JAEA-Technology 2010-047
J-PARC 10-03

JAEA-Technology

Technical Report on the Korea-Japan Software Collaboration

Yasuhiro INAMURA, Ji-Yong SO, Kenji NAKAJIMA, Jiro SUZUKI
Takeshi NAKATANI, Ryoichi KAJIMOTO, Toshiya OTOMO, Myung-Kook MOON
Chang-Hee LEE, Yoshiji YASU, Kazuo NAKAYOSHI, Hiroshi SENDAI
Uk-Won NAM, Je-Geun PARK and Masatoshi ARAI

Materials and Life Science Division
J-PARC Center

February 2011

Japan Atomic Energy Agency

日本原子力研究開発機構

本レポートは独立行政法人日本原子力研究開発機構が不定期に発行する成果報告書です。
本レポートの入手並びに著作権利用に関するお問い合わせは、下記あてにお問い合わせ下さい。
なお、本レポートの全文は日本原子力研究開発機構ホームページ（http://www.jaea.go.jp）
より発信されています。

独立行政法人日本原子力研究開発機構　研究技術情報部　研究技術情報課
〒319-1195　茨城県那珂郡東海村白方白根 2 番地 4
電話 029-282-6387, Fax 029-282-5920, E-mail:ird-support@jaea.go.jp

This report is issued irregularly by Japan Atomic Energy Agency
Inquiries about availability and/or copyright of this report should be addressed to
Intellectual Resources Section, Intellectual Resources Department,
Japan Atomic Energy Agency
2-4 Shirakata Shirane, Tokai-mura, Naka-gun, Ibaraki-ken 319-1195 Japan
Tel +81-29-282-6387, Fax +81-29-282-5920, E-mail:ird-support@jaea.go.jp

Technical Report on the Korea-Japan Software Collaboration

Yasuhiro INAMURA※, Ji-Yong SO[*1], Kenji NAKAJIMA, Jiro SUZUKI[*2],
Takeshi NAKATANI, Ryoichi KAJIMOTO, Toshiya OTOMO[*3], Myung-Kook MOON[*1],
Chang-Hee LEE[*1], Yoshiji YASU[*4], Kazuo NAKAYOSHI[*4], Hiroshi SENDAI[*4], Uk-Won NAM[*5],
Je-Geun PARK[*6] and Masatoshi ARAI

Materials and Life Science Division, J-PARC Center,
Japan Atomic Energy Agency
Tokai-mura, Naka-gun, Ibaraki-ken

(Received December 9, 2010)

Both Materials & Life Science Experimental Facility (MLF) of J-PARC and HANARO of KAERI started new neutron facility projects in 2002 and 2003, respectively. As part of their projects, both institutes began developments of new Time-of-Flight (ToF) spectrometer including DC-TOF of HANARO, 4SEASONS and AMATERAS of MLF. With this new instrument development, we saw an opportunity for collaboration between Korea and Japan regarding ToF software.

This Korea-Japan collaboration officially started in 2007 with an initially 6 items as its final goal. The 6 items include i) basic data reduction software, ii) informative visualization software, iii) data visualization software, iv) decision making & optimization software, v) single crystal alignment software, and vi) advanced analysis software. Using Manyo library developed at J-PARC as our software framework, we developed our software based on a combination of Python and C++ wrapped under SWIG.

In August 2008 we successfully released a beta-version of basic data reduction software which has been tested at the 2 beamlines of MLF; 4SEASONS and AMATERAS, and regularly updated. Other 2 beta-versions of informative visualization software and data visualization software have also been released and are successfully used during experiments at 4SEASONS and AMATERAS. Although we have had several discussions on the 3 remaining topics of the original goal of this collaboration, progress has been rather limited on these items. Therefore, we decided to consider them as the subject of the next Korea-Japan collaboration.

This report summarizes the 2-years (2007-2009) activities of Korea-Japan collaboration of chopper software development. Here we describe the background of the collaboration and the main part of our work. We also discuss briefly a future plan of our collaboration starting in 2010. Some of detailed descriptions on the activities of the collaboration as well as related information are given in appendix.

Keywords: Neutron Scattering, Chopper Spectrometers, J-PARC, HANARO, Data Acquisition, Data Reduction, Event Mode Data, Manyo Library, Software, Korea-Japan Collaboration

※ Research Staff on Loan
*1 Neutron Science Division, Korea Atomic Energy Research Institute, Daejeon, 305-353, Korea
*2 Computing Research Center, KEK, Tsukuba 305-0801, Japan
*3 Institute of Materials Structure Science, KEK, Tsukuba 305-0801, Japan
*4 Institute of Particle and Nuclear Studies, KEK, Tsukuba 305-0801, Japan
*5 Korea Astronomy and Space Science Institute, Daejeon 305-348, Korea
*6 Department of Physics & Astronomy, Seoul National University, Seoul 151-747, Korea

日韓中性子チョッパー型分光器用ソフトウェア共同開発についての技術報告書

日本原子力研究開発機構 J-PARCセンター
物質・生命科学ディビジョン

稲村 泰弘※、Ji-Yong SO[*1]、中島 健次、鈴木 次郎[*2]、中谷 健、梶本 亮一、大友 季哉[*3]、
Myung-Kook MOON[*1]、Chang-Hee LEE[*1]、安 芳次[*4]、仲吉 一男[*4]、千代 浩司[*4]、
Uk-Won NAM*[5]、Je-Geun PARK*[6]、新井 正敏

（2010年12月9日受理）

国内では、2002年に物質・生命科学実験施設（MLF）を含むJ-PARCの計画が開始され、韓国では、2003年に韓国原子力研究所のHANAROでの新たな中性子科学研究の計画が始動した。2つの計画の中には、HANAROにおいてはDC-TOF、そしてMLFでは、4SEASONS、AMATERAS等といった新しいチョッパー型分光器の建設が計画されていた。このような中で我々は日本と韓国の間でチョッパー型分光器のソフトウェアについての共同開発を行うこととなった。

この日韓中性子チョッパー型分光器用ソフトウェア共同開発は、2007年より開始され、当初6つの開発項目の達成が目標とされた。それらは、i) 基本となるリダクションソフトウェア、ii) 実験情報表示ソフトウェア、iii) データ可視化ソフトウェア、iv) 実験条件最適化、決定支援ソフトウェア、v) 単結晶軸立てソフトウェア、vi) その他先進的解析ソフトウェアである。J-PARCで開発されたManyo Libraryをフレームワークとし、PythonとSWINGでラップされたC++でソフトウェアを開発した。

2008年8月には、基本となるリダクションソフトウェアのβ版が完成し、MLFの四季分光器、アマテラスにて試験が開始された。実験情報表示ソフトウェアとデータ可視化ソフトウェアについても、これら2台のチョッパー型分光器での運用が開始された。残る3項目については、さらに引き続き日韓共同開発の項目として開発を続けている。

この報告書は、日韓中性子チョッパー型分光器用ソフトウェア共同開発について2007年から2009年の2年間の活動内容をとりまとめたものである。共同開発の背景、開発作業の内容等について述べている。また、2010年以降に続く次の共同開発についての議論にも触れる。付属する技術情報や活動記録等は、付録として収録している。

J-PARCセンター： 〒319-1195 茨城県那珂郡東海村白方白根2-4
※ 出向職員
*1 韓国原子力研究院 中性子科学研究部
*2 高エネルギー加速器研究機構 計算科学センター
*3 高エネルギー加速器研究機構 物質構造科学研究所
*4 高エネルギー加速器研究機構 素粒子原子核研究所
*5 韓国天文研究院
*6 国立ソウル大学

Contents

目次

1. Introduction

1.1 Background of collaboration

In 2002, High Energy Accelerator Research Organization (KEK) and Japan Atomic Energy Agency (JAEA) launched a joint project of J-PARC, Japan Proton Accelerator Research Complex. J-PARC is a research complex using secondary particles produced by 1 MW proton-beam (Figs. 1). One of the 4 main research facilities of J-PARC is Materials & Life Science Experimental Facility (MLF), at which broad range of investigations in the area of materials and life sciences will be carried out by using high-intensity pulsed neutron and muon beam. When the J-PARC project started, a number of new neutron instruments were proposed including 4SEASONS and AMATERAS (Fig. 2). 4SEASONS is a high-intensity chopper spectrometer constructed by a joint group from the members of JAEA, KEK, and Tohoku University [1]. AMATERAS is a cold-neutron disk chopper spectrometer constructed by JAEA [2]. Both projects of 4SEASONS and AMATERAS started in 2005 and 2006, respectively, and were expected to be completed as Day-1 instruments of MLF by the end of Japanese fiscal year 2008.

On the other hand, a cold neutron source project for a 30 MW research reactor, HANARO at Korea Atomic Energy Research Institute (KAERI) started in 2003 (Fig. 3). The project includes installing a cold-neutron source and constructing new experimental hall accommodating new cold instruments. DC-TOF is one of Day-1 instruments for the cold neutron project at HANARO and it is run by a joint group between SungKyunKwan University and KAERI with J. -G. Park as project leader (Fig. 4). The project officially began in 2003 and was due to be completed by April, 2010 [3]. However, some problems such as the global financial turmoil in 2008 and, to make things worse, a recent hike in ^{3}He price over 2008-2009 forced the DC-TOF team to revise the construction plan. The new plan is now: a first beam test by December 2010 and a full operation by April 2012.

In above situation, our Korea-Japan (KJ) collaboration began with discussions between 2 members of the collaboration team: M. Arai and J.-G. Park, in early 2007 at J-PARC. At that time, both teams (MLF, J-PARC team led by M. Arai and the DC-TOF team of HANARO, KAERI, led by J-G. Park) were very actively engaged in instrument developments as described below. Therefore, we saw the opportunity of the joint collaboration. Furthermore, we thought that some kind of closer collaboration between the 2 countries will be of mutual benefit once J-PARC and KAERI start the new neutron spectrometers.

Because both teams were by then already working on the hardware side of the project and each spectrometer has its own unique technical features, we thought that software will be a good topic for this collaboration. Apart from this, there are other advantages for us to choose the software development for the joint collaboration. The main reason was that once we have a clear plan and job assignments software can be coded by individual persons at one's own office with minimum off-line interaction. Although we wanted to have as frequent off-line meetings as possible for this collaboration from the beginning, our past experience told us that international collaborations based on off-line meeting often led to a very slow progress. As it turned out, we managed to have 16 monthly online meetings and 7 offline meetings. The offline meetings were done through either a joint collaboration meeting or visits by individual members to their own counterparts.

Based on this discussion, M. Arai and J.-G. Park organized the first meeting at J-PARC on 24th July 2007, which was attended by 12 people from both countries: J. -G. Park, M. -K. Moon, K. -C. Jin, J. -Y. So, M. Arai, K. Nakajima, T. Otomo, J. Suzuki, Y. Inamura, T. Nakatani, S. Itoh, T. Yokoo. After the meeting, we invited Kenji Nakajima to coordinate this collaboration and assigned J. -Y. So and Y. Inamura to the working group responsible for writing software. We chose the data reduction software as the key part of this collaboration particularly for the following reasons:

- It is the integral part of the software development
- It offers clear benefits to both sides' chopper projects
- We anticipated that a full development would take 2 years, which fitted well with the schedule of the hardware construction on both sides
- It was remained to be developed in the J-PARC project.

The first kick-off meeting was organized at J-PARC during 10th-14th September 2007 with the following attendees: K. Nakajima, J. -Y. So, Y. Inamura, M. Arai, J. -G. Park, T. Otomo, T. Nakatani, J. Suzuki, K. Tomiyasu. At the meeting, we agreed to develop the data reduction software using an event-mode format as the main object of this collaboration. A full list of items considered for this collaboration is as follows:

- Basic data reduction software
- Informative visualization (for decision of conditions)
- Single crystal alignment software
- Data visualization software
- Decision making and optimization.

One brief, noteworthy point about the data reduction software is that from the beginning we decided to write our own program based on an event mode format, for the first time for neutron scattering experiments. As discussed in detail subsequently in the report, this decision was shared by many others as a new direction of future software developments for neutron scattering. This new approach in the data reduction software is appropriate for the new chopper instruments of J-PARC. It would also offer much needed flexibility in data handling and new opportunities of doing neutron sciences because of the inherent time tagging feature.

Through the 2-years collaboration, we managed to release the beta version of the data reduction software by August 2008 with the first beam test subsequently made at the 4SEASONS beamline of J-PARC in September 2008. Subsequent upgrades have since been made with continual beam tests using 4SEASONS & AMATERAS.

As of writing this report, the data reduction software is not complete yet for the final release for users. But we will spend continuous effort to correct the codes for problems identified during the on-line use and to make it better. And this report serves as a status report and will be used as a reference material for a future full user's manual.

When we were preparing this collaboration, we were aware of similar efforts made at the ISIS, UK and the SNS, USA. Therefore, we thought that it would be useful to have exchanges of information as well as ideas. For this, we have invited Dr. T. G. Perring of ISIS and Dr. G. E. Granroth of the SNS to participate in some of our off-line meetings as documented in the appendix. In summary, there were altogether 3 of such kind meetings: twice at J-PARC and one at Fukuroda.

1.2 Organizational aspects of collaboration

Our collaboration was officially conducted within the existing MOU between JAEA, one of the 2 host organizations of J-PARC, and KAERI: "*ARRANGEMENT FOR COOPERATION IN THE FIELD OF PEACEFUL USES OF NUCLEAR ENERGY BETWEEN THE JAPAN ATOMIC ENERGY AGENCY AND THE KOREA ATOMIC ENERGY RESEARCH INSTITUTE*." The official name of the collaboration is the

Korea-Japan software collaboration.

The structure of the organization is shown in Fig. 5. M. Arai and J. -G. Park as leader initiated and guided the collaboration, while K. Nakajima as coordinator handled day-to-day management of the collaborations. And Y. Inamura and J. -Y. So formed a working group and were responsible for actual coding works. For the successful operation of the collaboration, K. Nakajima took the following responsibilities:

- Management of the working group
- Maintain the progress of the collaboration
- Monthly online meeting
- Offline meeting
- Running the International Advisory Team

The 2 members of the international advisory team: T. G. Perring of ISIS, UK and G. E. Granroth of SNS, USA, participated in the 3 offline meetings and presented the status reports of software developments of their own institutes. They also provided valuable inputs as well as suggestions regarding numerous technical problems occurred throughout the collaborations. We also had an expert group from both Korea and Japan. For example, our program heavily used Manyo-lib (Manyo library) developed by J. Suzuki and his co-workers of KEK while T. Otomo, T. Nakatani, R. Kajimoto, Y. Yasu, K. Nakayoshi and H. Sendai provided advices on the technical as well as architectural issues and M. -K. Moon and U. -W. Nam worked as experts for the Korean team.

1.3 Final target of collaboration

The aim of this collaboration is to develop software for time-of-flight (ToF) chopper spectrometers under construction at both KAERI and J-PARC. It was initially agreed that any level of software that was required for chopper spectrometers should be considered as parts of this Korea-Japan collaboration. After several times of discussions, it was decided that the following sets of software were included to the wider scope of the collaboration. Here, the item of "advanced analysis software" was added to the list at the first kick-off meeting.

i) Basic data reduction software.
Reduction software processing raw data, equivalent of Homer [4] which is the standard reduction software used at ISIS.

ii) Informative visualization software.
Software to visualize scan area, resolutions at given experimental condition to help users' decision.

iii) Data visualization software.
The software to visualize data in 3D, 2D and 1D plot with various functions of data manipulation like Mslice [4] which is the versatile data visualization software used at ISIS.

iv) Decision making & optimization software.
Software to justify or optimize the experimental conditions.

v) Single crystal alignment software.
Software used to align single crystals.

vi) Advanced analysis software.
Sophisticated analysis software including multiple scattering corrections, multiple phonon

correction, functions cooperating with simulation software etc.

Our final goal is to develop all the above software for our chopper spectrometers through this Korea-Japan collaboration. However, because of relatively higher urgency, we gave our priority to the data reduction software. The data reduction software is essential and indispensable parts of data acquisition for any chopper spectrometers. Furthermore, we decided to use event mode data acquisition, which was a quite new initiative with many conceptual challenges. We emphasize that there was no established data reduction software for chopper spectrometers using the event mode data in the world when we began this collaboration. We also recognize importance of facilitating better science through more effective software including better visualization program and decision making & optimization program. Software that would help users with single crystal experiments is also considered to enhance the efficiency of carrying out experiments. After finishing the above software, we will proceed to develop more advanced data analysis software.

The rough timeline was set as following. Our spectrometers will be ready in 2008-2009 (4SEASONS and AMATERAS) and in 2012 (DC-TOF). The data reduction software should be ready (at least in the form of a beta version) in 2008 before 4SEASONS starts its commissioning. The other parts of development should be finished before the chopper spectrometers of J-PARC start their full operation in 2010. It is our ultimate goal to ensure best performance of the advanced chopper spectrometers at J-PARC and KAERI with the full development of advanced software.

2. Background on programming

2.1 Basic information about ToF inelastic neutron scattering

Before starting to describe about detail coding of software, we will mention about basic information about ToF inelastic neutron scattering.

A schematic of a typical time of flight spectrometer is shown in Fig. 6. This type of a spectrometer produces monochromatic neutron pulses using an array of choppers, and measures energy ($\hbar\omega$) and momentum transfer ($\vec{Q}$) with high accuracy. Although both ToF and TAS (Triple-Axis Spectrometer) have more or less the same ($\vec{Q}, \hbar\omega$) coverage and are used for dynamic study of condensed matter sciences, ToF has an advantage of measuring a large ($\vec{Q}, \hbar\omega$) area simultaneously since a wide angle is covered by numerous detectors.

The basic principle of the ToF technique is shown in Fig. 7. In the case of neutron beams from reactors such as HANARO of KAERI, after the first chopper-set, neutrons become periodic pulses with a period of τ and a pulse width of Δt. Therefore, the first chopper-set is called a pulsing chopper. These pulsed neutrons have all wavelengths, and after passing through the second and third chopper-sets, they become monochromatic neutron pulses. In the case where τ is too small a value for good energy resolutions, a slowly rotating chopper with a period of $n \times \tau$ must be simultaneously used. Therefore, the second and third chopper-sets are often called a frame removal chopper and an order removal chopper, respectively. The forth chopper-set determines the pulse width in both time and wavelength (energy) domains [5]. On the other hand, in the case of spallation neutron source one does not need the first choppers since neutrons at spallation source are produced as bunches of pulses from the beginning. Otherwise, the remaining process is exactly same for ToF spectrometers at spallation neutron sources as ToF instruments at reactors.

Traditionally, the scattered neutrons are stored as histograms. However, we intentionally developed our data reduction software for a so-called event data format for reasons given later in the report. So in our case, all the scattered neutrons are stored with their own unique time tag, i.e. $I(\theta, t)$, where θ is the scattering angle of pixels of detectors and t is the time of each event of scatterings. These event data are then processed through our data reduction software, the product of this KJ collaboration, and converted into histograms. One can further treat the raw data in order to make absolute normalization process by using vanadium data measured with the same incident energy as a sample in question. In any case, $S(\vec{Q}, \hbar\omega)$ will be produced by using the histograms for further analysis. For more details about the mathematical formalism, please see the appendix.

2.2 Programming environments

In this development work, we chose to have as programming languages a combination of Python and C++ wrapped under SWIG (Fig. 8). One of the reasons behind our decision was that we use Manyo-lib, which was already under development as the framework of data structure and numerical routines before our collaboration. Details of Manyo-lib are described in later sections. The class libraries of Manyo-lib are written in C++ and wrapped by the Python interface that was created by SWIG, which is the major reason for us to choose the 2 languages as our programming languages. Also, these languages are quite ideal for our purpose for the following reasons. For example, Python, C++ and SWIG are robust, portable and well documented. They have been widely and successfully used in many scientific applications for more than 10 years. They also support many platforms – Microsoft Windows, Linux, OSX, etc. Sufficient information and documentations about the languages are available online.

Here, we shortly introduce Python, C++ and SWIG as well as their relations in our development.

Python [6]

Python is a programming language invented by Guido van Rossum. The official Python web page introduces Python programming language as below (http://www.Python.org/about/) :

Python is a remarkably powerful dynamic programming language that is used in a wide variety of application domains. Python is often compared to Tcl, Perl, Ruby, Scheme or Java. Some of its key distinguishing features include:

- Very clear, readable syntax
- Strong introspection capabilities
- Intuitive object orientation
- Natural expression of procedural code
- Full modularity, supporting hierarchical packages
- Exception-based error handling
- Very high level dynamic data types
- Extensive standard libraries and third party modules for virtually every task
- Extensions and modules easily written in C, C++ (or Java for Jython, or .NET languages for IronPython)
- Embeddable within applications as a scripting interface

In spite of several advantages of Python, native Python has its own drawback: performance for some application especially for numerical calculation is rather poor. For such scientific applications, it is necessary to glue Python with Fortran, C or C++. Python is called a "glue language" because one can easily glue many other programming languages or tools into Python environments. For this purpose, the most widely used method is SWIG as described later.

Another important feature of Python is that it is an interpreter programming language. Python provides its command line shell, and there exist other shells that have more useful functions, for example, IPython (http://www.iPython.org). Using these environments, we can easily check the results of our program line by line.

C++

C++ is one of the most popular programming languages. C++ is a general purpose, multi paradigm low level programming language invented by Bjarne Stroustrup. It succeeds C programming language and implements many new programming paradigms such as object-oriented, generic programming using template. C++ is also widely used in the area of numerical calculations because of its high computational power. One of the important features of C++ is that it is easy to implement C++ module into Python environments. Many methods have been so far developed and used with SWIG being one such example. Combination of Python and C/C++ is widely used in neutron scattering communities presently. For example, they are used for Manyo-lib, DANSE project of SNS, and MANTID project of ISIS [7, 8].

SWIG [9]

SWIG (Simplified Wrapper and Interface Generator) is a project to make a tool for building scripting language interfaces to C and C++. It supports many scripting languages including Python. It enables us to integrate C/C++'s high performance and Python's productivity and flexibility. By wrapping the class libraries of Manyo-lib using SWIG, one can access Manyo-lib via Python scripting language.

2.3 Manyo library as tool for data reduction

Manyo-lib is a framework for software development at MLF of J-PARC and provides general-purpose functions as common modules in manipulating data analysis [10]. It also provides basic environments for software development for each individual instrument. The development of Manyo-lib was managed by a group of T. Otomo and J. Suzuki of KEK. Manyo-lib was initially used at SWAN, a wide angle diffractometer at KENS of KEK.

By adapting this common framework of J-PARC for our collaboration, we could economically use the resources of our project. At the same time it made possible for us to share software between other instrument groups of J-PARC. This development is open not only to instrument scientists at J-PARC and HANARO, but also to any other persons who may be interested in sharing software through Manyo-lib. Common functions of Manyo-lib are written in C++, which can be used as coding templates for data correction, projection and merging.

All functions in Manyo-lib are wrapped into Python by SWIG so that users can easily learn how to use and execute procedures. By using Manyo-lib and Python, users will be able to write efficiently their own software. Since our software is based on this framework, we could initially write our codes without C++, at least at the beginning of the collaboration. We wrote our code in Python using the data container structure within the framework of Manyo-lib.

The structure of data containers [11]

Data container in Manyo-lib is named "ElementContainer," which can store several numerical arrays (vectors) for histogram data and Header information for meta-data about the histograms. The structure of the ElementContainer is shown in Fig. 9. ElementContainer contains meta-data. Therefore ElementContainer has 2 roles, 1) storing data and 2) storing parameter files.

Header information can include various types, i.e. real number, integer number, string, and arrays of them. It contains the information of histograms to be used during data reduction and data analysis. The information required in the data processing (the information transmitted between each component in the software) has a large variety of data from the information of detected events, information of instruments to parameters used in specific analysis. This meta-information can be handled by each data container. Data containers of Manyo-lib provide unified format for handling all the parameters used in the software. This feature is useful both to users and developers. Especially, when several people at different sites work together on one development as in our collaboration, the data container is also quite helpful in reducing efforts of managing I/Os or data formats in the program, which is often problematic in such cooperative works.

All numerical vectors of experimental data (for example, ToF, neutron counts detected by detectors) are stored in ElementContainer, from which user can define their own "histogram." Histograms as such defined from ElementContainer can be used for calculation by using functions of Manyo-lib. In our data reduction, a single ElementContainer corresponds to a single histogram data for a single pixel of PSD. A series of ElementContainers can be stored at upper-level container named as ElementContainerArray. This is suitable for treating all the data for the entire positions of a single PSD. ElementContainerArrays for each single PSD are then stored in ElementContainerMatrix for the whole data of a single measurement.

Calculation between histograms

In data reduction, operations between histograms are required. Containers in Manyo-lib have methods that can execute simple operations (+, -, *, /), although the operations are restricted between containers that have the same structure of contents.

Header information

Header, data structure in Manyo-lib, contains experimental information (run number, monitor counts, total number of events and incident neutron energy) and instruments parameters (flight path length between moderator (1st chopper) and sample, detector position and status). This Header information can be used and subsequently updated at various data reduction steps.

Functions using ElementContainer

Manyo-lib provides Python-wrapped commands to treat histogram and Header information. Users can easily learn and write Python codes for their own analysis and corrections. But the speed of executing Python codes is obviously slow, especially for a repeating process. It is clearly an issue of immense importance when dealing with a large size data. To speed up the performance, Manyo-lib provides users with an option to write their own routine in C++ using a function template to treat ElementContainers. This template makes it easy to share the products with other instruments at MLF and to use them for other developers. It is also allowed a user to add functions to Manyo-lib as new components. A Schematic structure of an analysis operation and a typical example of an analysis operation for projecting 3-dimensional histogram (3D histogram) into 2-dimensional histogram (2D histogram) are shown in Fig. 10 (taken after Ref. [11]).

