

Atacama Large Millimeter Array

ALMA-
Revision:1.3
2003-03-10
Jérôme Pety

AIPS++ Reuse Analysis Test: Report on Phase II

J. Pety^{1,2}, A. Baker³, A. Coulais²,
F. Gueth¹, D. Shepherd⁴, L. Testi⁵,
C. Wilson⁶

Support by D. Broguière¹, K. Golap⁴,
R. Lucas¹, G. Moellenbrock⁴, W. Young⁴

1. IRAM (France)
2. LERMA, Observatoire de Paris (France)
3. MPE, Garching (Germany)
4. NRAO (USA)
5. INAF - Osservatorio Astrofisico di Arcetri (Italy)
6. McMaster University (Canada)

Keywords:	
Author Signature: Jérôme Pety	Date: 2003-03-10
Approved by:	Signature:
Institute: NRAO, NRO, ESO	Date:
Released by:	Signature:
Institute:	Date:

Change Record

Revision	Date	Author	Section/ Page affected	Remarks
1	2003-03-04	Jérôme Pety	A to G	Questionnaire answers
2	2001-03-06	Jérôme Pety	1 to 7, H	Initial version of main document
3	2001-03-10	Jérôme Pety	1 to 7	Final version incorporating comments from testers

Contents

1	Introduction	6
2	Test charge as seen from Phase II	6
3	Recommendations	6
A	Summary	8
A.1	Introduction	8
A.2	Conditions of this test	8
A.2.1	Testers	8
A.2.2	Data sets	8
A.2.3	Time considerations	8
A.3	Results	9
A.3.1	Calibration with the <code>iramcalibrator</code> tool	9
A.3.2	Imaging	9
A.4	Degree of interactivity of AIPS++	10
A.4.1	Speed considerations	10
A.4.2	Interface implementation	10
A.4.3	Interface heuristics	11
A.4.4	Data visualization	11
A.4.5	Data inspection	11
A.5	AIPS++ flexibility and stability	12
A.5.1	Flexibility	13
A.5.2	Stability	13
A.6	Support	13
B	Questionnaire of Andrew Baker	14
B.1	Tester	14
B.2	Data set	14
B.3	Calibration inside AIPS++ by the <code>iramcalibrator</code> tool	14
B.4	Imaging inside AIPS++	17
B.5	First analysis and production of a publication quality plot inside AIPS++	17
B.6	Comparison of results obtained with AIPS++ and CLIC (Optional)	17
B.7	General	18
B.8	Support	18
B.9	Other remarks	19
C	Questionnaire of Alain Coulais	20
C.1	Tester	20
C.2	Data set	20
C.3	Calibration inside AIPS++ by the <code>iramcalibrator</code> tool	20
C.4	Imaging inside AIPS++	21
C.5	First analysis and production of a publication quality plot inside AIPS++	21
C.6	Comparison of results obtained with AIPS++ and CLIC (Optional)	22
C.7	General	22
C.8	Support	23
C.9	Other remarks	24

