

Sean Dobbs is a senior at Carnegie Mellon University, majoring in Physics and Computer Science. In addition to his appointment at Jefferson Lab, he works in the Carnegie Mellon Medium Energy Physics group, writing simulation programs to assist with data analysis for an experiment investigating the hypernuclear resonances of helium. Previous to this, he worked for Mitretek Systems, Inc., a not-for-profit company, writing weather radar data analysis systems for the National Weather Service. An experienced software developer, he hopes to lend his expertise to other programs in nuclear or particle physics in his graduate studies and beyond.

Jens-Ole Hansen received an undergraduate degree in physics from the University of Frankfurt, Germany, in 1989 and a Ph.D. in nuclear physics from Massachusetts Institute of Technology in 1995. Subsequently, he held a postdoctoral appointment at Argonne National Laboratory, working primarily on the HERMES experiment at DESY. He is currently a staff scientist at Jefferson Lab, where he is a member of the Hall A group. His scientific interests include the spin structure of the nucleon, nucleon electromagnetic form factors, and few-body nuclear physics. He has also been active in the development of scientific computing software.

COMPARISON OF TWO WIRE CHAMBER TRACK RECONSTRUCTION ALGORITHMS USED IN HALL A AT JEFFERSON LAB

SEAN DOBBS AND J. O. HANSEN

ABSTRACT

The Hall A collaboration at Jefferson Lab is currently in the process of redesigning and rewriting its physics analysis software in C++, using the ROOT libraries developed at CERN, to replace the Fortran-based ESPACE analyzer. In this paper, we carry out a detailed comparison of the wire chamber tracking results of both software packages. To this end, reconstruction of target quantities from detected tracks through the Vertical Drift Chambers (VDCs) in both spectrometer arms has been recently added to the C++ analyzer, to bring it to the same level as ESPACE. A study of the differences in the outputs of both analysis programs shows that while the output of the C++ analyzer is suitable for calibrations and studies that do not require high resolution, there are still many subtle differences that need to be resolved before it is ready for a production-quality release. The problems are most likely due to different cluster matching and track fitting algorithms used by both programs for tracks through the VDCs, as well as certain corrections missing from the C++ code. These problems and some possible solutions are discussed.

INTRODUCTION

The Hall A [1] collaboration at Jefferson Laboratory [2] is currently in the process of redesigning and rewriting its physics analysis software [3] in C++ using the ROOT libraries developed at CERN [4]. This software is intended to replace ESPACE [5], the current Fortran-based analysis software used in Hall A. Major goals of the new software are increased flexibility to support a variety of new detectors planned for future hall upgrades with a minimum of extra development and the ability to handle the larger data streams these detectors would generate.

Most nuclear physics at Jefferson Lab is conducted at the Continuous Electron Beam Accelerator Facility (CEBAF). CEBAF is able to deliver a continuous beam of electrons with energies ranging from 800 MeV to 6 GeV to three different experimental halls simultaneously. Experiments in the

halls use this beam to perform detailed measurements of nuclear interactions.

Hall A uses a pair of high resolution spectrometers (HRS) with superconducting magnets to measure the charged particles scattered from the target at various angles [6]. In each spectrometer, particle tracking information is obtained using Vertical Drift Chambers (VDCs) [7]. A VDC consists of a plane of sense wires sandwiched between conducting planes that are kept at a high voltage. This creates a strong electric field in the chamber which is constant and perpendicular to the wire plane throughout most of the chamber, though near the wires it becomes approximately radial. The chamber is filled with a gas that is easily ionized as charged particles pass through it. The electrons liberated from the gas when ionized drift towards the wires along the electric field lines at a nearly constant velocity. As these electrons drift into the region of very strong electrical field near the wires, they acquire enough energy to ionize other gas molecules, creating

an avalanche effect that greatly increases the number of ionized atoms generated by a particle track. The drift of the electrons towards the sense wires and that of the ions in the other direction induces a current in the sense wires, so the avalanche effect creates a signal that is easily measured with standard electronics [9, 10].