Using ElementContainer and following the formats are also advantageous when several people work in parallel to develop software like this KJ collaboration. In our case, we first determined what kind of information we wanted to store in Header at the beginning of this KJ collaboration, which is kept as meta-information in data container of Manyo-lib. Therefore we need not consider how to share all the information and how to connect them between software components written by different persons throughout our collaboration. ElementContainer and its format have been proven to be very effective in gluing each product after developed separately at the home institutes of participating groups in Korea and Japan.

2.4 Event mode data

Event Data Recording

The event data recording is a data recording method used in our facilities, J-PARC and HANARO. In this method, data are stored as neutron events (Fig. 11). In the event data recording method, the histogram data can be constructed from bunch of event data at any time after measurements. To obtain and use event data effectively, advanced developments on both electrical hardware and software of DAQ (data acquisition system) are required. Independently, J-PARC and HANARO decided to employ the event data recording method instead of the traditional histogram recording method. The event data recording method was quite, and is still relatively new in neutron scattering communities. In spite of the initial difficulties, we have benefitted in many ways as described below by choosing the event data recording method. And we learned during this KJ collaboration that new major facilities and instruments, for example at SNS, also employed this method of event data recording.

Advantage over histogram recording method

The histogram recording method is a traditional way of storing neutron signal as histograms, which DAQ generates for an individual pixel of each detector. Basically, this method makes it easy to treat data after data acquisitions and requires simpler hardware configurations in comparison with that used for the event-data recording method. On the other hand, there are serious problems as the binning of time and

position must be decided before measurements and cannot be changed afterwards once data are taken. In addition, since there is no information about absolute time in the histogram recording data, it is impossible to remove some part of the data even if one knows that they are corrupted afterwards. The only way left for the histogram recording is to throw away the whole data. But if such a situation occurs during the measurement, then one must simply pause and resume the data taking when one is sure that the source of corruption disappears.

In the event data recording method, the time and position information of when and where a single neutron count is detected at a detector are recorded in the raw data as an event. The resolution for the time and position of event recording is directly limited by hardware for an associated DAQ, not by the width of binning in the time and position ranges. On making histogram for analysis, user can freely decide or remake the binning of both time and position from event data after measurement. This has several favourable advantages as described below:

- One can remove bad frame due to problems of source, out-phasing of choppers and so on at any time after the measurements.
- One can produce several dynamic structure factor data from a single measurement with suitable ToF binning for the resolutions for each incident energy when using multi-E_i chopper.

Time information in event data keeps not only ToF from T_0 of spallation at a neutron source or the burst time of a pulsing chopper but also absolute time of each measurement. This enables users to track and see time-dependent phenomena of samples and to remove time-dependent impurity or noise signals from outer environment, such as out-phasing of chopper rotation, fluctuation for current of neutron by temperature shift in moderator and so on. Using this advantage, therefore, users can obtain data of better quality than available at the histogram recording method. One of disadvantage of the event data recording is that it is not easy to treat and understand the raw data directly. We need more than one step before actually being able to examine the data, i.e. translation from event data to histograms.

The data volume was also an issue considered during our initial discussion. Histogram recording requires a fixed memory on PC or a fixed file size, while the volume of data recorded in event mode is proportional to measurement time and increases with the flux of detected neutrons. In contrast, ToF data stored in histogram mode for a typical inelastic neutron scattering experiment have many zeros, resulting in a waste of memory on PC.

We considered actual data size for our instruments. For example, 4SEASONS, AMATERAS and DC-TOF each will be equiped with more than 300 1-dimensional PSD. Practically, we divided each of PSDs into more than hundred pixels. Therefore, a total number of detecting pixels is estimated to be in the order of 10^4~10^5. In addition, the effective number of time-of-flight binning for energy resolution with a single incident energy is in the order of 10^3. Assuming that the number of PSD is 300, 200 of the pixel number for a single PSD, 1000 of ToF binning, a typical histogram can easily reach about 1 – 2 Gbytes of the data size including ToF, counts and errors. We estimate there is a region of trade-off at about 100-200 million events as far as data size is concerned (Fig. 12).

Data format of event data

Actual data format of event data is shown in Fig. 13. We note that MLF of J-PARC and HANARO of KAERI use different formats. A major difference between the 2 formats is how to record the real time

information of the events. In the case of MLF format, the real time of events is calculated by combining the most recent T_0 data using Pulse ID or Instrument clock data that are recorded together with event data. In contrast, the KAERI format keeps the real time recorded as epoch time. The difference between the 2 formats is not a major problem in our software development, since both types of data can be handled in our software in the same manner with a simple wrapper.

3. Basic data reduction software development

The structure of basic data reduction software is shown in Fig. 14. When all neutron events are stored as series of events with their own unique time tags, they will be called into the first step of the data reduction software, "Event to Histogram." Here, it will be converted to histograms, which will be further checked to discriminate data from bad detectors by so-called masking process. Then the ToF of the data is converted to the energy transfer by using detector information embedded in a histogram data. After the conversion, the data of individual pixels of detectors will be constructed by using information described in a mapping file. For example, circular averaging (averaging on same $|\vec{Q}|$ value) can be done at this part. After that, background treatments (subtraction of empty can data and constant background correction), normalization by monitor counts, vanadium correction and correction of detector efficiency will be carried out. Then $S(\vec{Q}, \hbar\omega)$ will be obtained, which will be used in other analysis programs developed by ourselves or at other institutes.

3.1 Event data to histogram

Event data should be converted to histogram before being used at further analysis processes. In our data reduction software, the histogram data are stored in ElementContainer of Manyo-lib. Manyo-lib already has base functions for conversion from event data to histogram which are made by using functions in the GNU Scientific Library [12]. We have either modified the existing functions or added new functions for our own purpose. Moreover, the MLF computing group of J-PARC decided to use the XML file format to store all information, which should be communicated across different software using same DAQs and PSDs regardless of whether it is for inelastic spectrometers or diffractometers. Therefore parameters in the XML file contains following information:

- A number of detectors and a number of pixels for a single detector
- Type of detector, 1D (1) or 2D (2)
- How to number detector ID and pixel ID
- ToF binning patterns for each pixel ID
- Position binning pattern for each detector ID
- Parameters needed to calculate position from event data

An XML file with the above information will be imported to set parameters that are necessary for converting event data to histograms.

To define and import the binning information of ToF and position for PSD

In order to convert the raw event data with time and position information to histograms, we need to define binning in both ToF from T_0 time and position at each PSD. The binning pattern of time is simply given by the data described in an XML file. Numbers of types of time binning patterns are defined by the MLF computing group. As for the position information, the numbers of pixels of detectors are also described in the XML file. The whole length of a detector is simply divided by the number of pixels. Then the position at a PSD is given by identification number of the pixel, at which a neutron count is detected. Also, specific pattern of the position binning can be given, which is currently not used in our present software. This binning information will be retrieved from the XML file when necessary.

To cut out event data from an event data file by specifying the time period in absolute time

We can cut out event data, which were taken in certain time period, from a event data file as described below. There are 3 types of event data, i.e. T_0 events, instrument clock events and neutron events. T_0 events contain the IDs of proton beam. T_0 events are used to trigger the instrument clock, and then, instrument clock events are created. Instrument clock event contain time information which are used as common clock of a given instrument. From T_0 events and instrument clock events, we can determine absolute time of neutron events. This function is included in the base functions of Manyo-lib. as one of conversion functions. Users can pick up neutron events by specifying the event time in absolute time. Users can use this function when they want to perform time transient experiments. Also, if users know that the data which were taken at some certain time periods were bad (because of experimental conditions were out of permitted range, choppers were out of phase, the neutron source had some problems or etc.), they can remove them from original file. Actualy, such operations are carried out by using separate utility programs, which are under preparation and are not yet included in the current version of data reduction software.

To convert from event data to histogram

In this process, each event data is sorted into time bins of each pixel of detectors. Conversion from event data to histogram is carried out by using functions called from Manyo-lib. For this purpose, some of functions are modified or replaced from the original ones in Manyo-lib or newly prepared by us, and all these are implemented in Manyo-lib. In the process of conversion from event data to histogram, all histogram data are stored in a series of ElementContainers, which are constructed using Python code. All ElementContainers are provided with unique identification number, Pixel-ID and ElementContainerArrays are PSD-ID.

To put position information to each histogram

The position information at each pixel will be added to each histogram. The information is used in data reduction to calculate a flight path length from a sample to a pixel and a scattering angle. This information is also stored in an XML file as mentioned before. To calculate the position from Pixel-ID, coordinates of the origin point on each PSD and the unit vector along PSD are used. The coordinate system used is described in the appendix. After converting event data to histogram, necessary parameters are also imported from another XML file and the position is calculated, of which results are written into Header in each ElementContainer.

To enhance the executing performance

Although the main functions are written in C++, we found it quite time consuming to execute the above processes especially when working with large size data. We are afraid that it might give bad experiences to users. This problem with data handling will get exacerbated with increasing beam power at the MLF. Simple test runs at 4SEASONS have proven that it is the case. Therefore it was an important issue for us to find out an effective method to speed up the execution of process. We have tried to overcome this problem by using a parallel computing method and a multi-threading method. Base functions for both methods are already implemented in Manyo-lib. In the parallel computing method, Python codes are distributed to several servers, so-called 'Python servers' located at other PCs. Then the processes are executed in parallel. It is the easiest way for us. On the other hand, in the case of multi-threading method executing processes are divided into several threads to utilize multi core CPU effectively. To use this method, we had to rewrite the codes in C++ to work this method effectively. Although this required another development work for us, we succeeded in writing the code that became very easy to use.

Moreover, the multi-threading method allows resources of PC (memory or hard disc) to be kept lower than in the paralleled method. For actual systems, we should find a way of using these 2 methods effectively with limited resources. Discussion about this point is added later in the report. For example, the relative performance of the above methods can be found in Section 5.

3.2 Masking and mapping

Here we discuss the process of masking and mapping (Fig. 15). Masking is used to check whether spectrums in ElementContainerMatrix are suitable for further data processing. Following situations can be considered as reasons for applying masking process:

- Data from specific pixels are abnormally noisy.
- Data from specific pixels contain spurious peak.
- Data from specific pixels contain unexpected Bragg reflection in inelastic spectrum.
- Detector efficiency is considered to be changed between several vanadium measurements by any reasons.

In masking, vanadium measurement is used to check bad detectors or pixels. Due to its strong incoherent scattering cross-section, neutrons scattered from vanadium should be evenly distributed over the entire solid angle. Therefore, vanadium measurement can be used for this purpose. The choice of bad detectors can be done either manually or automatically. Automatic selection is just to compare each detector's total count (C_i) and median of all total count (C_m). If C_i is much larger or much smaller than C_m, it is classified as a bad detector. The comparison factors are given by parameters.

For automatic masking, *ChopperMaskAuto(ecm, lowerlimit = 10^{-4}, upperlimit = 10^{4}, writeonfile = True, filename = None)* function can be used. The input parameters are:

- *ecm* : ElementContainerMatrix
- *lowerlimit* : default value = $1.0\text{x}10^{-4}$
- *upperlimit* : default value = $1.0\text{x}10^{4}$
- *writeonfile* : True if user want to write masking file
- *filename* : filename of masking file.

It returns masked index or masking file.

The *lowerlimit* and *upperlimit* are the values used for checking bad spectrum. These limits can be changed by users, when necessary. Otherwise, the default values are set to 10^{-4} and 10^{4}, respectively. The processes are as below:

1. Calculate the median value of every detector pixels' total count: C_m
2. Compare C_m and pixel's total count. If total count of i-th pixel C_i < *lowerlimit* * C_m or C_i > *upperlimit*$*C_m$, then the i-th pixel is included in the masking index.
3. If *writeonfile* is true, it writes masking file as given in *filename* parameter. If file name parameters are not given, the default filename "YYYYMMDD_hhmmss.msk" is to be used. YYYY, MM, DD, hh, mm and ss are the year, month, day, hour, minute and second when the masking file is made.

The masking file is a simple ASCII file and an example is shown below.

```
# INSTRUMENT UNKNOWN
# DATATYPE ElementContainerMatrix
# TIME 2008 3 12 15 8 47

1.4 1.8
4.4 4.6   4.8
7
8
9.4
```

The line starting with # is a comment line, which does not affect data processing. One can add comments using any text editor. The numbers in the file is PSD ID and Pixel number. For example, "1.4" means pixel with PSD ID=1 and Pixel ID=4 to be masked. "7" means every pixels with PSD ID=7 will be masked.

ChopperMaskAuto function returns masked index as Python's tuple type or writes masking file. It does not mask the histogram but returns index to be masked.

Manual choice of masked pixel may be necessary because automatic selection is done by simply comparing the total count of pixels. If there are pixels that have spurious or noisy spectrum, they may not be masked by automatic masking. For the manual masking, we have prepared a program, ECMInspector. It can be used also for navigating through all spectrums in ElementContainerMatrix.

ECMInspector has following functions:

- Inspect all spectra in the ElementContainerMatrix
- Over-plotting of chosen spectra
- Normalized plots for comparison with different total count
- Selection of plotting range
- Saving the masking index

ChopperMask(ecm, masked, maskingfile) is a function for masking. If a masking file already exists, this can be given by the *maskingfile* parameter.

One should keep in mind that the masking process does not remove masked spectra from ElementContainerMatrix, but masking information is written in the Header. After masking, the value of key "MASKED" becomes 1 in the Header of ElementContainer and ElementContainerArray. ElementContainerArray's Header also has a list of maked ElementContainer as an Int4Vector.

Mapping is to collect and/or merge histograms by the description of detector geometry and other parameters necessary for data reduction. For the mapping process, we need 2 types of the description. One is a group of parameters to be given to each histogram.

- Sample to detector pixel distance
- Electronic time delay of detectors (not implemented yet)
- Detector pixel location
- Solid angle of pixel

These parameters should be stored in the Header information of each element container and are simply used for the collecting and the merging of histograms. In the present status, only the information of pixel location is stored in each ElementContainer during the process of converting event data to histogram. This information can be kept with several description types, the pixel position in rectangle coordinates and the angle in the spherical polar coordinates. From these stored value, software can calculate the distance between sample to detected position or scattering vector (momentum transfer) $\vec{Q}$ using the incident energy. The other way of producing a mapping file is for users to write their own description of the rules or conditions as to how to collect or merge histograms. But we did not produce a procedure for the second option yet because the development of mapping function is not clearly defined at the moment. So far we have succeeded in producing the merging function for powder or glassy samples, which collects and merges histograms with $|\vec{Q}|$ value at elastic scattering for a given $|\vec{Q}|$ range (circular averaging).

This function is based on the merging framework of Manyo-lib. The function in this framework first collects histograms suitable to given parameters and then merges them. Upon merging, the function just sums and averages the collected histograms if all of histograms have the same ToF binning. If they have different ToF binnings, the function calculates each value of time binning by the summation with weight reflecting the different binning width. The merging function for powder/glassy sample produces a single ElementContainerArray as a final result.

3.3 ToF to energy conversion

"ToF to energy conversion" is a process to determine neutron's energy transfer from the ToF measurement. (For the scattering geometry, see Fig. 16.) Let's assume that P is a position of a pixel in PSD and that R is the reference position as explained below.

The parameters required for ToF to energy conversions functions are:

v_0 : incident neutron velocity

L^P_{SD} : distance between sample to detector pixel

t_P : ToF measured at pixel

L_R : distance between reference position to sample

t_R : ToF at reference point.

Then the final neutron velocity (v) and energy transfer ($\hbar\omega$) (= incident neutron energy (E_i) - final neutron energy (E_f)) can be calculated as below,

$$v = \frac{L^P_{SD}}{t_P - t_R - L_R / v_0}, \tag{1}$$

$$\hbar\omega = E_i - E_f = \frac{1}{2} m_N v_0^{\,2} - \frac{1}{2} m_N v^2, \tag{2}$$

where m_N is the neutron mass.

The reference point might be a chopper or monitor. In the case of a chopper, t_R is the chopper opening time in each frame. If the reference is the monitor, then t_R is the elastic peak center of monitor spectrum in each frame. Sample to detector pixel distance (L^P_{SD}) is different among pixels. Therefore, it must be carefully considered in data reduction and analysis process that spatial resolutions and energy resolutions are different among pixels.

3.4 Background subtraction

Background in neutron scattering measurements comes from many sources such as sample environment, neighboring instruments, electrical noise, etc. There are 2 types of background subtraction. One is subtracting the background data taken by 'background run.' The other one is simply subtracting constant. A background run is, in general, an empty run – a measurement with the same sample environment but without the sample. In that case, the background subtracted spectrum is given by

$$I_P(t_P[i]) = I^0{}_P(t_P[i]) - \beta \cdot I^B{}_P(t_P[i]) - B_C, \quad (3)$$

where $I^0{}_P$ is the sample run, $I^B{}_P$ is the empty run, β is the subtraction ratio calculated from sample environment properties, and B_C is the constant background. If there is time independent constant background, our background subtraction function provides some ways of determining constant background. Although it is not described in Fig. 14, this function can be easily prepared also by users, if necessary. The parameters for these calculations are listed in Table 1.

3.5 Vanadium correction

White or monochromatic vanadium measurements are used for correcting detector efficiency fluctuation and for getting absolute physical value from neutron scattering experiments. As we have discussed for a masking process before, the strong incoherent scattering cross-section of vanadium also can be used for detector efficiency correction.

White vanadium data

White vanadium data (data from white beam vanadium runs) are used to perform corrections for the efficiency differences of each detector from the averaged value. The data are also used for the corrections for the solid angle defences of each pixel. The detail algebra of the neutron counts is described in Appendix v. We can introduce the parameter γ_P, the efficiency difference from the averaged value for the detector which contains the pixel P. Then the efficiency of the detector can be written as $\varepsilon(E_f) \times \gamma_P$, which should be replaced with $\varepsilon(E_f)$ in eqs. (A.1) and (A.4). Also, for the pixel P, the solid angle $\Delta\Omega$ in in eqs. (A.1) and (A.4) should be replaced with the solid angle for the pixel P, $\Delta\Omega_P$. Since both neutron counts of the sample runs and neutron counts of the white beam vanadium runs contain γ_P and $\Delta\Omega_P$, by dividing the counts from the sample by the counts from a white beam vanadium run, the differences in the sensitivity and solid angle on each pixel will be cancelled out.

Monochromatic vanadium data

To obtain absolute quantities from measured data, the monochromatic vanadium data is needed. Especially, the correction of the transmission of the components on the beamline (chopper and any other devices) is done by this data. The transmission of devices on the beamline at E_i is the term $T(E_i)$ in eqs. (A.1) and (A.3). Therefore, by dividing the measured intensity by the counts from a monochromatic vanadium run, the corrected intensity can be obtained.

Vanadium module

The diagram of the vanadium module is shown in Fig. 17. The actual module for this process consists of 2 different data treatments, which depend on data type to be processed. One is to calculate the integral of counts and the other is to correct the sample data by using this integral result.

In the "Prepare vanadium procedure" in Fig. 17, the raw event data of white vanadium and

monochromatic vanadium run can also be treated by the data reduction steps just as done for the sample data, i.e. conversion translating to histograms, masking, converting ToF to energy, mapping, normalization and subtracting background.

After processing these steps, the calculation is done for the integral of the intensity in histograms over the given energy region. The result is stored in ElementContainer with pixel-ID. In "Sample procedure", the vanadium module executes this correction with dividing sample data by the integral value stored in the ElementContainer, which is kept on memory or stored on the hard disc.

3.6 Detector efficiency correction

Detectors of chopper spectrometers use ^{3}He gas sealed by thin metal wall. Absorption cross-sections of materials have certain energy dependence, therefore detector efficiency also has similar energy dependence. The definition of detector efficiency is as following:

$$\varepsilon(E_f) = \frac{\text{Number of neutrons detected by detector}}{\text{Number of incident neutrons at detector}} \tag{4}$$

Detector efficiency depends on the average thickness of ^{3}He and detector wall. Therefore, if PSD stands vertically as above, efficiency also depends on the angle between the incident direction of neutrons and PSD direction (θ) (see Fig. 18).

Fig. 19 shows a cross-sectional view of a ^{3}He detector and the detector wall with respect to the incident neutrons. The detector efficiency is averaged over all cross-section with the following detector efficiency along neutron path (red line) :

$$\eta(x) = \exp(-n_{cell} \cdot \sigma_{cell} \cdot 2l_{cell}) \cdot (1 - \exp(-n_{^3He} \cdot \sigma_{^3He} \cdot 2l_{^3He}), \tag{5}$$

where n_{cell}, σ_{cell}, n_{He} and σ_{He} are numbers of atoms and cross sections of length of cell(l_{cell}) and length of ^{3}He (l_{3He}) are as below

$$l_{cell} = \frac{1}{\cos\theta}\left(\sqrt{d^2 - x^2} - \sqrt{(d - d_{cell})^2 - x^2}\right) \tag{6}$$

$$l_{^3He} = \frac{1}{\cos\theta}\left(\sqrt{(d - d_{cell})^2 - x^2}\right) \tag{7}$$

.

Absorption cross-section for ^{3}He has the following energy dependence:

$$\sigma_{^3He}(E_f) = \frac{5.024}{\sqrt{E_f\,(\text{meV})}}\,\sigma_{^3He}(25.247\text{meV}) \tag{8}$$

.

Therefore, energy dependent detector efficiency can be calculated from integration of equation (5) by using equations (6) - (8).

$$\varepsilon(E_f) = \frac{1}{2(d - d_{cell})} \int_{-(d-d_{cell})}^{d-d_{cell}} \eta(x) dx \quad (9)$$

.

Calculated detector efficiencies for some ^{3}He pressures and wall thicknesses are shown in Fig. 20.

4. Visualization software development

4.1 Informative visualization software

Informative visualization software is a utility to help users in their making decision on the experimental conditions such as chopper frequencies, choice of incident energies, choices of slit packages of Fermi-choppers or slit width of disk-choppers and so on. Using this software, before their measurements, users can consider experimental parameters by referring visual information in $\vec{Q}$-$\hbar\omega$ space, which is determined by certain crystal orientation and instrumental conditions given. This software can display $\vec{Q}$-$\hbar\omega$ space, expected flux at sample position and $\vec{Q}$ and $\hbar\omega$ resolution at the specific point. By using device control environments interfacing between the devices concerned and the software, this informative visualization software can also control devices directly or indirectly. At ISIS, there are already many handy utilities for determining such measurement conditions, e.g. CHOP and TOBYPLOT [13]. However, these utilities are completely independent from each other. User should use several software sets and should transfer the data resulted from one utility to the other by manual. And finally user should input the optimized conditions to the hardware controller again by manual. These situations cause uneasiness and, in sometime, pain to users, and are apparently not effective. In our new software development, we are aiming to make integrated decision making and optimization utility, which is combined with hardware control software tightly. Then setting-up the experimental conditions becomes easier.

Visualization of scan trajectory

At the beginning of a run, one should determine the incident energy. Scattered neutrons with some final energy E_f are detected by PSDs covering a large area of solid angle. The momentum transfer $\vec{Q}$ and energy transfer $\hbar\omega$ by scattering at the sample are calculated as below:

$$\vec{Q} = \vec{k}_i - \vec{k}_f , \tag{10}$$

$$\hbar\omega = \left(\frac{\hbar^2}{2m_N} \right) \left(k_i^{\,2} - k_f^{\,2} \right), \tag{11}$$

where $\vec{k}_i$ is the incident wave-vector and $\vec{k}_f$ the final wave-vector. Equation (11) is another expression of equation (2) describing the neutron energy transfer.

The accessible region of a pixel of a detector corresponds to a line (so-called scan trajectory) in the (Q_x, Q_y, Q_z, $\hbar\omega$) space, which depends on the chosen incident energy and the scattering angle of that pixel. By viewing the projection of scan trajectories from all pixels of all detectors on reciprocal lattice space defined by the sample, which can be calculated by this software, users see immediately which (Q_x, Q_y, Q_z, $\hbar\omega$) region in the Brillion zone can be covered by the measurement.

Features of functions

At the beginning of measurements, users may want to know an expected flux at the sample position and an expected energy resolution for a given setting. This information are required in choosing chopper conditions such as chopper frequencies, slit packages of Fermi-choppers or slit widths of disk-choppers. After selecting these chopper conditions on the left top panel of the main view in Fig. 21, several lines are plotted at the bottom of the left panel where horizontal axis indicates the incident energy. These lines show a series of the flux and energy resolution with selected chopper frequencies. The incident energy can be determined by typing values in the box at the left-middle panel or clicking on the plot directly.

Lattice parameters and orientations of the sample are shown on the top right panel. Then the sample information is used in calculating the scan trajectory on the right-bottom panel. This plot indicates the Brillion zone of sample and all scan trajectories projected on the horizontal plane including the sample position. By clicking a position on the plot and by pushing the "vertical plane" button, the scan trajectory on the vertical plane including this position will appear in another window. By pushing "Plot Resolution" button, the plot of $\vec{Q}$ and energy resolutions versus $\hbar\omega$ can also be drawn. If users want to change sample orientations and incident energy, they can write required values in each box or drag and drop directly the lattice point. Then both of the orientations of sample and incident energy are automatically calculated for the chosen point. As mentioned above, presently this software can be controlled by mouse. Using this software, therefore, the first step of each measurement can be carried out intuitively by not only expert users but also beginners.