D	Questionnaire of Frédéric Gueth	25
D.1	Tester	25
D.2	Data set	25
D.3	Calibration inside AIPS++ by the <code>iramcalibrater</code> tool	25
D.4	Imaging inside AIPS++	27
D.5	First analysis and production of a publication quality plot inside AIPS++	27
D.6	Comparison of results obtained with AIPS++ and CLIC (Optional)	27
D.7	General	27
D.8	Support	29
D.9	Other remarks	29
E	Questionnaire of Jérôme Pety	30
E.1	Tester	30
E.2	Data set	30
E.3	Calibration inside AIPS++ by the <code>iramcalibrater</code> tool	31
E.4	Imaging inside AIPS++	33
E.5	First analysis and production of a publication quality plot inside AIPS++	34
E.6	General	34
E.7	Support	35
E.8	Test charge	36
E.9	Other remarks	36
F	Questionnaire of Debra Shepherd	38
F.1	Tester	38
F.2	Data set	38
F.3	Calibration inside AIPS++ by the <code>iramcalibrater</code> tool	38
F.4	Imaging inside AIPS++	40
F.5	First analysis and production of a publication quality plot inside AIPS++	40
F.6	Comparison of results obtained with AIPS++ and CLIC (Optional)	40
F.7	General	41
F.8	Support	41
F.9	Other remarks	42
G	Questionnaire of Leonardo Testi	44
G.1	Tester	44
G.2	Data set	44
G.3	Calibration inside AIPS++ by the <code>iramcalibrater</code> tool	44
G.4	Imaging inside AIPS++	45
G.5	First analysis and production of a publication quality plot inside AIPS++	45
G.6	Comparison of results obtained with AIPS++ and CLIC (Optional)	45
G.7	General	46
G.8	Support	46
G.9	Other remarks	47
H	Questionnaire of Christine Wilson	48
H.1	Tester	48
H.2	Data set	48
H.3	Calibration inside AIPS++ by the <code>iramcalibrater</code> tool	48
H.4	Imaging inside AIPS++	49
H.5	First analysis and production of a publication quality plot inside AIPS++	50
H.6	Comparison of results obtained with AIPS++ and CLIC (Optional)	51
H.7	General	51

H.8	Support	52
H.9	Other remarks	53
I	Defect and enhancement summary	54
I.1	Problems fixed during calibration & imaging	54
I.2	Problems found and NOT yet fixed	54
I.3	Enhancements made during Phase II testing	55
I.4	Enhancements requested but not yet implemented	56

1 Introduction

The ‘‘AIPS++ Reuse Analysis Test’’ has been designed to address the ability of AIPS++ to be adapted to the needs of the ALMA reduction software. It has been divided into three parts:

Phase I Implementation of PdBI (Plateau de Bure Interferometer) calibration engines inside AIPS++;

Phase II Test of the calibration extensions implemented in Phase I and of the native AIPS++ capabilities (in particular, imaging capabilities) on a fair sample of PdBI data sets by a panel of testers outside the AIPS++ team;

Phase III Benchmarking of AIPS++ on ALMA-size data sets using the calibration extensions implemented in Phase I.

This report addresses every aspect linked to Phase II. Section 2 details the findings on the test charge as seen from Phase II. Section 3 is a collection of recommendations to the AIPS++ project. These findings and recommendations follow naturally from the experience gained during Phase II, which is summarized in Appendix A. In all sections of this appendix, the text gives only the mean of a response sample whose dispersion is sometimes large. We thus encourage the reader to go through all the questionnaires available in Appendices B through H, to obtain an idea of the full range of feedback.

2 Test charge as seen from Phase II

1. How can AIPS++ be adapted to reduce data of an instrument for which it was not initially designed?

In Phase II, all testers succeeded in reaching the end of the calibration steps. Calibration solutions seemed good, with two exceptions that can probably be quickly solved. However, many testers did not succeed in obtaining believable images during the short period in which they were attempting to do so. This is a surprising finding, as the imaging stage of interferometric data reduction is the part most independent of the instrument. It would be useful to close the loop with successful imaging to be sure the calibration was done correctly.

2. How long is the learning curve for developers who have sufficient experience in the processing of millimeter data, but no experience at all in the AIPS++ programming environment?

We can only answer here from the *user* point of view. A four-hour tutorial and a detailed cookbook are currently absolutely needed for the user to be able to do the basic calibration and continuum imaging steps. However, even a few days are insufficient for the user to be able to safely figure out (without help) any procedure deviating from the cookbook recipes.

3. Can we perform an end-to-end experiment on actual, real-life mm-wave spectroscopic data?

It is certainly possible to perform an end-to-end experiment on actual, real-life mm-wave spectroscopic data, as shown by one tester in Phase II. However, important examples of the science currently done with mm interferometers cannot yet be easily treated inside AIPS++, in particular due to its deficiencies in spectral line imaging and its lack of a *uv* fit tool.