Each HRS arm contains two pairs of VDCs. The two VDCs of each pair are labelled U and V. The separation between like pairs of VDC planes is 332.7mm for the left arm and 334.8mm for the right arm. Each VDC contains 368 sense wires, and the wire spacing and vertical extent of the chambers give a cell width of 4.24mm and a cell height of 26mm for each sense wire. The VDCs are filled with a gas mixture of 62% argon to 38% ethane and are operated at a voltage of 4.0kV. The VDCs are oriented horizontally in the laboratory frame, so that the central angle that tracks take through the VDCs is 45°. This means that a particle on its way through a VDC plane will travel through on average five cells. The sense wires for the U and V planes lie orthogonal to each other and 45° to the main axes of the HRS [8, 13]. Several coordinate

Next, in the VDC analysis, a series of consecutive wires with induced signals, called a cluster, is identified. Each cluster usually corresponds to a single track. The drift times for each hit in a cluster are typically converted to drift distances, and a linear fit is performed on these distances to determine the local cross-over point in the plane and the local angle of the track. However, the relation between drift time and perpendicular drift distance is not quite linear, in part because of radial nature of the field lines near the sense wires, which increases the proportion of events with short times, and also from the fact that some tracks leave the VDC plane before they leave the wire cell, which can produce excessively long drift times [10]. These aberrations are corrected by the analysis software.

The algorithms used by both ESPACE and the C++ analyzer to calculate quantities in the detector coordinate system differ slightly. ESPACE uses pairs of calculated cross-over points in the U and V planes to calculate the directions of the path in a coordinate system defined by the U and V planes. This system is rotated 45° around the z-axis from the detector coordinates, so ESPACE must then

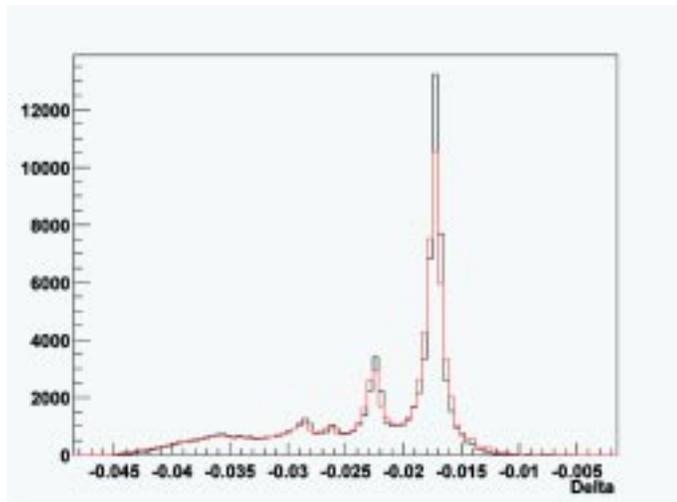


Figure 1: Right arm δ spectrum.

systems are used in the reconstruction algorithm. The reader interested in the exact definition of these systems is directed to references [8, 12] for a detailed discussion.

The basic algorithm for particle tracking in a VDC is straightforward. First, the so-called drift times are measured, that is, the time elapsed between the initial ionization in the sensing range of a wire and the creation of a signal in the wire. The time measurement is done with a time-to-digital converter (TDC) in “common-stop” mode, where the common stop signal is given by the delayed event trigger produced by the data acquisition unit, and the TDCs are started with the amplified and discriminated signals from the sense wires. Thus, higher TDC values correspond to shorter drift times.

$$y_{ig} = \sum_{i,j,k,l} C_{ijkl}^y x_{fp}^i \theta_{fp}^j y_{fp}^k \phi_{fp}^l \quad (1)$$

$$\theta_{ig} = \sum_{i,j,k,l} C_{ijkl}^T x_{fp}^i \theta_{fp}^j y_{fp}^k \phi_{fp}^l \quad (2)$$

$$\phi_{ig} = \sum_{i,j,k,l} C_{ijkl}^P x_{fp}^i \theta_{fp}^j y_{fp}^k \phi_{fp}^l \quad (3)$$

$$\delta = \sum_{i,j,k,l} C_{ijkl}^D x_{fp}^i \theta_{fp}^j y_{fp}^k \phi_{fp}^l \quad (4)$$

perform a further transformation to get the detector coordinates. The C++ analyzer, however, uses each related pair of U and V plane clusters to form a local cross-over point and direction using input from the cluster fitting procedure. These local directions are then used to match up clusters from the first VDC pair to similar clusters in the second VDC pair, from which the global track is calculated in detector coordinates. This algorithm was chosen because it was more natural to the object-oriented design used in the C++ analyzer.

