachelor of Science with

Honors College Graduate Distinction at Western Kentucky University

By

Jonathan Anthony Serpico
*****

Western Kentucky University
2014

CE/T Commitee:

Dr. Keith Andrew

Dr. Ivan Novikov

Dr. Gordon Baylis

Approved by

_____
Advisor
Department of Physics and Astronomy

ABSTRACT

Measurements of parity symmetry violation in nuclear reactions with polarized neutrons can provide valuable information on hadronic weak interaction. We have conducted an analysis of experimental error in the measurement of the parity violating effect in the *n3He* experiment. several experimental parameters have been optimized to minimize statistical error using numerical simulations. An analysis of systematic error due to differential cross-section dependence on energy as well as on false parity conserving asymmetries was also conducted. Our results suggest that the proposed parameters of the experiment will sufficiently suppress all sources of error under consideration. Furthermore, these approaches may be effectively applied to examine potential sources of error in other experiments using neutrons to measure P-violating or even CP-violating effects.

Keywords: neutrons, parity, polarized, hadron, nuclear, resonance

ACKNOWLEDGEMENTS

I would like to acknowledge my research advisor, Dr. Ivan Novikov, for his guidance in my research as well as in the process of developing my thesis. I would also like to thank the Honors College for helping to guide me through the thesis writing process. Finally I would like to thank the Applied Physics Institute and the FUSE Award Program for providing the funding that made my research possible.

VITA

April 14, 1993 ..... Born - Bamberg, Germany

Spring 2011 ..... Gatton Academy of Math and Science, Bowling Green, Kentucky

Fall 2012 ..... FUSE Award

FIELDS OF STUDY

Major Field: Physics

Minor Field: Mathematics

TABLE OF CONTENTS

List of Figures

CHAPTER 1

PARITY

1.1 Parity Symmetry

Parity symmetry describes symmetry under spatial-inversion, or parity transformation. Under parity transformation all three spatial coordinates are reversed. Parity symmetry is said to be discrete because it describes a non-continuous transformation of the system, in which one state changes to another without passing through intermediary states [1]

$$\mathcal{P}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -x \\ -y \\ -z \end{pmatrix} \tag{1.1}$$

Vectors change sign under parity while pseudo-vectors and scalars do not change sign. Quantities that do not change sign under parity transformation are said to be parity even while those that do change sign are said to be parity odd. Let us consider two examples. It is obvious that mass, a scalar, is unaffected by a change in spatial coordinates. However, consider angular momentum, a pseudo-vector, which is defined as the cross product of position and momentum. Under parity transformation both of these quantities change sign, however their vector product does not. Therefore angular momentum is a parity odd quantity

$$\mathcal{P}\left(\vec{L}\right) = \mathcal{P}\left(\vec{r} \times \vec{p}\right) = (-\vec{r}) \times (-\vec{p}) = \vec{L} \tag{1.2}$$

Other parity odd quantities include particle spin and magnetic field are also axial-vectors and as such are parity odd.

1.2 Parity Violation

The three discrete symmetries of the Standard Model are not exact. It is now known that under certain conditions these symmetries are violated. Theoretical predictions of parity violation in the weak interaction [2] motivated an experimental group led by C. S. Wu. In the experiment [3] the angular distribution of electrons emitted through the β-decay of polarized ^{60}Co nuclei was measured and then measured again when the polarization of the nuclei was reversed. If parity was conserved the angular distribution of the electrons would remain unchanged, instead a measurable asymmetry was observed thus confirming parity violation. It is possible to measure this parity violating (PV) asymmetry as a result of the dependence of the differential cross-section on the parity odd correlation. This correlation is given by

$$\mathcal{P}\left(\vec{J}\cdot\vec{k}_e\right) = -\left(\vec{J}\cdot\vec{k}_e\right) \tag{1.3}$$

where $\vec{J}$ is the spin of the cobalt nuclei and $\vec{k}_e$ is electron momentum.

Mathematically, the fact that parity is violated in the weak interaction may be described as

$$[V_W, \mathcal{P}] \neq 0 \tag{1.4}$$

where V_W is a weakly interacting potential. When considering the time evolution operator, $\hat{U}(t, t_0)$ and Hamiltonian, $\hat{H}$, given as

$$|\psi(t)\rangle = \hat{U}(t, t_0)\,|\psi(t_0)\rangle \tag{1.5}$$

$$\hat{U}(t, t_0) = exp\left(\frac{-i\hat{H}(t-t_0)}{\hbar}\right) \tag{1.6}$$

$$\hat{H} = \frac{p^2}{2\mu} + V \tag{1.7}$$

we can then see that the development of a system in time is determined primarily by its potential, V, with weakly interacting parity violating component V_W [4].

Due to the relatively small contribution of the weak interaction compared to that of the strong interaction it is sometimes useful to describe the eigenstates of the parity violating Hamiltonian as a mixing of eigenstates of the strong, parity conserving Hamiltonian [4] as in

$$|\phi\rangle = |\psi_+\rangle + \frac{\langle\psi_-| V_W |\psi_+\rangle}{\delta E} |\psi_-\rangle = |\psi_+\rangle + \epsilon |\psi_-\rangle \tag{1.8}$$

where δE is the difference in the strong Hamiltonian eigenvalues and psi_+ and psi_- are parity eigenstates. With this in mind an observable A of mixed parity, such as a difference in neutron cross-section, can be written as

$$A = A_{even} + A_{odd} \tag{1.9}$$

with an expectation value of

$$\langle\phi| A |\phi\rangle = \langle\psi_+| A_{even} |\psi_+\rangle + 2\epsilon Re\{\langle\psi_-| A_{odd} |\psi_+\rangle\} + O(\epsilon^2) \tag{1.10}$$

The value of the relative magnitude of the strong interaction and weak interaction components ϵ is small. Thus any attempted measurement of ϵ will require a carefully designed experiment in which high precision measurements may be taken.

CHAPTER 2

PARITY VIOLATION IN EXPERIMENTS WITH NEUTRONS

An accurate description of nucleon-nucleon potential is fundamental to our understanding of physics. In order to better understand the weak interaction component of hadron-hadron interactions, it is useful to exploit the fact that it uniquely violates parity. An interest has thus developed in experiments using low energy polarized neutrons to measure various PV asymmetries in hadron-hadron interactions. The bulk of these experiments can be divided into two classes, propagation experiments and reaction experiments

2.1 Propagation Experiments

The strong interaction component in nucleon interactions is many orders of magnitude larger than the weak interaction component, making measurement of weak interaction PV effects extremely difficult. One approach to this problem is to exploit resonance reactions of neutrons in heavy nuclei to enhance value of PV asymmetries [5]. In these experiments, a beam of polarized neutrons is fired at a thin target of heavy nuclei, for example ^{139}La [6]. A detector on the other side of the target and in the path of the incident beam then counts the number of neutrons which propagate through the target (N_+). The polarization of the neutron beam is then reversed and the experiment is repeated (N_-). The number of neutrons propagated through the target can be expressed as

$$N_\pm = N_0 e^{-\rho\sigma_\pm x} \tag{2.1}$$

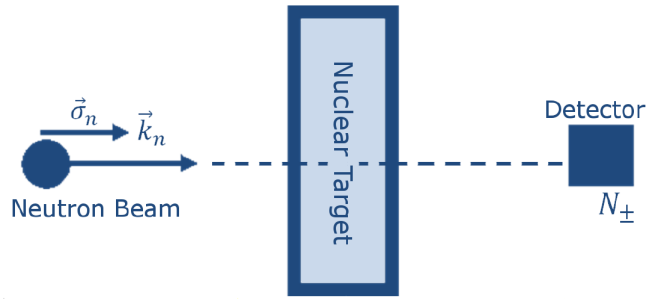

Figure 2.1: Diagram of propagation experiments

where ρ is target density ,$\sigma_{\pm}$ is the total cross-section and x is the thickness of the target.