3 Recommendations

- Additional time spent on `iramcalibrator` should now be minimized as other priorities are expected for AIPS++. Important exceptions are benchmarking and correction of bugs preventing a correct use of `iramcalibrator`, *i.e.*, bugs in the `almati filler` and `imager` tools.
- It is obvious that graphical elements such as `msplot` are too slow. Their functionalities should be moved to a much faster vehicle.

- In addition, other (non-graphical) parts of AIPS++ seem slower than those in other packages. This should be understood and corrected during benchmarking.
- AIPS++ should be made more robust against user input errors, and error messages should be uniformly presented so that users can easily diagnose the origin and severity of a particular problem. The protocol for bug reports should be documented more clearly (e.g., how specific "bugs" must be in order to be submitted, and how the AIPS++ team handles them upon receipt).
- This test was time-consuming and often frustrating for the individual testers. However, having a dedicated panel of testers was clearly useful for improving existing functions (*i.e.*, in `iramcalibrator`) and identifying areas where future improvement is needed (*i.e.*, in `imager`). We recommend that testing like this be repeated at regular intervals by an appropriate group of people, perhaps as parts of the package are thought to reach maturity. Moreover, as for every large software package, AIPS++ probably has a test suite to check its functionalities. This test suite should be extended to mm data sets and should be run *i)* on a regular basis, and *ii)* when changes are implemented in the master version of the software.
- The best data reduction software is more than a collection of algorithms, but also has an intuitive interface to help guide users through the different reduction steps. Having sensible defaults for input parameters that may take into account data properties, good documentation of the software features that are available, and a cookbook written by astronomers for the non-expert user are also mandatory. We encourage the AIPS++ team to pay careful attention to the heuristics which underlie the data reduction process.
- To help the design of new GUIs, it is advised that:
 - all GUIs (including plotters such as `viewer`) should be associated with a scripting language, *i.e.*, each GUI action should be associated with a script command;
 - GUIs should be developed around scripts which select intelligent defaults from inspection of the data. A representative example would be a script allowing the user to make a first, unoptimized image knowing just the source name, line name, and velocity resolution.

In conclusion, we recommend that astronomers be more closely involved not just in developing software requirements, but also in issues of implementation. There is a role for astronomers both at a leadership level and at lower levels in terms of definition, testing, and feedback in the software development. The recent appointment of a new AIPS++ Project Scientist is one positive step in this direction. The ultimate goal is to have a software package that is well adapted to the needs of the research astronomers who not only use it but also develop new modules inside it.

A Summary

A.1 Introduction

Sections A.2 and A.3 summarize the major issues identified by one or more of the testers concerning the Phase II aims. Because of the substantial time investment required to address the Phase II charge and learn how to reduce PdBI data in particular, the testers have also had the opportunity to familiarize themselves with many more general aspects of AIPS++. Sections A.4 and A.5 of this report detail some of the issues which arose in this category; these are offered in the expectation that they will complement the formal results of the AIPS++ Audit in providing useful feedback for the future development of AIPS++ as a possible ALMA software package¹.

In all sections of this executive summary, the text gives only the mean of a response sample whose dispersion is sometimes wide. We thus encourage the reader to go through all the questionnaires available in Appendices B through H, to obtain an idea of the full range of feedback.

A.2 Conditions of this test

A.2.1 Testers

Seven testers agreed to participate in Phase II. They comprised a panel of complementary backgrounds and experiences. Six are affiliated with the ALMA project (three SSR members, three Computing IPT members). One is a member of the AIPS++ North American User Group, one participated in the AIPS++ Audit, and the rest had little or no experience with AIPS++ before this test. Six have had extensive experience with calibration and imaging of mm interferometry data, while the seventh has developed/improved imaging and deconvolution algorithms. Four testers have fair to substantial experience with PdBI data, while the others have had little to none previous PdBI experience.