Both programs then convert detector coordinates into focal plane coordinates. Focal plane coordinates are defined in a so-called “rotating” reference frame. This frame is defined by a rotation of the detector coordinate system about the y-axis such that the z-axis points along the *local* central angle at a given point in the focal plane. The rotation angle is roughly 45°, but varies due to the spectrometer’s dispersion.

Focal plane coordinates present favorable numerical characteristics for target reconstruction as θ is always minimized. A set of zeroth-order matrix elements, such as those described in the following paragraphs, are used in the conversion to help approximate the local central angle [12].

After the particle tracks at the VDCs are computed at the focal plane, that information can be used to reconstruct the corresponding quantities at the target. From the focal plane, the transformation of the measured quantities through the magnetic elements of the spectrometer can be calculated using a technique similar to the TRANSPORT formalism [11].

For target reconstruction through the spectrometer, there are five coordinates to consider: the displacement and angular deviation in the dispersive direction, x and θ ; the displacement and angular deviation in the non-dispersive direction, y and δ ; and the fractional momentum difference, δ , which is defined as $p/p_0 - 1$, where p is the particle's momentum and p_0 is the central momentum of the detector. Since the VDCs only detect four quantities, x , y , θ , and δ , mathematically we can only reconstruct four quantities at the target. We choose those with the most physical relevance: y , θ , ϕ , and δ . x_{tg} is normally defined by the vertical beam position and can be measured independently. y_{tg} is essentially the interaction position along the beam. θ_{tg} and δ_{tg} are the in-plane and out-of-plane scattering angles respectively.

The focal plane coordinates are linked to the target coordinates by the Transport Tensor as follows: where C_{ijkl} are the coefficients describing the various matrix elements. In practice, the matrix elements are stored up to fifth order. The matrix elements are numerically calculated through a series of calibration experiments by fitting experimental data, and are then stored for later use [12].

The target coordinates can now be used to determine the physics of the experiment. The goal of this paper is to study differences between the results given by ESPACE and the C++ analyzer at the detector coordinate level and also at the very important target level. The next section will discuss the data and analysis techniques used. The section following that will present some of the more important differences between ESPACE and the new analyzer. The final section will discuss these differences in more detail and their implications. Areas of investigation and potential further improvements to the C++ analyzer will also be presented.

METHODS

The data used for this study is from an optics commissioning run (run number 1146 [14]) for Hall A experiment E97-111, taken on September 30, 2000. A sieve slit was installed in the spectrometer, in conjunction with a target consisting of nine thin ^{12}C foils placed 4cm apart. This setup is typical for collecting data to be used in optics calibrations. The in-

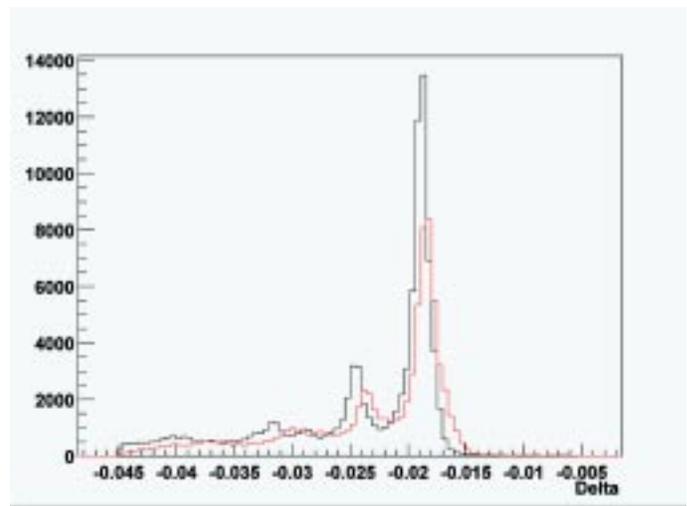


Figure 2: Left arm δ spectrum.

coming electron beam was at an energy of 825 MeV. The right spectrometer arm was set to an angle of 16.009° with a central momentum of 837.00 MeV. The left arm was set to an angle of 15.965° with a central momentum also of 837.00 MeV.