The cross-section of the polarized neutrons given by

$$\sigma_{\pm} = \sigma_0 \left(1 + \Delta_P(\vec{\sigma}_n \cdot \vec{k}_n)\right) \tag{2.2}$$

where Δ_P is the parity violating component of the neutron cross-section. The cross-section is dependent on the parity odd correlation

$$\mathcal{P}\left(\vec{\sigma}_n \cdot \vec{k}_n\right) = -\left(\vec{\sigma}_n \cdot \vec{k}_n\right) \tag{2.3}$$

where $\vec{\sigma}_n$ is the neutron polarization vector and $\vec{k}_n$ is neutron momentum. The theoretical parity violating effect is then

$$A_P^{(th.)} = \frac{\sigma_+ - \sigma_-}{\sigma_+ + \sigma_-} \tag{2.4}$$

In the case of a thin target where $x << 1$, (2.1) may be reduced to

$$N_{\pm} \approx N_0(1 + \rho\sigma_{\pm}x) \tag{2.5}$$

From this simplification the experimental parity violating effect may then be measured

as

$$A_P^{(exp.)} = \frac{N_+ - N_-}{N_+ + N_-} \approx \frac{(1 + \rho\sigma_0 x(1 + \Delta_P)) - (1 + \rho\sigma_0 x(1 - \Delta_P))}{(1 + \rho\sigma_0 x(1 + \Delta_P)) + (1 + \rho\sigma_0 x(1 - \Delta_P))} = x\rho\sigma_0\Delta_P \quad (2.6)$$

The magnitude of $A_P^{(exp.)}$ has been measured with a heavy nuclear target to be as large 10^{-1} relative to strong interaction effects. The many order of magnitude enhancement of the PV effect was explained in the framework of nuclear resonance theory [5].

The experimentally measured parity violating effect will vary from the theoretical value as

$$A_P^{(exp.)} = A_P^{(th.)} \pm \sigma_{A_P}^{(stat.)} \pm \sigma_{A_P}^{(sys.)} \quad (2.7)$$

where $\sigma_{A_P}^{(stat.)}$ and $\sigma_{A_P}^{(sys.)}$ are the statistical and systematic error respectively. In order to accurately measure the parity violating effect it is necessary to minimize the statistical error given by

$$\frac{\sigma_{A_P}^{(stat.)}}{A_P^{(exp.)}} = \sqrt{\frac{\Delta(N_+ - N_-)^2}{(N_+ - N_-)^2} + \frac{\Delta(N_+ + N_-)^2}{(N_+ + N_-)^2}} \quad (2.8)$$

Knowing that $|N_+ - N_-| << |N_+ + N_-|$, the statistical error may then be approximated as

$$\frac{\sigma_{A_P}^{(stat.)}}{A_P^{(exp.)}} \approx \sqrt{\frac{\Delta(N_+ - N_-)^2}{(N_+ - N_-)^2}} \approx \sqrt{\frac{N_+ + N_-}{(N_+ - N_-)^2}} \approx \frac{e^{x\rho\sigma_0}}{x\rho\sigma_0\Delta_P\sqrt{N_0}} \quad (2.9)$$

Thus the parity violating effect increases with target depth, x. However, a deep target decreases the total number of propagated neutrons resulting in a large uncertainty. Therefore, in order to accurately measure $A_P^{(exp.)}$, the target depth must be optimized in order to minimize the statistical error. We see that this occurs at $x = (\rho\sigma_0)^{-1}$, which we recognize as the mean free path of the neutrons.

2.2 Reaction Experiments

Reaction experiments seek to examine simple systems using nuclei with only a few nucleons. The resonance enhancement of the PV effects is lost in these experiments, resulting in a relative magnitude of 10^{-7}. As a result, reaction experiments require extremely precise measurement of the various asymmetries. Two such experiments are currently in progress, the *NPDGamma* experiment [7] and the *n3He* experiment which consider the reactions given by

$$\vec{n} + p \rightarrow d + \gamma \tag{2.10}$$

$$\vec{n} + {}^3He \rightarrow p + t \tag{2.11}$$

Both experiments are being conducted at the Fundamental Neutron Physics Beam Line (FNPB) at the Spallation Neutron Source at Oakridge National Lab (SNS ORNL) [8]. The data collection for the *NPDGamma* experiment is already complete while the *n3He* experiment is still being constructed.

The designs of these two experiments are very similar. In both a beam of polarized neutrons is fired at a large target chamber filled with gas. The neutrons propagate through the chamber and interact with the nuclei creating reaction products. The angular distribution of γ-rays is measured by a detector array in the case of the *NPDGamma* experiment. The angular distribution of proton and triton tracks is measured inside the target in the *n3He* experiment. Then in both cases the polarization of the neutrons is reversed and the experiment is conducted again. The difference in the angular distribution of the reaction products is then used to measure the PV asymmetry.

The neutron differential cross-section which determines the angular distribution

of the reaction products may be written

$$\frac{d\sigma}{d\Omega}_{\pm} = \frac{d\sigma}{d\Omega}_{0} \left(1 + \Delta_{nf}(\vec{\sigma}_n \cdot \vec{k}_f)\right) \tag{2.12}$$

where $\vec{k}_f$ is the reaction product momentum. The differential cross-section is dependent on the parity odd correlation given by

$$\mathcal{P}\left(\vec{\sigma}_n \cdot \vec{k}_f\right) = -\left(\vec{\sigma}_n \cdot \vec{k}_f\right) \tag{2.13}$$

where Δ_{nf} is the PV component of the differential cross-section and $\vec{k}_f$ is the reaction product momentum. The number of reaction products in a given solid angle for each neutron polarization can then be expressed as

$$\frac{dN}{d\Omega}_{\pm} = N_0 \frac{d\sigma}{d\Omega}_{0} \left(1 + \Delta_{nf}(\vec{\sigma}_n \cdot \vec{k}_f)\right) \tag{2.14}$$

The experimentally measured parity violating effect, A_{nf}^{exp}, can thus be extracted from the normalized difference of the measured angular distribution of the reaction products for forward and backward neutron polarization as in

$$A_{nf}^{exp} = \frac{\frac{dN}{d\Omega}_{+} - \frac{dN}{d\Omega}_{-}}{\frac{dN}{d\Omega}_{+} + \frac{dN}{d\Omega}_{-}} = \Delta_{n,f} \cos\theta_{n,f} \tag{2.15}$$

where θ_{nf} is the angle between $\vec{\sigma}_n$ and $\vec{k}_f$.

2.3 The *n3He* Experiment

Our goal is to use a combination of computational simulation and analytical calculation to analyse potential sources of experimental error in the ongoing *n3He* experiment. With this in mind a more thorough description of the proposed experimental configuration and an overview of the simulations developed to model it will be given below.