One or more of the testers has extensive experience in each of AIPS++, AIPS Classic, Difmap, GILDAS, IDL, IRAF, the ISO reduction software, MMA, and MIRIAD. All seven testers participated in the AIPS++ tutorial at the Phase II kickoff meeting in Garching at the end of January 2003.

A.2.2 Data sets

During Phase II, six different PdBI data sets were used, in addition to the data set used for phase I. Four testers were already familiar with their data sets, as they were PIs or co-PIs of the projects. The data sets provided fair coverage of both the scientific range and the technical capabilities of the PdBI. They included single-field detection or mapping of extragalactic sources with large velocity width in line and/or continuum at 3mm and/or 1mm, and mosaicing and single-field mapping of Galactic sources in line and/or continuum at 3mm and/or 1mm.

Because of its complexity, the mosaic data set was assigned to two testers– one used to mosaicing techniques inside AIPS++, and the other used to mosaicing techniques inside GILDAS. Modifications needed to calibrate and image mosaics inside AIPS++ were implemented by D. Broguière and R. Lucas of IRAM, and K. Golap of NRAO.

A.2.3 Time considerations

Phase 2 started at a face-to-face meeting in Garching (27th-31st January 2003). It was agreed at that time that the final document should be delivered by March 10th, to be available for the Preliminary Design Review of the Computing IPT. To leave enough time for writing the document, the deadline for finishing the testing was thus set to March 3rd. This left approximately one month for AIPS++ installation, testing, and bug correction.

¹By way of example, our comments address– among others– requirements OL-1.2-R2, OL-2.2-R1, OL-2.2-R3, and OL-2.2-R6 in Version 1.4 of the ALMA Offline Data Processing Requirements document.

The total time spent on the testing is estimated to be 29.5 FTE days (including calibration, imaging, and filling out the questionnaire, but not counting installation of AIPS++) unevenly spread among the testers. IRAM and NRAO support must be added to this amount. Due to other constraints, it turned out that five of the testers worked mainly in the last week before the deadline. Some bugs were thus identified but not corrected during the time frame of the test.

A.3 Results

A.3.1 Calibration with the `iramcalibrator` tool

All testers succeeded in using the `iramcalibrator` tool up to its final stage, *i.e.*, application of the calibration curves to the source data. This represents a clear improvement over the state of the `iramcalibrator` tool at the kickoff meeting. The calibration solutions seemed correct when plotted over the calibrator data, except for two cases.² There is no doubt that these problems can be corrected. However, it is difficult to say with confidence that the calibration process has been successful as long as imaging has not also been completed successfully.

The quality of the observations (weather conditions, system temperatures, pointing errors, tracking errors) are either not filled or not used. This is an important deficit, as this information helps the user select good data points according to the needs of the project (*i.e.*, the impact of high tracking errors is less severe for a detection project than for a mapping project). Acceptable use of this information would only require production of a dozen plots as done inside GILDAS. Moreover, in the `iramcalibrator` tool, graphics are not displayed by the software. Plots are just output to PostScript files, which limits the interactivity. These PostScript files have explicit names but no internal captions, limiting the user's ability to identify each plot with the corresponding day/band after printing. Axis scales are sometimes confusing due to use of different units (seconds vs. hours and degrees vs. radians). Such details make it more difficult to check calibration solutions. Finally, a unified paper report from the PS and ASCII files already available would be very useful for archiving reasons. As these functionalities are not fundamental in obtaining calibration solutions, however, it is understandable that they have not yet been implemented inside the `iramcalibrator` tool.

During Phase II, bug fixes and improvements to the `iramcalibrator` tool were implemented quickly (See Appendix I for a list of the fixed and outstanding bug and improvement requests generated during Phase II).

