The $Q_{weak}$ experiment measured a parity violating asymmetry of longitudinally polarized electrons scattered off an unpolarized liquid hydrogen target, which enabled the determination of the weak charge of the proton. The $Q_{weak}$ experiment measured the most precise asymmetry ever recorded in electron-proton scattering, leading to the most precise determination of the weak charge of the proton. However, there were several sources of background uncertainty in the experiment. The largest source of background uncertainty comes from the aluminum casing of the liquid hydrogen target; instead of scattering off the liquid hydrogen, a fraction of the polarized electrons scattered off the aluminum casing surrounding the liquid hydrogen, adding to the events of interest in the asymmetry measurement. In order to obtain a more accurate measurement of the weak charge of the proton, the background signal fraction due to scattering off the aluminum casing must be removed from the $Q_{weak}$ data. An initial determination of the background signal fraction resulted in a value of 2.85% ± 0.2%. A new approach to this correction is proposed here, which uses the rates from the target when it was filled with a low density hydrogen gas and then scaled up to the rate of the liquid hydrogen target. Though this method introduced a new error - the uncertainty of the density of the hydrogen gas - it removes the electronic dead time, accidentals, and beam current calibration uncertainties, and should result in the same background signal fraction as the previous method, but with a smaller error. This presentation states the current results of this analysis.
1 Introduction

1.1 The Standard Model

The Standard Model is a fundamental theory in particle physics. It encompasses three of the four fundamental forces, electromagnetic force, strong nuclear force, and weak force, but does not include gravity. The Standard Model does not include either dark matter or dark energy; it also does not explain the large difference in interaction strength between the various forces. The Standard Model is likely a low energy approximation.

1.1.1 Weak Interactions and Weak Charge

This paper will focus on only one of the three fundamental forces described by the Standard Model, the weak interaction. A weak interaction can only occur over a short range (10^{-15} m). In the Standard Model, a weak interaction is caused by the absorption of emission of one of two force carriers, either the W boson or the Z boson. The Z boson is transferred during the interaction of an electron and a proton. The interaction between an electron and a proton, and the transfer of the Z boson, is shown in Figure 1.

![Figure 1: The interaction between an electron and a proton and the transfer of a Z boson.](image)

The weak charge of a particle is the neutral weak analog of its electric charge. The Standard Model predicts the value of the weak charge for many particles.

1.2 The $Q_{weak}$ Experiment

The $Q_{weak}$ experiment took place in Hall C of Jefferson Lab from 2010 to 2012. It determined the weak charge of a proton by scattering longitudinally polarized electrons off an unpolarized liquid hydrogen target. The parity violating
asymmetry of the scattering was used to determine the weak charge of the proton. The $Q_{weak}$ experiment measured the smallest and most precise asymmetry ever recorded in electron-proton scattering—leading to the most precise calculation of the weak charge of a proton.\(^3\)

### 1.2.1 The $Q_{weak}$ Experimental Apparatus

A diagram of the $Q_{weak}$ experimental apparatus is shown in Figure 2.

![Diagram of the $Q_{weak}$ experimental apparatus](image)

Figure 2: A diagram of the $Q_{weak}$ experimental apparatus. The liquid hydrogen target cell is too small to see.\(^4\)

The $Q_{weak}$ experiment used a beam of longitudinally polarized electrons, that were accelerated to 1.16 GeV. This beam passed through a liquid hydrogen target, which acted as the proton source of the electron-proton scattering. The liquid hydrogen target was encased in aluminum, with thin regions where the beam entered and exited the target cell. A series of collimators and a toroidal magnet allowed only the electrons which scattered off the liquid hydrogen in the target cell to enter into one of eight quartz bar detectors. This eight fold azimuthal symmetry was important to the experimental design because it minimized experimental errors. When the electrons entered the quartz bars, they were moving faster than the speed of light in quartz, so they produce Cherenkov radiation— a bluish glow. At each end of the quartz bars, there were photomultiplier tubes, which detect the Cherenkov radiation and, in turn, send a signal to a computer which registers the signal as a "hit" on that quartz bar.\(^5\)

### 1.2.2 The $Q_{weak}$ Results

The $Q_{weak}$ experiment measured the smallest and most precise asymmetry ever recorded in an electron-proton scattering experiment. This lead to the
most precise measurement of the weak charge of a proton. The weak charge as predicted by the Standard Model is $0.0710 \pm 0.0007$. The weak charge of the proton as measured by the $Q_{\text{weak}}$ experiment is $0.064 \pm 0.012$, which is in agreement with the Standard Model. This result was obtained by analyzing only 4% of the total data.\(^3\)

### 1.3 Aluminum Background Signal Fraction

A background signal fraction, or a dilution, is simply removing part of a data set which has been corrupted by some source of error. The liquid hydrogen target is encased in aluminum. The areas where the electron beam entered and exited the casing were made of a thinner aluminum. These areas are called windows, not because they are transparent, but because they are made of very thin aluminum. The window located where the electron beam enters the target cell is known as the upstream window, and the window where the beam exits is known as the downstream window. A fraction of the electrons scattered from the aluminum windows, instead of from the protons which interferes with the results of the experiment. Electrons scattering off the aluminum windows instead of off the liquid hydrogen is the single largest source of background error in the $Q_{\text{weak}}$ experiment. The fraction of electrons which scattered off the aluminum windows as opposed to off the protons needs to be known precisely so that it can be removed from the data. This calculation is known as the aluminum dilution.\(^2\)

#### 1.3.1 The First Approach

The first attempt at the aluminum dilution was performed by Joshua Magee in his doctoral thesis. The dilution Magee used was simply the detector rate normalized to beam current of electrons which scatter off the windows as opposed to the rate of electrons which scattered off the liquid hydrogen in the target. The target cell when filled with liquid hydrogen is known as a full target. The rate was measured when the target cell was full, and when a vacuum was pulled on the target. Both of these rates were compared to get just the rate of scattering off of the aluminum windows. The aluminum dilution obtained by Magee was $3.2\% \pm 0.2\%$. However, this result needs to be multiplied by ten to get the true scattering, leading to a total error of 2.85%.\(^1\)

This error comes from four different sources: accidentals, beam current calibrations, electronic dead times, and radiative effects. An accidental occurs when both of the photomultiplier tubes on each end of a quartz tube fire, but there is not an electron in the quartz tube producing the signal. There is an uncertainty in the calibration of the electron beam. A discriminator is used to remove electronic noise from the signal, but it has a dead time so that if two electrons pass through the detector within 100 ns of each other, only the first electron is counted. The radiative effect occurs when an electron losses energy as it passes through the hydrogen in the target cell. Only electrons with the correct range of energies will make it to the quartz bar detectors. If an electron
A new approach to determine the aluminum dilution has been created which proposes a reduced error when compared to Magee's method, but should give the same dilution. Instead of just comparing the rates from the empty and full target, this method does not use the rate from the full target; the target was filled with hydrogen gas at one of three low densities. The rate due to the low density hydrogen can be compared to the rate of the empty target, and this result can be scaled up to the rate of the liquid hydrogen. The calculations for this method are shown below.\(^6\)

\[ Y = \text{"yield"} = \frac{\text{detector rate}}{\text{beam current}} \left( \frac{kH z}{\mu A} \right) \] (1)

\[ Y_{MT} = \text{yield from empty target} \] (2)

\[ Y_{0\text{Window}}^{0} = Y_{MT} \] (3)

\[ Y_{\text{gas}} = Y_{H2}^{\text{gas}} + Y_{\text{Window}}^{\text{gas}} \] (4)