As part of this project, the target reconstruction algorithm as described in the previous section (eqs. 1-4) was implemented in the C++ analyzer. The first 200,000 physics events were run through both analysis programs and their

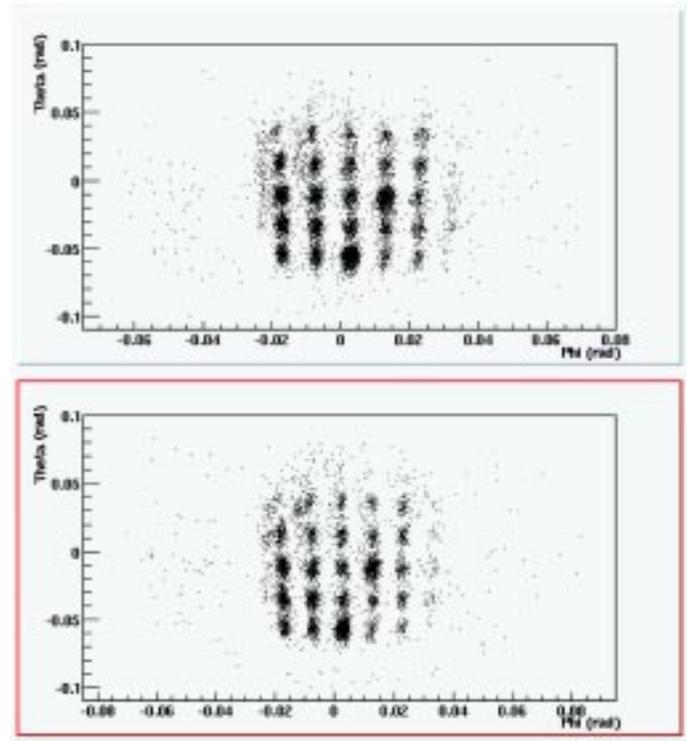


Figure 3: A plot of θ_{tg} against ϕ_{tg} for the left arm for tracks using a cut of $9.5\text{mm} < y_{tg} < 15.5\text{mm}$ to pick out one foil. The top graph is generated by the C++ analyzer, while the bottom graph is generated with ESPACE.

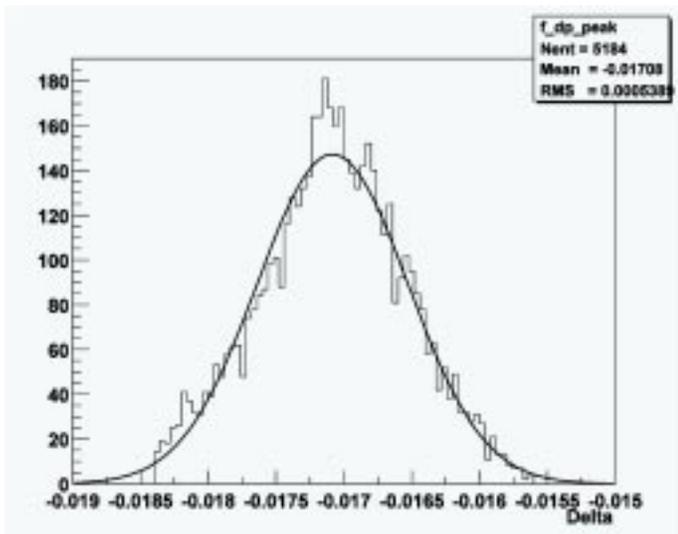


Figure 4: The elastic peak in the right arm ESPACE δ spectrum with kinematic corrections added. A gaussian curve has been fit to the peak.

output was compared. The matrix elements used for both programs are calibrated for analysis with ESPACE and have been found to be stable for analysis with that program [12].

The main analysis was performed on a PIII 700MHz Dell PC, with 768 MB of RAM, running Red Hat Linux 7.2. ESPACE version 2.9.2 was used along with a development version of the ROOT/C++ analyzer based on version 0.65.

RESULTS

We only show selected results in this paper. Many areas were studied, including simple comparisons between ESPACE and C++ analyzer values, correlations between related values, such as x_{fp} and y_{fp} or x_{fp} and θ_{fp} , and event-by-