A.3.2 Imaging

All testers ran into problems during the imaging part. Only four continuum images have been produced, of which one had stripes and two had spurious detections due to the impossibility of easily extracting the lines contaminating the continuum. Only two line cubes were produced, one of which had spurious negative residuals due to a bug that was encountered while setting the deconvolution regions. Four reasons explain these results: *i*) Possible calibration and/or editing problems; *ii*) Complexity of imaging commands (added to sometimes inadequate documentation), preventing a good use of the input parameters; *iii*) Bugs in the masking routine; *iv*) missing features. Bugs and missing features were in fact observed to be present; the most sorely lacking features were the abilities to select arbitrary channels in a spectral window for imaging with multi-frequency synthesis (mfs), and to perform mfs on spectral windows with different velocity resolutions. The one tester who succeeded in obtaining believable line and continuum images (in the complex mosaic case) needed several work-arounds, which she was able to find due to her 2.5-year past experience in AIPS++.

²*i*) For one project at 1 mm, the phase bandpass was wrong by a constant offset of $\sim 15^\circ$ for several baselines. *ii*) For another project, the phase and amplitude calibration solutions were wrong for all antennas when data from one antenna were absent at the beginning or end of the observations (due to, e.g., shadowing).

A.4 Degree of interactivity of AIPS++

A.4.1 Speed considerations

Pure computing speed performance will be addressed by benchmarking (partly in Phase III). However, those benchmarks will address only batch computation. In interactive mode, other speed limitations must be taken into account— optimization to decrease the number of user inputs, quick display methods, and so on. Here are several important points to be addressed by the AIPS++ team:

- There is a consensus among testers that the graphical part (in particular, the `mplot` tool used for flagging) is much too slow³.
- In an attempt to measure the interactivity speed as objectively as possible, one of the testers tried to recalibrate his data set knowing exactly what to do in both AIPS++ and GILDAS, and starting from the raw data format for each package to avoid the filler steps (no by-hand flagging was done at all, as this appears to be the most time-consuming step in AIPS++). The AIPS++ calibration took a bit more than three times longer than the GILDAS calibration (57 min. compared to 16 min.). Such a factor, while perhaps acceptable for the occasional user, is completely prohibitive for daily use.
- Why does it take from 30 to 60s on a powerful Linux PC or a new SunBlade system just to start AIPS++?

A.4.2 Interface implementation

The underlying object-oriented programming structure is systematically presented to the user. For instance, when the user clicks on a software functionality programmed as an object, he or she first has to create the object and give it a name. This is i) lost time and ii) confusing to the user who may not be aware of object-oriented programming. The user is more inclined to think of data reduction operations as actions to be performed on his or her data; the creation of objects, though necessary in scripts, should be better hidden from the GUI user.

The names of the requested input parameters precisely reflect the underlying variables. For instance, depending on which tool is used, the source name is solicited as `FieldName` or `FieldID`. The answer must be an integer number that may be either absolute or relative. There is thus no uniform way for the user to interact with the system. Furthermore, several testers had trouble remembering to click GO after setting all the options in a function before moving on to the next one.

There is not enough integration between the different GUIs. For instance, the number of windows generated by AIPS++ quickly becomes hard to manage. When the user enters AIPS++, three different windows appear (`manager`, `logger`, and `glish`). The user must then open at least one window per tool, the needed ones being the `catalog`, `mplot` (for flagging), `iramcalibrator`, `imager`, and the `viewer` windows. To go from one of those GUI windows to another typically requires going back to the main GUI (`manager`). Essential information can arrive through the `logger` window, through the `glish` window, or sometimes as PS or ASCII files on disk (for which it is necessary to open `ghostview` or a text editor). Finally, another optional (but very interesting, as it allows the user to learn how to write scripts) window is the `scripter`. All this means that the user must very often hunt for the correct window in order to understand what is happening or to execute the next reduction step.