\text{Assume: } Y_{\text{Window}}^{\text{gas}} \approx Y_{\text{Window}}^{0} \] (5)

\text{Then: } Y_{H2}^{\text{gas}} = Y_{\text{gas}} - Y_{\text{Window}}^{0} \] (6)

\text{Desired Diution:} \] \[ \frac{Y_{\text{Window}}^{\text{full}}}{Y_{H2}^{\text{full}}} \] (7)

\[ Y_{H2}^{\text{full}} = Y_{H2}^{\text{gas}} \ast \left( \frac{\rho_{\text{full}}}{\rho_{\text{gas}}} \right) \ast R_{\text{H2/gas}}^{\text{loss}} \] (8)

\[ Y_{\text{Window}}^{\text{full}} \ast \frac{Y_{\text{Window}}^{0 \text{Window}}}{R_{\text{H2/0}^{\text{loss}}}} \] (9)

On March 14, 2011, while the \(Q_{\text{weak}}\) experiment was running, the pressure and temperature inside the target cell were varied, which caused liquid hydrogen to transition into hydrogen gas. The pressure and temperature were varied in a way to produce three distinct densities of hydrogen gas. Because each density is relatively independent of the other two, the aluminum dilution can be calculated independently three times with this method, leading to a greater accuracy.\(^2\)

The advantage of this method is it all but eliminates the error due to the accidentals and the beam current calibration, since the gas target runs where taken on the same day - March 14, 2011 - at approximately the same beam current. Unfortunately, this method also introduces a new source of error, which is the largest source of error for this method.\(^6\) There is some uncertainty in the measurement of the density of the hydrogen gas. A comparison of the errors obtained from each method, Magee’s "Standard" method and the new "Gas Target" method, is shown below in Table 1.\(^7\)
Table 1: A comparison of the errors generated by the two methods for calculating the aluminum dilution

<table>
<thead>
<tr>
<th>Error Source</th>
<th>Standard</th>
<th>Gas Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gas Density Determination</td>
<td>XXXX</td>
<td>1%</td>
</tr>
<tr>
<td>BCM Calibration</td>
<td>1.5%</td>
<td>&lt; 0.08%</td>
</tr>
<tr>
<td>Simulation</td>
<td>1.5%</td>
<td>1.5%</td>
</tr>
<tr>
<td>Statistics and Accidentals</td>
<td>0.8%</td>
<td>0.15%</td>
</tr>
<tr>
<td>Dead time</td>
<td>1.5%</td>
<td>0.3%</td>
</tr>
<tr>
<td>Target Boiling</td>
<td>0.2%</td>
<td>0.2%</td>
</tr>
<tr>
<td>Total</td>
<td>2.72%</td>
<td>1.93%</td>
</tr>
</tbody>
</table>

1.4 The Density of the Hydrogen Gas

In order for the aluminum gas target dilution method to have a lower error than the standard method for calculating the dilution - while obtaining the same dilution - the density of the hydrogen gas inside the target cell and the error on that calculation must be known exactly.\(^6\)

The density of a gas is calculated from the pressure and temperature of the gas. The pressure and temperature inside the target cell were each measured with multiple sensors.

Both the pressure sensor and the thermometers have errors on their measurements. The errors on both sensors must be known before the density of the hydrogen gas can be calculated. The pressure sensor has an error reported by the manufacturer of ±0.25%.\(^8\) The thermometers had a calibration offset, which until this research, has not been calculated.

To calculate the thermometer calibration offset, pressure and temperature data were taken from a target cell warm up period on May 13, 2011. During this period, the pressure and temperature readings were on the vapor pressure curve for hydrogen. The vapor pressure curve is a precisely known relationship between pressure and temperature when they are in equilibrium, which only happens when the liquid is absorbing the latent heat of vaporization in making the transition to the gas phase. The vapor pressure curve for hydrogen is given by Equation (10).\(^2\)

\[
P(T) = 14.69595 * 10^{2.00662 + (-0.99708 \times 10^{-1} + 0.017484957 T)}
\]  

1.4.1 Methods for Calculating the Density of a Gas

Three methods for calculating density were explored in this research. The first is plugging a temperature and pressure measurement into the National Institute for Standards and Technology (NIST) database, located at http://webbook.nist.gov/chemistry/fluid/. Of the three, this method is the most widely accepted and has the smallest uncertainty in density at .25%. However, the only way of calculating the density with this method is by plugging the temperature and
pressure readings into a graphical interface, which can be time consuming if the density needs to be calculated at many data points.\textsuperscript{2}

The second method for calculating the density is through the use of the ideal gas law. The equation for calculating density from the ideal gas law is presented below in Equation (11), where $P$ is pressure, $MW$ is the molecular weight of the gas, $T$ is temperature, and $R$ is the ideal gas constant, $0.080206 \text{ atm} \cdot \text{L}^2 / \text{mol} \cdot \text{K}$.

$$\rho = \frac{P \cdot MW}{R \cdot T}$$

While this method is the simplest, during the gas target run on March 14, 2011, the hydrogen gas did not meet the conditions for an ideal gas, so this method also gives the largest error on the density calculation.

The third method for calculating the density of a gas is the method of virial coefficients, which is outlined below in Equations (12) - (15), where $P$ is pressure and $T$ is temperature.\textsuperscript{9}

$$\rho = \frac{P}{83.0597T + B(P) + C'(P)^2}$$

$$B = \frac{2939.77417 - 192795.22 + \frac{2289005.1}{P} + \frac{11094088}{P^2}}{82.0597}$$

$$C' = \frac{C - B^2}{T}$$

$$C = 388.682 * e^{\frac{4.5}{T}} \cdot [1 - e^{0.6(1 - (\frac{P}{T})^4}$$

This method has an error on the density calculation similar to that of the NIST database at 0.26%, and, since it is a series of equations, it is easily incorporated into a program which makes it quick to find the density at multiple data points.

A comparison of the three methods for finding density is shown below in Table 2, with the pressure and temperature pairs representative of the conditions inside the target cell during the gas target run.

<table>
<thead>
<tr>
<th>Pressure (psia)</th>
<th>68.3</th>
<th>41</th>
<th>18.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature (K)</td>
<td>71.5</td>
<td>70</td>
<td>64</td>
</tr>
<tr>
<td>NIST Density $(\frac{g}{cm^3})$</td>
<td>.0016146</td>
<td>.00098634</td>
<td>.00047760</td>
</tr>
<tr>
<td>Ideal Gas Law Density $(\frac{g}{cm^3})$</td>
<td>.001596799</td>
<td>.000979087</td>
<td>.000475365</td>
</tr>
<tr>
<td>Virial Coefficients Density $(\frac{g}{cm^3})$</td>
<td>.0016146</td>
<td>.00098627</td>
<td>.00047754</td>
</tr>
</tbody>
</table>

Table 2: A comparison of the three methods for finding the density of a gas given a pressure and a temperature

Since the method of virial coefficients has a similar error to the NIST database and gives similar densities, and it is easier to use with large data sets, it is the method with is used to calculate densities in this research.
2 Procedure

2.1 Calculating the Density of the Hydrogen Gas

2.1.1 Accessing the Raw Data

The raw data needed to calculate the density of the hydrogen gas is the pressure and temperature inside the target cell on March 14, 2011. This data is stored in EPICS files. Instructions for accessing the EPICS files can be found in Appendix A.