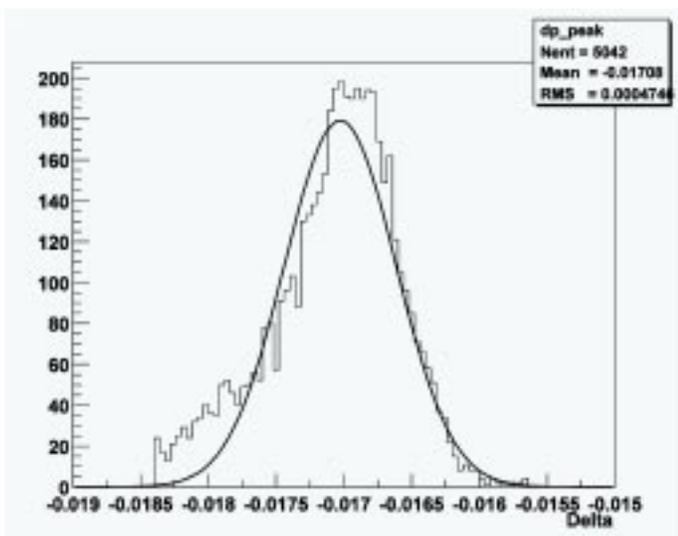


Figure 5: The elastic peak in the right arm C++ analyzer δ spectrum with kinematic corrections added. A gaussian curve has been fit to the peak.

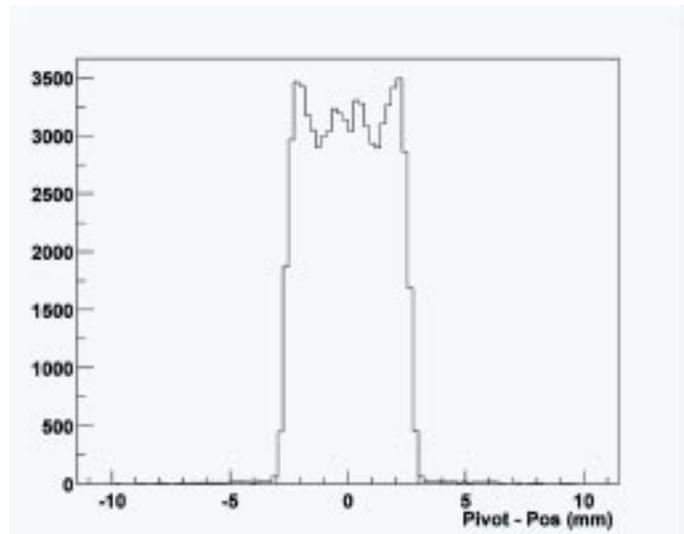


Figure 6: The difference in the cluster pivot wire and the crossing-over point for a track in the U1 VDC plane in the left arm as determined by ESPACE.

event cross-correlations and differences between ESPACE and the C++ analyzer.

Tables 1 and 2 give the results of an analysis of the difference in target values as calculated by ESPACE and the C++ analyzer for both HRS arms. Table 3 gives the results of a similar analysis for values at the focal plane. The differences were calculated on an event-by-event basis. The resulting distributions are essentially gaussian, except for x_{fp} and θ_{fp} on both arms, and θ_{tg} and δ on the left arm, which all have sizeable tails. Figures 1 and 2 give the δ spectrum for the right and left arm respectively. In all such graphs, the ESPACE results are given by the light-colored lines.

The tall peak in the δ spectrum is known as the elastic peak, as it results from elastic scattering of electrons from

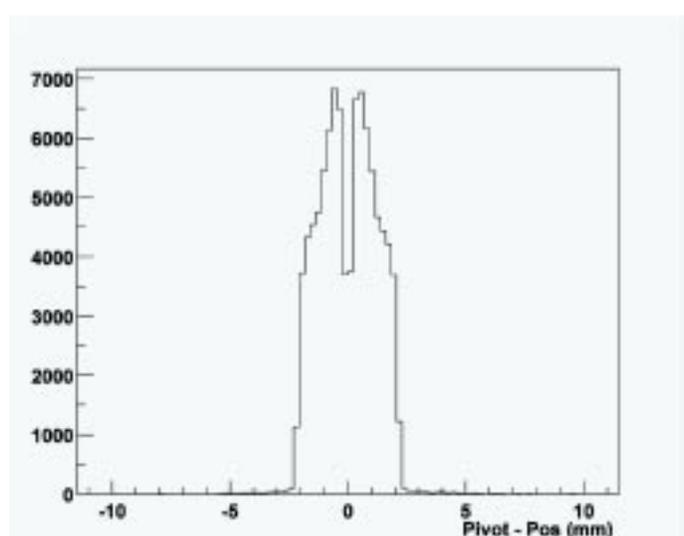


Figure 7: The difference in the cluster pivot wire and the crossing-over point for a track in the U1 VDC plane in the left as determined by the C++ analyzer.

