Nuclear Instruments and Methods in Physics Research A 803 (2015) 24–28

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

Event-based processing of neutron scattering data

Peter F. Peterson*, Stuart I. Campbell, Michael A. Reuter, Russell J. Taylor, Janik Zikovsky

Neutron Data Analysis and Visualization, Oak Ridge National Laboratory, Oak Ridge, TN, USA

ARTICLE INFO

Article history:
Received 26 May 2015
Received in revised form
1 September 2015
Accepted 8 September 2015
Available online 16 September 2015

Keywords:
Neutron
Software
Data Analysis

ABSTRACT

Many of the world's time-of-flight spallation neutrons sources are migrating to recording individual neutron events. This provides for new opportunities in data processing, the least of which is to filter the events based on correlating them with logs of sample environment and other ancillary equipment. This paper will describe techniques for processing neutron scattering data acquired in event mode which preserve event information all the way to a final spectrum, including any necessary corrections or normalizations. This results in smaller final uncertainties compared to traditional methods, while significantly reducing processing time and memory requirements in typical experiments. Results with traditional histogramming techniques will be shown for comparison.

1. Introduction

The Spallation Neutron Source (SNS) at Oak Ridge National Laboratory [1,2] measured first neutrons on April 28, 2006. Besides the promised neutron flux increase over existing sources and novel instrument design, the biggest feature advertised was event mode data acquisition. Event mode provides three pieces of information for each detected neutron: detector pixel identifier, total time-of-flight from source to detector, and wall clock time of the proton pulse the neutron is associated with. The full list of events can be stored for later processing. The time-of-flight is the time from the neutron to be emitted from the moderator until it is measured by the detector. The detector pixel identifier, or pixel id, is a unique index to the detector pixel in the instrument. This differs from the traditional data acquisition mode of measuring histograms where each detected neutron is given a pixel identifier and time-of-flight then collected into histograms, without information about the pulse the neutron is associated with.

Limited memory and bandwidth of computers when time-of-flight neutron scattering was being developed is the reason for histogram mode [3–5]. Generally the experimenter made a choice of either having more detector pixels or time-of-flight bins, balancing them as anticipated for the experiment. With event mode, the neutrons are recorded at full resolution of the data acquisition electronics. At SNS this is 100 ns for time-of-flight and each individual pixel. While this new acquisition mode was available since the first neutrons were measured, techniques for processing the events had not been developed. For the first few years of operation data was reduced by first histogramming the events, then processing them using traditional techniques. Note that this is still an available option, but much of the data reduction at SNS has been converted to use the techniques described below.

2. Histogramming

Before the SNS, most experiments on time-of-flight neutron sources, data was collected in what is known as Histogram Mode. What do we mean by this? Histograms divide the total range of time-of-flights ($t_{min} \rightarrow t_{max}$) into a number of channels or bins. When a neutron is detected, its time-of-flight (*TOF*) is determined and the scaler counter for the corresponding histogram channel is incremented.

The time-of-flight (*TOF*) of a neutron recorded into the time channel N can be defined as

$$t_{min} \le t_{min} + t_N \le TOF < t_{min} + t_{N+1} \le t_{max} \quad (1)$$

where t_N is the starting time boundary for the Nth histogram channel. Fig. 1 illustrates this. For the Nth histogram channel, the most likely time-of-flight for the neutron is given by

$$TOF = t_{min} + \tfrac{1}{2}(t_{N+1} - t_N) \quad (2)$$

From this we can see that the time-of-flight accuracy of a given feature in the collected spectrum will depend on the width of the time channels that we use in the data acquisition. Furthermore, an increase in the channel width would also move the position of any feature by half the increase in channel width. For a more detailed discussion of choosing histogram bin widths can be found in [3].

* Corresponding author.

http://dx.doi.org/10.1016/j.nima.2015.09.016
0168-9002/

3. Dynamic histogramming

One great advantage of event-based processing is that histogramming can be performed dynamically, using the raw event data. In histogrammed data, the distribution of neutrons' times-of-flight within a single bin is unknown, that information has been discarded, requiring that assumptions be made when rebinning. Generally the assumption is that the events are evenly distributed within a histogram bin. As a result, rebinning always smooths the data. By retaining the distribution of events, one can continue using the notation of rebinning, when all that is really being done is histogramming. Fig. 2 shows the ability to rebin event-based data (left) as compared to histogram data (right). As can be seen, the diffraction pattern is revealed as the bins are made smaller. With traditional histogram data, there is an effective smoothing of the data as seen in the figure. As mentioned in Section 1, at any point data processing can continue with the histogram version of the data. In practice this can be done making a copy of the dynamic histogram.