4.2 Data visualization software

Visualization for data reduction

Visualization software helps users to treat and visualize any data at data reduction steps, i.e. the simple ToF data and the $S(\vec{Q},\hbar\omega)$ data measured in inelastic neutron scattering instruments, which have the 5 dimensions (3 axes for Q_x, Q_y, Q_z,, 1 axis for $\hbar\omega$ and 1 axis for the intensity). Functions of visualization program are mainly consisted of several parts, 1) 1D-Plotter (X and Y axes), 2) 2D-Plotter (X, Y axes and the intensity) and 3) the data management including slicing a higher dimensional data into lower dimensional ones. These 2 plotters are independent software and can also be controlled individually by Python command line or other software.

$S(\vec{Q},\hbar\omega)$ data from all detectors, which are produced by the data reduction software, are loaded by this software both to visualize and to analyze themselves. Some parameters are read from Header information of ElementContainerMatrix. By inputting sample lattice parameters and parameters for projection, users can slice and plot the data. At the present status, this software can only slice and plot the data. Any functions for advanced analysis, e.g. fitting and convolution of resolution are not prepared yet and will be parts of future project.

Functions and interface

In the main panel, users can handle up to 5 individual data sets of $S(\vec{Q},\hbar\omega)$. See the left figure in Fig. 22. Users can choose each data by clicking tabs on GUI. In addition, it is possible to calculate arithmetic operation between 2 data sets and to put the result into another tab. Slicing and plotting functions are also included in this panel. For given parameters, projecting and slicing are carried out to send the results to the outer 2D-Plotter for display on the screen.

2D-Plotter can plot not only the sliced data of single crystal sample but also the powder/glassy sample by given ElementContainerArray. Using the 2D-plotter, users can cut the 2D data by mouse on GUI and the resulting data is then sent to the outer 1D-plotter for display on screen.

1D-Plotter, called MPlot, is usually used just to plot the histogram in ElementContainer. As other ordinary plotters, this plotter has several basic functions such as changing types of lines and markers, scaling plot region and so on. Also this plotter can read and write text data file to import/export any histogram data. More advanced functions for data analysis will be implemented in future.

Future plan

Most parts of the software are ready to use and user can handle the data processed by the reduction software. However, functions are still limited to basic ones and may not be sophisticated enough for more

advanced requests of users. Also, the software has several other problems to be improved. At this moment, we can point out following problems with the current version of the software:

- An input procedure of sample parameters is still primitive. Presently user manually provides the lattice parameters and the lattice vector to indicate the sample direction. This information should be shared with other software, for example the informative visualization software or experimental information management system. And we need to improve how to input the setting parameters.
- Cut and sliced data are independent from the command line system. This series of software can be called from the command line interface on Python. Currently, the interchangeability is rather low between the command line system and GUI. To solve this problem, we should connect GUI to command line system directly. However, at this moment, it is not clear whether our users prefer to use the visualization functions from the command line interface.
- More functions for plotting are required depending on the requests from users.
- We should implement 3D visualization plot.

5. Performance test of software

5.1 Event data to histogram

From the beginning of Manyo-lib developments the MLF computing group has considered the issues of handling large data produced at MLF with high flux of neutrons. Then, the computing group prepared a function to distribute some processes of data treatment over several servers, called "Python Server," in Manyo-lib. This server can be executed in multiple PCs connected to a host PC by networks. In this way, one can basically use many severs at the same time. While developing our software, we realized that the process of converting raw event data to histograms takes particularly long time for real data. Therefore, we developed another converting method using a multi-threading technique, which was then implemented for Manyo-lib.

We were particularly interested in the speed of these methods for the converting process. In Table 2, we compare the 2 methods to identify which new coding is required.

We checked the performance of both methods in terms of speed and memory resources with conditions as given below.

- Test environment:
  PC: Quad-core with 2.0 GHz.
  Data size: 8 raw event data files (in total 1.0 GB on hard disc)

- Program environment:
  Distributed processes: 4 Python Servers on 1 PC
  Multh-threading: 4 threads on 1 PC

We show the test results with different conditions about ToF binning.
Condition 1: Number of Pixels: 9600, Number of ToF bins: 400 (Table 3)
Condition 2: Number of Pixels: 9600, Number of ToF bins: 8000 (Table 4)

For each test, we included a single thread results as a reference. In terms of speed both methods are 2-3 times faster than a single thread with a distributed processes method being a little faster than multi-threads for Condition 1. On the other hand, there is a big difference in the memory size. A distributed processes method occupies 2-3 times larger memory size than that required for a multi-threading. The reason for the large memory for the distributed process is simply explained by the fact that 4 processes are executed in parallel in this method.

For Condition 2, the multi-thread method becomes better in terms of both speed and memory. One disadvantage with the distributed processes is that it should have additional file access process in order to combine several data produced by each server, as shown in Fig. 23. Table 3 and 4 show the file access time. It is obvious that if file access process gets much faster, the distributed method might become better.

For practical purposes, we may use both methods together. For example, we can distribute several process of multi-thread conversion into several servers on other PCs. Most effective way to use both methods should be found in near future.

5.2 Masking

Testing conditions are following.

- OS: OSX 10.5

- CPU: Intel Core 2 Duo 2.4 G Hz
- Memory: 2GB, 667 MHz DDR2 SDRAM

A masking function is tested using dummy vanadium data. Each pixel has similar spectrum except for some pixels to be masked. We assumed that 300 PSDs are installed and every PSD has 100 pixels. Each pixel has a spectrum with 600 energy bins. For the test of the automatic masking function, we arbitrarily choose 50 ElementContainers and put extremely small or large values in to them. We expected that these 50 pixels would be automatically masked. Automatic masking written in Python took just 0.5 seconds to find bad pixels and to generate a masking file. The process speed of the Python module is satisfiable.

5.3 Mapping

For the test of the mapping function, we prepared a function including collection and merging of histograms for data taken on a powder or glassy sample along a Debye-Sherrer ring. This is a built-in function of Manyo-lib, which is written in C++.

Brief descriptions of a sample data and conditions for merging are as shown below.

Test environment: Quad-core with 2.0 GHz, 8GB of memory

Sample data:

- The number of pixels = 16000
- The number of ToF bins at each pixel = 1500

Merging:

- to merge 16000 pixels into 54 pixels.

This merging process took 9.1 seconds.

5.4 ToF to energy conversion

For this process, we have prepared a function in Python. However, we found that this function took much longer than we expected for actual measured data. So we then converted it into a C++ code, which made significant improvement in speed.

To show the performance of the C++ code, we used the sample data as shown below and checked the performance.

Test environment: Quad-core with 2.0 GHz, 8GB of memory

Sample data:

- The number of pixels = 16000
- The number of ToF bins at each pixel = 1000

Conversion steps:

Step 1: Conversion from each ToF to energy transfer

Step 2: Re-binning them so that all pixels have same array of energy transfer

Results are shown in Table 5.

In step 1, the code of C++ is 100 times faster than that of Python while there is a small difference in step 2. This difference comes from the way of coding with Manyo-lib commands (see notes below). This

result indicates that developers should be more careful about execution time when they develop new coding using Manyo-lib.

Notes: In step 2, both coding call the same function written in C++, and the difference is where the function is called. On the other hand, in step 1, the codes must treat the data in ElementContainer. There, the C++ code can directly treat data of ElementContainer on memory, while the Python code cannot do it and needs to copy the data for further treatments. Although Python takes longer executing time than C++, it also has an advantage that there is no risk of destroying the original data. For general users, therefore, we recommend to use Python for safety.

5.5 Background subtraction

In order to test the function for background subtraction, we used the same testing condition as Masking. We considered 2 kinds of background subtraction. The first one is to subtract one ElementContainerMatrix from another ElementContainerMatrix directly. The other way is to calculate constant background and subtract it from an ElementContainerMatrix.

Direct subtraction using 2 ElementContainerMatrix's is provided by *SubtractTwoECM(ecm1, ecm2, factor)* function. The results are *ecm1-factor*ecm2*, where *ecm1* and *ecm2* are ElementContainerMatrix's and *factor* is a real number. To test the speed of both methods, 2 ElementContainerMatrix's are made with 300 PSDs, where each PSD has 100 pixels and each pixel has 100 bins. It took 14.3 seconds to perform *SubtractTwoECM(ecm1, ecm2, 0.3)*. The obtained results are shown in Fig. 24.

A function of finding constant background was also tested using the same size of ElementContainerMatrix. The results can be summarized below:

- Background subtraction by given constant: 18 seconds
- Background subtraction by calculating constant background for each spectrum. Number of points at both ends to be used for the estimation of constant background are given: 42 seconds
- Background subtraction by calculating constant background for each spectrum. Range of spectrum used for the estimation of constant background is given: 120 seconds.

5.6 Detector efficiency correction

We used the same testing conditions as the Masking process. Detector efficiency correction using 300 PSDs, 100 pixel per PSD and 100 energy binnings per pixel took about 8 min. Figure 25 shows an example after detector efficiency corrections. We note that our current code written in Python for detector efficiency correction took much longer than we would like. Therefore, we are considering writing a new function using C++ in order to speed up the process.

Overall, we conclude that the current masking code is sufficiently fast but both background correction and detector efficiency correction need to be improved probably by writing new codes in C++.

5.7 Beta test using pulsed neutrons at MLF

In June 2009, we measured a single crystal of $CuGeO_3$ on 4SEASONS at MLF, J-PARC. Using these real data, we checked parts of the data reduction process. The data used consists of separated 20 event files with a total size of 2.0 GB. The following steps of data reduction were executed in this test:

- Event data to histogram
- Translation from ToF to energy transfer
- Correction for k_f/k_i term

Results are shown in Table 6. In "Event Data to Histogram," we used 2 Python servers running at 2 different PCs for distributed methods (Fig. 26). After these processes, results stored in ElementContainerMatrix are sent to Data visualization software to be visualized. It took 2 seconds to import data into Data visualization software. Once imported, the projection, slicing and plotting using this software is sufficiently fast enough for users to use. Figure 27 shows an example plot from this visualization. In the lower part of plot, we show data for 4 different incident energies, which were obtained in a single measurement. For each incident energy of neutrons, we need to repeat these processes one by one to get the final data set.

6. Discussion

6.1 Status of progress and remaining issues

As stated in Section 1.3, we originally had 6 development items as targets for this collaboration. The achievements of these development works are summarized as following:

- Basic data reduction software: Beta-version released
- Informative visualization software: Beta-version released
- Data visualization software: Beta-version released
- Decision making & optimization software: Conceptual study done
- Single crystal alignment software: Conceptual study done
- Advanced analysis software: Conceptual study done

Most parts of "Basic data reduction software" have already been completed. And a beta-version is currently running on 4SEASONS and AMATERAS of J-PARC. However, the following components are still missing.

- Monitor counter normalization-related components
- Vanadium normalization
- Check of histogram for masking

A beta-version of "Informative visualization software," was already released. We consider that the beta-version is only for a demonstration of our concept and needs to be upgraded in order to be used for real experiments.

A beta-version of "Data visualization software" already has most of functions we wanted to implement. For example, users of 4SEASONS already used this software for experiments and published their results. Remaining issues concerning with "Data visualization software" are to improve the interface and add more functions.

We have not yet started coding of "decision making & optimization software" and "single-crystal alignment software." We only finished conceptual design of both sets of software. We have also discussed with the single-crystal diffractmeter team of J-PARC how to initiate collaboration with them for the development of "single-crystal alignment software."

"Advanced analysis software" is also left for future work. This part is one of the main targets of our next collaboration work.

In summary, we released beta-version software for 3 items, but they have some remaining issues to be solved. The other 3 items are left for future development.

6.2 Future plan of KJ collaboration

Future plan of KJ collaboration software development has been discussed during both offline and on-line meetings on several occasions. Since our development work is not yet completed, finishing current work and preparing the documents have been agreed as next targets of this current collaboration. We have also agreed on continuing this KJ collaboration in software development as long-term targets.

Short-term plan: We should correct problems already identified by the recent test results of the data reduction software. Some of them have been found during the commissioning of 4SEASONS. By the end of 2009, we performed further extensive commissioning on both 4SEASONS and AMATERAS of J-PARC with real neutron events and checked some problems. By using these test results, we have improved the performance of the software by writing some of the core codes in C++ instead of Python. This work includes upgrading the current programs. We can envision that it will take most of our time until proven to be fully user friendly, which is expected to be sometime in 2010. In parallel with efforts on both tests and improvements, we will prepare 2 separate manuals, one for users and another for professional developers as reference (technical) manual.

Mid- and long-term plan: The key plan of the future direction of this KJ collaboration was discussed by core members during an offline meeting on 9 July 2009, which was accepted and was broken down into the action plan during the following on-line meeting attended by all KJ collaboration members. With the 1st phase of the current development nearing to the end, we have more or less a complete set of basic software, which can be used to produce $S(\vec{Q}, \hbar\omega)$ from raw data. So, the next target of this KJ collaboration will be to develop advanced analysis software, which is strongly science-oriented. Furthermore, we agreed that it requires some strategic thinking as to which unique tools we want to develop by ourselves. Also, during our discussion, we came to conclusion that our level of software development is still relatively low compared with the situations at ISIS and ILL. Therefore, we agreed to adapt a catching-up approach for some years until we are confident enough that our level of software are mature enough to begin true collaborations with institutes outside the current KJ collaboration. We also decided that we should soon bring together core groups of collaborating team from outside for both science and software developments.

Here is the summary of our conclusion for the next phase of the KJ collaboration:

- We decided that for some time we will continue in the current mode.
- For the beginning, we will start from general analysis software, not too advanced or specific one.
- We plan to involve new members into this KJ collaboration in addition to J. -Y. So and Y. Inamura since they are expected to become busier with responsibilities at J-PARC and HANARO beginning in 2010.
- During user program that already started at J-PARC, we will have to look for "good" potential-collaborators regarding advanced software development.
- After the end of the second phase of this KJ collaboration, we will discuss our strategy again regarding the future direction of analysis software development with other foreign institutes.

Acknowledgement

We acknowledge J-PARC and KAERI for their generous supports for this Korea-Japan collaboration on ToF software development. Throughout this collaboration, we have benefited from discussions with numerous people. In particular, we should thank T. G. Perring of ISIS and G. E. Granroth of SNS for their helpful comments and discussions. One of us (J. -G. Park) acknowledges supports by the KAERI through the CNRF project for DC-TOF. This work was supported by a Grant-in-Aid for Specially Promoted Research (No. 17001001) from the Ministry of Education, Culture, Sports, Science and Technology of Japan and the Cold Neutron Research Facility project of the Ministry of Education, Science and Technology, Korean government.

References

[1] R. Kajimoto, T. Yokoo, K. Nakajima, M. Nakamura, K. Soyama, T. Ino, S. Shamoto, M. Fujita, K. Ohoyama, H. Hiraka, K. Yamada, M. Arai, J. Neutron Res. **15** (2007) p. 5
[2] K. Nakajima, M. Nakamura, R. Kajimoto, T. Osakabe, K. Kakurai, M. Matsuda, M. Metoki, S.Wakimoto, T. J. Sato, S. Itoh, M. Arai, K. Yoshida, and K. Niita: J. Neutron Res. **15** (2007) p. 13.
[3] KAERI Technical Report, KAERI/RR-3156/2009, "Development of Cold Neutron Research Facility and Utilization Technology"; KAERI Commission Report , KAERI/CM-1241/2009, "Development of inelastic neutron spectrometer (DC-TOF) & utilizations."
[4] Detail of Mslice and Homer can be found at the software reference site of the excitation group at ISIS, http://www.isis.stfc.ac.uk/groups/excitations/software/.
[5] R. E. Lechner, Physica B **180&181** (1992) p. 973.
[6] *See* http://www.Python.org/about/.
[7] *See* http://wiki.cacr.caltech.edu/danse/.
[8] *See* http://www.mantidproject.org.
[9] *See* http://www.swig.org.
[10] J. Suzuki, K. Murakami, A. Manbe, S. Kawabata, T. Otomo, M. Furusaka, Nucl. Instrum. Methods Phys. Res., A **534** (2004) p. 175.
[11] J. Suzuki, T. Nakatani, T. Ohhara, Y. Inamura, M. Yonemura, T. Morishima, T. Aoyagi, A. Manabe, T. Otomo, Nucl. Instrum. Methods Phys. Res., A **600** (2009) p. 123.
[12] *See* http://www.gnu.org/software/gsl.
[13] *See* http://www.isis.stfc.ac.uk/groups/excitations/downloads/downloads4639.html.
[14] M. Nakamura, R. Kajimoto, Y. Inamura, F. Mizuno, M. Fujita, T. Yokoo, M. Arai, J. Phys. Soc. Jpn **78** (2009) p. 093002.
[15] R. Osborn, R. S. Eccleston, S. M. Bennington, B. G. B. Kitchener, Data Reduction on HET and MARI, HET & MARI User Manual, ISIS Report (1995).

Figures

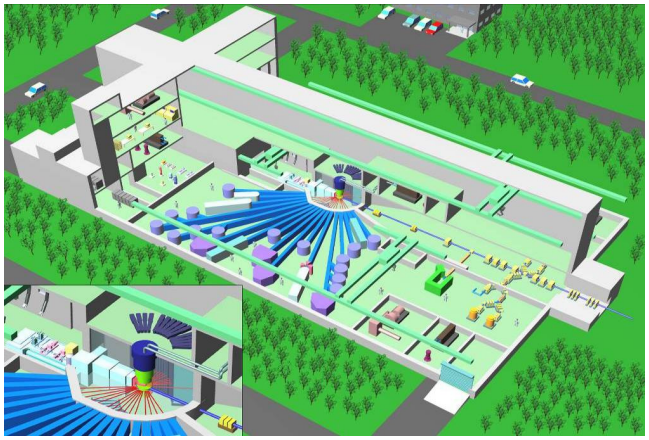
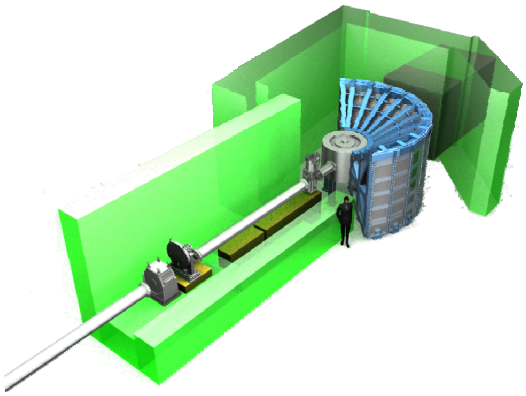
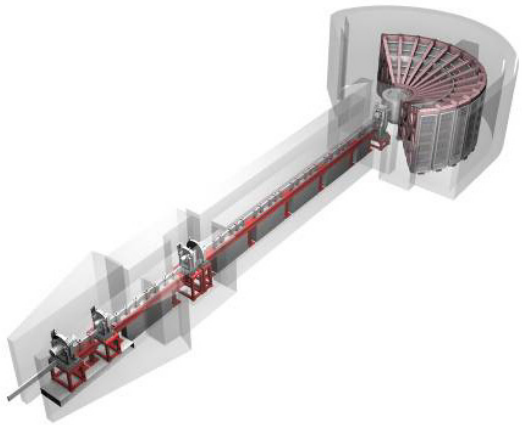


Fig.1 Overview of J-PARC (upper panel) and Materials & Life Science Facility (lower panel).



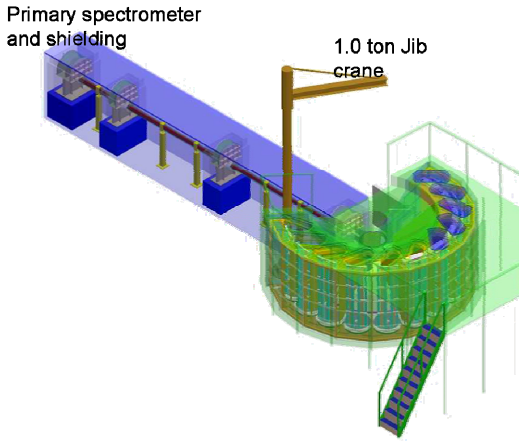
4SEASONS (JAEA, KEK, Tohoku Univ.)
Construction 2006 ~ 2008
L(moderator-sample)=18m
L(sample-detector)=2.5m
E_i=5~300 meV
Energy Resolution: $\Delta E/E_i$=5~6%
Horizontal Scat. Ang.: -30˚ < 2θ < 120˚
Vertical Scat. Ang.: -25˚ < 2θ < 26.7˚



AMATERAS (JAEA)
Construction 2006 ~ 2009
L(moderator-sample)=30m
L(sample-detector)=4m
E_i=1~80 meV (at its best @ 1~20 meV)
Energy Resolution: $\Delta E/E_i$>1% @ E_i=20meV
Horizontal Scat. Ang.: -40˚ < 2θ < 140˚
Vertical Scat. Ang.: -16˚ < 2θ < 22˚

Fig. 2 4SEASONS (upper panel) [1] & AMATERAS (lower panel) at J-PARC. [2].

Fig. 3 30 MW research reactor, HANARO (left) and new HANARO cold-neutron guide hall.



DC-TOF (KAERI & Sungkyunkwan Univ.)
Construction 2006 ~ 2010
L(1st chopper-sample)=10.5m
L(sample-detector)=2.5m
E_i=1~20 meV
Energy Resolution: $\Delta E/E_i$=2~10%
Horizontal Scat. Ang.: -91.4° < 2θ < 136.6°
Vertical Scat. Ang.: -21.8° < 2θ < 21.8°

Fig. 4 DC-TOF at HANARO cold-neutron guide hall [3].

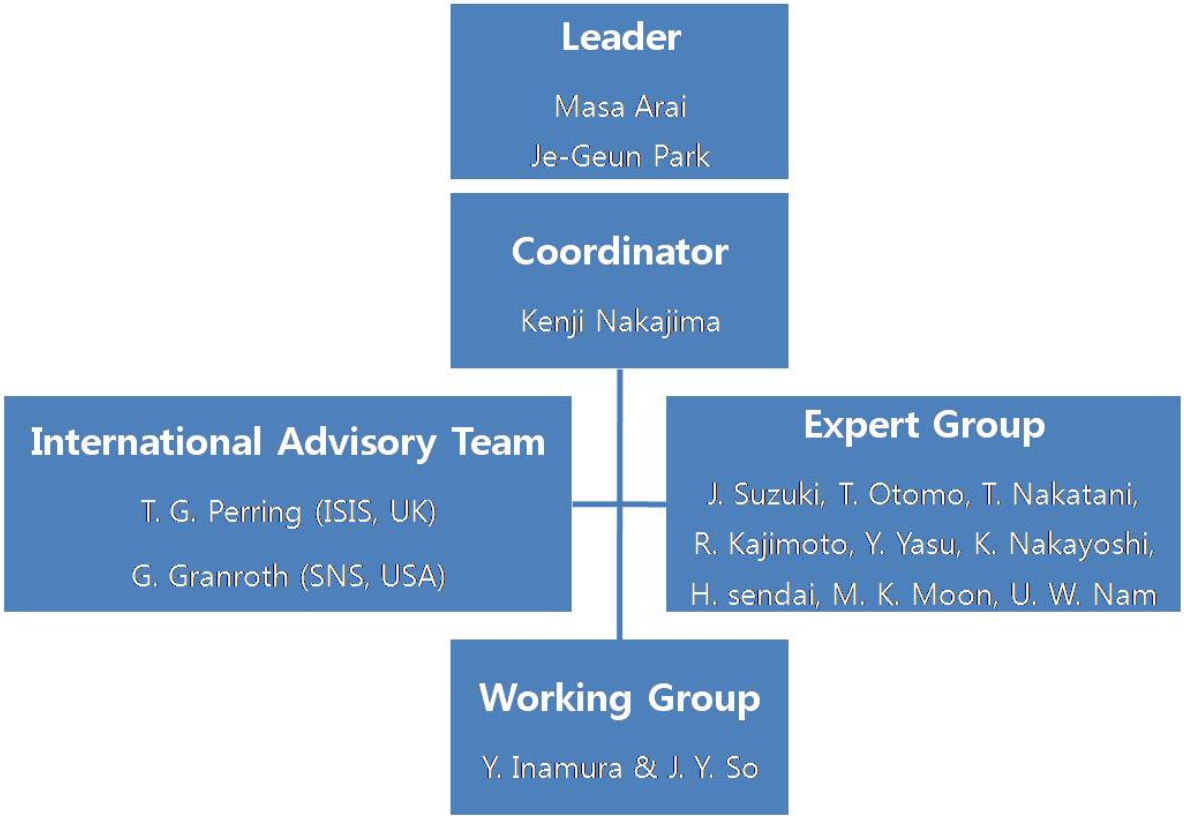


Fig. 5 Organization chart of the collaboration team.

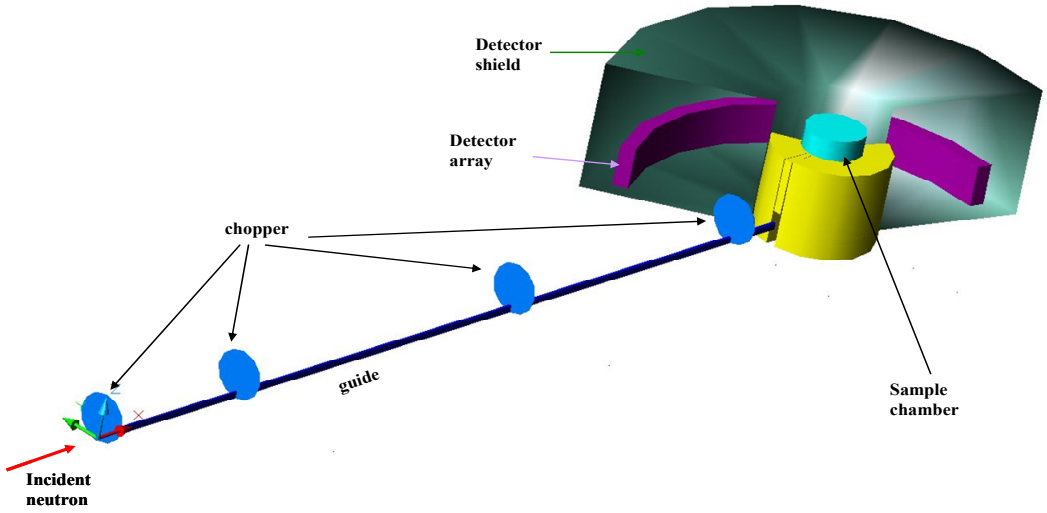


Fig. 6 Schematic figure of time of flight spectrometer.