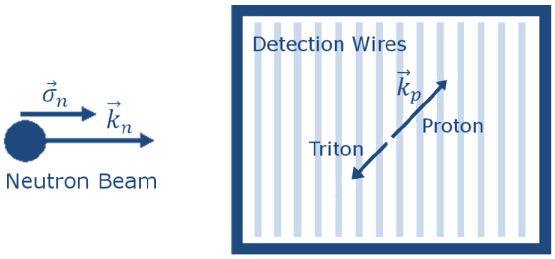

Figure 2.2: Diagram of *n3He* experiment

2.3.1 Configuration of the Experiment

The n3He experiment is to be carried out at the SNS ORNL. The neutron flux reaching the target is anticipated to be 10^8 neutrons/s/cm and the experiment will run for 10^7 seconds. The neutrons will have an energy between roughly 1 meV and 80 meV and a polarization of approximately 96%. It is expected that the polarization and beam axes will be aligned to the detector axis with a precision of 10 mrad each. The proposed target chamber is a cube with side lengths of roughly 20 cm. The chamber is filled with ^{3}He gas and a very small percentage of nitrogen gas at near room temperature and atmospheric pressure.

The target chamber is filled with a lattice of wires with alternating planes of wires carrying a high voltage and the others serving as detectors. There will be between 20 and 40 planes of detector wires in the chamber. When the incident neutrons react a proton and triton are created with kinetic energies of 573 keV and 192 keV respectively and with anti-parallel momentum. As the proton and triton travel through the target they ionize the helium creating a current between the high voltage and detector wires [9].

The purpose of the nitrogen is to suppress showers of ionization that would trigger detectors far from the proton and triton paths. The detector wires are arranged in such a way as to divide the chamber into cells, making it possible to track the path of the reaction products as they pass through the chamber. This process is made easier by the nature of the proton and triton themselves. Most of the triton energy is lost

at the beginning of its track, while most of the proton energy is lost at the end of the track. These two points of high energy deposition mark the beginning and end of the proton track, making calculation of its scattering angles easy.

2.3.2 Simulation of the Experiment

Just as in the propagation experiments, $\frac{\sigma_{A_{np}}^{(stat.)}}{A_{np}^{(exp.)}}$ must be minmized in order to accurately measure the parity violating effect. In the *n3He* experiment, the parity violating effect, $A_{np}^{(exp.)}$, depends on a large number of experimental parameters in a complicated way. As a result it is not possible to use analytical calculation, as in the case of propagation experiments. Therefore, a numerical approach was utilized.

We have developed simulations which model the *n3He* experiment and analyse the resulting data. In this way we numerically calculate the PV effect and systematic error and their dependence on the experimental parameters. The code for these simulations is included in Appendix A.

Our simulation models the experiment using Markov Chain Monte Carlo methods [10] to simulate the neutron reactions with the ^3He target and to generate the resulting reaction products. The program uses a coordinate frame aligned with the target chamber, with the z-axis being into the target chamber. A neutron is generated by assigning it an x and y coordinate within the beam from a uniform distribution on a disk of given size. The neutron energy and polar angle, θ, of the proton created in the reaction are generated by using the Metropolis-Hastings algorithm to sample from their respective distributions. The algorithm samples from these distributions by making a random walk over the the parameter space:

1. Choose an initial value x_t with high probability in your target distribution $P(x)$

2. Generate a random number from a Gaussian distribution, $N(0, \sigma_{step})$, and add it to x_t to obtain a new proposal value, x'

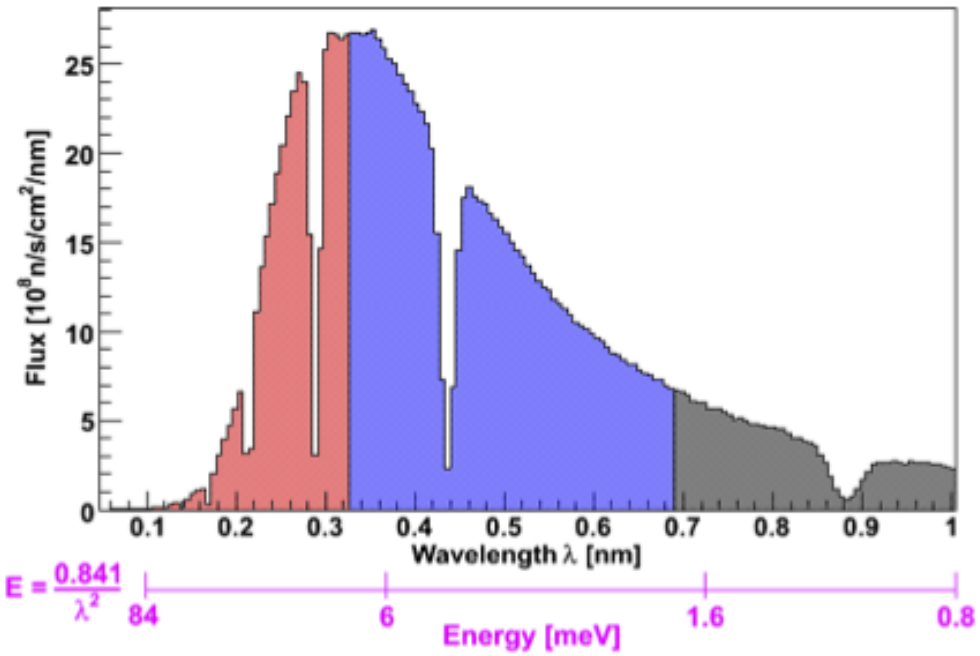

Figure 2.3: Neutron energy distribution at the FNPB at SNS [9]

3. Calculate $a = \frac{P(x')}{P(x_t)}$

4. If $a \geq 1$ let $x_t + 1 = x'$, repeat from 2.

5. If not choose r from a uniform distribution from 0 to 1

6. If $a \geq r$ let $x_t + 1 = x'$, repeat from 2.

7. If $a < r$ let $x_t + 1 = x_t$, repeat from 2.

The neutron energy is chosen from the distribution in Figure 2.3. The angle θ is sampled from the distribution

$$f(\theta) = 1 \pm \Delta_{np} \cos\theta \tag{2.16}$$

where $\Delta_{pn}^{(th.)}$ is the theoretical magnitude of the parity violating asymmetry that the

simulation will attempt to measure. The distribution uses a plus when the simulation is running with the polarization defined as $\sigma_n = +\hat{z}$ and minus when the simulation is running with the polarization defined as $\sigma_n = -\hat{z}$. The angle ϕ is sampled from a uniform distribution.

The depth into the chamber at which the neutron reacts is then calculated using

$$I = I_0 e^{-N\sigma z} \tag{2.17}$$

where the neutron cross section, σ, is calculated from neutron energy using values from the National Nuclear Data Center Evaluated Nuclear Data File [11]. The molecular density, N, is given by

$$N = \frac{N_a \rho}{M} \tag{2.18}$$

where ρ is the density, and M is the molar mass of the target. Exponential distributions may be calculated analytically and thus do not require the use of MCMC methods.

With the position of the neutron reaction in three dimensional space and the angles of the proton path defined the program now propagates the proton through the target. This is simulated by the proton taking a series of small steps through the chamber. At each step the amount of energy the proton loses is calculated using stopping power calculations as seen in Figure 2.4. Stopping power is typically calculated using the Bethe formula. The Bethe formula uses the characteristics of the particle and medium to determine the stopping power of the particle in the medium, as in [12]. However, the formula is not accurate enough at the low energies present in the *n3He* experiment. For this reason we have used data taken from the SRIM software package, which combines many equations and experimental data to accurately calculate stopping power [13]. The program continues to make steps until the proton's kinetic energy reaches zero. The same process is repeated for the triton, which takes a path anti-parallel to the proton path.