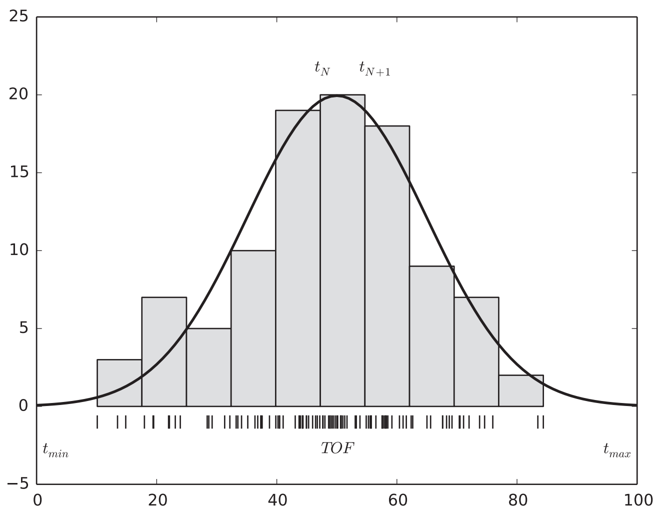

Fig. 1. Schematic diagram of histogramming. Dashes below are individual events. Boxes are the histogram representation. Curved line is the normal distribution for reference.

In order to efficiently histogram the data, it is best to sort the events for the histogram by time-of-flight. Then the algorithm iterates through the sorted event list, adding events to a bin, then moving on to the next bin when the next event's time-of-flight goes to a new bin. While the initial sort of n events has a computational complexity of order $O(n) = n \log n$, any subsequent rebinning is performed in approximately linear time $O(n) = n$. Histogram errors are calculated to be simply equal to the square root of the number of counts in each bin as according to Poisson statistics [7,8].

4. Propagation of uncertainties

4.1. Existing histogram mode data processing

To provide a framework for understanding event data processing, we will first describe processing histograms. A histogram can be made from event data by ignoring the wall clock time and summing together all events within histogram bin boundaries together. The term "wall clock time" refers to the absolute time of the proton pulse on target. If we sum the wall clock time and the time-of-flight, we will get the absolute time the neutron was detected. The uncertainties are just the square root of the measured intensities, $\sigma^2 = I$, as traditionally assumed. The four types of arithmetic that one can generally do are add, subtract, multiply and divide. When doing these operations for individual histogram bins uncertainties are propagated according to traditional

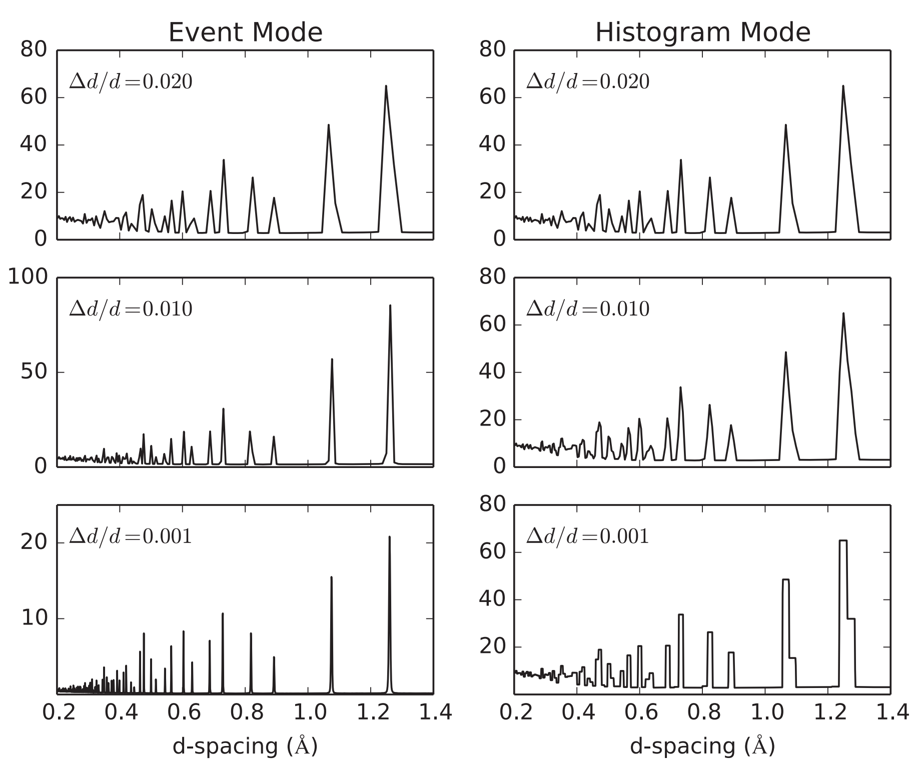

Fig. 2. A comparison of the relative differences observed in diffractograms arising from the rebinning of histogram or event data. Each lower figure is a rebinned version of the one above it. All four are done with constant $\Delta T/T$ binning with the left column being the event data and the right being historgram, for comparison. The data presented is from a Diamond sample measured on the instrument POWGEN [6].