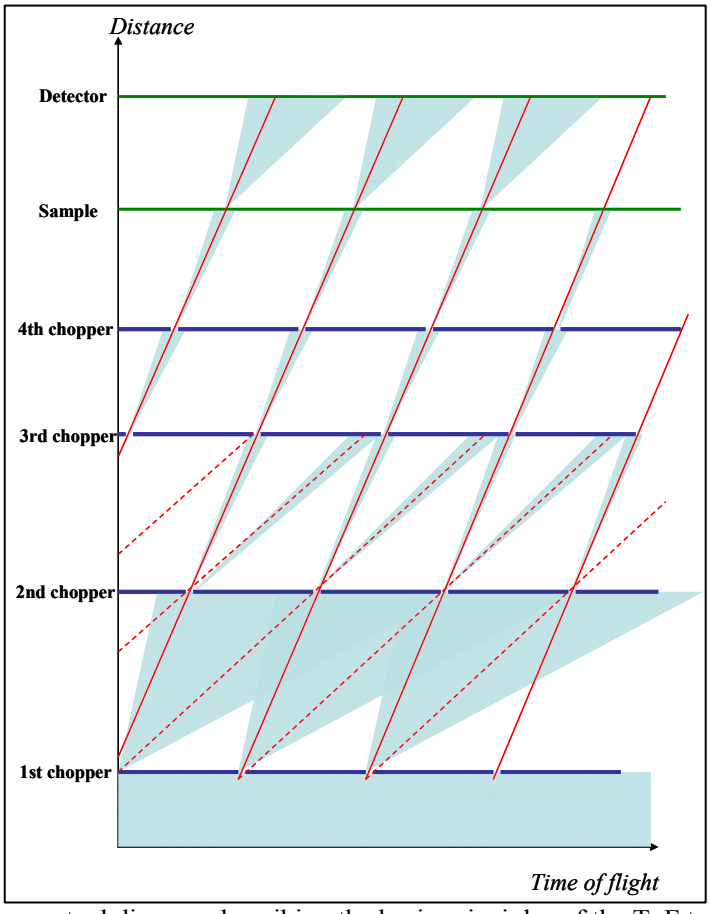


Fig. 7 Conceptual diagram describing the basic principles of the ToF technique.

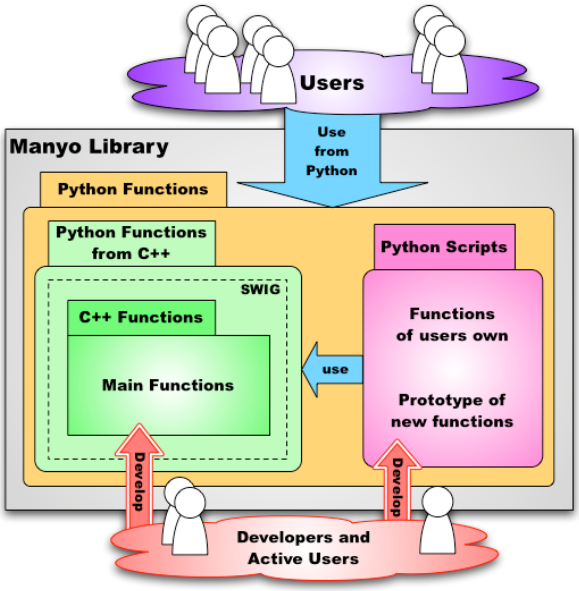


Fig. 8 Roles of Python, C++, SWIG and Manyo-lib in our software development.

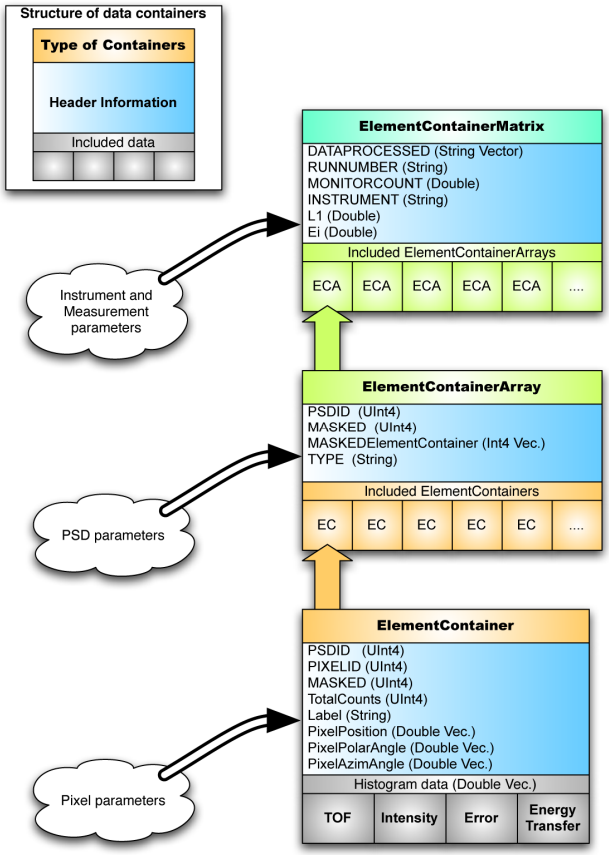


Fig. 9 Structure of a data container in Manyo-lib. Data container has a hierarchical structure of ElementContainer<ElementContainerArray<ElementContainerMatrix. This hierarchical class structure makes it possible that each of containers can hold numerous lower level containers with easy access to the lower level containers. Each data container can also store the meta-information of subjected containers in the Header. The lowest level container is ElementContainer that contains arrays of histogram data (ToF, intensity, errors and energy transfer values).

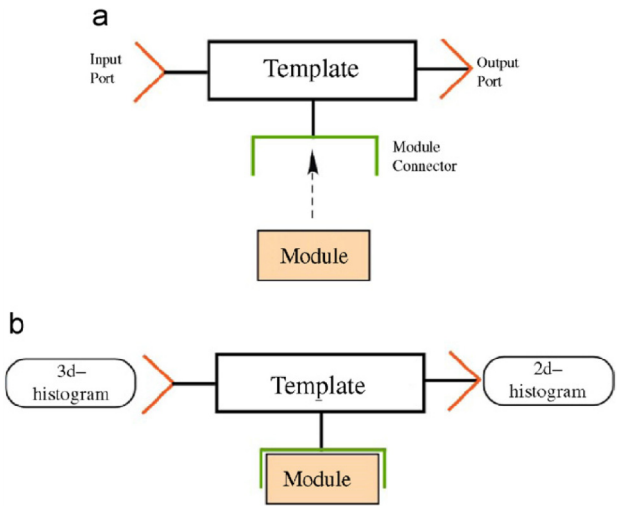


Fig. 10 (a) A schematic structure of an analysis operation. The template in the diagram is a template class, and consists of input and output ports and a module connector. The module describes the operation procedure. (b) A typical example of an analysis operation for projecting 3-dimensional histogram (3D histogram) into 2-dimensional histogram (2D histogram), taken after Ref. [11].

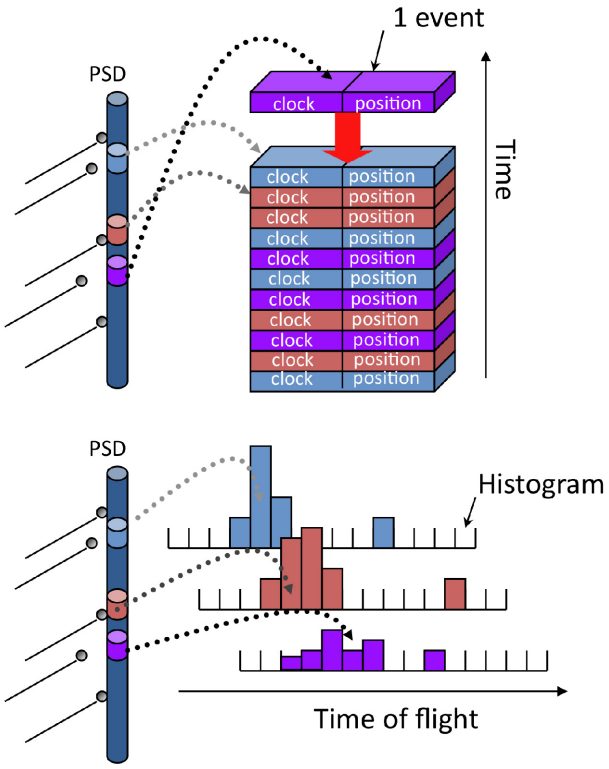


Fig. 11 Diagram of event data recording method (upper panel) and histogram recording method (lower panel).

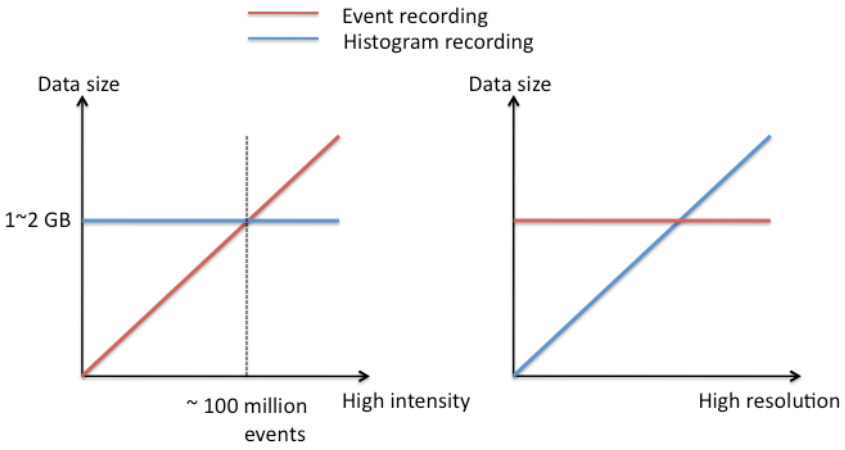
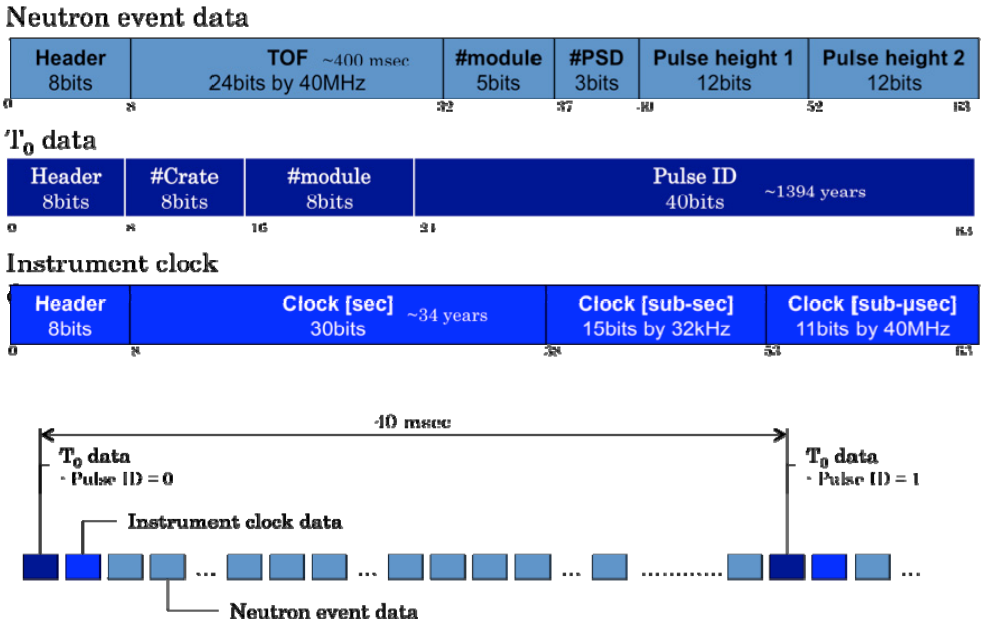


Fig. 12 Comparison of data size for event recording (red) and histogram recording (blue).

MLF Event data Format

8bytes (64bits) for one event



Header: discrimination for event type
#module, #PSD: discrimination for amp box and psd
Pulse height: A/D converted electric signal from both terminals of PSD
Pulse ID: Unique ID of each occurrence of neutron

KAERI Event data Format
Raw data : 14 bytes for 1 event

0	1	2	3	4	5	6	7	8	9	10	11	12	13
STX	ID		TOF			PL		PR		E. time			

STX: Event data frame
ID: PSD ID (1 to 352)
ToF: Time-of-flight (50 nsec unit)
PL, PR: Positions of pixel
E. time: Epoch time (real time in sec)

Fig. 13 Event data formats used at MLF of J-PARC and KAERI.

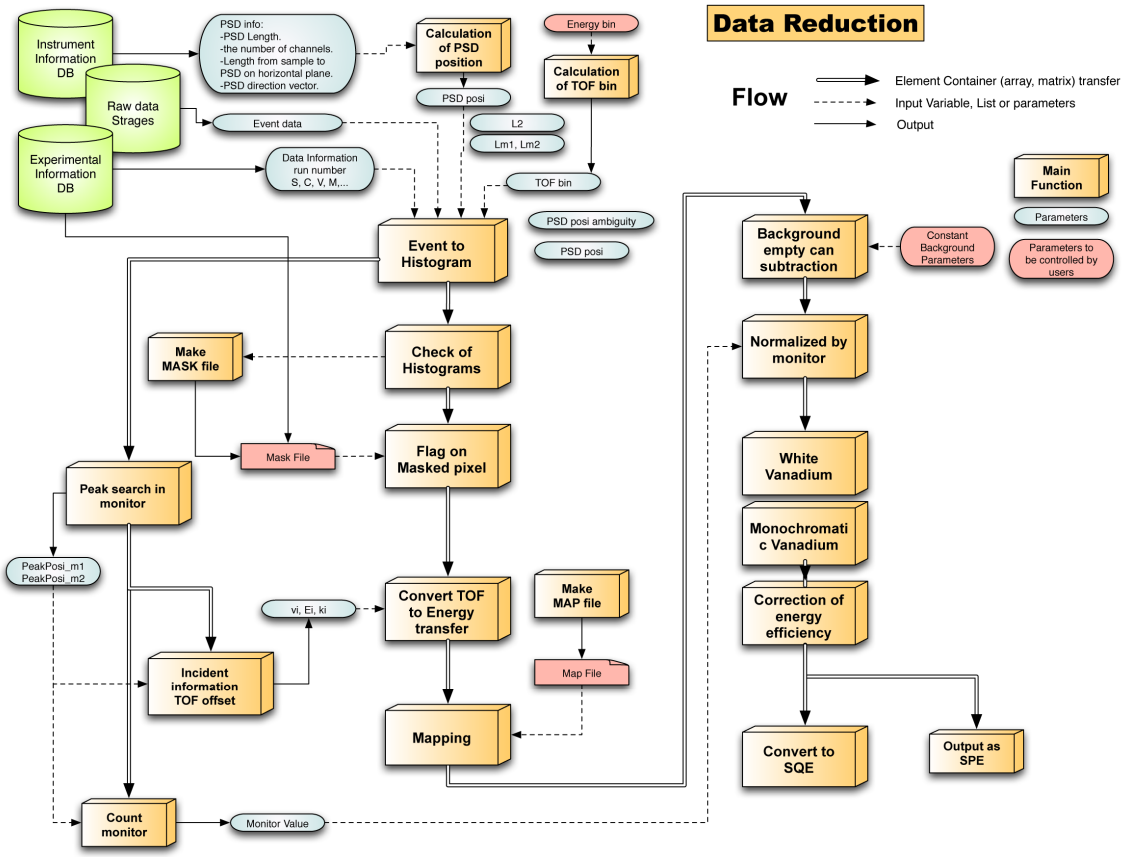


Fig. 14 Structure of basic data reduction software

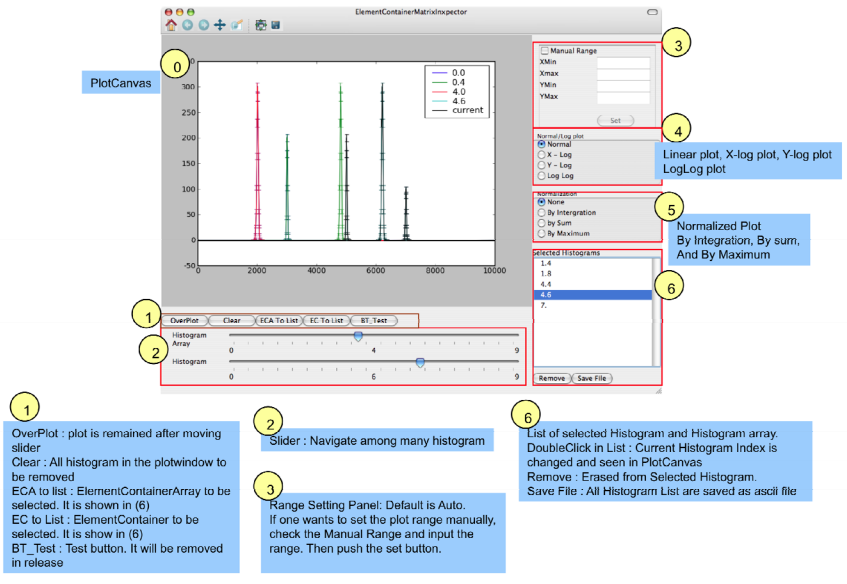


Fig. 15 Figure for Masking and Mapping processes.

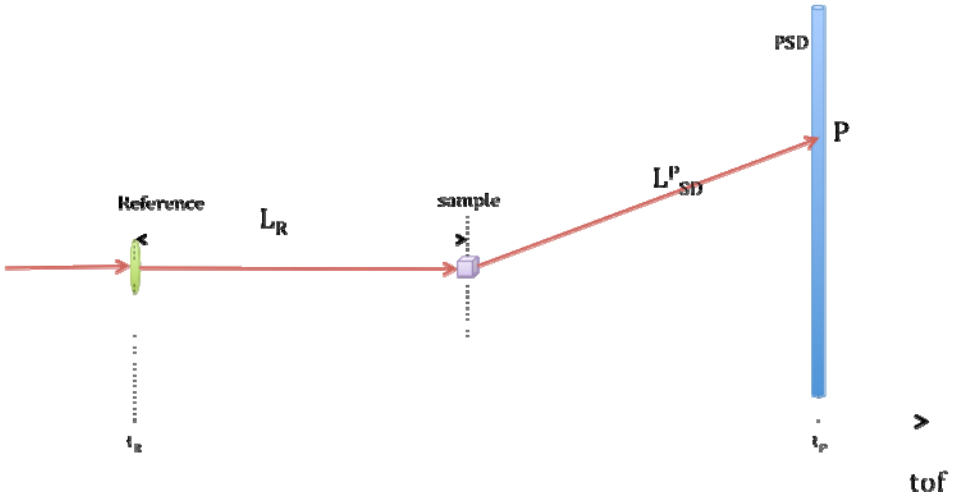


Fig. 16 Schematic diagram for experimental configuration.

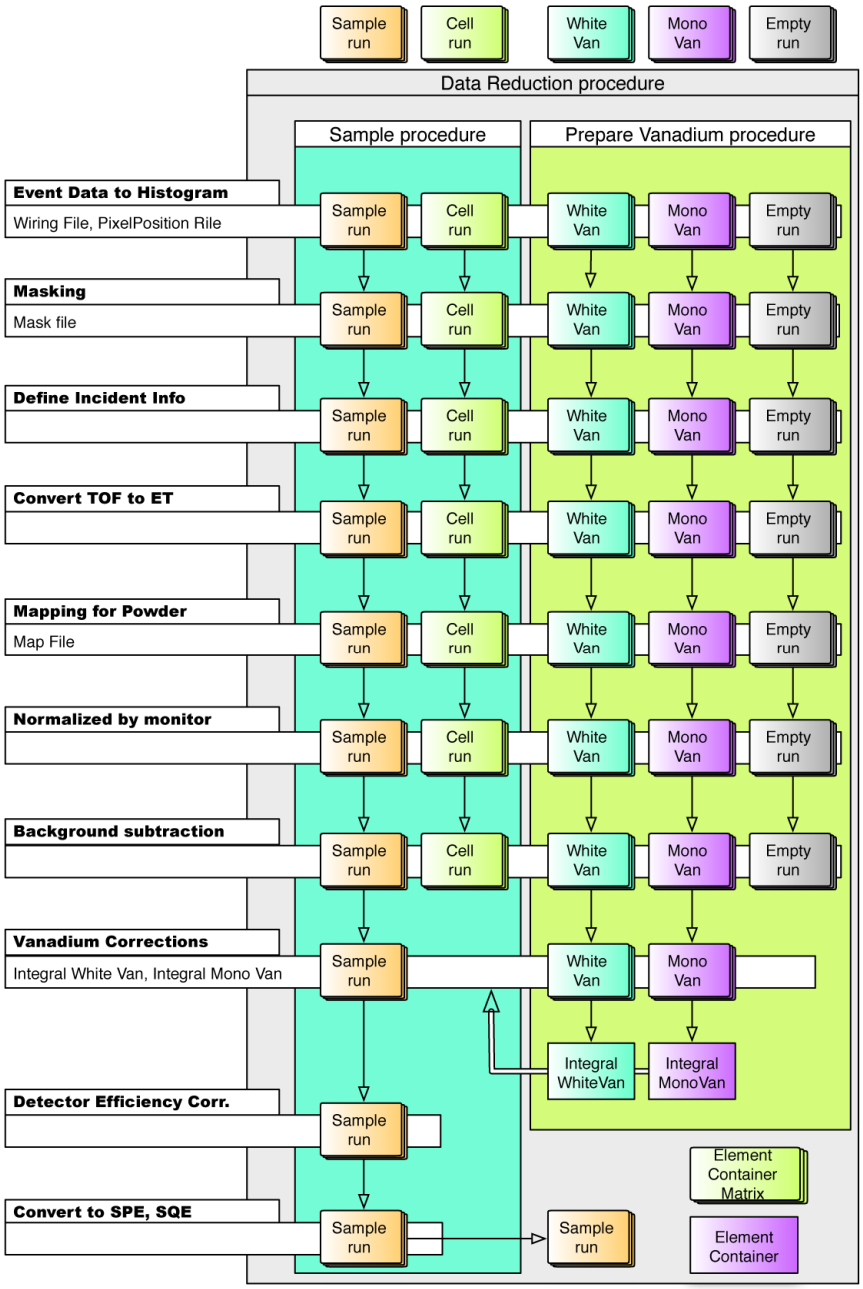


Fig. 17 Diagram of procedures of modules for data reduction process.

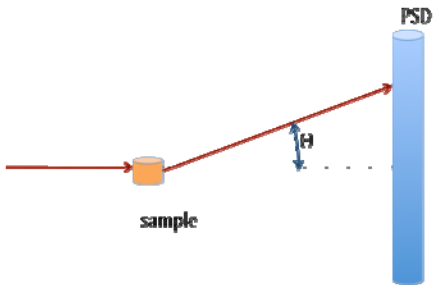


Fig. 18 Detector geometry used in the software.

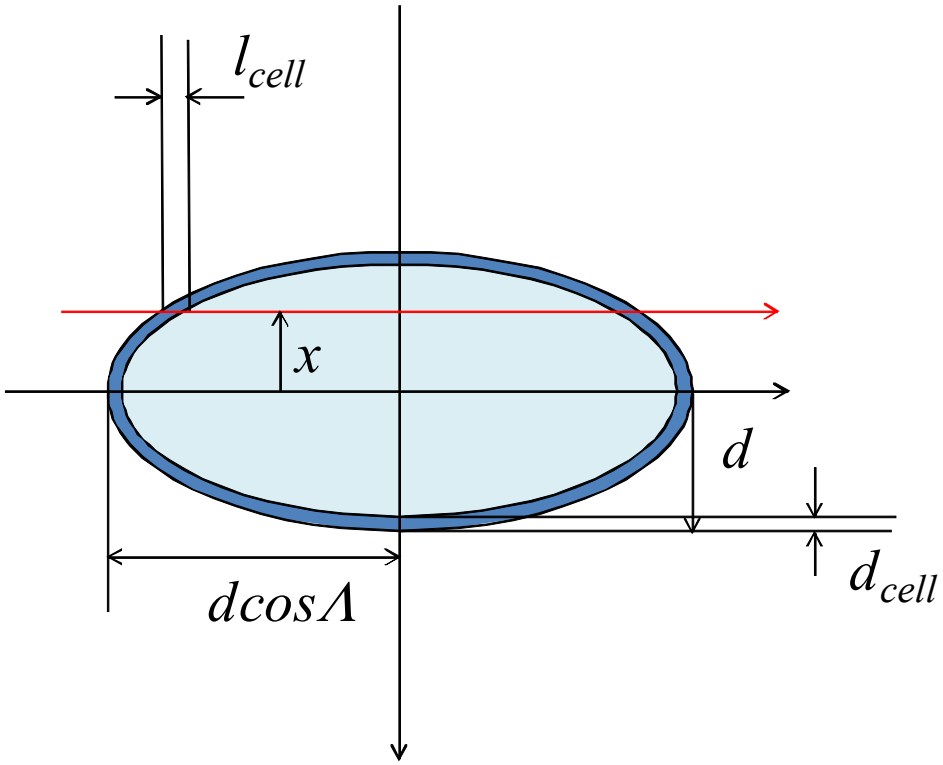


Fig. 19 Diamgram of ^{3}He detectors.