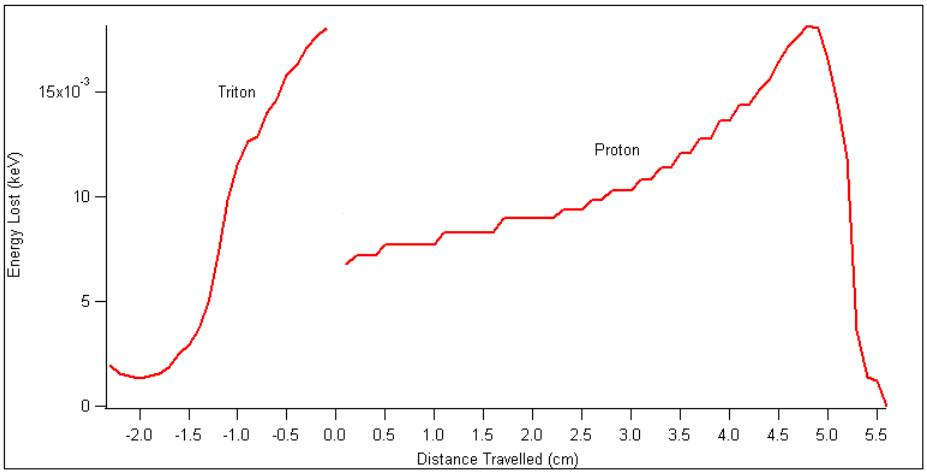

Figure 2.4: Proton and Triton energy loss as a function of distance calculated using SRIM software [13]

The target chamber is simulated as an array of cells defined by the detector wires. Each cell is defined as a region in the chamber and by an energy which is initially zero. The energy lost at each step of the proton and triton is added to the total energy of the cell that the particle is currently within. If a particle reaches the predefined edge of the chamber before all of its energy is lost the energy is not placed in any cell.

The *n3He* experiment will run for 10^7 seconds at a flux of 10^8 neutrons/second/cm. If we let the beam cross-sectional area be one square centimeter, the total number of neutrons will be roughly 10^15. In order to efficiently simulate such a large number of neutrons a tremendous amount of computational power is necessary. To meet this demand, we chose to utilize NVIDIA's CUDA architecture [14] to implement our program in parallel on an NVIDIA GTX670 GPU. GPUs have are capable of achieving a tremendous amount of computing power at a very low cost both in terms of hardware and energy, however, they can only achieve their full potential in situations in which it is possible to massively parallel-ize the computation. In our program this is implemented by utilizing the maximum number of processes a GTX670 can carry out simultaneously with each process calculating the energy loss and cell position for an individual step of of 1024 separate particles. This totals to 32768 simultaneous

processes.

To analyze the results of these simulations, data is taken of total energy deposited in each cell for both forward and backward polarization of the neutron beam. The calculated PV effect, $A_{np}^{(sim.)}$, in the ith cell can then be calculated from the cell energy using

$$A_{np,i}^{(sim.)} = \frac{E_{i,+} - E_{i,-}}{E_{i,+} + E_{i,-}} \tag{2.19}$$

and the relative statistical error can then be calculated as

$$\frac{\sigma_{A_{np,i}}^{(stat.)}}{A_{np,i}^{(sim.)}} = \sqrt{\frac{E_{i,+} + E_{i,-}}{(E_{i,+} - E_{i,-})^2}} \tag{2.20}$$

where $E_{i,\pm}$ is the energy lost in the ith cell for the forward or backward polarization. With these two values we can observe in which regions of the target chamber the statistical error is low and the measured effect is near the expected value. Using this information we have concluded that the most sensitive regions of the chamber are at a depth of 4 cm to 7 cm and within a centimeter or two of the initial neutron beam path. These conclusions seem to agree well with those presented in [9]. However, our results suggest that the proposed target depth of 20 cm could be reduced by several centimeters without loss of accuracy.

Beyond this primary result, our program has also been used to analyze several other features of the experiment. For example, temperature and pressure of the target are defined within our program and it is thus possible to observe the effects on distribution of neutron reaction depth as well as proton and triton energy loss over a range of these variables and optimize for desirable results. Cursory exploration has revealed room temperature and atmospheric pressure are relatively strong choices for these values.

Another useful measurement is percentage of proton and triton energy deposited in the cells. It is desirable that this value be large as if the particles are not able to

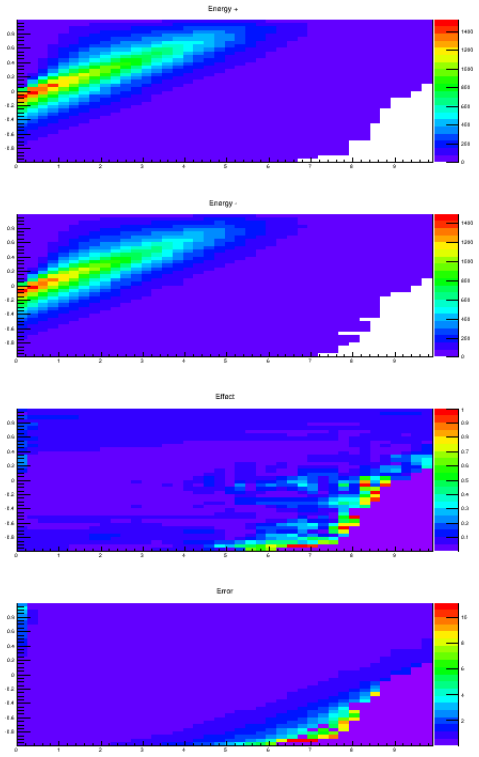

Figure 2.5: Deposited energy, parity violating effect, and statistical error as a function of target depth and proton angle

lose all of their energy within the chamber before striking the edge statistical accuracy of the measurement is being lost.

Furthermore, the design of these simulations is very flexible and in the future could easily be reconfigured to perform similar analysis of statistical error and experiment optimization on other neutron PV experiments. For example, many propagation experiments as well as other reaction experiments could readily be examined with only minor modification to the structure of these programs.

CHAPTER 3

ANALYSIS OF POTENTIAL SOURCES OF SYSTEMATIC ERROR

In experiments requiring high precision measurements, potential sources of systematic error must be carefully considered. In order to obtain meaningful results these sources must be identified and their effect on the measurements taking place must either be eliminated through modification of the experiment or measured with greater accuracy than the desired measurement. In the case of the n3He experiment, there are two major sources of potential systematic error that we have identified. The first is the dependence of both the strong interaction and weak interaction contributions to the experimentally measured values on neutron energy. The second is the contribution of several additional parity even and parity odd correlations to the differential cross-section. We have examined the potential contribution of both of these sources in detail.