The temperature and pressure data over the March 14, 2011 run is shown below in Figure 3. The regions of relatively stable pressure are shown in the blue boxes. These are the regions where the density of the hydrogen gas needs to be calculated.

![Figure 3: A graph of pressure and temperature on March 14, 2011, the day of the gas target run. The blue boxes show areas of relatively stable pressure, where the density of the hydrogen gas will be calculated.](image)

2.1.2 Formatting the Raw Data

The raw data from the EPICS files includes a time stamp for each data point. However, the time stamp is formatted with colons and a decimal place, which makes it difficult to analyze the time data with a Python script. To solve this problem, the EPICS data was imported into a Python script, and the
time stamp was separated from its corresponding data point. The time stamp was then formatted to remove the colons and decimal point. The script then returned two lists, one of the formatted time stamps and one of the data, with the index of a time stamp in the list corresponding to the index of the data point in the data list taken at that time. This script is shown in Appendix A.

2.1.3 Calculating the Thermometer Offset

Equation (10) was solved for $T$ and a pressure measurement was plugged in to get a theoretical temperature. The theoretical temperature was compared to the measured temperature at the time of the pressure measurement to obtain the calibration offset. The method for finding the thermometer calibration offset is shown below in Equations (16) - (17).

$$
T_{\text{theoretical}} = -1748495 P^2 + (\log\left(\frac{P}{14.69595} - 2.0062\right) - 2.0237) P + 1.0044 \times \log\left(\frac{P}{14.69595}\right) + 48.08205 \tag{16}
$$

$$
\text{offset} = T_{\text{measured}} - T_{\text{theoretical}} \tag{17}
$$

This calculation was performed for the pressure sensor QW\_PT3, which was the only pressure sensor functioning on March 14, 2011, and the thermometers QWT\_moA, QWT\_moB, and QWT\_hphiA. These three thermometers are located after the target cell, but before the heater, as shown in Figure 4.

Figure 4: A diagram of the target cell and recirculating setup in the $Q_{\text{weak}}$ experimental apparatus. The location of the three thermometers, QWT\_moA, QWT\_moB, and QWT\_hphiA, are shown in the yellow box.

2.1.4 Calculating a Line of Best Fit for the EPICS Data

The data during the March 14, 2011 run, especially the pressure data, was quite noisy. In addition, there are restriction on the beam current and the magnet current such that only small slices of the regions in the blue boxes in
Figure 3 can be used. The beam current restriction is $1 \mu A \pm 0.2 \mu A$ and the magnet current restriction is $8921 A \pm 2 A$. Figure 5 shows a period of time which corresponds to the first blue box in Figure 3. Though the period is five hours long, only three small slices can be used. The three slices are identified with the labels "1", "2", and "3".

Figure 5: A graph of a five hour time slice from March 14, 2011, roughly corresponding to the first blue box in Figure 3. The top graph shows the pressure as measured by the sensor QW,PT3, and the temperature, as measured by the sensors QWT_moA, QWT_moB, and QWT_hphiA. The bottom graph shows the magnet current, measured by ibcm1, and the beam current, measured by qw:qtmpsidcct. The regions denoted by "1", "2", and "3" show the only slices in the time range where the magnet current was 8921 A and the beam current was $1 \mu A$.

The solution to the noisy data over a small period of time is to create a line of best fit for the pressure and temperature data, and then use that to determine the equation for the time versus density from 5:00 to 23:30 on March 14, 2011. Each of these equations is bounded by a ten minute period because the data fluctuates so much. The code used to calculate the lines of best fit is shown in Appendix A.
2.2 GEANT4 Simulations

2.2.1 Checking out the QwGEANT4 Source Code

All simulations in this research were performed using the QwGEANT4 software package. Directions for obtaining and building this software is found in Appendix B.

2.2.2 Defining New Materials

Three new materials were defined for this research in the materials source file, located at QwGEANT4/src/QweakSimMaterials.cc. The three materials defined was diatomic hydrogen gas at densities $1.158515 \times 10^{-3} \text{ g/cm}^3$, $.97991 \times 10^{-3}$ $\text{g/cm}^3$, and $.47577 \times 10^{-3}$ $\text{g/cm}^3$, which were representative of the densities inside the target cell during the March 14, 2011 run. The code added to the material source file is shown in Appendix B.

2.2.3 Running a Simulation on Jefferson Lab’s ifarm and Caching the simulations and Calculating Rates

All simulations presented in this paper were ran on the computer farm at Jefferson Lab, known as ifarm. Directions for running simulations on the ifarm, as well as analyzing the output file is shown in Appendix B, along with an example of the scripts used to run simulations of the farm.

Each simulation depicted 100,000 events, with an event being defined as an electron passing through the $Q_{\text{weak}}$ experimental apparatus. Simulations were run in eight conditions. The target cell was filled with either one of the three densities of hydrogen gas, or there was a vacuum pulled on the target cell. Each of these four target cell options were simulated with focus on either the upstream window or the downstream window. One hundred simulations of 100,000 events each were performed for each of the eight conditions, though, due to various problems, not all runs completed successfully. Only the results from the successful runs are presented in the results section of this paper.

Each simulation produced a .root file, which was analyzed using the directions in Appendix B to obtain the relative rate, which is the detector rate normalized to beam current, measured in $\text{kHz} \mu\text{A}^{-1}$, for all eight octants. This rate was then divided by eight to obtain the average rate for one octant. The relative rate for all successful simulations were averaged to obtain a final predicted relative rate for each condition.

2.2.4 Calculating the Uncertainty on Multiple Simulations

Each relative rate has an uncertainty associated with it. To determine to total uncertainty over multiple simulations, the below equation is used, where $n$ is the total number of simulations and $U_i$ is the uncertainty calculated from
simulation i, where i ranges from 1 to n.

\[ \frac{\sum_{i=1}^{n} U_i}{n} \sqrt{\frac{1}{n}} \]  

(18)

3 Results

3.1 Calculating the Density of the Hydrogen Gas

3.1.1 Calculating the Thermometer Calibration Offset

Table 3, shown below, shows the offsets calculated for all three thermometers, using two different pressure sensors.

<table>
<thead>
<tr>
<th></th>
<th>QW_PT3</th>
<th>QW_PT4</th>
</tr>
</thead>
<tbody>
<tr>
<td>QWT_moA</td>
<td>0.03256 K ± 0.00375593 K</td>
<td>0.046204 K ± 0.00598988 K</td>
</tr>
<tr>
<td>QWT_moB</td>
<td>-0.0609267 K ± 0.00474626 K</td>
<td>-0.047177 K ± 0.006099 K</td>
</tr>
<tr>
<td>QWT_hpHiA</td>
<td>-0.03962 K ± 0.0018099 K</td>
<td>-0.0276362 K ± 0.0056414</td>
</tr>
</tbody>
</table>

Table 3: The results of calculating many temperature calibration offsets for six different pressure/temperature sensor pairs and averaging the offsets. The overall temperature sensor offset was determined to be 30mK, with a generous uncertainty of 10mK.