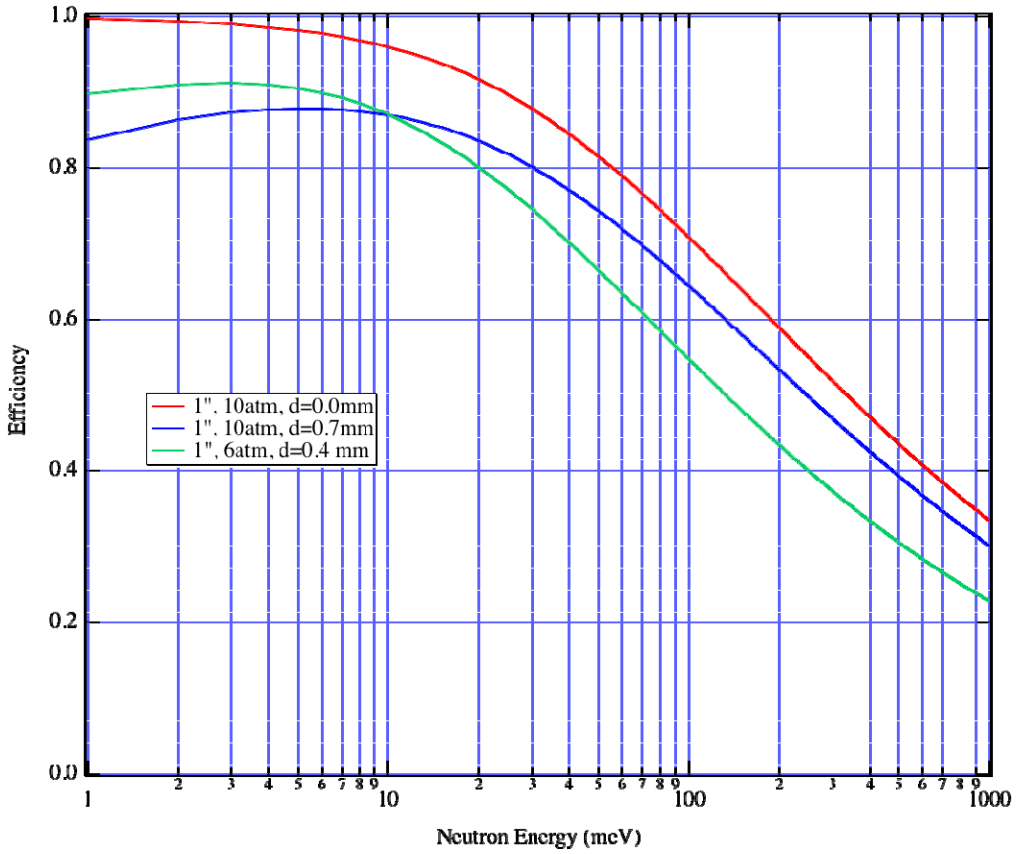


Fig. 20 Detector efficiency calculation results.

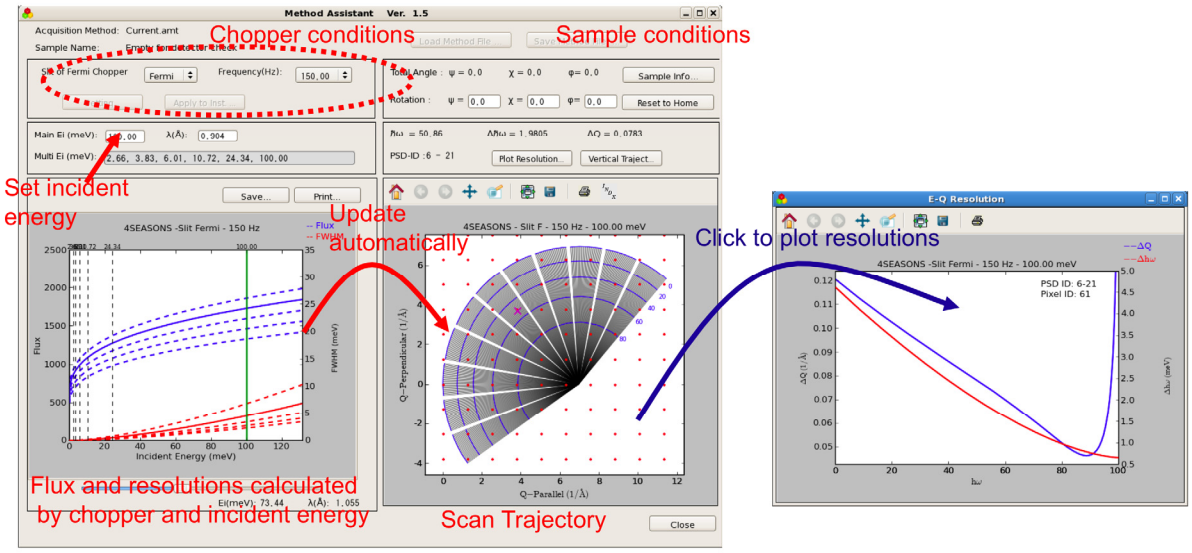


Fig. 21 Screenshots of (left) flux, resolution, and *Q*-coverage for horizontal detectors; (right) *Q* & energy resolutions at the selected position of PSD.

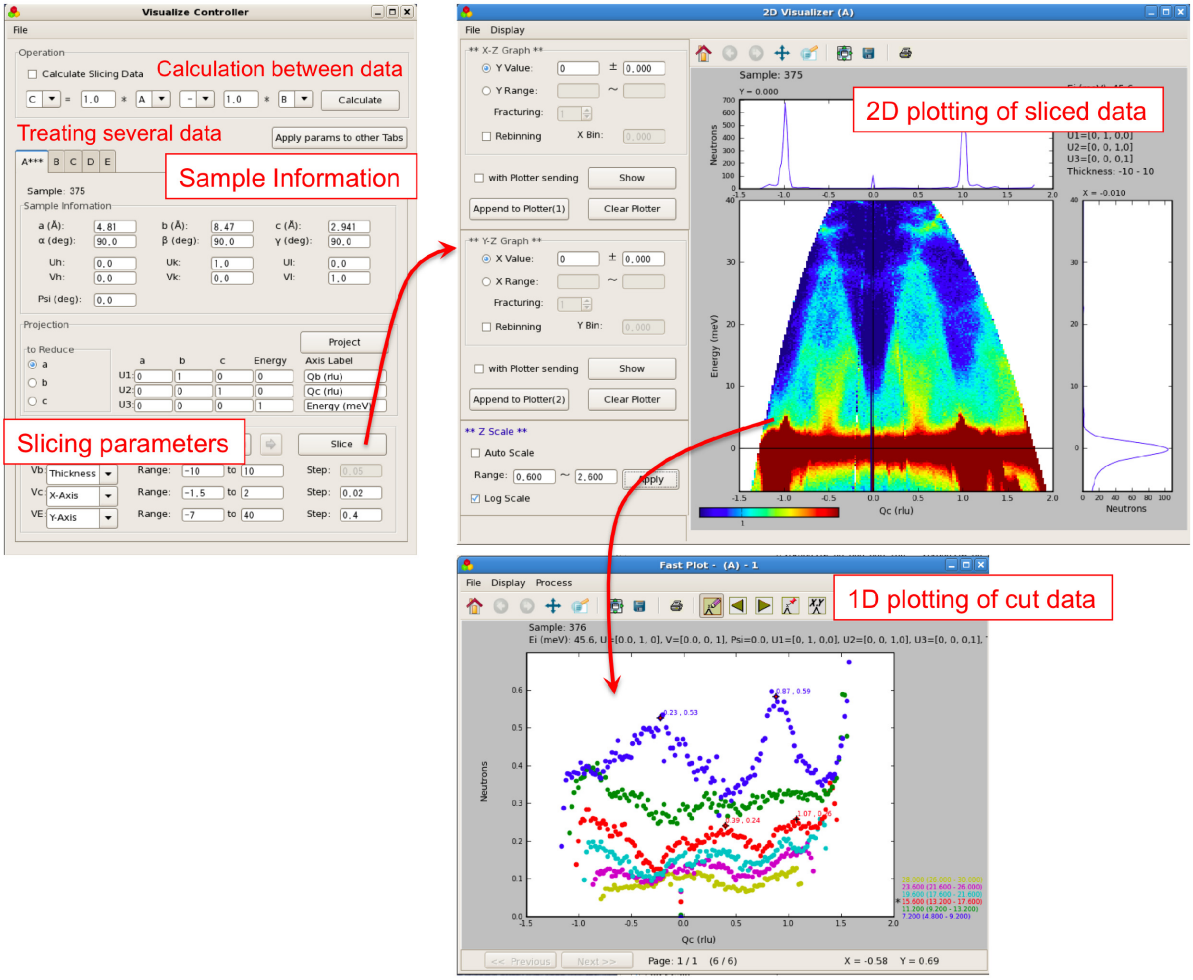


Fig. 22 Screenshot of Data visualization program.

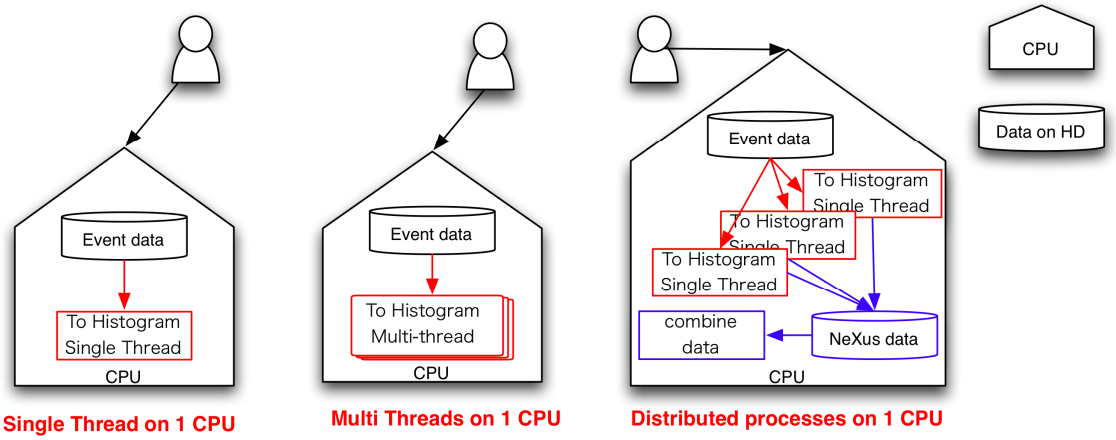


Fig. 23 Diagram for methods of converting event data to histogram.

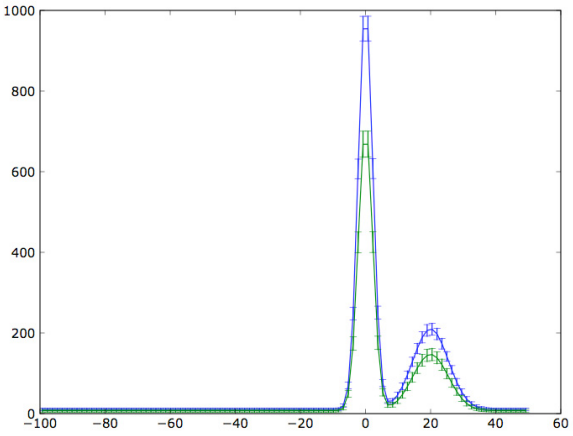


Fig. 24 The blue line is the original spectrum and the green line is subtracted one.

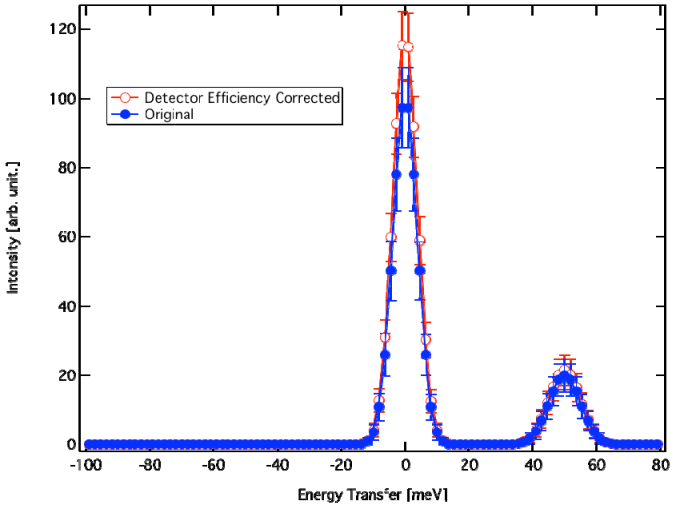


Fig. 25 Data before (open) and after (closed) detector efficiency correction.

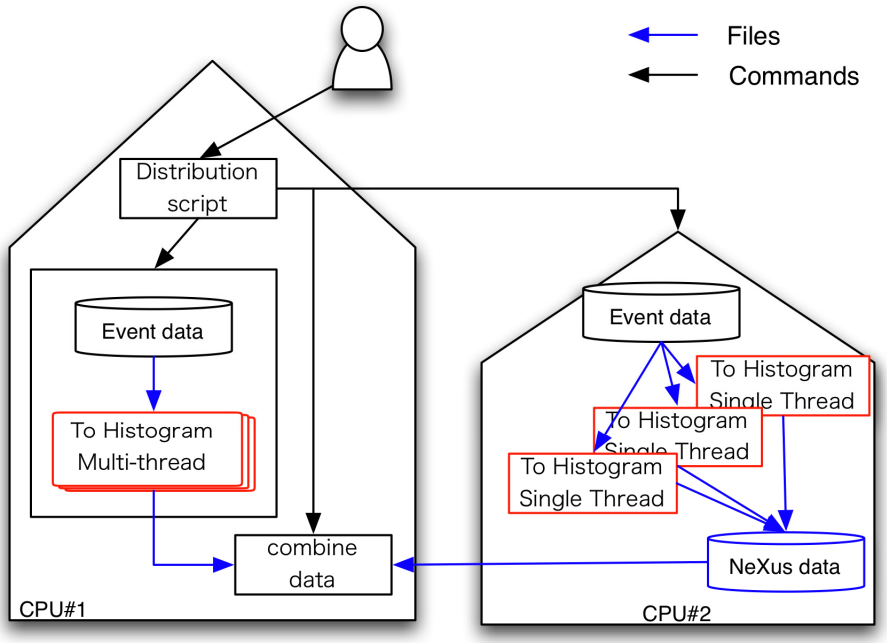


Fig. 26 Structure for the sequence of "Event data To Histogram" step in 4SEASONS. We use 2 PCs: CPU #1 is a host PC to run multi-thread methods and collect the results from distributed processes. CPU #2 runs 4 processes to execute commands distributed from the host PC.

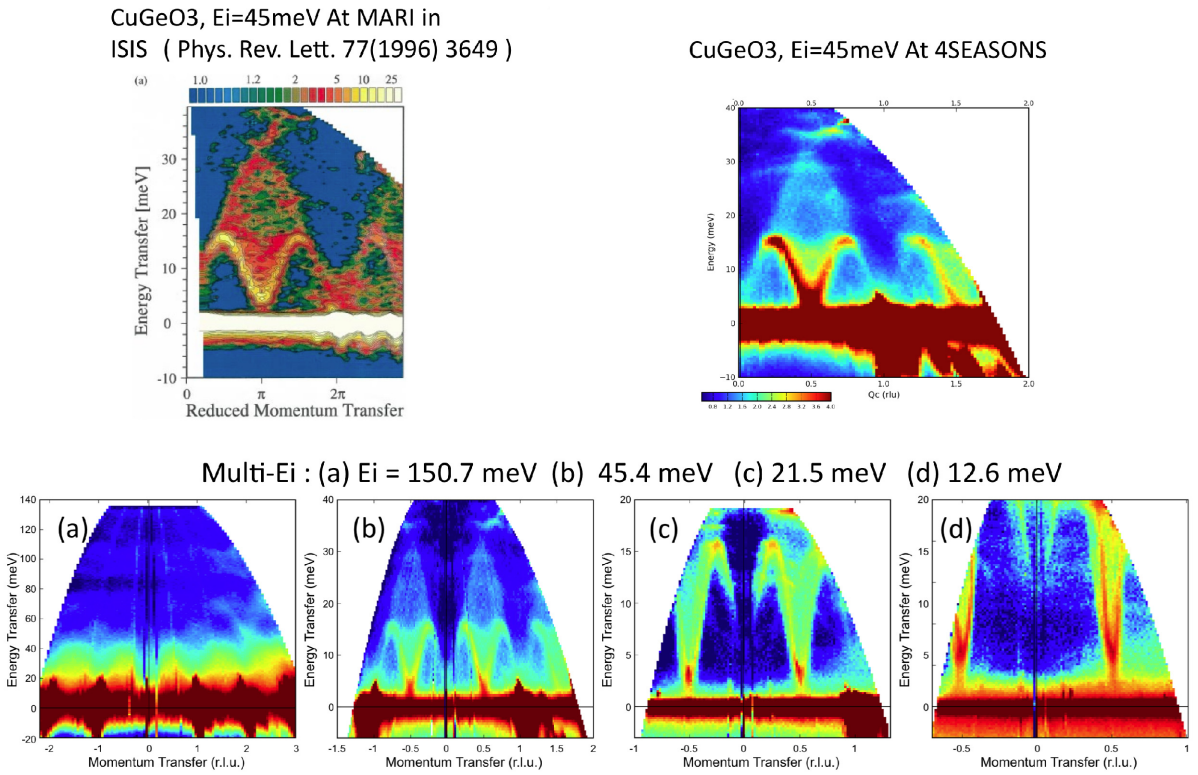


Fig. 27 Data of $CuGeO_3$ taken at 4SEASONS using Data visualization software for different E_i of RRM [14].

Tables

Table 1: The parameters for calculations of "Background subtraction" in data reduction.

Method No.	Parameter	Description
0	Double	Constant background value are given by parameter
1	Int, Int	Constant background is calculated by average of both ends point. The first, last integer parameters are number of points in lower and higher ends.
2	Float, Float	Constant background is the average value calculated from given range.
3	DoubleVector or array	Backgrounds are given by DoubleVector or array with the same size as the ElementContainer.

Table 2: Comparison of "Distributed process" method and "Multi-threading" method in view of new cording to be required.

	New coding in C++	New coding in Python	Multipurpose
Distributed processes	not required	Required	yes
Multi-threading	required	not required	no

Table 3: Details of the results of performance check on condition 1 (Number of Pixels: 9600, Number of ToF bin: 400).

	Single thread	Multi-thread	Distributed	File access
time(sec)	707	210	198	67
memory(MB)	350	300	900	-

Table 4: Details of the results of performance check on condition 2 (Number of Pixels: 9600, Number of ToF bin: 8000).

	Single thread	Multi-thread	Distributed	File access
time(sec)	743	248	302	160
memory(MB)	2400 – 4200	2400 - 4200	3000 - 6000	-

Table 5: Results of performance check of "ToF to energy conversion."

	Python	C++
Step 1 (Conversion)	79.0 sec	0.7 sec
Step 2 (Re-binning)	17.8 sec	15.8 sec

Table 6: Results of performance check using real data.

Reduction step	Time (sec)
Event data to histogram	320
ToF to energy transfer	16
Correction for k_f/k_i	2
Total	338

Appendix

i. Glossary
ii. Records of meetings and personnel exchange
iii. Summary materials of meetings
iv. Choice of coordinates
v. Mathematical background
vi. List of publications and presentations

i. Glossary

4SEASONS:
A Fermi-chopper spectrometer at MLF, J-PARC. Operation has been started from 2008.

AMATERAS:
A disk-chopper spectrometer at MLF, J-PARC. Operation has been started from 2009.

Binning:
Data handling process to sorting the data by specific physical quantity (in time, energy, wave length and etc.) into legs with certain width. Each leg is called 'bin.' By this binning, a set of histogram will be made.

CHOP:
A tool program to calculate energy resolution and neutron flux at sample position for given chopper conditions developed at ISIS.

Chopper:
Device to cut the neutron beam at certain time window. At a chopper spectrometer, a chopper with a short burst time is the key component of the spectrometer to select the incident neutron energy.

Chopper Spectrometer:
One of types of instruments performing inelastic neutron scattering experiments. With pulsed neutron beam (at steady neutron source, a pulse generating chopper is included in an instrument), spectroscopy is measured by ToF technique. At least one chopper is place before the sample to select the incident neutron energy.

CUI:
Character-based user interface.

DC-TOF:
A ToF spectrometer at a cold-neutron guide hall at HANARO. It is expected to start operation from 2010.

Event Data:
As opposed to the conventional data taking in histogram, each event of neutron detections is recorded individually with its own unique time tag.

Framework:
In software engineering, framework means a collection of reusable class and library providing API (Application Protocol Interface), executive files and other resources for the specified purpose. It is also referred as development environment.

GUI:
Graphical user interface.

HANARO:

High-flux Advanced Neutron Application ReactOr. A multipurpose research reactor in Korea constructed and owned by KAERI. Completion of construction is in April of 1995. Operating power is 30MW. From 2003, upgrading program has been lunched and a cold-neutron source and a related experimental hall are under construction.

Homer:

Basic reduction software at ISIS. It converts raw detector data into $S(\phi, \hbar\omega)$ on inelastic scattering measurements. Various corrections depending on instruments are included.

ISIS:

The experimental facility at the Rutherford Appleton Laboratory in United Kingdom providing pulsed muon and neutron beam. The first neutron beam was emerged in 1985. Recently, ISIS has started the operation of 2nd target station (TS2) from 2009 in addition to original 1st target station (TS1) and has increases its proton beam power. Current operation mode is that 240 kW proton beam produces 50 Hz (40 Hz for TS1 and 10 Hz for TS2) pulse neutron beam which is delivered to 18 (TS1) + 18 (TS2) beam ports.

JAEA:

Japan Atomic Energy Agency.

J-PARC:

Japan Proton Accelerator Research Complex. A Japanese joint project of JAEA and KEK. J-PARC has 3 proton accelerators and 4 experimental facilities, one of which is MLF. The project has been started from 2002 and user program has been started from 2008.

KAERI:

Korea Atomic Energy Research Institute.

KEK:

High Energy Accelerator Research Organization.

KENS:

Neutron Science Laboratory of KEK. KENS operated an experimental facility with a spallation neutron source in KEK site, which was closed in 2006.

MLF:

Materials & Life Science Experimental Facility. One of facilities of J-PARC providing pulsed muon and neutron beam for experiments. The neutron source at MLF is 1 MW (in proton beam power) at the final goal with the repetition of 25Hz providing neutrons to 23 beam ports. The first neuron at MLF has emerged in 2008.

Mslice:

Visualization software for inelastic scattering measurement at ISIS. This software is used to display

and visualize data analysed on Homer.

Pixel:

Separated area of a PSD. Minimum detecting area.

PSD:

Position Sensitive Detector.

SNS:

A spallation neutron source facility in Oak Ridge National Laboratory in the United States. Proton beam power is 1.4 MW at the final goal. Repetition rate is 60Hz. Number of beam ports is 18 (some of beam ports are spilt in to 2 beamlines and 24 beamlines are planned). The first neuron at MLF has emerged in 2006.

SWAN:

A small and wide angle scattering neutron instrument which existed at a neutron facility at KEK. The data from SWAN are often used in developments of software including Manyo-lib. and sometimes it acted as a test bed for new data acquisition systems. The spectrometer was decommissioned in 2006 when a neutron facility at KEK was closed.

T_0:

Because of the time structure inherent at spallation neutron sources, the experiment clock is set by the event of proton hitting the target. This time is defined as T_0. In case of DC-TOF in reactor source, master pulse signal from chopper generate T_0.

TAS:

Triple-Axis Spectrometer. One of typical neutron instruments installed at continuous neutron sources. Since it has 3 primal rotating axes for a monochrometor crystal, sample and an analyzer crystal, it is called a triple-axis spectrometer. Versatile instruments performing inelastic, quasi-elastic and elastic neutron scattering experiments. TAS can measure $S(\vec{Q},\hbar\omega)$ precisely with well studied resolution function but measurements on TAS is point-by-pint in $\vec{Q}$-$\hbar\omega$ space (recently, many of TASs with multiple analyzer-detector systems to overcome this disadvantage).

TOBYFIT:

A utility program for simulations and fitting of single crystal neutron scattering data developed at ISIS.