3.1 Neutron Energy Dependence

The dependence of both strongly and weakly interacting terms on neutron energy is significant primarily because of the small magnitude of the parity violating effect relative to the strong interaction term. As a result, even a relatively small dependence of the strong interaction term on neutron energy could overwhelm the parity violating effect. Consider the differential cross-section given by

$$\frac{d\sigma}{d\Omega}_{\pm}^{(exp.)} = \langle B(E_n)\rangle + \langle C_{np}(E_n)\rangle \, (\vec{\sigma}_n \cdot \vec{k}_p) \tag{3.1}$$

where B is a parity even term and C_{np} is a parity odd term. These quantities are averaged over the neutron energies of a single measurement. The quantity, given by

$$\frac{(B(\langle E_n\rangle_1) + C_{np}(\langle E_n\rangle_1)\cos\theta) - (B(\langle E_n\rangle_2) - C_{np}(\langle E_n\rangle_2)\cos\theta)}{(B(\langle E_n\rangle_1) + C_{np}(\langle E_n\rangle_1)\cos\theta) + (B(\langle E_n\rangle_2) - C_{np}(\langle E_n\rangle_2)\cos\theta)} \neq \frac{C_{np}}{B}\cos\theta \quad (3.2)$$

is then no longer equivalent to the parity violating effect in the case where the average neutron energy of the two measurements is not the same.

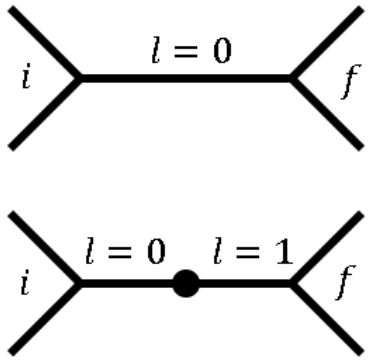

Figure 3.1: Parity violating (top) and parity conserving (bottom) R-Matrix elements

In order to analyze this potential source of systematic we have used the resonance approach presented in [15] and [5]. Using this approach the differential cross-section may be considered in terms of

$$\frac{d\sigma}{d\Omega} = \frac{\pi}{k^2} Tr(\hat{R}\hat{\rho}\hat{R}^\dagger) \quad (3.3)$$

where $\hat{R} \propto \left(\hat{S} - \hat{\mathbb{1}}\right)$ is the reaction matrix and $\hat{\rho}$ is the density matrix, whose elements define each possible state of the system.

Using results derived in [16], the differential cross-section can be calculated from the PV and PC R-matrix elements given by

$$\langle s'l'|\, R^J\, |sl\rangle = -\frac{iw\left(\Gamma_l^n(s)\Gamma_{l'}^p(s')\right)^{1/2}}{(E - E_l + i\Gamma_l/2)\,(E - E_{l'} + i\Gamma_{l'}/2)} e^{i(\delta_l^n + \delta_{l'}^p)} \quad (3.4)$$

E_r (MeV)	J^π	l	T	Γ_n (MeV)	Γ_n^0 (eV)	Γ_p (MeV)	Γ_p
-0.211	0+	0	0		954.4	1.153	1.153
0.430	0-	1	0	0.48		0.05	0.53
3.062	1-	1	1	2.76		3.44	6.20
3.672	1-	1	0	2.87		3.08	6.10
4.702	0-	1	1	3.85		4.12	7.97
5.372	1-	1	1	6.14		6.52	12.66
7.732	1+	0	0	4.66		4.725	9.89
7.92	1-	1	0	0.08		0.07	3.92
8.062	0-	1	0	0.01		0.01	4.89

Table 3.1: List of resonance energies with associate quantum numbers, partial, and total widths [16].

$$\langle s'l'|\, R^J\, |sl\rangle = \frac{i\left(\Gamma_l^n(s)\Gamma_{l'}^p(s')\right)^{1/2}}{(E - E_l + i\Gamma_l/2)} e^{i(\delta_l^n + \delta_{l'}^p)} \tag{3.5}$$

where E is neutron energy, E_l and $E_{l'}$ are resonance energies, the Γ are partial widths of the resonances, the δ are small phases, and w is the parity violating mixing element given by

$$w = -\int \phi_l V_W \phi_{l'} d\tau \tag{3.6}$$

as the result of mixing the two resonance states ϕ_l and $\phi_{l'}$ where V_W is the parity violating weak interaction potential presented in (1.8). These values are given for the significant reaction resonance energies in Table 3.1. The value of the differential cross-section is then approximately given by

$$\frac{d\sigma}{d\Omega}_{\pm} \propto \left|\langle s'l'|\, R^J\, |sl\rangle\right|^2 \tag{3.7}$$

where the matrix element represents a sum over resonance R-matrix elements.

By examining (3.3) and (3.4) it can thus be seen that the neutron energy dependence contributes as a term in the denominator of the R-matrix elements. It is thus clear that for neutron energies near or greater than the resonance energies, neutron energy dependence will present a significance source of substantial error in the event that $\langle E_n\rangle_1$ is sufficiently different from $\langle E_n\rangle_2$.

In the case of the n3He experiment, the distribution of energy in the neutron beam is from roughly 1 meV to 80 meV as seen in Figure 2.1 and the resonance energies, shown in Table 3.1, contributing to the differential cross-section range from 0.4 MeV to 8.0 MeV. This gives a minimum value of $\frac{E}{E_r} = 2 \times 10^{-7}$. These values have been calculated to contribute a maximum systematic error of approximately $A_E = 10^{-9}$, which is a full order of magnitude less than the desired measurement precision of 10^{-8}. This value will be further reduced by convergence of the average neutron energies to be many orders of magnitude less. As a result of these calculation, it can be concluded that neutron energy dependence is not a significant source of systematic error in the *n3He* experiment. However, it is also clear that in other experiments using higher neutron energies, neutron energy dependence could serve as a substantial source of systematic error. It would also likely be significant in attempts to measure CP violating effects, which are typically many orders of magnitude less than corresponding PV effects.

3.2 Parity Conserving Asymmetry

The second source of potential systematic error is from the additional correlations contributing to the differential cross-section. The differential cross-section has been described using (2.10). However, this equation is incomplete, in reality there are a total of eight parity violating and parity conserving correlations and each one has the potential to contribute some amount of asymmetry. These correlations are listed in Table 3.2. Many of these correlations, though, can be eliminated for various reasons.

The correlations which the include nuclear spin, $\vec{I}$, will all average to zero over the course of the experiment. The term $(\vec{k}_n \cdot \vec{k}_p)$ will contribute nothing, as it's value is initially small and will come close to cancelling for high statistics. The $(\vec{\sigma}_n \cdot \vec{k}_n)$ is initially small and does not affect the angular distribution of the protons. With these

PV	PC
$(\vec{\sigma}_n \cdot \vec{k}_p)$	$(\vec{k}_n \cdot \vec{k}_p)$
$(\vec{\sigma}_n \cdot \vec{k}_n)$	$(\vec{\sigma} \cdot [\vec{k}_n \times \vec{k}_p])$
$(\vec{I} \cdot \vec{k}_p)$	$(\vec{\sigma} \cdot \vec{I})$
$(\vec{I} \cdot \vec{k}_n)$	$(\vec{I} \cdot [\vec{k}_n \times \vec{k}_p])$

Table 3.2: List of parity violating and parity conserving quantities contributing to the n3He reaction [16].

quantities eliminated we are left only with the parity violating asymmetry that is intended to be measured, and a parity conserving asymmetry given by $(\vec{\sigma} \cdot [\vec{k}_n \times \vec{k}_p])$.

the magnitude of the PC effect has been calculated using the nuclear resonance approach applied to calculating A_{np} [16]. The effect is energy dependent but has a maximum value of approximately 10^{-4}. It may then be observed from considering the correlation that in the case where $[\vec{k}_n \times \vec{\sigma}] = 0$, the PC asymmetry disappeared. This observation can then be generalized to say that, for a polarization axis and beam axis very near the z-axis, the averaged asymmetry can be given by

$$A_{PC} \approx 10^{-4}(\vec{\sigma} \cdot [\vec{k}_n \times \vec{k}_p]) \approx 10^{-4}\theta_{\sigma_n,k_n} \sin\theta_{k_p,\hat{z}} \tag{3.8}$$

