

Standards for Data Analysis Software (WP6)

Emmanuel Farhi

Computing for Science, ILL

March 2012

Partners: ILL, ISIS, PSI, FRM2, JCNS, GKSS, HZB

Our collective approach to data reduction and analysis



This way please

What requirements should a data reduction/analysis software fulfill ?

You all have your own ideas, right ?

*Let's see what is requested from the Data, by Users and
by Programmers...*

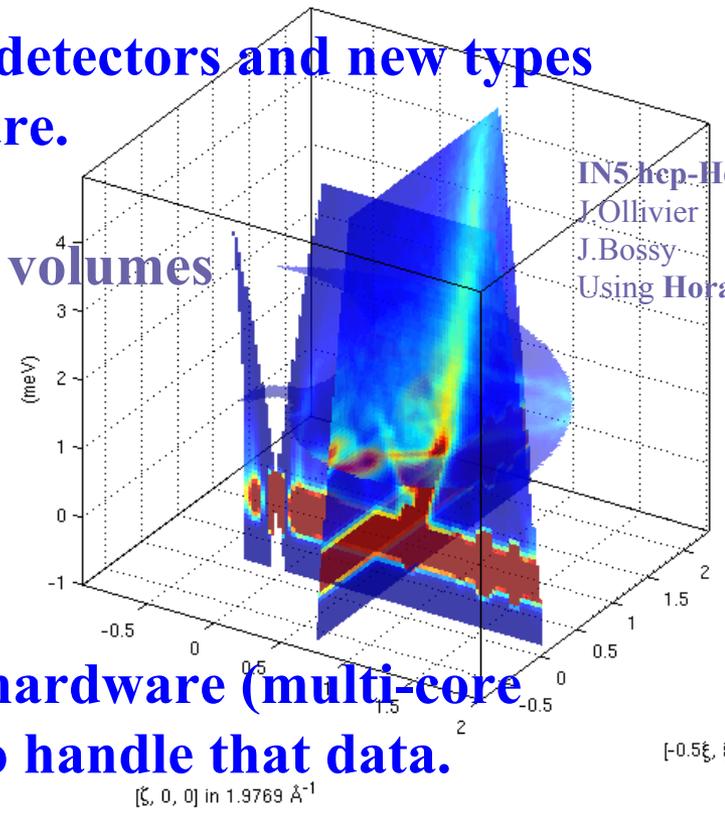




D:\users\ollivier\6-01-321\spe\He_25p387bars_0p3K_3A_3rd_crystal_HR.sqw
-0.025 ≤ η ≤ 0.025 in [0, 0, η]
ξ = -0.9875:0.025:2.0125 in [ξ, 0, 0], ξ = -0.8375:0.025:2.3375 in [-0.5ξ, ξ, 0], E = -1.0089:0.00

Shorter experiments, rapid data rates, large detectors and new types of instruments make new demands on software.

BUT: this is just as in the old days, with **larger volumes** → **require fast processing.**



New software (languages and methods) and hardware (multi-core processors, GPU's, etc) offer new solutions to handle that data.

BUT: never forget that the **old codes** are the most valuable and hold the scientific knowledge (stable, fast...) → **re-use ideas and implementations.**

Plenty of available good software with *many* users.

- Satisfy the need today (mostly) !
- (Easy) install and (Easy) usage
- Reliability (bugs have been fixed before)
- A community



“What do you say ?”

Requests from users ?

Data treatment survey ILL News #52 June 2010

8000 emails sent, 250 answers.

<<http://www.ill.eu/quick-links/publications/ill-news/>>



“COMMON”

What do “they” want ?

- ★ **Integrated** software platform
- ★ **Common** language
- ★ Software compatibility/**inter-operability**
- ★ **Common** data formats
- ★ **Home/lab/facility** equivalence
- ★ **Documentation**
- ★ Efficiency → **both interfaces & scripts** → separated
- ★ **Fitting**
- ★ **Quality graphics**
- ★ **Open source**



She: “I want this !”



He: “I want this !”

★ *Prepare the coffee ?*

The existing codes are nice (knowledge, reliability)

BUT:

- As many programming styles as individuals.
- Usually created for a specific application.
- Single points of failure.
- Issues regarding maintainability (abandoned software ?).
- Many features are common to all packages (the *kernel*):
load, plot, subtract, group, fit, save...

What counts for Developers: software life-time and maintenance

- Lightweight project.
- Ensure portability.
- Centralize common functionalities (no duplication).
- Define standards (format, procedures ... and keep them !).
- Separate scripts and interfaces

Any 'new' project should start by an evaluation from past attempts:

- LAMP
- Grumtree
- DANSE
- Horace
- Mantid
- ROOT (CERN)
- ... (I certainly forget some)

What is “*good, bad and ugly*” in these ?

Do we need to re-invent the wheel ?

How to optimize your investment ?

As the others, we will start by a review.



NMI3-II (the Revenge) Data analysis software work-package

Funding: 24+ (wo)man months starting in May 2012.

1. Review existing data analysis software and practices of software developers (2)
2. Review existing solutions for a common data analysis infrastructure (2)
3. Develop *prototype* software in chosen solution for representative applications (14)
4. Evaluate *prototype* software (3).

Proposed prototype:

- S(q,w) 4D data from reactor based multiplexed Xtal instruments

One candidate:

- Mantid <www.mantidproject.org>
- VATES is the Mantid 4D extension, based on Horace/LibISIS.

But we should be aware of *other solutions* as well