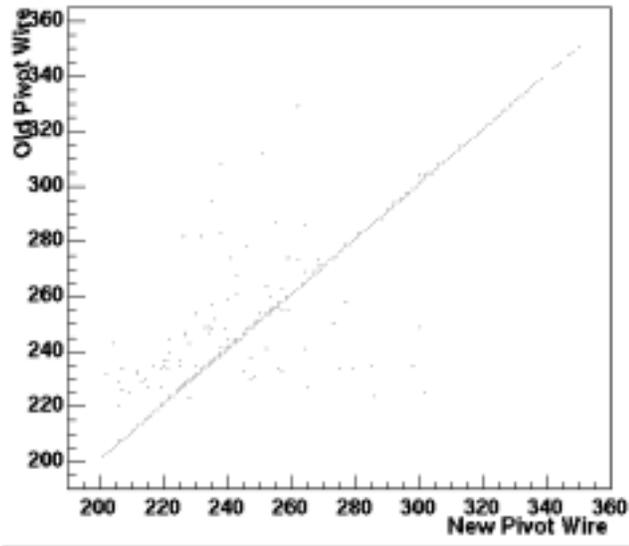


Figure 8: A plot of the cluster pivot wire for the U1 VDC plane in the left arm as determined by ESPACE against the pivot wire determined by the new C++ analyzer.

the ^{12}C nucleus. It is theoretically infinitesimally small, but its finite width is due to spectrometer resolution and radiative effects. The Δ resolution was estimated by fitting a gaussian curve to this peak, after corrections for kinematical broadening were applied. No radiative correction was made, so the estimated resolution is worse than the actual value. The peak was taken to range from a Δ value of 0.0156 to 0.00184. Figures 3 and 4 show the result of these curve fits for the right arm.

Figure 5 gives a comparison of a plot of Δ_{ig} against Δ_{ig} for tracks using a cut of $9.5\text{mm} < y_{ig} < 15.5\text{mm}$ to pick out one foil for both ESPACE and the C++ analyzer. The sieve slit pattern is clearly visible in this graph, indicating that the target reconstruction is essentially working in the C++ analyzer and agrees substantially with the results from ESPACE.

Figures 6 and 7 give a plot of the difference between the pivot wire position and calculated cross-over position for clusters in the U1 VDC plane of the left arm. Figure 8 gives a cross-correlation plot of the pivot wire number for clusters in

	Mean Difference	RMS	Optics Comm.	Design Target
Left arm y_{ig}	-0.3 mm	3.1 mm	4.0 mm	1.5 mm
Right arm y_{ig}	-0.4 mm	2.5 mm		
Left arm θ_{ig}	-1.7 mrad	7.9 mrad	6.0 mrad	2.0 mrad
Right arm θ_{ig}	-0.9 mrad	5.4 mrad		
Left arm ϕ_{ig}	0.2 mrad	1.6 mrad	2.0 mrad	0.6 mrad
Right arm ϕ_{ig}	-0.5 mrad	1.3 mrad		

Table 1. The differences in reconstructed target values between ESPACE and the C++ analyzer were calculated for each arm and a gaussian curve was fit to each difference. Given are the mean and RMS values for each distribution. For comparison to the RMS, the resolutions obtained at optics commissioning and the maximum design values for the HRS are given (both FWHM).

	Elastic Peak	RMS	Optics Comm.	Design Target
Left arm δ_{ESPACE}	-0.018589	0.000299	0.000250	0.000100
Left arm $\delta_{\text{C++}}$	-0.019009	0.000315		
Right arm δ_{ESPACE}	-0.017079	0.000539		
Right arm $\delta_{\text{C++}}$	-0.017076	0.000475		

Table 2. The location and RMS width of the elastic peak seen in ^{12}C scattering in the Δ spectrum in both arms for both ESPACE and the C++ analyzer. The locations of the peaks are different in each arm due to the different positioning of the arms. For comparison to the RMS, the resolutions obtained at optics commissioning and the maximum design values for the HRS are given (both FWHM).

the U1 left arm VDC plane between ESPACE and the C++ analyzer. The line is slightly inclined due to a trivial difference in numbering between Fortran and C++. For these plots, only clusters belonging to tracks that produced one cluster in all 4 VDC planes were chosen.