3.1.2 Calculating a Line of Best Fit for the EPICS Data

Figures 6 - 8 graphs all of the lines of best fit density over the region from 5:00 to 23:30 on March 14, 2011.

![Figure 6: A graph of the lines of best fit for density calculated using QW_PT3 and QWT_moA.](image)
3.2 GEANT4 Simulations

Table 4, below, shows the rates and uncertainties obtained from averaging multiple simulations. Results for both the upstream and downstream window are shown, with simulations ran using a vacuum as well as hydrogen gas at densities $1.15815 \times 10^{-3}$ g cm$^{-3}$, $0.97991 \times 10^{-3}$ g cm$^{-3}$, and $0.47577 \times 10^{-3}$ g cm$^{-3}$.
<table>
<thead>
<tr>
<th></th>
<th>Downstream</th>
<th></th>
<th></th>
<th>Upstream</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rate $kHz \mu A$</td>
<td>Uncertainty $kHz \mu A$</td>
<td>Rate $kHz \mu A$</td>
<td>Uncertainty $kHz \mu A$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vacuum</td>
<td>52.34912299</td>
<td>.06727608</td>
<td>35.07450096</td>
<td>.064910022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1.59815 \times 10^{-3}$ g cm$^3$</td>
<td>52.24989594</td>
<td>.064722881</td>
<td>34.97516278</td>
<td>.063791258</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$.97991 \times 10^{-3}$ g cm$^3$</td>
<td>52.36311394</td>
<td>.062073953</td>
<td>34.90095074</td>
<td>.063753183</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$.47577 \times 10^{-3}$ g cm$^3$</td>
<td>52.37373766</td>
<td>.075766346</td>
<td>34.85906648</td>
<td>.067508105</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: The average rates and uncertainties obtained from all successful simulations.

Below are graphs visualizing the data in Table 3. Figure 9 is a graph of density versus relative rate for the downstream window and Figure 8=10 is the same graph, but for the upstream window.

Figure 9: A graph of density, in g cm$^{-3}$, versus relative rate, in $kHz \mu A$, for vacuum and three densities of hydrogen gas for the downstream window.
Figure 10: A graph of density, in $\frac{g}{cm^3}$, versus relative rate, in $\frac{kHz}{\mu A}$, for vacuum and three densities of hydrogen gas for the upstream window.

4 Discussion

4.1 Calculating the Density of the Hydrogen Gas

4.1.1 Calculating the Thermometer Calibration Offset

The calculated thermometer calibration offset of 30 mK ± 10 mK was smaller than expected, which reduced the uncertainty in the density of the hydrogen gas from 1.0 % 0.5%. This, in turn, reduced the total error from the gas target method from 1.93 % to 1.63 %. These changes in the errors are presented below in Table 5.

<table>
<thead>
<tr>
<th>Gas Density Determination</th>
<th>Standard</th>
<th>Gas Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXX</td>
<td>0.5%</td>
<td></td>
</tr>
<tr>
<td>BCM Calibration</td>
<td>1.5%</td>
<td>&lt; 0.08%</td>
</tr>
<tr>
<td>Simulation</td>
<td>1.5%</td>
<td>1.5%</td>
</tr>
<tr>
<td>Statistics and Accidentals</td>
<td>0.8%</td>
<td>0.15 %</td>
</tr>
<tr>
<td>Dead time</td>
<td>1.5%</td>
<td>0.3%</td>
</tr>
<tr>
<td>Target Boiling</td>
<td>0.2%</td>
<td>0.2%</td>
</tr>
<tr>
<td>Total</td>
<td>2.72%</td>
<td>1.63%</td>
</tr>
</tbody>
</table>

Table 5: A comparison of the errors generated by the two methods for calculating the aluminum dilution, with the gas target errors reflecting the calculation of the thermometer calibration offset.
4.1.2 Calculating a Line of Best Fit for the EPICS Data

This method seems to be producing lines of best fit for density which are giving the densities expected at each time period. Further work needs to be done to incorporate the data in Appendix C into a program which make it easy to calculate the density from the gas target run at any time during March 14, 2011.

4.2 GEANT4 Simulation

It was predicted that the graph of density versus relative rate for both the upstream and downstream windows would be linear. This was not the result. It is unclear at this point why the graphs are not linear, and requires further investigation.

5 Conclusion

While the gas target method for calculating the aluminum dilution introduces a new error into the calculation of the aluminum dilution - the uncertainty in the density of the hydrogen gas - it removes the beam current calibration errors, dead time errors, and accidental errors which plague the standard method for calculating the aluminum dilution. The error obtained by the gas target method was thought to be 1.93%. However, by carefully calculating the thermometer calibration offset, the error has been reduced to 1.63%. The line of best fit method for calculating densities seems promising and appears to be giving correct calculations for the density of the hydrogen gas at a particular time. The predicted rates calculated through GEANT4 simulations are not giving the expected result, so this requires further investigation. Overall, the gas target method for calculating the aluminum dilution produces a smaller error than the standard method for calculating the aluminum dilution, and is on track to produce the same dilution.

6 Future Work

Plans for future work include performing more GEANT4 simulations under all eight conditions to reduce the uncertainty of the predicted rate calculated for each condition, as well as improving the method for calculating the lines of best fit for the density. These two plans should help reduce the error on the aluminum dilution. In addition, the measured rate needs to be calculated from the gas rate data collected on March 14, 2011. This requires finding regions of data with meet certain beam current and magnet current requirements. The nonlinearity of the plots of density versus relative rate also needs to be investigated further.
Appendix A

Accessing the Raw Data

1. Open a web browser and go to the url userweb.jlab.org/ smithg/.
2. Under the heading "Targets", click "EPICS Log"
3. The gas target data is located under the link "Qweak2Archive".
4. The date of the gas target run was March 14, 2011, from 5:00 to 23:30.