ToF:

Time-of-flight.

ii. Records of meetings and personnel exchange

A. Offline meetings

No	Title	Date	Venue	Participants
1	KJ-Collaboration Kick-off Meeting	24 July, 2007	J-PARC, Tokai, Japan	M.-K. Moon, K.-C. Jin, S.-Y. So, J.-G. Park, M. Arai, K. Nakajima, Y. Inamura, T. Nakatani, T. Otomo, Jiro Suzuki, S. Itoh (J-PARC), T. Yokoo (J-PARC)
2	K-J Software Collaboration Working Week 2007	10-14 September, 2007	J-PARC, Tokai, Japan	S.-Y. So, J.-G. Park, M. Arai, K. Nakajima, Y. Inamura, T. Nakatani, T. Otomo, Jiro Suzuki, K. Tomiyasu (Tohoku Univ.)
3	1st International mini Workshop on Chopper Software Development	10-13 December, 2007	J-PARC, Tokai, Japan	S.-Y. So, J.-G. Park, M. Arai, K. Nakajima, Y. Inamura, T. Nakatani, M. Nakamura, Jyunichi Suzuki (J-PARC), T. Otomo, Jiro Suzuki, S. Itoh (J-PARC), K. Tomiyasu (Tohoku Univ.), M. Takahashi (Tsukuba Univ.), T. G. Perring (ISIS), G. E. Granroth (SNS)
4	K-J Software Collaboration Tokyo Meeting	5 April, 2008	JAEA, Tokyo, Japan	S.-Y. So, J.-G. Park, M. Arai, K. Nakajima, Y. Inamura, T. Nakatani, R. Kajimoto
5	2nd International mini-Workshop on Chopper Software Development	26-27 May, 2008	J-PARC, Tokai, Japan	S.-Y. So, J.-G. Park, M. Arai, K. Nakajima, Y. Inamura, T. Nakatani, R. Kajimoto, K. Shibata(J-PARC), N. Takahashi, T. Aoyagi (JAEA), T. Otomo, T. G. Perring (ISIS), G. E. Granroth (SNS)
6	3rd International mini-Workshop on Chopper Software Development	8-9 December, 2008	Fukuroda, Ibaraki, Japan	S.-Y. So, J.-G. Park, M. Arai, K. Nakajima, Y. Inamura, T. Nakatani, R. Kajimoto, T. Otomo, N. Kawakita (Kyushu Univ.), T. G. Perring (ISIS), G. E. Granroth (SNS)
7	KJ Collaboration Core Meeting	9 July, 2009	IQBRC, Tokai, Japan	J.-G. Park, M. Arai, K. Nakajima

*IQBRC: Ibaraki Quantum Beam Research Center


K-J Software Collaboration Tokyo Meeting (5 April, 2008, JAEA, Tokyo, Japan)

K-J Software Collaboration Tokyo Meeting (5 April, 2008, JAEA, Tokyo, Japan)


2nd International mini-Workshop on Chopper Software Development (26-27 May, 2008, J-PARC, Tokai, Japan)

B. Online meetings

No	Time / Date	Participants
1	10 August, 2007	J.-G. Park, J.-Y. So, Y. Inamura, K. Nakajima *There is no record about other participants.
2	10:00-12:00, 5 September, 2007	J.-G. Park, J.-Y. So, Y. Inamura, K. Nakajima *There is no record about other participants.
3	10:30-11:30, October 9, 2007	J.-G. Park, J-Y. So, M. Arai, Y. Inamura, T. Nakatani, K. Nakajima
4	13:25-14:45, November 14, 2007	J.-G. Park, J-Y. So, Y. Inamura, T. Nakatani, K. Nakajima
5	13:45-15:30 January 22, 2008	J.-G. Park, M. Arai, Y. Inamura, T. Nakatani, K. Nakajima
6	10:00-11:30 February 13, 2008	J.-G. Park, M. Arai, Y. Inamura, T. Nakatani, K. Nakajima, J.-Y. So
7	13:30-15:00 March 13, 2008	J.-G. Park, J.-Y. So, M. Arai, Y. Inamura, Jiro Suzuki, K. Shibata, K. Nakajima
8	13:30-14:50 April 21, 2008	J.-G. Park, J.-Y. So, R. Kajimoto, Y. Inamura, K. Nakatani, K. Nakajima
9	13:30-15:00 May 19, 2008	J.-G. Park, J.-Y. So, R. Kajimoto, Y. Inamura, K. Nakajima
10	16:00-17:30, July 29, 2008	J.-G. Park, J.-Y. So, M. Arai, T. Nakatani, R. Kajimoto, Y. Inamura, K. Nakajima
11	10:00-12:00, August 25, 2008	J.-G. Park, J.-Y. So, M. Arai, T. Nakatani, R. Kajimoto, Y. Inamura, K. Nakajima
12	15:00-17:00, September 18, 2008	J.-G. Park, J.-Y. So, Y. Inamura, K. Nakajima *There is no record about other participants.
13	17:15-18:15, November 27, 2008	J.-Y. So, M. Arai, T. Nakatani, Y. Inamura, K. Nakajima
14	14:05-15:15 April 2, 2009	J.-G. Park, J.-Y. So, M. Arai, R. Kajimoto, Y. Inamura, K. Nakajima
15	13:30-14:30 June 29, 2009	J.-Y. So, R. Kajimoto, Y. Inamura, T. Nakatani, K. Nakajima
16	10:00- July 22, 2009	J.-G. Park, J.-Y. So, M. Arai, R. Kajimoto, Y. Inamura, T. Nakatani, K. Nakajima

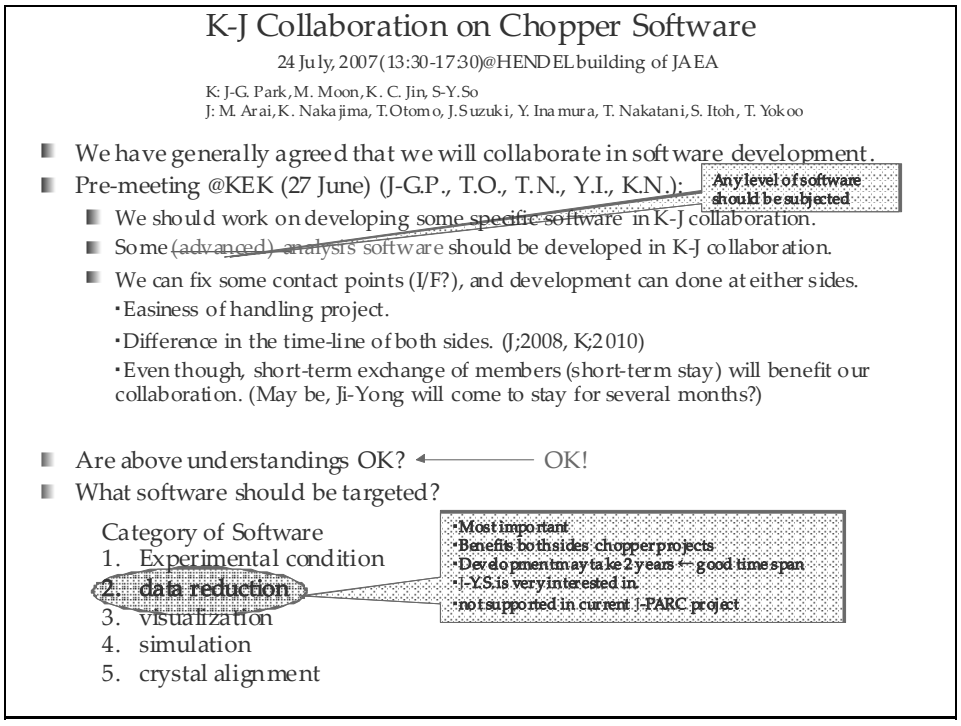
C. Exchange of staff supported by J-PARC & KAERI

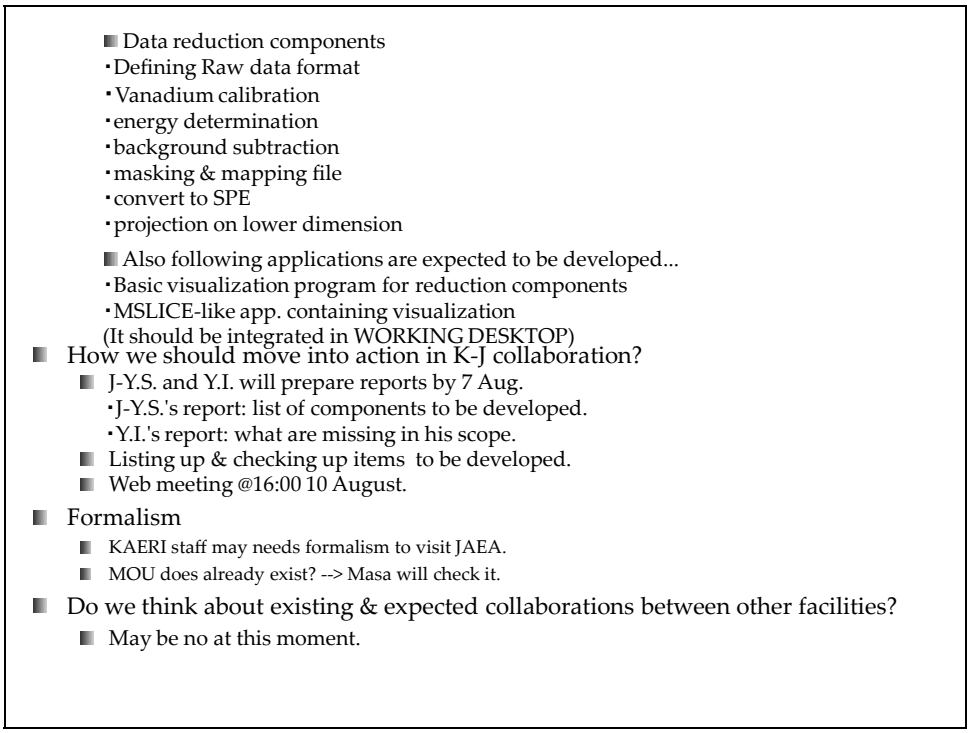
Ji-Yong So (KAERI) visited to J-PARC: 9-15 September, 2007
Ji-Yong So (KAERI) visited to J-PARC: 9-15 December, 2007
Yasuhiro Inamura (JAEA) visited to KARI: 19-22 February, 2008
Ji-Yong So (KAERI) visited to J-PARC: 25-29 May, 2008
Ji-Yong So (KAERI) visited to J-PARC: 7-15 December, 2008

iii. Summary materials of meetings

Here we give show materials used at the summary sessions of offline meetings. These materials describe summary of discussion we had during meetings.

A. KJ-Collaboration Kick-off Meeting
Date: 24 July, 2007
Venue: J-PARC, Tokai, Japan





B. K-J Software Collaboration Working Week 2007
Date: 10-14 September, 2007
Venue: J-PARC, Tokai, Japan

K-J Software Collaboration Working Week 2007

Sep. 10 - 14, 2007 at JAEA

Reported by Kenji Nakajima, Ji-Yong So and Yasuhiro Inamura

Participants

- **M. Arai**
- **Je-Geun Park**
- **T. Otomo**
- **T. Nakatani**
- **J. Suzuki**
- **K. Tomiyasu**

Contents

1

Draft Schedule of K-J Collaboration Week '07
(Sept. 10-14, 2007)

	Monday	Tuesday	Wednesday	Thursday	Friday
9:00-10:00					**Major Meeting IV Summary**
10:00-11:00	**Major Meeting I** -Schedule (Agenda) -Ground Design of MLF Software (by T.O.) -Concept (by Y.I. & ALL)	Daily Meeting I	Working Time (J-Y. S)	Daily Meeting II	
11:00-12:00					
12:00-13:00	Lunch	Lunch	Lunch	Lunch	Lunch
13:00-14:00	**Major Meeting II**	Working Time (J-Y.S, Y.I., K.N.)	Working Time (J-Y.S, Y.I., K.N.)	Working Time (J-Y.S, Y.I., K.N.)	
14:00-15:00					
15:00-16:00			**Major Meeting III**		
16:00-17:00	Working Time (J-Y.S, Y.I., K.N.)				
17:00-18:00					

*** Major Meeting will call All Major Participants.**
*** All Major Meetings will be held at the Small Meeting Room on the 2nd Floor of HENDEL BLDG.**
*"Working Time" includes i) practical work, ii) study on J-side materials (Manyo-lib., Frame-Work, KCS results & etc.) and iii) extraordinary meetings.

*J-side's fixed schedule: i) 13:30-15:30 of Sept. 11: Commissioning Task Meeting, ii) 10:00-12:00 of Sept. 11: Chopper Task Meeting, iii) 15:30-17:30 of Sept. 12: Grand Design Meeting, iv) 10:00-12:00 of Sept. 14: Neutron Science Section Meeting, v) 16:

2

K-J Software Collaboration Working Week 2007

Meetings

4 Major Meetings & 2 Daily Meetings

Major Meeting I & II

10 Sept., 2007
10:00-12:00, 13:30-16:00
J-Y.S., Y.I., M.A., J-G.P., T.O., T.N., J. S., K.T., K.N.

- Frame Work Architecture (T.O.) [1]
- Image of Our Chopper Software [2]
  - >> What Components should be added to FW?
    - >> S-Xtal Arraignment, Decision Makign & Exp. Optimization, Infromative Visualization, Adv. MSLICE.....
- etc.

Tuesday's Daily Meeting

11 Sept., 2007
10:30-12:00
J-Y.S., Y.I., M.A., J-G.P., T.O., K.N.

- Where required components should be accommodate in FW
  - -> almost software are covered by FW but need checking whole functions [2]
- Discussion about 'Standard Window' -> discussion after
- J-Y.S. & Y.I. started to summarize the inside of each components (functions).[3]
- We confirmed that Discussion about Concept might be over.

3

Wednesday's Daily Meeting

12 Sept., 2007
9:15-10:20
J-Y.S., Y.I., M.A., J-G.P., T.N., K.N.

- Brief look at J-Y.S. & Y.I.'s works
- >> Show Flow diagram for data reduction part [4]
- >> add "Estimate resolution to decide time-binning" function

Major Meeting III

12 Sept., 2007
17:00-19:00
J-Y.S., Y.I., M.A., J-G.P., T.O., T.N., K.N.

- What should be done by Friday? -> refer to "Future…." page
- need to discuss about Event Data Format
- Data Reduction & Analysis Component
  We have to add some functions (Resolution Calc., Channel Width Decision...)
  Event Data Treatment
  Background Trick...
- Show flow diagram for Single Crystal Alignment software
  We need functions for "peak search", "fitting whole PSDs data"
  try to search the results of Single Crystal group in J-PARC
- Collaboration between ISIS
- Man Power -> core development members are J-Y.S., Y.I, Minakawa(KCS).
- Schedule of monthly TV-meetings & real meetings -> refer to "Future…" page
- Communication Tools (SW Development Env.)
  -> We can use environments prepared by MFL Comp. Group. (T.O.)

4

- Thursday's Daily Meeting

13 Sept., 2007
15:30-16:30
J-Y.S., Y.I., M.A., J-G.P., K.N.

- We asked J-Y.S. & Y.I. to prepare the material for the final meeting.
  List or Map of SW items with allocation, priority, due dates....

- Major Meeting IV

14 Sept., 2007
9:30-10:45
J-Y.S., Y.I., M.A., J-G.P., T.N., J.S., K.N.

- What SW do we need; listed up them by J-Y.S. & Y.I. [5]

five software :
1) Homer-like Software
2) Informative visualization for decision of conditions
3) Single crystal alignment software
4) MSLIC-like software
5) Decision making and optimization

- Functions, due date, allocation, priorities of Homer-like SW part
- How to manage future works.
  -> refer to "Future…" page 4
- Discussion & Inquirer of necessary information for J-Y.S. & Y.I.'s actual work
  -> refer to "Future…" page 4

5

Future...

- Job-allocation & Schedule

We will start from HOMER-like part
Detail is Listed by J-Y.S. & Y.I. and discussed
Informative visualization: Done by KCS (Y.I.)
S-Xtal alignment: Y.I. will take initiative
Decision making & optimization: Difficult. Done by task team. (K.N.)
MSLICE: need study & discussion, ISIS people will be involved (M.A.)
We need manager!! K.N. will watch J-Y.S. & Y.I.' s progress
*Mac will be used in Working-level KJ Collaboration (*MLF supports Intel Mac only)*

- How to communicate each other

We can use development environments prepared by MFL comp. G. (T.N.)
Web-server, mailing-list, software repository...
Of course, we have to prepare the contents by ourselves!!

- Monthly TV-meetings

In the Afternoon On Every 1st Wednesday (K.N.)
Time should be determined for each meetings
We'd better to have caretaker of the meetings. Who?

- Real Meetings

We will have 2 Real Meetings before Day-1 of J-PARC. (K.N.)
1st: 10th week of next December 2007
2nd: April 2008

6

References

[1] Frameworks Architecture (T. Otomo) … p8

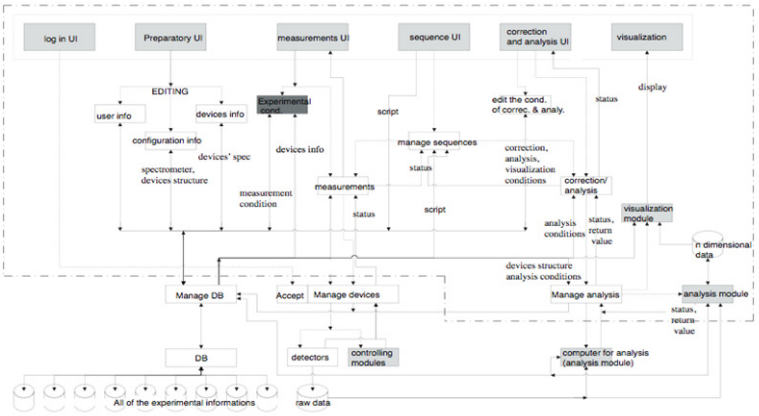
[2] Image of our chopper software (Y. Inamura)
… p9 : “User Login”, “ Spectrometer Control and monitor of devices”,
“Supports for decision of experimental Conditions”, “Sequence Control”
… p10 : “RAW data to SQE on Manyo-Lib”, “Analysis and Visualization”

[3] Listed functions (JY.So, Y. Inamura)
… p11 : required information
… p12 : Function list for data reduction and analysis
… p13 : Virtual experiments and optimization, Single crystal alignment

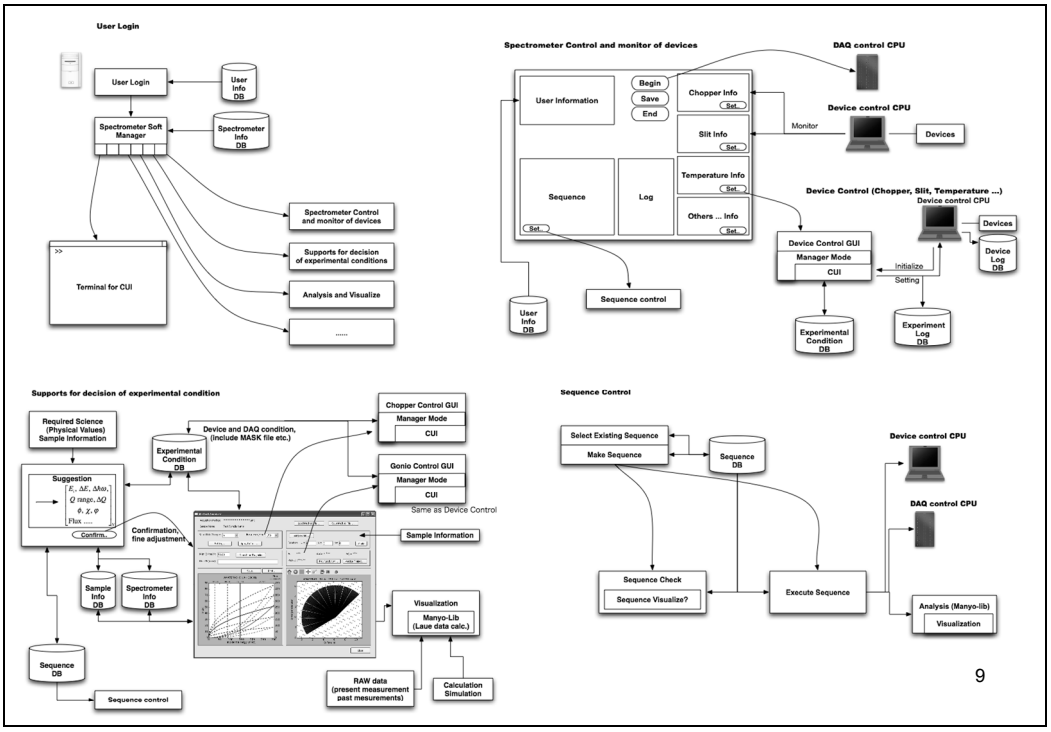
[4] Diagram and data flow for data reduction and single crystal alignment (JY.So, Y. Inamura)
… p14 left : Diagram for data reduction
… p14 right : Diagram for single crystal alignment

[5] What SW do we need in J-K collaboration (JY.So, Y. Inamura)
… p15 : List of software developed for J-K collaboration
… p16-17 : Pseudo code for data reduction functions in developer’s view
… p18 : Allocation of data reduction functions

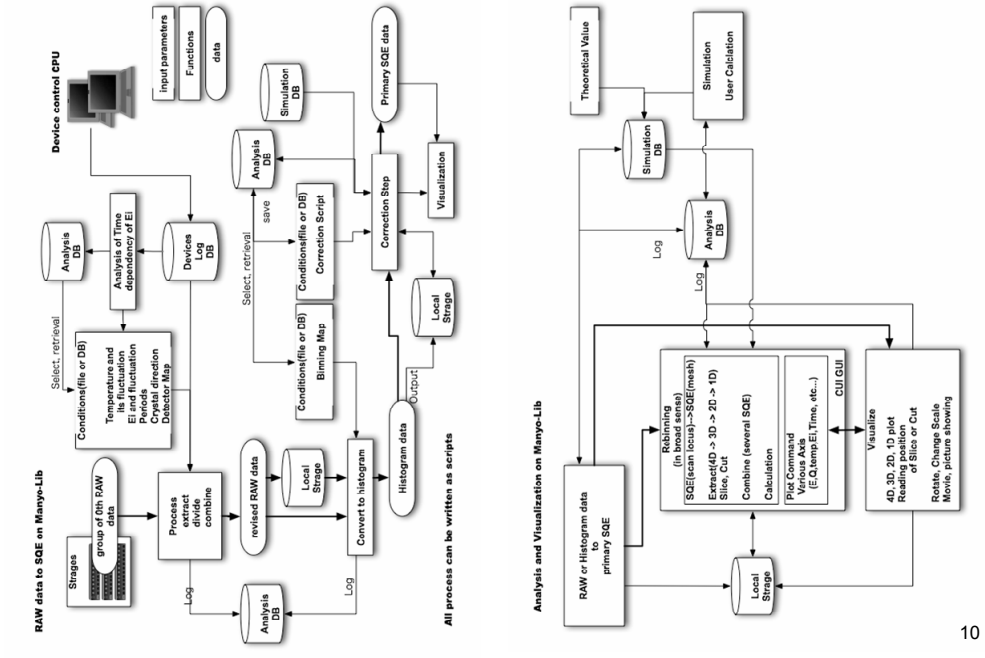
7



8


User Login
User Login
User Info DB
Spectrometer Soft Manager
Spectrometer Info DB
Spectrometer Control and monitor of devices
Supports for decision of experimental conditions
Analysis and Visualize
.....
Terminal for CUI
Spectrometer Control and monitor of devices
DAQ control CPU
User Information
Begin
Save
End
Chopper Info
Slit Info
Temperature Info
Others ... Info
Set.
Monitor
Device control CPU
Devices
Sequence
Log
Device Control (Chopper, Slit, Temperature ...)
Device control CPU
Devices
Device Log DB
Device Control GUI
Manager Mode
CUI
Initialize
Setting
Sequence control
User Info DB
Experimental Condition DB
Experiment Log DB
Supports for decision of experimental condition
Required Science (Physical Values) Sample Information
Device and DAQ condition, (include MASK file etc.)
Experimental Condition DB
Suggestion
Confirm.
Confirmation, fine adjustment
Chopper Control GUI
Manager Mode
CUI
Gonio Control GUI
Manager Mode
CUI
Same as Device Control
Sample Information
Sample Info DB
Spectrometer Info DB
Visualization
Manyo-Lib (Laue data calc.)
Sequence DB
Sequence control
RAW data (present measurement past measurements)
Calculation Simulation
Sequence Control
Select Existing Sequence
Make Sequence
Sequence DB
Device control CPU
DAQ control CPU
Sequence Check
Sequence Visualize?
Execute Sequence
Analysis (Manyo-lib)
Visualization

9


Device control CPU
Input parameters
Functions
data
Simulation DB
Primary SQE data
Analysis DB
Visualization
save
Select, retrieval
Analysis DB
Analysis of Time dependency of Ei
Devices Log DB
Conditions(file or DB) Correction Script
Correction Step
Local Strage
Select, retrieval
Conditions(file or DB)
Temperature and Ni monitor
Ei and fluctuation
Periods
Crystal direction
Detector Map
Conditions(file or DB) Binning Map
Output
RAW data to SQE on Manyo-Lib
group of 0th RAW data
Process extract divide combine
revised RAW data
Local Strage
Convert to histogram
Histogram data
Strages
Log
Analysis DB
Log
All process can be written as scripts
Theoretical Value
Simulation User Calclation
Simulation DB
Analysis DB
Log
Log
Rebinning (in broad sense) SQE(ecan locus)->SQE(mesh)
Extract(4D -> 3D -> 2D -> 1D) Slice, Cut
Combine (several SQE)
Calculation
Plot Command Various Axis (E,Q,temp,Ei,Time, etc...)
CUI GUI
Visualize 4D, 3D, 2D, 1D plot Reading position of Slice or Cut
Rotate, Change Scale Movie, picture showing
Analysis and Visualization on Manyo-Lib
RAW or Histogram data to primary SQE
Local Strage

10

Information required for functions

Instrument Information	
Instrument Name	
Primary Flight Path	
Secondary Flight Path	
Location of the Chopper	
Detector Geometry	
Chopper Geometry	Radius, Slit width, Slit Phase
Other Instrument Informations	
Experiment Information	
User Information	Name, Affiliation, ID number
Local Contact Name	
Sample Information	Mass, Elements, Volume, Shape, Orientation, Lattice Parameters, Cross-section etc
Time	
Proton beam current(?)/Monitor Count	*for the estimation of flux*
Real time information of accelarater(mode)	*moderator condition(tempereture)*
Run Numbers	
Chopper Configuration	Chopper Speed, Phase
Assumed Incident Energy	
Temperature	
External Field	
Pressure	
Polarized Neutron Conditions	
Other Experimental Conditions	