The error on the measurement of this quantity can then be expressed through

$$\delta_{A_{PC}} \approx A_{PC}\theta_{\sigma_n,\hat{z}}\theta_{k_n,\hat{z}} \approx 10^{-4}\theta_{\sigma_n,k_n}\theta_{\sigma_n,\hat{z}}\theta_{k_n,\hat{z}} \sin\theta_{k_p,\hat{z}} \tag{3.9}$$

From these equations we see that the measured parity conserving asymmetry has a magnitude of 10^{-4} and is further suppressed by a factor of $\theta_{\sigma_n,k_n}\theta_{\sigma_n,\hat{z}}\theta_{k_n,\hat{z}}$. Thus we can conclude that in order to accurately measure the parity violating asymmetry in the *n3He* experiment, the product of alignments must be such that $\theta_{\sigma_n,k_n}\theta_{\sigma_n,\hat{z}}\theta_{k_n,\hat{z}} \leq 10^{-4}$. Current specifications for the experiment suggest each of these angles can be kept to 10 mrad for a total suppression of 10^{-6}. This value is well within the necessary suppression and so we may conclude that although the magnitude of the PC effect is

large relative to PV effect, the alignment of experimental components is sufficiently precise to accurately measure the PV asymmetry. However, the value of A_{PC} is greater than the value of A_{np} and thus the parity conserving false asymmetry will have to be carefully subtracted from the measurements in order to achieve an accurate measurement of A_{np}.

CHAPTER 4

CONCLUSIONS

In conclusion, The class of experiments using cold neutrons to measure parity violation in hadron-hadron interactions may be described using a nuclear resonance approach. In this model, the system occasionally transitions between resonances of opposite symmetry, thus violating parity. When applying this approach to measurement of a given observable, the parity violating resonance reactions contribute a parity odd component to the observable. Therefore the measurement of this parity violating component can be used to examine the parity violating weak interaction contribution to the hadron-hadron resonance reactions. We have used this approach to minimize experimental error in the *n3He* experiment using both analytical and numeric methods.

The statistical error was minimized as a function of experimental parameters using a numerical MCMC approach. This has led us to conclude that the most accurate measurements of the PV effect can be taken at a depth of 4 cm to 7 cm into the target and in a region within a couple centimeters of the path of the incident neutron beam. We have also demonstrated that systematic error from PC asymmetries and neutron energy dependence are suppressed sufficiently as to be insignificant through an analytical consideration of the reaction matrix. Our measurements do, however, suggest that it is important to make an accurate measurement of the parity conserving asymmetry so that it may be removed from calculations of the parity violating effect.

In summary, all of our calculations suggest that under the proposed conditions, it will be possible to measure the parity violating effect in the *n3He* experiment to an accuracy of approximately 10^{-8}. The specific methods of calculating systematic error and for simulating the experiment which we have develop may also be effectively

applied as constraints when developing future experiments to measure parity violation, of even CP-violation with neutrons.