Formatting the Raw Data

The following script, shown in Listing 1, is used to format the raw data imported from an EPICS log.

```python
import csv

def getTimesAndData ( filename ) :

    ""
    Pulls the time and value data from text files, removes times where data is not available, formats the time for ease of graphing and returns two arrays, time and corresponding value
    ""

    filename = filename + '.txt'

    # pulls the data in row by row
    results = []
    with open ( filename ) as inputfile:
        for row in csv.reader ( inputfile ) :
            results . append ( row )

    times = []
data = []

    # removes the empty element at the beginning
    # of the list
    results . pop ( 0)

    # separates the date, time, and value from each
    # other
    for x in results:  
        temp = str(x)
date = temp [2:12]
time = temp [13:31]
value = temp [33:len(temp)-2]
        if value[1:len(value)] != 'N/A':
            times . append ( time)
data . append ( value)
```

18
Calculating a Line of Best Fit for the EPICS Data

Listing 2 shows the Python script used to calculate lines of best fit for the pressure and temperature data from the EPICS log. It also calculates a line of best for the density of the hydrogen gas over the time period.

```python
import csv
import numpy
from math import exp
def getTimesAndData(filename):
    
    
    filename = filename + '.txt'

    results = []
    with open(filename) as inputfile:
        for row in csv.reader(inputfile):
            results.append(row)

    times = []
data = []

    # removes the empty element at the beginning
    # of the list
    results.pop(0)

    # separates the date, time, and value from each
    # other
    for x in results:
        # pulls the data in row by row
        results = []
        with open(filename) as inputfile:
            for row in csv.reader(inputfile):
                results.append(row)

        times = []
data = []

        # removes the empty element at the beginning
        # of the list
        results.pop(0)

        # separates the date, time, and value from each
        # other
        for x in results:
            
Listing 1: The script used to format the raw data from an EPICS log.

Calculating a Line of Best Fit for the EPICS Data

Listing 2 shows the Python script used to calculate lines of best fit for the pressure and temperature data from the EPICS log. It also calculates a line of best for the density of the hydrogen gas over the time period.
temp = str(x)
date = temp[2:12]
time = temp[13:31]
value = temp[33:len(temp)-2]
if value[1:len(value)] != 'N/A':
times.append(time)
data.append(value)

# removes the '.' and ':' from the times
# leaves just a number
newTimes = []
for x in times:
    temp = str(x).replace('.', '')
    temp = temp.replace(':', '')
    newTimes.append(temp)
times = newTimes

return times, data

def lineOfBestFit(xData, yData):
    # Finds the line of best fit given some x and y data and returns
    # an array of the coefficients of the line i.e if
    # y = mx + b is the line of best fit, this method returns
    # [m, b]
    xData = [float(x) for x in xData]
yData = [float(x) for x in yData]
    coefficients = numpy.polyfit(xData, yData, 1)
polynomial = numpy.poly1d(coefficients)
    ys = polynomial(xData)
    return coefficients

def findC(temperature):
    # Finds the C variable from the method of virial coefficients
    # for a given inputted temperature
    return 388.682*exp(45.5 / temperature)*(1 - exp(.6 * (1 - pow(temperature / 20, 4))))

def findB(temperature):
    # Finds the B variable from the method of virial coefficients
    # for a given inputted temperature
    # if the temperature is zero, quit the program to avoid a divide by
    return
# zero error
if (temperature == 0):
    print("Temperature is zero, divide by zero error")
    print("Temperature: ", temperature)
    exit()

return ((1939.7741 * temperature + -192795.22 + (-2289005.1 / temperature) + (11094088 / pow(temperature, 2))) / 82.0597) / temperature

def findCPrime(temperature):
    #Finds the C' variable for the method of virial coefficients
    #for a given inputted temperature
    return ((findC(temperature) - pow(findB(temperature), 2)) / 82.0597) / temperature

def findDensity(temperature, pressure):
    #Returns the density found through the method of virial coefficients for a given inputted pressure/temperature pair
    #
    #print(findB(temperature))
    #print(findC(temperature))
    #print(findCPrime(temperature))
    pressure = pressure / 14.695950254
    return (pressure / (82.0597 * temperature + findB(temperature) * pressure + findCPrime(temperature) * pow(pressure, 2))) * 2.0159

def findCorrespondingTimes(Time1, Data1, Time2, Data2):
    #Compares the arrays Time1 and Time2 and returns the arrays Time1Short and Time2Short which contain only the times which were the same in both arrays along #of the corresponding data points
    Time1Index = []
    Time1Short = []
    Data1Short = []
    Time2Short = []
    Data2Short = []
    # Time1Index = []
    Time2Index = []
    minIndex = 0
    maxLength = 0
    if (len(Time1) > len(Time2)):
        maxLength = len(Time1)
        minIndex = 0
        maxIndex = 0
    while (minIndex < minlength and maxIndex < maxLength):
        if (Time1[maxIndex] == Time2[minIndex]):
Time1Short.append (Time1[maxIndex])
Data1Short.append (Data1[maxIndex])
Time2Short.append (Time2[minIndex])
Data2Short.append (Data2[minIndex])
Time1Index.append (maxIndex)
Time2Index.append (minIndex)
maxIndex = maxIndex + 1
minIndex = minIndex + 1
elif (Time1[maxIndex] > Time2[minIndex]):
    minIndex = minIndex + 1
else:
    maxIndex = maxIndex + 1
# print (maxIndex, '\t', minIndex)
else:
    maxlength = len (Time2)
    minlength = len (Time1)
    maxIndex = 0
    minIndex = 0
while (minIndex < minlength and maxIndex < maxlength):
    if (Time2[maxIndex] == Time1[minIndex]):
        Time2Short.append (Time2[maxIndex])
    Data2Short.append (Data2[maxIndex])
    Time1Short.append (Time1[minIndex])
    Data1Short.append (Data1[minIndex])
    Time2Index.append (maxIndex)
    Time1Index.append (minIndex)
    maxIndex = maxIndex + 1
    minIndex = minIndex + 1
elif (Time2[maxIndex] > Time1[minIndex]):
    minIndex = minIndex + 1
else:
    maxIndex = maxIndex + 1
# print (maxIndex, '\t', minIndex)
return Time1Short, Data1Short, Time2Short, Data2Short

# names of the pressure and temperature files without the ".txt"
pressureFilename = "QW_PT3_Gas"
temperatureFilename = "QWT_hphiA_Gas"

# formats the time stamp and separates it from the data for both
# the pressure and data
pressureTimes, pressureData = getTimesAndData (pressureFilename)
temperatureTimes, temperatureData = getTimesAndData (temperatureFilename)
pressureIndex = 0
temperatureIndex = 0

# start time
startTime = 50000000000000

# 10 minutes after the start time
d endTime = 51000000000000

count = 0
print(len(pressureTimes), len(temperatureTimes))

# loops through all possible 10 minute intervals
while (pressureIndex < len(pressureTimes) - 1 and temperatureIndex < len(temperatureTimes) - 1):
    pressureTimesShort = []
    pressureDataShort = []
    temperatureTimesShort = []
    temperatureDataShort = []
    
    # get only the pressure data between start time and end time
    while (float(pressureTimes[pressureIndex]) <= endTime):
        pressureTimesShort.append(pressureTimes[pressureIndex])
        pressureDataShort.append(pressureData[pressureIndex])
        pressureIndex = pressureIndex + 1

    # get only the temperature data between start time and end time
    while (float(temperatureTimes[temperatureIndex]) <= endTime):
        temperatureTimesShort.append(temperatureTimes[temperatureIndex])
        temperatureDataShort.append(temperatureData[temperatureIndex])
        temperatureIndex = temperatureIndex + 1

    count = count + 1

    # if hour has not passed
    if (count != 6):
        # move the end time back 10 minutes
        endTime = endTime + 1000000000000

    # if one hour has passed
    else:
        # move the start time up one hour
        startTime = startTime + 1000000000000

        # move the end time to 10 minutes after the start time
        endTime = startTime + 1000000000000
        count = 0

    # count = count + 1

pressureLength = len(pressureTimesShort)
temperatureLength = len(temperatureTimesShort)

# prints the formatted pressure time range
print("Pressure Time Range:")
print(pressureTimesShort[0] [0:2],
      ",
      "
      pressureTimesShort[0] [2:4],
      "
      ( pressureTimesShort[0] % 2)
}
pressureTimesShort[0][4:6],
pressureTimesShort[0][6:len(pressureTimesShort[0]) − 1],
pressureTimesShort[pressureLength − 1][0:2],
pressureTimesShort[PressureLength − 1][2:4],
pressureTimesShort[pressureLength − 1][4:6],
pressureTimesShort[pressureLength − 1]

# prints the formatted temperature time range
print("Temperature Time Range:")

print(temperatureTimesShort[0][0:2],
      temperatureTimesShort[0][2:4],
      temperatureTimesShort[0][4:6],
      temperatureTimesShort[0][6:len(temperatureTimesShort[0]) − 1],
      temperatureTimesShort[temperatureLength − 1][0:2],
      temperatureTimesShort[temperatureLength − 1][2:4],
      temperatureTimesShort[temperatureLength − 1][4:6],
      temperatureTimesShort[temperatureLength − 1]

# finds the times in the pressure and temperature data which are the
# same and their corresponding data

pressureCoefficients = lineOfBestFit(pressureTimesShort, pressureDataShort)
temperatureCoefficients = lineOfBestFit(temperatureTimesShort, temperatureDataShort)

# prints the pressure line of best fit, formatted with math
# symbols
print("Pressure Line of Best Fit:")
print("P(t) = ", pressureCoefficients[0], "∗t + ",
      pressureCoefficients[1])

# prints the temperature line of best fit, formatted with math
# symbols
print("Temperature Line of Best Fit:")
print("T(t) = ", temperatureCoefficients[0], "∗t + ",
      temperatureCoefficients[1])

print("Density Line of Best Fit:")

# finds the times in the pressure and temperature data which are the
# same and their corresponding data
pressureTimesSame = []
pressureDataSame = []
temperatureTimesSame = []
temperatureDataSame = []

pressureTimesSame, pressureDataSame, temperatureTimesSame, temperatureDataSame = findCorrespondingTimes(
    pressureTimesShort, pressureDataShort, temperatureTimesShort, temperatureDataShort)

# if there are no corresponding times in a 10 minute interval, quit
# the program
if (len(pressureTimesSame) == 0):
    print("No Corresponding Times")
    exit()

newDensity = []

# finds the density for each pressure/temperature pair where the time
# stamps were the same
for i in range(0, len(pressureTimesSame)):
    newDensity.append(findDensity(float(temperatureDataSame[i]), float(pressureDataSame[i])))

# finds the line of best fit for the densities found above
densitySameCoefficients = lineOfBestFit(pressureTimesSame, newDensity)

# print the density line of best fit, formatted with math symbols
print("D(t) = \), densitySameCoefficients[0], "t + "
    densitySameCoefficients[1])

print()
Appendix B

Checking Out the QwGEANT4 Source Code

1. Open terminal.
2. Type "ssh -X username@login.jlab.org", and replace "username" with jlab username. Enter jlab password when prompted.
3. Type "ssh -X ifarm" and enter jlab password when prompted.
4. Type "svn checkout https://qweaksvn.jlab.org/repos/simulation/trunk/QWFEANT4"
5. Wait for it to finish downloading, then type "nano .login" and add the following lines to the end of the file:
   "setenv JLAB_VERSION 1.3"
   "setenv JLAB_ROOT /site/12gev_phys"
   "source ${JLAB_ROOT}/celjlab.csh"
6. Use Control-O to save the changes and Control-X to exit nano.
7. Type "cd QWGEANT4"
8. Type "mkdir build"
9. Type "cd build"
10. Type "cmake .."
11. Type "make"

Note: If changes are made to the source code, the following commands will need to be ran:

1. Type "cd build"
2. Type "make"

Defining New Materials

The below code, shown in Listing 3, was added to the file QWGEANT4/src/QweakSimMaterial.cc to diatomic hydrogen gas at three different densities.