11

Function List for Reduction and Analysis

	Funtion List	Input	Output
Before Experiment	Virtual Experiment		
	Condition Optimization(Ei, Resolution, Flux etc)		Scripts
	Determine Experiment Condition		Scripts
	Single Crystal Alignment		Scripts, parameter files
Experiment			RD, INI, EXI, Log files
	Incident Energy Correction(Determine Precise Ei, Considering RRM)	RD,INI,EXI	
	Make Histogram(Considering RRM)	USI, RD, INI, EXI, Log files	HIST0
After Experiment	TOF correction [tof align, t to E conversion]	USI, HIST0, INI, EXI	HIST1
	Make Mask File *[Decision of Good Detectors]*		Masking file
	Make Map File		Map file
	Detector Efficiency Correction		
	Vanadium Correction		
	Background Subtraction		
	Self Shielding Correction		
	Transmission/Absorption Correction		
	Multiple Scattering Correction		
	Absolute Value Normalization		
Make SPE	Make SPE		SPE file
Make SQE	Make SQE		SQE file

RD=Raw Data, INI=Instrument Information,
EXI=Experiment Information, USI=User Input,
HIST=Histogram

12

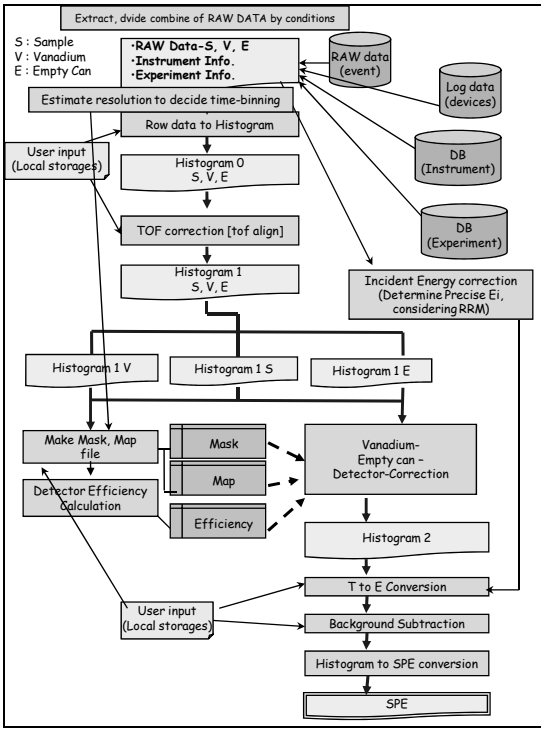
Virtual Experiments and optimization

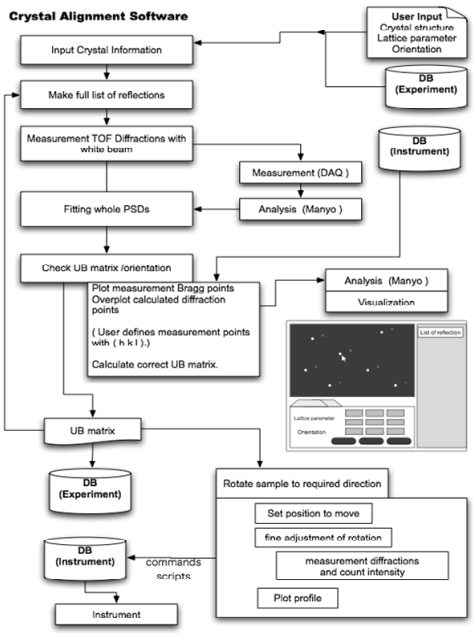
Function List	Input	Output
Calculate Energy, Momentum Range	INI, USI	
Calculate Flux/Resolution	INI, USI	
Calculate Momentum Range in Reciprocal Space [Q, (hkl)]	INI, USI, Sample Informations	
Listing Possible Experimental Conditions	INI, USI, Sample informations	Chopper condition for Experiments

Single Crystal Alignment and Check

Function Lists	Inputs	Outputs
Calculate UB matrix and refinement		
Calculate goniometer angles to measure specified points in Momentum Space		
Calculation and Visualization of Momentum Range [(hkl)]		
Making Script for Goniometer Control		
Comparing Expecting and Experimental Results(Overplot)		

13





14

J-K collaboration for software

[1] HOMER-like software

1. List up required functions
2. Diagram for relations between functions
3. Allocate functions and start development

[2] Informative visualization (for decision of conditions)

Already started with company for first version.

[3] Single crystal alignment software

1. List up required functions
2. Diagram for relations between functions but not enough.
3. Shall start to search and contact with Single Crystal group's software

[4] MSLICE-like software

1. Start list up required functions
2. Need training and experiences to analysis real data of single crystal

[5] Decision making and optimization

1. This is heavy task. If we want, we need to make task team.
2. Need concept, need people who have experiences about real experiments

15

```
#! /usr/bin/env python

############################################################
# This is not executable python file.
# Just Pseudo code
# For Considering Functions required for Data Reductions
# Made by Y. Inamura and J-Y So
############################################################

import Manyo
import GUImodule

class PSDdata( object ):
    """ class for the raw data of single PSD"""

    def __init__(self):
        ID_detector
        Position=[]
        Tof=[]

class Rawdata(object):
    # Included in Manyo Library
    def __init__(self):
        Datas = [ PSDdata_class ]

    def showGUI(self):
        # show Raw data and information using GUImodule
        return


class Container_RawData(object):
    # Included in Many Library
    def __init__(self):
        Datas=[ list_of_Rawdata_class ]
        Info_Instruments={}
        Info_Experiments={}

    def Read_Raw_Data( self, raw_data_file ):
        """read raw data and save it in container class"""
        return 0

    def Read_Instrument_Informations( self, instrument_information_file ):
        """read instrument information file : XML and save it in container class"""
        return 0

    def Read_Experiment_Informations( self, experiment_information_file ):
        """read Experiment information file(XML) and save in in container class"""
        return 0

    def Convert_to_Histogram( self, external_parameter ):
        """
        convert raw data to Histogram Data
        It needs some external parameters
        """
        return Container_Histogram_Class

    def showGUI(self):
        # show Raw data array using GUImodules
        return

class Container_Histogram(object):
    # basic functions are defined in Manyo Library

    def __init__(self):
        Datas=[list_of_Histogram]
        Info_Instruments={}
        Info_Experiments={}

    def showGUI(self):
        # show Histogram using GUI modules
        return

    def __add__(self, ops):
        return

    def __sub__(self, ops):
        return

    def __mul__(self, ops):
        return

    def __div__(self, ops):
        return

    def Do_TOF_Corrections( self, some_parameters ):
        """
        tof_corrections
        """
        # shift and align tof spectrum according to its position, Ei, etc
        return

    def Make_Mask_Files( self, some parameters):
        """
        Make Mask Files

        """
        # It Requires GUI

        return Make_Informations
```

16

```
def Normalizations(self, val):
        """
        make itself as  Normalized_Histogram
        """
        return 0

    def Time_to_Energy_Conversions(self, parameters , Ei, Mode):
        """
        perform time to energy conversions
        """
        if Mode == "Tof_Aligned":
            # do t2e conversion with previous tof-aligned process

        else :
            # do t2e conversion without previous tof-alignment process
            # In this case, t_0(tof of elastic peak center) is different among PSDs
            # Then, we should have informations about each PSD's t_0
        return 0

    def Background_Corrections(self, parameters_background):
        #
        # In first phase, just consider constant background
        # after beam coming, we must consider non-linear background
        #
        return 0

def Determine_Incident_Neutron_Energy( Informations of Monitor)
    # Calculate Incident Neutron Energy
    return Ei

def Read_Datas_From_DB( some_parameters )
    get_RawDataFile_From_DB( run_number)
            # Raw data Files : Sample , Vanadium, Empty Can, etc Measurements,
    get_InstrumentInformations_From_DB( instrument_name)
    get_ExperimentsInformations_From_DB (run_number )
    get_ExperimentsLog_From_DB (run_number )

    return Container_RawData_Class

def Make_Map_Files( some_parameters) :
    """
    Make Map Files
    """
    return Map_Informations

def Make_Detector_Efficiency( some parameters ):
    """
    make detector efficiency tables
    using White Vanadium Data
    """
    return Efficiency_Table
```

```
def Empty_Can_Corrections( Hist0, Hist_E):
    # code for Empty can corrections
    Result_Hist= Hist0-Hist_E
    return Result_Hist

def Detector_Efficiency_Corrections ( Histograms, Masks, Detector_Efficiency, parameters ):
    Result_Hist = Correct_Detector_Efficiency_Using_Histogram_Mask_parameters
    return Result_Hist

def Absolut_Normalize(Hist_Sample, Hist_Vanadium):
    # code for Absolute Normalization
    return Normalized_Hist

def MakeSPE(Histogram, Mapfiles):
    SPE = Manyo.ElementContainer()
    # calculate SPE from Histogram
    return SPE_Datas
```

```
############################################################
# from this, codes are commands
############################################################
Ei = Calculate_Incident_Neutron_Energy ( monitor_info)
RawData_Sample= get_RawData_From_DB( sample_run_number)
RawData_Vanadium= get_Raw_Data_From_DB( vanadium_run_number)
RawData_Empty = get_Raw_Data_From_DB(empty_can_run_number)
Hist_S= RawData_Sample.Convert_to_Histogram( parameters )
Hist_V= RawData_Vanadium.Convert_to_Histogram( parameters )
Hist_E= RawData_Empty.Convert_to_Histogram( parameters )

Hist_S.Do_TOF_Correnctions( *args )
Hist_V.Do_TOF_Correnctions( *args )
Hist_E.Do_TOF_Correnctions( *args )

Masks = Hist_V.Make_Mask_Files( *args)
Maps = Make_Map_Files( *args)
Effs = Make_Detector_Effiency( *args )

Hist_S1= Empty_Can_Corrections(Hist_S, Hist_E)
Hist_S2= Detector_Efficiency_Corrections(Hist_S1, Masks, *args)
Hist_S2.Time_to_Energy_Conversions( *args, Ei)
Hist_S2.Background_Correction( *args)
Hist_S3= Absolute_Normalize(Hist_S2, Hist_V, Effs)

SPE = Make_SPE(Hist_S2, Maps)
```

17

Functions	rank	Who	Due time
Do_TOF_Corrections	2	J	2008.01
Make_Mask_Files	3	K	2008.01
Normalizations	1	J	2008.01
Time_to_Energy_Conversions	1	K	2008.01
Background_Corrections	1	K	2008.02
Determine_Incident_Neutron_Energy	3	J	2008.02
Read_Datas_From_DB	2	J	2008.02
Make_Map_Files	3	K	2008.02
Make_Detector_Efficiency	2	K	2008.02
Empty_Can_Corrections	1	J	2008.03
Detector_Efficiency_Corrections	2	K	2008.03
Absolut_Normalize	2	J	2008.03
MakeSPE	3	JK	2008.04

18

C. 1st International mini Workshop on Chopper Software Development
Date: 10-13 December, 2007
Venue: J-PARC, Tokai, Japan

December '07 Chopper Software Meeting
(Dec. 10-13, 2007)

	10-Dec	11-Dec	12-Dec	13-Dec
10:00-11:00	**Opening Meeting** (@small meeting room at HENDEL bldg.)	**Meeting II** (@small meeting room at HENDEL bldg.)	**Meeting III** (@small meeting room at HENDEL bldg.)	
11:00-12:00	1. Welcome 2. Status & Issues 2-1. K-J collaboration			
12:00-13:30	Lunch	Lunch	Lunch	Lunch
13:30-14:30	**Opening Meeting cont. ~18:00** (@small meeting room at HENDEL bldg.) 2-2. ISIS 2-3. SNS	**J-PARC & Related Facilities Tour**	**Meeting IV** (@small meeting room at HENDEL bldg.)	**Chopper Hardware Meeting** (@small meeting room or rm. 103 at HENDEL bldg.)
14:30-15:30			**Summary** (@small meeting room at HENDEL bldg.)	
15:30-16:30				
16:30-17:30				
			Get Together Party@Shu-Sen-Bou 19:00~	

The 1st Workshop on Chopper Software Development for Next Generation
@J-PARC Center, 10-13 December, 2007

Day-1 (10:00-12:15, 13:30-19:00, 10 December, 2007)
S. Itoh, T. Otomo, J-Y. So, M. Arai, J-G. Park, T. Perring, G. Granroth, Y. Inamura, T. Nakatani, M. Nakamura, J. Suzuki , K. Nakajima

- Welcome (by MA)
- Status of Korea-Japan Collaboration Work (KN, YI, JYS)
  - ✓Korea-Japan Collaboration
  - ✓Software Components & Structure Planned
  - ✓Objectives & Schedule of KJ Collaboration
  - ✓Data Reduction & Manyo-Lib.
- Status of ISIS Excitation Group (TP)
  - ✓Existing Soft wares & those under development (planned)
    HOMER, Mslice, Tobyfit
    LIBISIS, Horace, Grig Tobyfit (Matlib & Fortran 90)
  - ✓Mantid Project
  - ✓Don't reinvent the wheel→Network & Communication
  - ✓Discussion on Sharing works, Sharing Codes in World Neutron Community
- Status of SNS (GG)
  - ✓Over View of SNS Software
  - ✓Portal
  - ✓Data Reduction Software
  - ✓Advanced Analysis Software

The 1st Workshop on Chopper Software Development for Next Generation
@J-PARC Center, 10-13 December, 2007

Day-2 (10:00-11:30, 11 December, 2007)
J-Y. So, J-G. Park, T. Perring, G. Granroth, Y. Inamura, K. Nakajima

- Summary of Day-1
- Development Environment
  Matlab -> ISIS Exc. G
  Python+C^{++} --> SNS, J-PARC, Mantid
  MATLAB is perfect
  Many good public domain soft
  Python is Key (glue) to share the resources
- J-PARC Tour

Day-3 (10:10-11:45, 13:30-16:00 12 December, 2007)
M. Takahashi, K. Tomisasu, J-Y. So, J-G. Park, T. Perring, G. Granroth, Y. Inamura, M. Arai, M. Nakamura, K. Nakajima

- Use AMATERAS as a Single Crystal Diffractometer (MT)
- Single Crystal equivalent Data from Powder Sample (KT)
- Summary
  Next Gen. Chopper Software Development Activity in World Neutron Community

Key Words in the Presentations

- Single Crystal → Arignment, Reduction, Visualization, Analysis ....
  Miwako's Talk
  Keisuke's Talk
- Generalization (data format & applications)
- Error (definition, propagation)
- Development Environment (do we use commercial software?)
- Data Size
- TeraGrid
- Network & Communication
- Sharing Basic Codes
- Sharing development work
- RRM or Any multiple measurements
- Advanced Data Correction (absorption, multiple, phonon....) & Simulation

Matlab, Python+ Public Domain...
↓ ↓
LIBISIS, Horace, Grig Tobyfit... SNS, Mantid, MLF, Manyo-lib.

Genrarization in Data Fromat
Python is Key!

We will keep in touch.
Next meeting in May 2007
Exchange Persons
etc.

Memo for
Next Gen. Chopper Software Development Activity in World Neutron Community

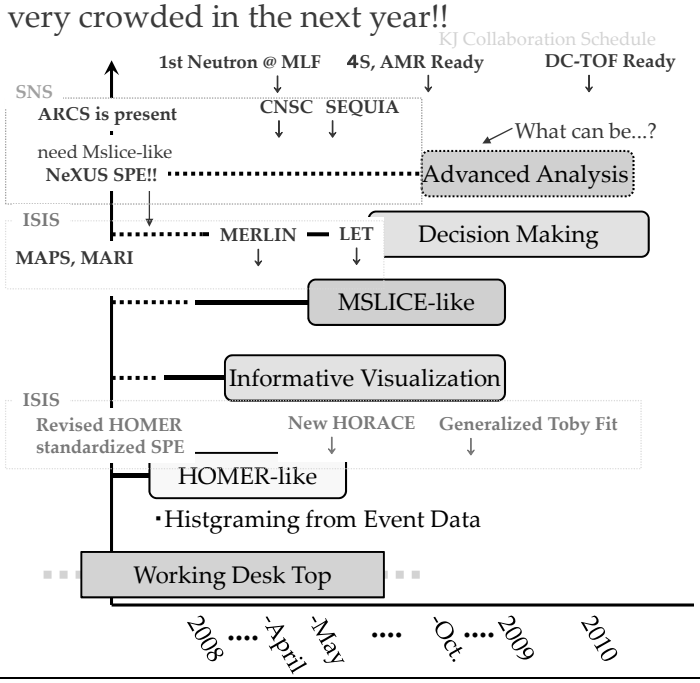
- Where we should go...?
- What we should do...?

✓What we can do together...?
✓What we should develop...?
"Don't reinvent the wheel."

- How we can keep in touch...

✓ Communication
  ✓Mailing List
  ✓Web Site
  ✓Web meeting
✓ Real Meeting
✓ Exchange Persons
✓ Other Ideas...

Time Line of Chopper Group very crowded in the next year!!



Efficiency is important
Advices rather than co
Individual effort unti
Someone between us
Software Work Shop at
Exchange People for a
Advisory Meeting for
Several Levels of Comm
Difference in level of p
(J&K are at bottom, s
Communication is very
meetings, exchange p
Software Review at SN
lack in the lower leve

We will keep in touch!
1. Next Meeting :26-27
2. Exchange Persons :
list up items to be dis
short term till by ne
3. Formalism would be
4. Contact Persons
TB, GG, JGP, KN, MA
5. Use KJ Web site
Issue Accounts for G

The 1st Workshop on Chopper Software Development for Next Generation

- Next Meeting :26-27 May
- Exchange Persons :
  Some of persons of KJ Group will visit SNS, ISIS.
  Short term visit.
  It should be done till next March.
  Long term visit should be discussed in future.
- Formalism would be OK
  We will use existing agreements.
- Contact Persons of this activity:
  TB, GG, JGP, KN, MA, JYS, YI
- KJ Collaboration Web site will be used for this Activity.
  Accounts and Password will be issued to GG & TP. (by YI)

D. 3rd International mini-Workshop on Chopper Software Development
Date: 8-9 December, 2008
Venue: Fukuroda, Ibaraki, Japan

UK-US-K-J Collaboration; the 3rd Chopper Software Mini-Workshop

8-9 December, 2008
Kajika-en at Fukuroda, Ibaraki

8 December, 2008

Persons in Present: T. Perring., G. Granroth, J-G. Park, J-Y. So, M. Arai, Y. Inamura, T. Nakatani, R. Kajimoto, T. Otomo, S. Kawamura, K. Nakajima.

J-PARC & KAERI DC-TOF Status (by K.N. & J-G.P.)

- J-PARC had the first neutron beam on 30 May
- 7 instruments have already start their commissioning and 5 will join soon

- Construction of DC-TOF at KARI is progressing and the spectrometer will be ready in the beginning of 2010
- Especially, significant effort has been done in DAQ-electronics by KAERI in-house people.

J-PARC new DAQ (by T.N.)

- New DAQ system has been developed for J-PARC neutron instruments.
- New DAQ system is using new technology, such as SiTCP and RT middleware.
- Some issues about event data have been discussed.
  - Pulse ID delivered by the accelerator should be used.
  - Some consideration should be done for data format.

SNS status (by G.G.)

- Status of 4 choppers in SNS was reported.
- Dr CHOP tools
- Code developed by SNS data analysis group
- PDOS analysis tool developed by DANSE
- Software modules for single crystal experiments
- New software show its power in some of measurements including Iron superconductor systems

- Discussion on NeXus file format has been done
  - NeXus for general distribute files
  - Internal files may have original formats

ISIS status (by T.P.)

- Status of Chopper Status: MARI, MAPS, MERLIN(~July, 2008), LET(~Oct., 2009)
- Libisis
- MSLICE & HORACE: HORECE is porwerful tool for 4D visualization
- CASTEP: ab initio Phonon calculation software with compatibility to available visualization tools
- Tobyfit: already available, well tested, distributed computing capability
- Mantid Project: Whole ISIS project; next generation of OpenGENIE

- KJ Collaboration (Y.I. & J-Y.S.)

- Current Status of KJ Collaboration Work
- HOMER-like software: 80% ready
- Informative visualization (for decision of conditions): beta version is available
- Single crystal alignment software: Some modules are borrowed from SXD group
- MSLICE-like software (Visualization): beta version is available
- Commissioning tools developed in 4SEASONS commissioning
- Some of Remaind issues

- Event Mode Data Reduction for Chopper Spectrometer
  - Considerations on event data collection on Chopper Spectrometer
  - Reduce the file size, process time and also error bars!

UK-US-K-J Collaboration; the 3rd Chopper Software Mini-Workshop

8-9 December, 2008

Kajika-en at Fukuroda, Ibaraki

- 9 December, 2008

Persons in Present: T. Perring., G. Granroth, J-G. Park, J-Y. So, M. Arai, Y. Inamura, T. Nakatani, R. Kajimoto, Y. Kawakita, S. Kawamura, F. Mizuno, K. Nakajima.

- 4SEASONS current status & commissioning

- KCS Software Demo
  -Commissioning tools including visualization (detector calibration, sequencer tester)
  -Single Crystal Software

- Remained Issues
  -Single Crystal Issues
  -Combining multiple runs to cover large Q, E space
  -User support tools to find good experimental conditions

- Future Collaboration

- Summary

UK-US-K-J Collaboration; the 3rd Chopper Software Mini-Workshop

8-9 December, 2008
Kajika-en at Fukuroda, Ibaraki

- Future Collaboration
  - The level of KJ collaboration might be rising a little bit...
  - Actual production of specific software in the frame work of UK-US-K-J Coll.
    - Single Crystal Treatment
    - Experimental Condition
    - Multi-orientation measurements to cover large Q, E

      :
    - Multiple Scattering
    - Multi Phonon

      } for single crystal; to be done in future after finishing the development of basic software we can start thinking....

    Keeping sharing the information
  - Regular meeting; once per 6 months
    - Next meeting will be held in USA (Thanks, Garrett!)
  - We can open and enlarge this collaboration
    - Open meeting in the next WINS (ICNS)
    - Current core members will keep spending effort in core-activity of this collab.
    - Some of new members can join

UK-US-K-J Collaboration; the 3rd Chopper Software Workshop @ Fukuroda

Thank you for your participation!!

iv. Choice of coordinates

We are using 2 coordinates systems to represent a position of the instrument as shown in Fig. A1. One is the rectangle coordinates system, in which the direction of incident beam is on z-axis, anti-gravity direction on y-axis and x-axis are defined by a right-hand system. The other is the polar coordinates system, in which the position is described by a distance from the sample position and the polar angle from the z-axis of the rectangle coordinates system. Both definitions have been used in many fields, especially for the measurement system of X-ray diffraction. ISIS group has also adopted the polar coordinate to express the position and ambiguities for the area of detecting neutrons in data reduction and analysis. Following the popular use of both coordinates in various fields, we also chose these 2 coordinate systems for our software.

On the other hands, it is not a simple problem to decide which expression is better from a view point of users' understanding and of accurate data treatments. We considered the ambiguities of detector position in our instruments, whose PSDs are standing along perpendicular to horizontal (x-z) plane. We considered 3 expressions for the center position and ambiguities of the area of detectors viewed by scattered neutrons (Fig. A2). Each definition has its own advantages and disadvantages.

(i) Rectangular coordinates

Advantage:

-It achieves the most correct expression of ambiguities whose shape is very close to that of actual detector, sample or other object on spectrometer.

-It looks easy to calculate errors on $\vec{Q}$-vector (Q_x,Q_y,Q_z) by direct translation.

Disadvantage:

-It is not simple to get actual image about the position from this information.

(ii) Pseudo-spherical polar coordinates

Advantage:

-It is easy for users to image the actual position used in the experiment. And it gives more or less correct description when converting the shape of PSD to that of ambiguities on the polar coordinates.

Disadvantage:

-At large elevation angle, a difference between the actual shape of PSD and calculated error shape get increased.

(iii) Spherical polar coordinates

Advantage:

-This choice of coordinate directly includes the scattering angle theta and is the same as the ISIS format. It looks easy to treat data for circular averaging and to correct some errors such as the divergence of incident beam.

Disadvantage:

-The error shape is not suitable for PSD standing along y-axis.

Fig. A3 shows examples for the difference of ambiguity area at one pixel on a PSD between coordinate systems of (ii) and (iii). Different 2 ambiguity-areas at a pixel of a PSD expressed with each

coordinates systems are projected on the same sphere surface (corresponding to a solid angle) with different colours.

It is obvious that the ambiguity expression with (ii) is more suitable for our arrangement of PSD than that with (iii). After the discussions, we concluded that (ii) is the most preferable for our works. On the other hand, we are also aware of the fact that (ii) is rather difficult to be used for actual coding. Therefore, currently we decided to use (iii).