APPENDIX A: SIMULATION CODE

```
//Simulation of the n3He experiment using CUDA
#include <iostream>
#include <cuda.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <unistd.h>
#include "PMrand.h"
#include <cuda_runtime.h>

//point info object
typedef struct {
   float x;
   float y;
   float z;
   float t;
   float p;
   float e;
} point;

//probability of a neutron with a given energy in meV
float n_energy[61]={7.750276, 7.480738, 7.225020,
   6.982153, 6.751368, 6.531802, 6.322811, 6.123692,
   5.933801, 5.752637, 5.579646, 5.414314, 5.256250,
   5.105008, 4.960176, 4.821445, 4.688454, 4.560870,
   4.438443, 4.320880, 4.207909, 4.099329, 3.994880,
   3.894390, 3.797644, 3.704444, 3.614648, 3.528078,
   3.444567, 3.364000, 3.286227, 3.211107, 3.138547,
   3.068418, 3.000602, 2.935021, 2.871567, 2.810138,
   2.750669, 2.693059, 2.637249, 2.583157, 2.530702,
   2.479838, 2.430492, 2.382597, 2.336111, 2.290973,
   2.247123, 2.204526, 2.163130, 2.122883, 2.083754,
   2.045698, 2.008668, 1.972641, 1.937569, 1.903430,
   1.870184, 1.837797, 1.806250};
float n_prob[61]={26.764706, 26.764706, 26.705882,
   26.764706, 26.882353, 10.352941, 25.882353, 25.294118,
   25.000000, 24.411765, 23.882353, 23.470588, 23.058824,
   22.588235, 22.352941, 20.882353, 15.529412, 7.294118,
   2.294118, 6.882353, 14.529412, 16.117647, 17.823529,
   18.117647, 17.470588, 17.294118, 17.117647, 16.411765,
   16.294118, 15.882353, 15.529412, 15.000000, 14.529412,
   14.235294, 13.705882, 13.000000, 12.823529, 12.470588,
```

```
   12.294118, 11.764706, 11.529412, 11.294118, 10.941176,
   10.470588, 10.294118, 10.176471, 9.941176, 9.647059,
   9.411765, 9.117647, 8.764706, 8.705882, 8.411765,
   8.117647, 8.176471, 7.823529, 7.647059, 7.529412,
   7.235294, 7.352941, 6.941176};
24
25
26 //energy deposited at each step for the proton and triton
   measured in MeV
27 float p_list[28]={0.013964, 0.014204, 0.014463, 0.014753,
   0.015050, 0.015352, 0.015711, 0.016081, 0.016459,
   0.016897, 0.017361, 0.017838, 0.018401, 0.019013,
   0.019675, 0.020428, 0.021278, 0.022243, 0.023344,
   0.024603, 0.026109, 0.027950, 0.030005, 0.032656,
   0.035326, 0.035906, 0.027325, 0.000606};
28 float t_list[11]={0.035875, 0.034491, 0.031680, 0.027903,
   0.024764, 0.017798, 0.007616, 0.004060, 0.002811,
   0.002830, 0.001173};
29
30 //target
31 float t_side = 20.0; //target side length in cm
32 float c_side = 1.0; //cell side length in cm
33 float t_temp = 300.0; //target temperature in K
34 float t_pres = 1.0; //target pressure in atm
35 float m3He = 3.016; //target molar mass in g/mol
36
37
38 //other parameters
39 float step = 0.2; //distance between depositions in cm
40 float R = 82.05746; //(atm*cm^3)/(mol*K)
41 float Na = 6.022e23; //atoms/mol
42 //for box-muller transform
43 float hold = 10.0;
44
45 //method to handle part of x,y positioning using cuda
46 __global__ void define(point *u){
47    int i = (blockIdx.y * 32) + blockIdx.x;
48     int j = threadIdx.x;
49     //calculate initial x and y position
50     if(j == 0)
51    {
52          float u1 = u[i].x;
53          float u2 = u[i].y;
54       float t1 = 2.0*M_PI*u2;
55       t1 = cos(t1);
```

```
56        float t2 = -2.0*log(u1);
57        t2 = sqrt(t2);
58        t1 = t1*t2;
59        float z1 = t1;
60        t1 = 2.0*M_PI*u2;
61        t1 = sin(t1);
62        float z2 = t1*t2;
63
64          u[i].x = z1;
65          u[i].y = z2;
66     }
67      //calculate depth of interaction
68      if(j == 1)
69      {
70          float e = u[i].e * 1.0e-12;
71          float cs = 38528.29128;
72          cs = cs - 1.809682225e13 * e;
73          cs = cs + 6.077844749e21 * pow(e,2);
74          cs = cs - 1.206615936e30 * pow(e,3);
75          cs = cs + 1.295176648e38 * pow(e,4);
76          cs = cs - 5.809728335e45 * pow(e,5);
77          cs = cs*1.0e-24;
78          float c = 1.0*((1.0*6.022e23)/(300.0*82.05746))*cs
               ;
79          float r = u[i].z;
80          r = 1.0-r;
81          r = log(r);
82          r = -1.0*r/c;
83
84          u[i].z = r;
85      }
86      __syncthreads();
87 }
88
89 //method to deposit energy in the appropriate cell
90 __global__ void deposit(point *u, float *v, float step,
       float t_side, float c_side, int cd){
91 int i = (blockIdx.y * 32) + blockIdx.x;
92 int j = threadIdx.x;
93 float p_list[28]={0.013964, 0.014204, 0.014463, 0.014753,
       0.015050, 0.015352, 0.015711, 0.016081, 0.016459,
       0.016897, 0.017361, 0.017838, 0.018401, 0.019013,
       0.019675, 0.020428, 0.021278, 0.022243, 0.023344,
       0.024603, 0.026109, 0.027950, 0.030005, 0.032656,
       0.035326, 0.035906, 0.027325, 0.000606};
```

```
94 float t_list[11]={0.035875, 0.034491, 0.031680, 0.027903,
     0.024764, 0.017798, 0.007616, 0.004060, 0.002811,
     0.002830, 0.001173};
95
96     float r,x,y,z,e;
97     int k1,k2,k3,k;
98     if(j < 28){
99         r = (j+1)*step;
100        x = r*sin(u[i].t)*cos(u[i].p);
101        x = x + u[i].x + t_side/2.0;
102        y = r*sin(u[i].t)*sin(u[i].p);
103        y = y + u[i].y + t_side/2.0;
104        z = r*cos(u[i].t);
105        z = z + u[i].z;
106        e = p_list[j];
107    }
108    else{
109        r = -1.0*(j+1-28)*step;
110        x = r*sin(u[i].t)*cos(u[i].p);
111        x = x + u[i].x + t_side/2.0;
112        y = r*sin(u[i].t)*sin(u[i].p);
113        y = y + u[i].y + t_side/2.0;
114        z = r*cos(u[i].t);
115        z = z + u[i].z;
116        e = t_list[j-28];
117    }
118    k1 = 0;
119    while(k1*c_side < x)
120        k1++;
121    k2 = 0;
122    while(k2*c_side < y)
123        k2++;
124    k3 = 0;
125    while(k3*c_side < z)
126        k3++;
127    if( k1 < cd && k2 < cd && k3 < cd){
128        k = k3*cd*cd + k2*cd + k1;
129        v[k] =  v[k] + e;
130    }
131    __syncthreads();
132 }
133
134 //zeroes float array
135 __global__ void cuda_zero(float *v, int n){
136    int i = (blockIdx.y * 32 * 39) + (blockIdx.x * 39) +
```

```
      threadIdx.x;

    if(i < n)
        v[i]=0;
}

//zeroes int array
__global__ void cuda_zero(int *v, int n){
   int i = (blockIdx.y * 32 * 39) + (blockIdx.x * 39) +
      threadIdx.x;

    if(i < n)
        v[i]=0;
}

//box-muller transform for metropolis algorithm
float transform(float sigma){
   float r;
   if(hold != 10.0){
      r = hold;
      hold = 10.0;
   }
   else{
      float u1 = (float)PMrand()/RAND_MAX;
      float u2 = (float)PMrand()/RAND_MAX;
      float t1 = 2.0*M_PI*u2;
      t1 = cos(t1);
      float t2 = -2.0*log(u1);
      t2 = sqrt(t2);
      t1 = t1*t2;
      float z = t1;
      t1 = 2.0*M_PI*u2;
      t1 = sin(t1);
      hold = t1*t2;
      r = z;
   }
   return r*sigma;

}

//distributions for metropolis algorithm
float funcP(float t, int pm, float ap){
   float f;
    int pol = (1 - 2*pm);
   if(t > M_PI || t < 0.0){
```

```
180         f = 0;
181     }
182     else{
183         //plus or minus to change polorization
184         f = 1.0 + pol*ap*cos(t);
185      }
186     return f;
187 }
188 //neutron energy function
189 float funcN(float e_n){
190      float m,b,f;
191      int i = 0;
192      if( e_n > n_energy[0] || e_n < n_energy[60])
193          f = 0;
194      else{
195          while(n_energy[i] > e_n)
196              i++;
197          m = (n_prob[i-1] - n_prob[i]) / (n_energy[i-1] -
                n_energy[i]);
198          b = n_prob[i] - n_energy[i]*m;
199          f = e_n*m + b;
200      }
201      return f;
202 }
203
204 int main(int argc, char **argv){
205
206
207     //seed the random number generator
208     sPMrand();
209
210      //keep track of how long it takes
211      clock_t begin2, end2;
212      float time_spent2, time_sum;
213      time_sum = 0.0;
214      begin2 = clock();
215
216      //magnitude of effect: 1 + ap*cos(t)
217      float ap = 0.1;
218      int pm = 0;
219
220          //n sets of m*1024 neutrons for each polarization
221          int n = 100;
222          int m = 1e6;
223          point *pointsh = (point *)malloc(sizeof(point)
```

```
            *1024);
224
225         //make cell array
226         int cd = (int)(t_side/c_side);
227         int cV = pow(cd,3);
228         float *cellsh = (float *)malloc(sizeof(float)*cV);
229         float *cells2h = (float *)malloc(sizeof(float)*cV)
               ;
230         int i;
231         for(i=0; i<cV; i++){
232             cellsh[i] = 0.0;
233             cells2h[i] = 0.0;
234         }
235         float *cellsd;
236         cudaMalloc((void **)&cellsd,sizeof(float)*cV);
237
238         //setup cuda thread stuff
239         dim3 threadBlockRows(39, 1, 1);
240         dim3 blockGridRows(32, 32, 1);
241
242         //determine theta and neutron energy for points
               using metropolis algorithm
243         float t = 1.5;
244         float e = 4.8; //in meV
245         float r,t2,e2,ab,f1,f2;
246         int j,k;
247         long int accept,reject;
248         accept = 0;
249         reject = 0;
250         cudaMemcpy(cellsd,cellsh,sizeof(float)*cV,
               cudaMemcpyHostToDevice);
251         point *pointsd;
252         cudaMalloc((void **)&pointsd,sizeof(point)*1024);
253         for(k = 0; k < n; k++){
254
255             if(pm == 0  && k == 0){
256                 printf("simulating %e neutrons, polarized
                       parallel!\n", (float)n*m*1024);
257             }
258             if(pm == 1 && k == 0){
259                 printf("simulating %e neutrons, polarized
                       anti-parallel!\n", (float)n*m*1024);
260             }
261
262
```

```
263             //keep track of how long it takes
264             clock_t begin, end;
265             double time_spent;
266             begin = clock();
267 
268             for(i = 0; i < m; i++){
269                 for(j = 0; j < 1024; j++){
270                     r = (float)PMrand()/RAND_MAX;
271                     t2 = t + transform(2.0);
272                     e2 = e + transform(2.0);
273                     f1 = funcP(t2,pm,ap)*funcN(e2);
274                     f2 = funcP(t,pm,ap)*funcN(e);
275                     ab = f1/f2;
276                     if (r <= ab){
277                         t = t2;
278                         e = e2;
279                         accept++;
280                     }
281                     else{
282                         reject++;
283                     }
284                     pointsh[j].t = t;
285                     pointsh[j].p = (float)PMrand()/
                           RAND_MAX * 2.0*M_PI;
286                     pointsh[j].e = e;
287                     pointsh[j].x = (float)PMrand()/
                           RAND_MAX;
288                     pointsh[j].y = (float)PMrand()/
                           RAND_MAX;
289                     pointsh[j].z = (float)PMrand()/
                           RAND_MAX;
290                 }
291                 cudaMemcpy(pointsd,pointsh,sizeof(point)
                       *1024,cudaMemcpyHostToDevice);
292                 define<<<blockGridRows, threadBlockRows
                       >>>(pointsd);
293                 cudaMemcpy(pointsh,pointsd,sizeof(point)
                       *1024,cudaMemcpyDeviceToHost);
294                 deposit<<<blockGridRows, threadBlockRows
                       >>>(pointsd,cellsd,step,t_side,c_side,
                       cd);
295                 cudaMemcpy(cellsh,cellsd,sizeof(float)*cV,
                       cudaMemcpyDeviceToHost);
296                 for(j = 0; j < cV; j++){
297                     cells2h[j] = cells2h[j] + cellsh[j];
```

```
298                 }
299                 cuda_zero<<<blockGridRows, threadBlockRows
                       >>>(cellsd,cV);
300             }
301
302             //write data to files
303             FILE *file1;
304             int index, i2, j2, k2;
305             char str[20];
306             sprintf(str, "cells_%d/cells_%d.txt", pm, k);
307             file1 = fopen(str,"w");
308             for(i2 = 0; i2 < cd; i2++){
309                 for(j2 = 0; j2 < cd; j2++){
310                     for(k2 = 0; k2 < cd; k2++){
311                         index = i2*cd*cd + j2*cd + k2;
312                         fprintf(file1,"%d\t%d\t%d\t%f\n",
                               k2,j2,i2,cells2h[index]);
313                     }
314                 }
315             }
316             fclose(file1);
317
318             //print out run-time
319             end = clock();
320             time_spent = (float)(end - begin) /
                   CLOCKS_PER_SEC;
321             time_sum = time_sum + time_spent;
322             if (time_spent < 120.0) {
323                 printf("time elapsed for set %d_%d: %f s\n
                       ",pm,k,time_spent);
324             }
325             else if(time_spent < 60.0*120.0){
326                 printf("time elapsed for set %d_%d: %f min
                       \n",pm,k,time_spent/60.0);
327             }
328             else{
329                 printf("time elapsed for set %d_%d: %f hrs
                       \n",pm,k,time_spent/(3600.0));
330             }
331             if (time_sum < 120.0) {
332                 printf("total: %f s\n",time_sum);
333             }
334             else if(time_sum < 60.0*120.0){
335                 printf("total: %f min\n",time_sum/60.0);
336             }
```

```
337                 else{
338                     printf("total: %f hrs\n",time_sum/(3600.0)
                           );
339                 }
340 
341                 if(pm == 0 && k == 9){
342                     pm = 1;
343                     k = -1;
344                     t = 1.0;
345                     e = 6.8;
346                 }
347             }
348             printf("accepted/rejected: %f\n",(float)accept/
                   reject);
349 
350             //cleanup
351             free(cellsh);
352             free(cells2h);
353             free(pointsh);
354             cudaFree(cellsd);
355             cudaFree(pointsd);
356 
357         printf("simulations completed!\n");
358 
359         //print out run-time
360         end2 = clock();
361         time_spent2 = (float)(end2 - begin2) / CLOCKS_PER_SEC;
362         if (time_spent2 < 120.0) {
363             printf("total time elapsed: % fs\n",time_spent2);
364         }
365         else if(time_spent2 < 60.0*120.0){
366             printf("total time elapsed: %f min\n",time_spent2
                       /60.0);
367         }
368         else{
369             printf("total time elapsed: %f hrs\n",time_spent2
                       /(3600.0));
370         }
371 
372 }
```