CONCLUSIONS

As can be seen from Tables 1 and 2, the deviations of the results obtained using the C++ analyzer from those obtained with ESPACE are almost all well within the error of their distributions. Ideally, the distributions of these differences between the analyzers should be delta functions, showing no difference. However, the C++ analyzer does not include some of the corrections that ESPACE does, such as corrections for time-of-flight between the VDCs and the trigger scintillator, which should lead to purely statistical fluctuations, yielding a gaussian peak. Indeed, for nearly every variable, we find this to be the case. The resolutions of the target values are on the same order as those calculated with ESPACE during detector commissioning, so they seem to be quite reasonable. This suggests that the C++ analyzer is ready to be used for basic analysis that does not require very high resolution.

However, there are still some lingering problems. Looking at Table 1, we see that the left arm Δ_{ig} has by far the largest deviation and an unusually large distribution. The main cause of this can be traced back to deviations in the related values x_{fp} and Δ_{fp} , given in Table 3. Δ_{fp} does not have a very large deviation, however, Δ_{ig} has a complicated dependence

	Mean Difference	RMS		Mean Difference	RMS
Left arm x_t	-0.7 mm	0.7 mm	Left arm x_{fp}	-0.7 mm	0.7 mm
Right arm x_t	0.5 mm	0.8 mm	Right arm x_{fp}	0.5 mm	0.3 mm
Left arm y_t	-0.3 mm	0.7 mm	Left arm y_{fp}	-0.3 mm	0.7 mm
Right arm y_t	0.6 mm	0.9 mm	Right arm y_{fp}	0.9 mm	0.3 mm
Left arm θ_t	-1.5 mrad	1.9 mrad	Left arm θ_{fp}	0.8 mrad	3.1 mrad
Right arm θ_t	-0.3 mrad	1.2 mrad	Right arm θ_{fp}	-0.5 mrad	0.8 mrad
Left arm ϕ_t	0.02 mrad	2.1 mrad	Left arm ϕ_{fp}	0.04 mrad	2.1 mrad
Right arm ϕ_t	-0.3 mrad	1.2 mrad	Right arm ϕ_{fp}	0.2 mrad	0.9 mrad

Table 3. The differences in reconstructed focal plan values between ESPACE and the C++ analyzer were calculated for each arm and a gaussian curve was fit to each difference. Given are the mean and RMS values for each distribution. The t subscript signifies values in the standard Transport coordinate system at the focal plane, while the fp subscript signifies values at the focal plane in the rotating coordinate system.

on θ_{fp} , which through the Transport Tensor can consist of a number of high-order polynomials. Also, the deviations in x_{fp} can cause inaccuracies in the reconstruction procedure, since the polynomials used depend on powers of x_{fp} . Such dependences could easily explain the unusual distribution for θ_{fp} in the left arm and can lead to larger errors than would otherwise be expected.

Another unusual effect can be seen by comparing Figures 1 and 3. The distributions for the right arm match up well, whereas the peaks in the left arm are shifted to negative energies by an amount that appears to depend on the value of δ . As δ also has a similar complicated dependence on θ_{fp} and x_{fp} , the slight difference noted above could easily explain the distorted peaks. The cause of these shifts in θ_{fp} and x_{fp} are currently not well understood, but a possible cause is mentioned below.

Another problematic effect can be seen in Figure 3. The holes in the sieve slit form a grid of straight lines, and this should be readily apparent in the given plot. However, the lines from the C++ analyzer in the top plot show slight diversions with increasing θ_{fp} , unlike the ESPACE results on the bottom. This effect is likely related to the above-mentioned errors in focal-plane quantities, but it is very minor, and overall this figure is a prime example of the C++ analyzer's readiness for use.

An interesting difference is seen when the elastic peaks measured in the right arm with both programs are compared. Both programs give a value for the center of the peak that agrees within 3 keV (0.001%). However, The ESPACE results, shown in Figure 4, give a broad, nearly gaussian peak. The C++ analyzer, as shown in Figure 5, gives a narrower peak, but one that has a distinct tail. This could be the radiative tail of the elastic peak, or an inaccuracy in the C++ analyzer algorithm. Modeling of the expected radiative tail would help clarify the source of this difference.