Finally, there does not seem to be a scripting language associated with the `viewer` tool. For instance, the `scripter` does not log anything when this tool is in use, and it is hard to find anything about this issue in the documentation. This makes it difficult to redo a previous reduction from scratch without having to do everything interactively— very important when the user wants to make slight modifications to the reduction or to publication-quality plots.

³The most experienced developer among the testers suggests that speed problems most likely result from the use of TCL/TK and of the bounding PGplot inside `glish` (<http://aips2.nrao.edu/released/docs/reference/Glish/node98.html>).

A.4.3 Interface heuristics

When entering AIPS++ to reduce interferometric data, all tools from Single-Dish to modeling are available at the same level. In each tool, all the functionalities are listed in alphabetical order without hierarchical organization. One exception is the `imager` tool. However, here the structure is so awkward that even the tutorial suggests it is useful to eliminate this organization by selecting the option of viewing all the functionalities at once.

The user very often must interact with AIPS++, which repeatedly asks for the source name and the spectral windows used. This interaction is also not straightforward, as AIPS++ often asks for integer instead of human-readable information, *i.e.*, the user must first translate a source name or sideband (*e.g.*, 1mm USB) into an integer through the `catalog` tool. On the other hand, the level of control is not always fine enough, *i.e.*, the user would sometimes like to interact more with AIPS++.

There are no short cuts. For instance, a novice user wants an image. It does not matter initially whether this image is optimized, *i.e.*, whether the deconvolution method is the right one (a true novice probably does not even want to know that deconvolution is a needed step). There should thus be a short cut that is able to make an image without any input parameters other than the line name (and maybe the velocity resolution)⁴. Currently the number of mouse clicks required to produce a map is significantly larger than the number of major conceptual steps involved. Moreover, the advice function is at best misleading. This function gives the user an upper limit on the pixel size and a lower limit on the image size, but does not give *good* advice, as it specifies only the pixel size yielding one cell across the beam. Nyquist sampling requires two pixels per beam, and deconvolution asks for three to five pixels per beam.

Current users thus face the following choice: either they follow the (excellent) cookbook without understanding what they are doing (in which case the software should be able to do exactly the same steps by itself), or they try to devise a variation from the cookbook and they run into problems. This is exactly what happened in Phase II with line imaging, which is not yet part of the cookbook.

Some of these problems have already been addressed in the `iramcalibrator` tool as it was developed, but testers used to reducing PdBI data still find this tool far below the equivalent GILDAS tool (*i.e.* CLIC). For instance, inside CLIC, the six calibration steps are offered to the user in the order required by the calibration logic, the bandpass calibrator is automatically chosen, each calibration solution is displayed in a graphical window, and there is a pause allowing the user to interactively change the fit parameters if an initial solution is not satisfactory.

A.4.4 Data visualization

One important way to interact with a reduction software package is through graphics. All testers pointed out that data visualization is a weak point of AIPS++. Only remarks about the `viewer`, which has been part of AIPS++ for some time, will be made here. In the `viewer`, the default is to plot one plane of cube at a time, in black and white, without axes, contours, a color scale, or a header describing the content. To produce a paper-quality plot requires a good dozen mouse clicks. The `viewer` is very hard to configure. Adding text and symbols is difficult, and making multi-plot figures is almost impossible. It is only barely adequate for plotting a continuum image.

A.4.5 Data inspection

The `Catalog/Summarize` function allows the user to determine much information about the observations. Here are the relevant parameters for one data set:

Fields: 5				
ID	Name	Right Ascension	Declination	Epoch
1	3C345	16:42:58.81	+39.48.37.00	J2000
2	MWC349	20:32:45.54	+40.39.36.64	J2000
3	3C454.3	22:53:57.75	+16.08.53.56	J2000

⁴This kind of short cut is available inside GILDAS for single-field mapping.