```cpp
// Gas H2 .00159815 Non-STP Diatomic
name = "H2Gas_00159815_Diatomic_NonSTP";
density = 0.00159815*g/cm3;
nelements = 1;
temperature = 71.5*kelvin;
pressure = 470911.9*pascal;
G4Material* __attribute__((unused)) matGasHydrogen_00159815_Diatomic_NonSTP = new G4Material(name, 
density, nelements, kStateGas, temperature, pressure);
```
Running Simulations on ifarm and Analyzing the Resulting .root File

Running Simulations on ifarm

1. Create a file in the folder "QWGEANT4/macros/" modeled after Listing 4. It defines the name of the configuration file as well as the name of the resulting .root file. This script should have the extension ".in".

2. Create a configuration file in the folder "QWGEANT4/" modeled after Listing 5. This file defines all the neccessary configurations for the Q\textsubscript{weak} experiment. The only lines in this file which need to be
changed are 286, 278, 281, and 296. Line 286 sets the type of interaction which will take place inside the target cell. Line 278 sets the material inside the target cell. This material is defined in the file QWGEANT4/src/QweakSimMaterial.cc. Line 281 needs to be set to ”LH2” so the material chosen in line 286 fills the interior of the target cell. Line 296 chooses which region of the target cell the simulation should focus on. Entrance target window is the upstream window and exit target window is the downstream window. This file should have the extension ”.mac”.

3. The final file needed to run a simulation on ifarm should be created in the folder ”QWGEANT4/scripts” and should be modeled after Listing 6. The number of simulations to be ran on ifarm can be set in this file by changing lines 6 and 8. The number of simulated events is also set in this file. This file should have the extension ”/mac”.

4. The simulation is then submitted to the ifarm by typing the following command in the terminal from the QWGEANT4/ folder: bash scripts/-fileName.mac