A potential low-level problem in the VDC tracking algorithm of the C++ analyzer is suggested by looking at the differences in the pivot wire position and the calculated cross-over position for both ESPACE, shown in Figure 6, and the C++ analyzer, shown in Figure 7. The cross-over point is calculated by fitting the drift distances in the cluster to a line, which is done after cluster identification. Since the density of tracks in a wire cell is approximately constant, this spectrum of distance variation should be close to flat, as it is with ESPACE. However, the C++ analyzer produces a highly structured spectrum. The reasons for this are not completely known, but could come from incorrectly calculated drift times, non-linearities in the drift spectrum, or errors in the fitting code. The cross-over positions are used to calculate locations in the focal plane, so errors in their calculation could be the cause of the deviations in x_{fp} . Also, since the angles of the tracks are highly dependent on the cross-over positions calculated in this way, these errors might be the cause of the shift in θ_{fp} that was discussed above.

Figure 8 points to two related problems in the cluster identifying algorithm. Because this graph looks simply at the pivot wires of clusters, it does not depend on such compli-

cated line-fitting functions as are used later in analysis, and therefore the line should be very nearly straight. However, while most clusters belong to only one track, if two tracks get close enough to each other, their clusters overlap. This explains some of the variation, as ESPACE handles this case, but the C++ analyzer as yet does not. However, since the average size of a cluster is only five wires, this explanation only works for those wires within a certain range of the pivot. There are some differences in this plot that are much larger, which are most likely due to some misidentification of the clusters.

Besides the above-mentioned problems that need to be investigated, there are a few other improvements that could be made to increase the accuracy of the C++ analyzer. The event trigger that stops the TDCs is derived from scintillator signals. This introduces some slight inaccuracies, which ESPACE corrects for, but the C++ analyzer does not. A correction can also be added for the timewalk of the scintillator timing. Also, effective cluster reconstruction algorithms, for tracks that do not create clusters in each plane, can improve analysis efficiency.

ACKNOWLEDGEMENTS

The research described in this paper was carried out at the Thomas Jefferson National Accelerator Facility in Newport News, VA, over the summer of 2002 under the auspices of the Energy Research Undergraduate Laboratory Fellowship Program.

Many thanks go to my mentor, Ole Hansen, for patiently explaining things to me and putting up with my mistakes, as well as the rest of the Hall A Collaboration. Thanks also go to Jan Tyler and the rest of the gang in the Science Education Department.

Thanks go to the United States Department of Energy, Office of Science for creating, organizing and funding this program. Also, thanks to the National Science Foundation for their assistance in funding this program.

REFERENCES

- [1] <http://hallaweb.jlab.org>
- [2] <http://www.jlab.org>
- [3] <http://hallaweb.jlab.org/root/>
- [4] <http://root.cern.ch/>
- [5] <http://hallaweb.jlab.org/espace/>
- [6] <http://hallaweb.jlab.org/equipment/HRS.htm>

- [7] Sauli, F. (1977). *Principles of Operation of Multiwire Proportional and Drift Chambers*. Internal Report 77-09, CERN (unpublished).
- [8] Fissum, K.G., W. Bertozzi, J.P. Chen, D. Dale, H.C. Fenker, J. Gao, A. Gavlya, S. Gilad, C.R. Leathers, N. Liyanage, R.O. Michaels, E.A.J.M. Offermann, J. Segal, J.A. Templon, R. Wechsler, B. Wojtsekhowski, J. Zhao. (2001). *Vertical Drift Chambers for the Hall A High-Resolution Spectrometers at Jefferson Lab*. Nuclear Instruments and Methods in Physics Research A 474, 108-131.
- [9] Leathers, C. (1996) *Efficiency Measurements of the CEBAF Hall A VDCs*. Undergraduate Thesis, Massachusetts Institute of Technology (unpublished).
- [10] Wechsler, R. H. (1996) *Drift-time Properties of the CEBAF Hall A Vertical Drift Chambers*. Undergraduate Thesis, Massachusetts Institute of Technology (unpublished).
- [11] Brown, K.L., F. Rothacker, D.C. Carey, Ch. Iselin. (1983). *Transport: A Computer Program for Designing Charged Particle Beam Transport Systems*. SLAC-91, Rev. 3. UC-28.
- [12] Liyanage, N. (n.d.). Notes on Optimization. Retrieved July 9, 2002 from <http://hallaweb.jlab.org/espace/optimization.ps.gz>
- [13] The Jefferson Lab Hall A Collaboration. (June 10, 2002). *Basic Instrumentation for Hall A at Jefferson Lab*. (preprint) Retrieved July 29, 2002 from <http://hallaweb.jlab.org/equipment/NIM.ps>
- [14] http://hallaweb.jlab.org/adaq/log/html/0009_archive/000930104120.html