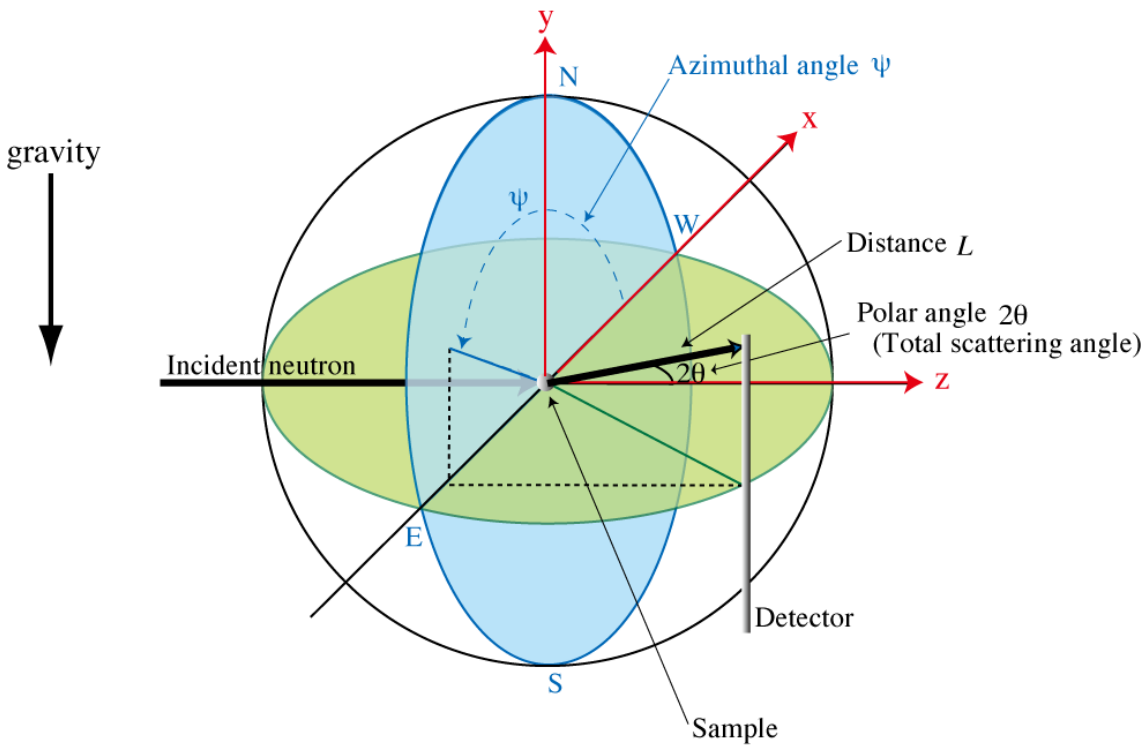


Fig. A1 Coordinates systems used in MLF, J-PARC.

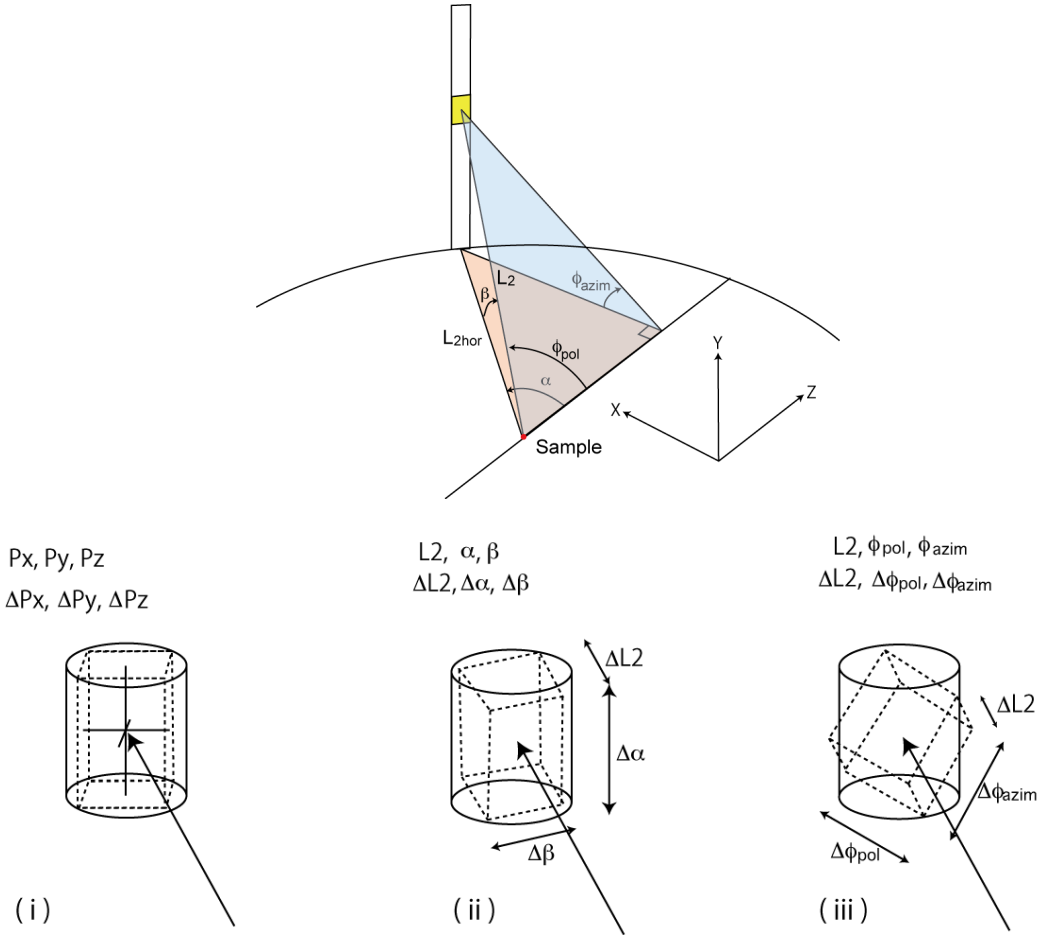


Fig. A2 Possible coordinates systems.

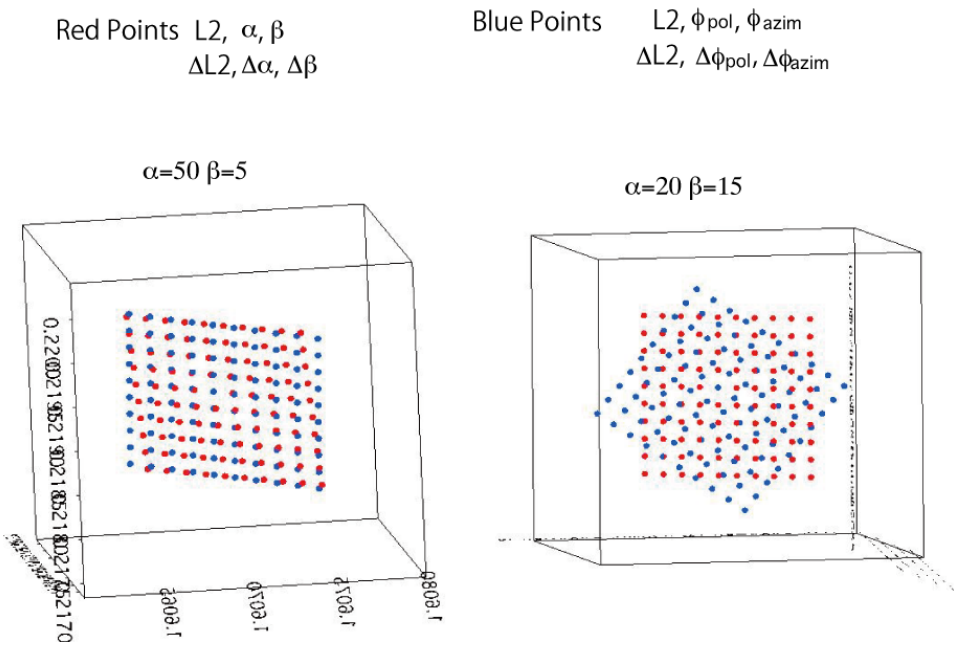


Fig. A3 Ambiguity area at one pixel on PSD in the coordinate systems of (ii) (left) and (iii) (right) as used in Fig. 29

v. Mathematical background

We note that the following discussion is adapted and modified from an ISIS report [15] for our own descriptions. The neutron counts at arbitrary neutron detector pixel are given by

$$C = \Phi(\tau)\cdot\phi(E_i)\cdot T(E_i)\cdot\Delta\tau\cdot N\cdot f(E_i,E_f,\phi)\cdot\frac{k_f}{k_i}\cdot S(\phi,\omega)\cdot\Delta\Omega\cdot\Delta E\cdot\xi(\tau)\varepsilon(E_f), \tag{A.1}$$

where $\Phi(\tau)\varphi(E_i)$ is the neutron flux just before neutron monitor in n/sr/cm^2/s, ΔE_i is the spread in incident energy, $\Delta\tau$ is the counting time, $T(E_i)$ is the transmission due to neutron optics device such as chopper, guide, etc, which are located after monitor. Multiplying the double differential cross section are N, the number of scattering centers in the sample, and $f(E_i, E_f, \varphi)$, the self-shielding factor. $\xi(\tau)\varepsilon(E_f)$ is the efficiency of the detectors measuring over the solid angle $\Delta\Omega$ with energy bin ΔE.

The terms in the above formula can be divided as

Incident neutron flux: $\Phi(\tau)\cdot\phi(E_i)\cdot T(E_i)\cdot\Delta\tau$

Sample correction term: $N\cdot f(E_i,E_f,\phi)$

Scattering Function: $\frac{k_f}{k_i}\cdot S(\phi,\omega)\cdot\Delta\Omega\cdot\Delta E$

Detector correction term: $\xi(\tau)\varepsilon(E_f)$.

Neutron monitor count can be describes as

$$M = \Phi(\tau)\cdot\Delta\tau\cdot\xi(\tau)\int_{Measure}\phi(E_i)\cdot\varepsilon_M(E_i)dE_i = \alpha_M\cdot\Phi(\tau)\cdot\Delta\tau\cdot\xi(\tau), \tag{A.2}$$

where α_M is the integral part.

Monochromatic vanadium run for normalization is used as integrated over all energy transfer.

$$C_V = \Phi(\tau)\cdot\phi(E_i)\cdot T(E_i)\cdot\Delta\tau\cdot N_V\cdot f_V(E_i,\phi)\cdot\frac{\sigma_V}{4\pi}\cdot\Delta\Omega\cdot\xi(\tau)\varepsilon(E_f). \tag{A.3}$$

White beam vanadium runs are used for detector efficiency estimations.

$$\begin{aligned} W &= \Phi(\tau)\cdot\Delta\tau\cdot N_V\cdot\frac{\sigma_V}{4\pi}\cdot\Delta\Omega\cdot\xi(\tau)\int_{Measure}\phi(E_i)\cdot T_0(E_i)\cdot f(E_i,\phi)\cdot\varepsilon(E_f)dE_i \\ &= \alpha_W\cdot\Phi(\tau)\cdot\Delta\tau\cdot N_V\cdot\frac{\sigma_V}{4\pi}\cdot\Delta\Omega\cdot\xi(\tau), \end{aligned} \tag{A.4}$$

where α_W is a constant including an integral part and $T_0(E_i)$ is the transmission for white vanadium measurement. After normalization using measuring time, white beam vanadium and monochromatic vanadium the results become:

$$R(\Omega,t)=\frac{\dfrac{C_S}{\Delta\tau_S}\dfrac{\Delta\tau_{MS}}{M_S}\cdot\left[\dfrac{M_W}{\Delta\tau_M}\dfrac{\Delta\tau_W}{W}\right]_S}{\dfrac{C_V}{\Delta\tau_V}\dfrac{\Delta\tau_{MV}}{M_V}\cdot\left[\dfrac{M_W}{\Delta\tau_M}\dfrac{\Delta\tau_W}{W}\right]_V}=\frac{N_S}{N_V}\frac{4\pi}{\sigma_V}\frac{f_S(E_i,E_f,\phi)}{f_V(E_i,\phi)}\cdot\frac{k_f}{k_i}S(\phi,\omega)\cdot\Delta E_f\cdot\varepsilon(E_f)\,, \tag{A.5}$$

where $\Delta\tau$ is measurement time for sample ($\Delta\tau_S$), monochromatic vanadium ($\Delta\tau_V$), white vanadium ($\Delta\tau_M$), monitor for sample run ($\Delta\tau_{MS}$), for monochromatic vanadium run ($\Delta\tau_{MV}$). $\dfrac{M_W}{\Delta\tau_M}\dfrac{\Delta\tau_W}{W}$ is the ratio between white vanadium run and its monitor count normalized with its counting time for sample run (subscription S) and monochromatic vanadium run (subscription V). Therefore, the scattering function *S(ϕ,ω)* can be calculated from the following equations,

$$S(\phi,\omega)=\frac{N_V}{N_S}\frac{\sigma_V}{4\pi}\frac{f_V(E_i,\phi),}{f_S(E_i,E_f,\phi)}\varepsilon(E_f)\cdot\frac{k_i}{k_f}R(\Omega,\omega)\Delta\omega\,, \tag{A.6}$$

where $\hbar\omega=E_i-E_f$ is the energy transfer.

This formula means that 3 independent measurements (Sample, white vanadium, and monochromatic vanadium) with monitor counts and self attenuation factor are enough to calculate scattering function $S(\phi,\hbar\omega)$. This can be transformed to a general scattering function $S(\vec{Q},\hbar\omega)$ by converting polar and azimuthal angle to momentum transfer $\vec{Q}=\vec{k}_i-\vec{k}_f$.

vi. List of publications and presentations

1. "Current Status of Software Development on Chopper Spectrometers for MLF, J-PARC"
   Y. Inamura, K. Nakajima, R. Kajimoto, T. Nakatani and M. Arai, KAERI J. -Y. So, J. -G. Park
   NOBUGS 2008 Conference
   3-5 November, 2008
   Australia

2. "Current Status of Chopper Spectrometer Software Development at MLF, J-PARC"
   2008 Meeting of the Japanese Society of Neutron Science
   Y. Inamura, K. Nakajima, R. Kajimoto, M. Arai, J. -Y. So, J. -G. Park
   1-2 December, 2008
   Nagoya, Japan

3. "Event Mode Data Reduction and Analysis for Time-of-flight Neutron Spectrometer"
   J. -Y. So
   Hanaro Symposium 2009
   Daejon, Korea

4. "Development of DC- TOF: Current Status and Perspective"
   J. -Y. So
   The 9-th Japan-Korea Meeting on Neutron Science
   Busan, Korea

5. "Korea-Japan Collaboration on Software Development for New Chopper Spectrometer at HANARO & J-PARC"
   Y. Inamura, J. -Y. So, K. Nakajima, J. Suzuki, T. Nakatani, R. Kajimoto, T. Otomo, M. -K. Moon, C. -H. Lee, Y. Yasu, K. Nakayoshi, K. Sendai, U. W. Nam, J. -G. Park and M. Arai
   Trend in Cold Neutron Time-of-Flight Spectroscopy Workshop
   ILL, Grenoble, France

国際単位系（SI）

表１．SI 基本単位

基本量	SI 基本単位	
	名称	記号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質量	モル	mol
光度	カンデラ	cd

表２．基本単位を用いて表されるSI組立単位の例

組立量	SI 基本単位	
	名称	記号
面積	平方メートル	m^2
体積	立法メートル	m^3
速さ，速度	メートル毎秒	m/s
加速度	メートル毎秒毎秒	m/s^2
波数	毎メートル	m^{-1}
密度，質量密度	キログラム毎立方メートル	kg/m^3
面積密度	キログラム毎平方メートル	kg/m^2
比体積	立方メートル毎キログラム	m^3/kg
電流密度	アンペア毎平方メートル	A/m^2
磁界の強さ	アンペア毎メートル	A/m
量濃度(a)，濃度	モル毎立方メートル	mol/m^3
質量濃度	キログラム毎立法メートル	kg/m^3
輝度	カンデラ毎平方メートル	cd/m^2
屈折率(b)	（数字の） 1	1
比透磁率(b)	（数字の） 1	1

(a) 量濃度（amount concentration）は臨床化学の分野では物質濃度（substance concentration）ともよばれる。
(b) これらは無次元量あるいは次元１をもつ量であるが、そのことを表す単位記号である数字の１は通常は表記しない。

表３．固有の名称と記号で表されるSI組立単位

組立量	SI 組立単位			
	名称	記号	他のSI単位による表し方	SI基本単位による表し方
平面角	ラジアン(b)	rad	1 (b)	m/m
立体角	ステラジアン(b)	sr(c)	1 (b)	m^2/m^2
周波数	ヘルツ(d)	Hz		s^{-1}
力	ニュートン	N		m kg s^{-2}
圧力，応力	パスカル	Pa	N/m^2	m^{-1} kg s^{-2}
エネルギー，仕事，熱量	ジュール	J	N m	m^2 kg s^{-2}
仕事率，工率，放射束	ワット	W	J/s	m^2 kg s^{-3}
電荷，電気量	クーロン	C		s A
電位差（電圧），起電力	ボルト	V	W/A	m^2 kg s^{-3} A^{-1}
静電容量	ファラド	F	C/V	m^{-2} kg^{-1} s^4 A^2
電気抵抗	オーム	Ω	V/A	m^2 kg s^{-3} A^{-2}
コンダクタンス	ジーメンス	S	A/V	m^{-2} kg^{-1} s^3 A^2
磁束	ウエーバ	Wb	Vs	m^2 kg s^{-2} A^{-1}
磁束密度	テスラ	T	Wb/m^2	kg s^{-2} A^{-1}
インダクタンス	ヘンリー	H	Wb/A	m^2 kg s^{-2} A^{-2}
セルシウス温度	セルシウス度(e)	℃		K
光束	ルーメン	lm	cd sr(c)	cd
照度	ルクス	lx	lm/m^2	m^{-2} cd
放射性核種の放射能(f)	ベクレル(d)	Bq		s^{-1}
吸収線量，比エネルギー分与，カーマ	グレイ	Gy	J/kg	m^2 s^{-2}
線量当量，周辺線量当量，方向性線量当量，個人線量当量	シーベルト(g)	Sv	J/kg	m^2 s^{-2}
酸素活性	カタール	kat		s^{-1} mol

(a)SI接頭語は固有の名称と記号を持つ組立単位と組み合わせても使用できる。しかし接頭語を付した単位はもはやコヒーレントではない。
(b)ラジアンとステラジアンは数字の１に対する単位の特別な名称で、量についての情報をつたえるために使われる。実際には、使用する時には記号rad及びsrが用いられるが、習慣として組立単位としての記号である数字の１は明示されない。
(c)測光学ではステラジアンという名称と記号srを単位の表し方の中に、そのまま維持している。
(d)ヘルツは周期現象についてのみ、ベクレルは放射性核種の統計的過程についてのみ使用される。
(e)セルシウス度はケルビンの特別な名称で、セルシウス温度を表すために使用される。セルシウス度とケルビンの単位の大きさは同一である。したがって、温度差や温度間隔を表す数値はどちらの単位で表しても同じである。
(f)放射性核種の放射能（activity referred to a radionuclide）は、しばしば誤った用語で"radioactivity"と記される。
(g)単位シーベルト（PV,2002,70,205）についてはCIPM勧告2（CI-2002）を参照。

表４．単位の中に固有の名称と記号を含むSI組立単位の例

組立量	SI 組立単位		
	名称	記号	SI 基本単位による表し方
粘度	パスカル秒	Pa s	m^{-1} kg s^{-1}
力のモーメント	ニュートンメートル	N m	m^2 kg s^{-2}
表面張力	ニュートン毎メートル	N/m	kg s^{-2}
角速度	ラジアン毎秒	rad/s	m m^{-1} s^{-1}=s^{-1}
角加速度	ラジアン毎秒毎秒	rad/s^2	m m^{-1} s^{-2}=s^{-2}
熱流密度，放射照度	ワット毎平方メートル	W/m^2	kg s^{-3}
熱容量，エントロピー	ジュール毎ケルビン	J/K	m^2 kg s^{-2} K^{-1}
比熱容量，比エントロピー	ジュール毎キログラム毎ケルビン	J/(kg K)	m^2 s^{-2} K^{-1}
比エネルギー	ジュール毎キログラム	J/kg	m^2 s^{-2}
熱伝導率	ワット毎メートル毎ケルビン	W/(m K)	m kg s^{-3} K^{-1}
体積エネルギー	ジュール毎立方メートル	J/m^3	m^{-1} kg s^{-2}
電界の強さ	ボルト毎メートル	V/m	m kg s^{-3} A^{-1}
電荷密度	クーロン毎立方メートル	C/m^3	m^{-3} sA
表面電荷	クーロン毎平方メートル	C/m^2	m^{-2} sA
電束密度，電気変位	クーロン毎平方メートル	C/m^2	m^{-2} sA
誘電率	ファラド毎メートル	F/m	m^{-3} kg^{-1} s^4 A^2
透磁率	ヘンリー毎メートル	H/m	m kg s^{-2} A^{-2}
モルエネルギー	ジュール毎モル	J/mol	m^2 kg s^{-2} mol^{-1}
モルエントロピー，モル熱容量	ジュール毎モル毎ケルビン	J/(mol K)	m^2 kg s^{-2} K^{-1} mol^{-1}
照射線量（X線及びγ線）	クーロン毎キログラム	C/kg	kg^{-1} sA
吸収線量率	グレイ毎秒	Gy/s	m^2 s^{-3}
放射強度	ワット毎ステラジアン	W/sr	m^4 m^{-2} kg s^{-3}=m^2 kg s^{-3}
放射輝度	ワット毎平方メートル毎ステラジアン	$W/(m^2$ sr)	m^2 m^{-2} kg s^{-3}=kg s^{-3}
酵素活性濃度	カタール毎立方メートル	kat/m^3	m^{-3} s^{-1} mol

表５．SI 接頭語

乗数	接頭語	記号	乗数	接頭語	記号
10^{24}	ヨタ	Y	10^{-1}	デシ	d
10^{21}	ゼタ	Z	10^{-2}	センチ	c
10^{18}	エクサ	E	10^{-3}	ミリ	m
10^{15}	ペタ	P	10^{-6}	マイクロ	μ
10^{12}	テラ	T	10^{-9}	ナノ	n
10^9	ギガ	G	10^{-12}	ピコ	p
10^6	メガ	M	10^{-15}	フェムト	f
10^3	キロ	k	10^{-18}	アト	a
10^2	ヘクト	h	10^{-21}	ゼプト	z
10^1	デカ	da	10^{-24}	ヨクト	y

表６．SIに属さないが、SIと併用される単位

名称	記号	SI 単位による値
分	min	1 min=60s
時	h	1h =60 min=3600 s
日	d	1 d=24 h=86 400 s
度	°	1°=(π/180) rad
分	’	1’=(1/60)°=(π/10800) rad
秒	”	1”=(1/60)’=(π/648000) rad
ヘクタール	ha	$1ha=1hm^2=10^4m^2$
リットル	L, l	$1L=11=1dm^3=10^3cm^3=10^{-3}m^3$
トン	t	$1t=10^3$ kg

表７．SIに属さないが、SIと併用される単位で、SI単位で表される数値が実験的に得られるもの

名称	記号	SI 単位で表される数値
電子ボルト	eV	$1eV=1.602\ 176\ 53(14)\times10^{-19}J$
ダルトン	Da	$1Da=1.660\ 538\ 86(28)\times10^{-27}kg$
統一原子質量単位	u	1u=1 Da
天文単位	ua	$1ua=1.495\ 978\ 706\ 91(6)\times10^{11}m$

表８．SIに属さないが、SIと併用されるその他の単位

名称	記号	SI 単位で表される数値
バール	bar	$1\ bar=0.1MPa=100kPa=10^5Pa$
水銀柱ミリメートル	mmHg	1mmHg=133.322Pa
オングストローム	Å	$1\ Å=0.1nm=100pm=10^{-10}m$
海里	M	1 M=1852m
バーン	b	$1\ b=100fm^2=(10^{-12}cm)2=10^{-28}m^2$
ノット	kn	1 kn=(1852/3600)m/s
ネーパ	Np	SI単位との数値的な関係は、対数量の定義に依存。
ベル	B	
デジベル	dB	

表９．固有の名称をもつCGS組立単位

名称	記号	SI 単位で表される数値
エルグ	erg	$1\ erg=10^{-7}$ J
ダイン	dyn	$1\ dyn=10^{-5}$N
ポアズ	P	1 P=1 dyn s cm^{-2}=0.1Pa s
ストークス	St	1 St =$1cm^2$ s^{-1}=$10^{-4}m^2$ s^{-1}
スチルブ	sb	1 sb =1cd cm^{-2}=10^4cd m^{-2}
フォト	ph	1 ph=1cd sr cm^{-2} 10^4lx
ガル	Gal	1 Gal =1cm s^{-2}=$10^{-2}ms^{-2}$
マクスウェル	Mx	1 Mx = 1G cm^2=10^{-8}Wb
ガウス	G	1 G =1Mx cm^{-2} =10^{-4}T
エルステッド(c)	Oe	1 Oe ≙ $(10^3/4\pi)$A m^{-1}

(c) ３元系のCGS単位系とSIでは直接比較できないため、等号「 ≙ 」は対応関係を示すものである。

表10．SIに属さないその他の単位の例

名称	記号	SI 単位で表される数値
キュリー	Ci	$1\ Ci=3.7\times10^{10}Bq$
レントゲン	R	$1\ R=2.58\times10^{-4}C/kg$
ラド	rad	$1\ rad=1cGy=10^{-2}Gy$
レム	rem	$1\ rem=1\ cSv=10^{-2}Sv$
ガンマ	γ	1γ=1 nT=10-9T
フェルミ		1フェルミ=1 fm=10-15m
メートル系カラット		1メートル系カラット = 200 mg = 2×10-4kg
トル	Torr	1 Torr = (101 325/760) Pa
標準大気圧	atm	1 atm = 101 325 Pa
カロリー	cal	1cal=4.1858J（「15℃」カロリー），4.1868J（「IT」カロリー）4.184J（「熱化学」カロリー）
ミクロン	μ	1 μ =1μm=10^{-6}m

（第8版，2006年改訂）

この印刷物は再生紙を使用しています