```bash
/control/execute
 myQweakConfiguration_H2Gas_00159815_Diatomic_NonSTP.mac
/control/execute macros/noTracking.mac
/random/setSavingFlag 0
/random/setDirectoryName random/%name%_%jobid%
/random/setSeeds %seedA% %seedB%
/TrackingAction/TrackingFlag 3
/Target/SetCenterPositionInZ −652.466 cm
/Target/SetTargetMaterial Vacuum
/MagneticField/SetActualCurrent 8920.10 A
/EventGen/SetBeamEnergy 1.16 GeV
/PrimaryEvent/SetBeamDirectionX −0.3 mrad
/PrimaryEvent/SetBeamDirectionY −0.41 mrad
/Trigger/Disable all
/Trigger/Enable cer
/Trigger/Show
#set the root file name
/Analysis/RootFileName rootfiles/%name%_%jobid%_%workdir/%name%_%jobid%.root
#number of events
/run/beamOn %nevents%
```

Listing 4: Example of a script in ”QWGEANT4/macros”
# Macro file myQweakConfiguration.mac
# Base configuration of the Qweak Apparatus
# Defaults to production LH2 simulation

# Store Tracks
/tracking/storeTrajectory 1

# Octant Selection
# Valid values range from 0 to 12
# default 0: events thrown in all octants
# 1−8: events thrown in corresponding octant
# 9−12: events thrown in two octants (9 for 1+5, 12 for 4+8)
/EventGen/SelectOctant 11

# Set beam rastering region
/PrimaryEvent/SetRasterXmin -2.0 mm
/PrimaryEvent/SetRasterXmax 2.0 mm
/PrimaryEvent/SetRasterYmin -2.0 mm
/PrimaryEvent/SetRasterYmax 2.0 mm

# Detector Hut related
/DetectorHut/CollimatorWall/SetCollimatorWallMaterial ShieldingConcrete

# Cerenkov Detector related
# Air : without Cerenkov light production
# Quartz : with Cerenkov light production
# Angle is defined along vertical (y) axis
# - a tilt angle of 0.0*deg means vertical
# - negative angle will tilt detector towards target
/Cerenkov/SetCerenkovMaterial Quartz
/Cerenkov/SetTiltingAngle 0.0 degree
/Cerenkov/SetPreradiatorMaterial PBA

# Define the number of Cerenkov Bars
/Cerenkov/SetNumberOfDetectors 9

# Move Cerenkov Detectors to actual position
/Cerenkov/Cerenkov1/SetCenterPositionInY 334.724 cm
# ELOG Detector /74: Positions of background detectors

For practical reasons (MD9 is in the same octant as MD1) the background detector is here in the opposite octant than it actually was in (behind MD5).

PRELIMINARY: More accurate numbers will require both the final survey and the local survey of the monument with respect to the detector itself.

"Det. Origin" is (assumed to be) the center of the back of the bar, outer edge.

Z position of Det. Origin = 737.86 cm
- 1 cm (half thickness) = 736.86 cm (Z in MC)
X position of Det. Origin = 377.84 cm
- 9 cm (half width bar) = 368.84 cm (Y in MC)
Y position of Det. Origin = 72.86 cm (X in MC)

# Trigger Scintillator related

Air : without Cerenkov light production
Quartz : with Cerenkov light production

angle is defined along vertical (y) axis
- a tilt angle of 0.0° deg means vertical
- positive angle will tilt detector against target
SETSCINTILLATOR/SETSCINTILLATORMATERIAL Polyethylene

# TS SetCenterPosition works using local coordinates. Set values as if in top octant.
/TS/SETCENTERPOSITIONINX 0.0 cm
/TS/SETCENTERPOSITIONINY 328.0 cm
/TS/SETCENTERPOSITIONINZ 545.0 cm

# Example of setting position of second TS below. Note that above commands
# by default only translates TriggerScintillator1
/TS/SETCENTERPOSITIONINZ 540.0 cm

/TriggerScintillator/SETTILTINGANGLE 0.0 degree

# = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
# GEM related (Region1)
# = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
/GEM/SETFRONTCENTERPOSITIONINX 0.0 cm
/GEM/SETFRONTCENTERPOSITIONINY 25.0 cm
/GEM/SETFRONTCENTERPOSITIONINZ -543.170 cm
/GEM/SETBACKCENTERPOSITIONINX 0.0 cm
/GEM/SETBACKCENTERPOSITIONINY 25.0 cm
/GEM/SETBACKCENTERPOSITIONINZ -535.0 cm
/GEM/SETFRONTCAMBERROTATIONANGLEINPHI 90.0 degree
/GEM/SETBACKCAMBERROTATIONANGLEINPHI 90.0 degree
/GEM/SETMASTERCONTAINERMATERIAL Air
/GEM/SETSUBCONTAINERMATERIAL ArCO2
/GEM/SETFRAMEMATERIAL NemaG10
/GEM/SETFOILMATERIAL Kapton

# = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
# HDC related (Region2)
# = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
/HDC/SETFRONTCENTERPOSITIONINX -0.276 cm
/HDC/SETFRONTCENTERPOSITIONINY 51.104 cm
/HDC/SETFRONTCENTERPOSITIONINZ -337.355 cm
/HDC/SETBACKCENTERPOSITIONINX -0.141 cm
/HDC/SETBACKCENTERPOSITIONINY 54.872 cm
/HDC/SETBACKCENTERPOSITIONINZ -294.677 cm

# Set Phi angle for HDC Package 1
# Package 2 HDC automatically 180.0 in opposite octant
/HDC/SETROTATIONANGLEINPHI 90.0 degree
/HDC/SETMASTERCONTAINERMATERIAL Air
/HDC/SETSUBCONTAINERMATERIAL ArCO2
/HDC/SETFOILMATERIAL Mylar
/HDC/SETFRAMEMATERIAL NemaG10
/HDC/ SetWirePlaneMaterial ArCO2

#Disable Package 2 HDC
/HDC/HDC2/ SetFrontCenterPositionInY 200.104 cm
/HDC/HDC2/ SetBackCenterPositionInY 204.872 cm

# VDC related (Region3)

# here you can set the position of the Front VDC (upstream)
# The Back VDC will automatically repositioned relative to the Front VDC
/VDC/SetMeanTrackAngle 20.57 degree
/VDC/SetFrontBackDistance 50.0 cm
/VDC/ SetFrontCenterPositionInX −0.08 cm
/VDC/ SetFrontCenterPositionInY 272.255 cm
/VDC/ SetFrontCenterPositionInZ 444.61 cm

/VDC/ SetVDC_SubContainerMaterial Vacuum
/VDC/ SetVDC_SubContainerMaterial Ar−C2H6 40−60

# Angle Orientation in Phi (Input Package 1)
# Package 2 automatically 180.0 degrees in opposite octant
# 90.0 degree = octant 1 in 12 o'clock position
# 45.0 degree = octant2
/VDC/ SetRotationAngleInPhi 90.0 degree
/VDC/ SetRotationAngleInPhi 45.0 degree
/VDC/ SetRotationAngleInPhi 0.0 degree

# angle in respect to SHORT Frame side
# 90−26.57=63.43
/VDC/ DriftCell/ SetWireAngleFront 65.57 degree
/VDC/ DriftCell/ SetWireAngleBack −65.57 degree
/VDC/ DriftCell/ SetWireAngleFront 45.0 degree
/VDC/ DriftCell/ SetWireAngleBack −45.0 degree
/VDC/ DriftCell/ SetMaterial Ar−C2H6.40−60
/VDC/ DriftCell/ SetMaterial NemaG10
/VDC/ DriftCell/ SetFullHeight 70.0 cm
/VDC/ DriftCell/ SetDCNumberPerPlane 341

# Put Package 1 VDC in contraction position (Must have phi of 90 degree)
/VDC/VDC2/ SetRotationAngleInPhi 90.0 degree
/VDC/VDC2/ SetFrontCenterPositionInY 115.225 cm

# Put Package 2 VDC in contraction position (Must have phi of 270 degree)
/VDC/VDC2/ SetRotationAngleInPhi 270.0 degree
/VDC/VDC2/ SetFrontCenterPositionInY −115.225 cm