BIBLIOGRAPHY

[1] S.V. Jablan. Symmetry, ornament and modularity. *K & E Series on Knots and Everything*, 30, 2002.

[2] T.D. Lee and C.N. Yang. Question of parity conservation in weak interactions. *Physical Review*, 104(1), 1956.

[3] C.S. Wu. Experimental test of parity conservation in beta decay. *Phys. Rev.*, 105:1413–1415, 1957.

[4] R.C. Gillis. *3 He Ionization Chambers as Neutron Beam Monitors for the NPDGamma Experiment*. PhD thesis, University of Manitoba, 2006.

[5] V.E. Bunakov. Fundamental Symmetry Breaking in Nuclear Reactions. *Physics of Particles and Nuclei*, 26:115 – 147, 1995.

[6] V.W. Yuan, C.D. Bowman, J.D. Bowman, J.E. Bush, P.P.J. Delheij, C.M. Frankle, C.R. Gould, D.G. Haase, J.N. Knudson, G.E. Mitchell, et al. Parity nonconservation in polarized-neutron transmission through la 139. *Physical Review C*, 44(5):2187–2194, 1991.

[7] N. Fomin. First results from the npdgamma experiment at the spallation neutron source. *arXiv preprint arXiv:1209.1622*, 2012.

[8] Oak Ridge National Lab. Fundamental neutron physics beam line at sns. `http://neutrons.ornl.gov/fnpb/`, 2014.

[9] C. Crawford. The $n - {}^3He$ parity violation experiment. `http://battlestar.phys.utk.edu/wiki/images/0/05/N3he_nsac_2011-04-16.pdf`, 2011.

[10] A.D. Sokal. Monte carlo methods for the self-avoiding walk. *Nuclear Physics B-Proceedings Supplements*, 47(1):172–179, 1996.

[11] M.B. Chadwick, M. Herman, P. Obložinskỳ, M.E. Dunn, Y. Danon, A.C. Kahler, D.L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, et al. Endf/b-vii. 1 nuclear data for science and technology: cross sections, covariances, fission product yields and decay data. *Nuclear Data Sheets*, 112(12):2887–2996, 2011.

[12] H. Bethe and J. Ashkin. *Experimental Nuclear Physics*. Wiley, New York, 1953.

[13] James F Ziegler, MD Ziegler, and JP Biersack. Srim–the stopping and range of ions in matter (2010). *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 268(11):1818–1823, 2010.

[14] D. Kirk. Nvidia cuda software and gpu parallel computing architecture. In *ISMM*, volume 7, pages 103–104, 2007.

[15] V.E. Bunakov and V.P. Gudkov. Parity Violation and Related Effects in Neutron-Induced Reactions. *Nuclear Physics A*, 401(1):93 – 116, 1983.

[16] V.P. Gudkov. Parity Violation in the $n + {}^3He \to {}^3H + p$ Reaction: Resonance Approach. *Phys. Rev. C*, 82:065,502, Dec 2010.