# Thickness = 64/64 inch
/VDC/ DriftCell/ SetFullThickness 25.4 mm

# Thickness = 68/64 inch
# Event generator related

# Usually this will be changed by a user macro that calls this configuration macro.

# List of Generators (Included in src/QweakSimEPEvent.cc):
# 0 – Combination of all reactions
# 1 – LH2 elastic (default)
# 2 – Al elastic
# 3 – Al quasi–elastic (proton)
# 4 – Al quasi–elastic (neutron)
# 5 – LH2 inelastic (delta resonance)
# 6 – Moller scattering
# 7 – LH2 radiative lookup table (3.35 GeV)
# 8 – Al quasi–elastic, Bosted
# 88 – pion
# 9 – Al nuclear inelastics SPS
# 10 – Al GDR
# 11 – Al inelastic (Bosted)
# 12–17 – Al alloy (Zn, Mg, Cu, Cr, Fe, Si, respectively)

/EventGen/SelectReactionType 2

# Target related
# LH2 Target by default

/Target/SetCenterPositionInZ -652.67 cm

/Target/SetTargetCellMaterial Aluminum

/Target/SetTargetLength 34.346 cm

/Target/SetTargetMaterial H2Gas

# Target Selection: LH2 (default), USA11p, USA12p, USA14p, USC, DSA12p, DSA14p, DSA18p, DSC

/Target/SetTarget LH2

# Selects a region of target:

# 1 - LH2 in target, no Al end caps (default)

# 2 - Entrance target window

# 3 - Exit target window

# 4 - 1% US Al Dummy Target

# 5 - 2% US Al Dummy Target

# 6 - 4% US Al Dummy Target

# 7 - 2% DS Al Dummy Target

# 8 - 4% DS Al Dummy Target

# 9 - 8% DS Al Dummy Target

# 10 - US C Dummy Target

# 11 - DS C Dummy Target

/EventGen/SelectReactionRegion 3

# Magnetic Field related

# The preferred way to load a field map is to use

# the macros in macros/peiqing_*.mac

/Control/execute macros/peiqing_2011.mac

# Older Magnetic Field map that is no longer used.

/#control/execute macros/peiqing_2007.mac

# Random Seed Handling

# Inform the RunManager to systematically save the seed

/random/setSavingFlag 0

# Copy the following command line to your user macro and uncomment it

# to use the seeds of the last event in last run

#/random/resetEngineFrom currentEvent.rndm

# Or use specific seeds from file 'myseed.rndm', for example

#/random/resetEngineFrom myseed.rndm

# Pion Wall related

# Comment out the line below to leave the Pion Wall in (defaults to 4" thick)
# Uncomment the following line to change the Pion Wall thickness to 2"
#/PionWall/SetThicknessInZ 5.08 cm

# Tungsten Shutters related
#
# Comment out the following two lines to enable the Tungsten Shutters.
# By default they are disabled.
#/WShutters/WShutters1/Enable
#/WShutters/WShutters2/Enable

# LeadGlass related
#
# Comment out the line below to leave the LeadGlass in
#/LeadGlass/Disable

# PMTOnly related
#
# Comment out the line below to leave the PMTOnly in
#/PMTOnly/Disable

# Lumi related
#
# Comment out the line below to leave the Lumi in
#/Lumi/USDisable

Listing 5: Example of a configuration file

ame=myRun_H2Gas_00159815_Diatomic_NonSTP_farm

# first job number (can change if just want to add
# on to the get more simulations)
firstjob=51

# last number of jobs one would like done
lastjob=56

# number of events simulated in each Geant4 file
nevents=100000

# batch system
batch=xml
qweant4=~/QWGEANT4

let jobid=${firstjob}
while [[ ${jobid} -le ${lastjob} ]]; do
  echo "Job ${jobid}"
# create a random seed so that the simulations are not all the same
unixtime = 'date +%s'
nanosecond = 'date +%N'
let seedA=10#$
{
unixtime}
+
{
jobid}
let seedB=10#$
{
nanosecond}
+
{
jobid}
mkdir -p random/$
{
name}
{
jobid}
mkdir -p jobs/$
{
batch}
mkdir -p macros/jobs
# replace the variables in the file macros/$
{
basename}
.in
# which creates the mac file that Geant4 uses
sed -e "s|%nevents%|${nevents}|g" 
- e "s|%qwgeant4%|${qwgeant4}|g" 
- e "s|%seedA%|${seedA}|g" 
- e "s|%seedB%|${seedB}|g" 
- e "s|%jobid%|${jobid}|g" 
- e "s|%name%|${name}|g"
macros/$
{
name}
.in
> macros/jobs/$
{
name}
{
jobid}
.mac
# replace the variables in the file $
{
basename}
.in
# which creates the job file that is submitted to the farm
sed -e "s|%nevents%|${nevents}|g" 
- e "s|%qwgeant4%|${qwgeant4}|g" 
- e "s|%seedA%|${seedA}|g" 
- e "s|%seedB%|${seedB}|g" 
- e "s|%jobid%|${jobid}|g" 
- e "s|%name%|${name}|g"
jobs/job.$
{
batch}
.in
> jobs/$
{
batch}
{
name}
{
jobid}
.
{
batch}
#define where the root file will be stored, check to see if a root file
# exists, if not then submit the job if it is there do not submit it
# if so then move to the next job
rootfile=/$
{
USER}
/scratch/$
{
name}
.
{
jobid}
.root
if [ -e $rootfile ] ; then
echo "File $rootfile already exists on mss."
else
echo "File $rootfile doesn’t exist. Will submit job."
if [ "$batch" == "xml" ] ; then
jsub -xml jobs/$
{
batch}
{
name}
{
jobid}
.
{
batch}
fi
if [ "$batch" == "sh" ] ; then
qsub jobs/$
{
batch}
{
name}
{
jobid}
.
{
batch}
"
qsub jobs/$
{
batch}
{
name}
{
jobid}
.
{
batch}
fi
let jobid=${jobid}+1
Analyzing the Resulting .root File

To analyze a .root file produced from a simulation run on ifarm using the above format, use the following commands with the terminal navigated to the QWGEANT4/ folder.

1. Type `"jcache get /mss/hallc/qweak/misc/username/rootfile/path/to/.root"`
2. After waiting a few minutes for the file to be cached, type `"cp /cache/hallc/qweak/misc/username/rootfiles/path/to/.root ."`
3. Type `"./build/QweakSimRoot"`
4. Type `"new TBrowser"`
5. When the TBrowser window appears, double click on the .root file in the file navigator located on the left side of the window. Then, click back onto the terminal.
6. In the root terminal, press enter.
7. Type `"TH1D * h1 = new TH1D ("h1", "Title", 100, 1., 1.)"`
8. Type `"h1->Sumw2()"`
9. Type `"QweakSimG4.Tree.Draw("PrimaryOriginVertexKineticEnergy <<h1", "Cerenkov.PMT.PMTTotalRate * (Cerenkov/PMT.OMTLeftNbOfPEs > 0 && Cerenkov.PMT.PMTRightNbOfPEs > 0)")"`
10. Type `"Double_t rate; Double_t error"`
11. Type `"rate = h1->IntegralAndError (1, 99, error)"`
12. Type `"rate / 8"`
13. Type `"error / 8"`
Acknowledgements

This work was supported in part by the National Science Foundation under Grant No. PHY-1359364.
Works Cited


2. Private Correspondence

3. Androic, D; at.el. First Determination of the Weak Charge of the Proton. PRL. 2013, 111.


5. Allison, T.; at.el. The $Q_{\text{weak}}$ Experimental Apparatus.