4	2230+114	22:32:36.41	+11.43.50.90	J2000	
5	PSS2322+19	23:22:07.20	+19.44.23.00	J2000	
Data descriptions: 32 (32 spectral windows and 1 polarization setups)					
ID	Ref.Freq	#Chans	Resolution	TotalBW	Correlations
1	90029 MHz	1	145990 kHz	145990 kHz	XX
2	93429 MHz	1	-145990kHz	145990 kHz	XX
3	90029.6MHz	128	1250 kHz	160000 kHz	XX
4	93428.4MHz	128	-1250 kHz	160000 kHz	XX
5	90159 MHz	1	152000 kHz	152000 kHz	XX
6	93299 MHz	1	-152000kHz	152000 kHz	XX
7	90159.6MHz	128	1250 kHz	160000 kHz	XX
8	93298.4MHz	128	-1250 kHz	160000 kHz	XX
9	90289 MHz	1	152000 kHz	152000 kHz	XX
10	93169 MHz	1	-152000kHz	152000 kHz	XX
11	90289.6MHz	128	1250 kHz	160000 kHz	XX
12	93168.4MHz	128	-1250 kHz	160000 kHz	XX
13	90419 MHz	1	152000 kHz	152000 kHz	XX
14	93039.1MHz	1	-152000kHz	152000 kHz	XX
15	90419.6MHz	128	1250 kHz	160000 kHz	XX
16	93038.4MHz	128	-1250 kHz	160000 kHz	XX
17	227339 MHz	1	145990 kHz	145990 kHz	XX
18	230718 MHz	1	-145990kHz	145990 kHz	XX
19	227339 MHz	128	1250 kHz	160000 kHz	XX
20	230717 MHz	128	-1250 kHz	160000 kHz	XX
21	227469 MHz	1	152000 kHz	152000 kHz	XX
22	230588 MHz	1	-152000kHz	152000 kHz	XX
23	227469 MHz	128	1250 kHz	160000 kHz	XX
24	230587 MHz	128	-1250 kHz	160000 kHz	XX
25	227599 MHz	1	152000 kHz	152000 kHz	XX
26	230458 MHz	1	-152000kHz	152000 kHz	XX
27	227599 MHz	128	1250 kHz	160000 kHz	XX
28	230457 MHz	128	-1250 kHz	160000 kHz	XX
29	227729 MHz	1	152000 kHz	152000 kHz	XX
30	230328 MHz	1	-152000kHz	152000 kHz	XX
31	227729 MHz	128	1250 kHz	160000 kHz	XX
32	230327 MHz	128	-1250 kHz	160000 kHz	XX

This listing has simultaneously too much and too little detail for an end user. In the source list, it would be useful to have flux estimates and indications of which are the bandpass calibrators, the flux calibrators, and the phase and amplitude calibrators. The spectral information is also at a more fundamental level than needed to reduce the data. An experienced user needs at least five minutes to understand it. The end user merely needs to know that there have been continuum and/or line observations at 1mm and/or 3mm using the USB and/or LSB.

It is also difficult to know what is the state of the data. Has it been flagged? Has it been calibrated? Where is the calibration information? How can some of this information be re-used?

A.5 AIPS++ flexibility and stability

Flexibility and stability are here separated from interactivity because they have more to do with the overall design of the reduction software than with its interface to the outside world.

A.5.1 Flexibility

Flexibility may be defined as the ability to do something for which the software has not been initially designed. The flexibility of AIPS++ is high in theory, as may be seen from the number of available tools. In practice, there are serious deficiencies in spectral analysis, in particular the inadequate multi-frequency synthesis behavior for lines. These deficiencies will need to be addressed if AIPS++ is to be used for ALMA data.

A.5.2 Stability

All testers experienced a reasonable number of glitches, hangs, and crashes. Some of these were inside the `iramcalibrator` algorithms that have been the subject of heavy recent development; indeed, these bugs were often quickly corrected, explaining why all testers succeeded in finishing the calibration of their data. However, many of them were in preexisting tools of AIPS++ (flagging in `msplot`, `imager`, `viewer`). There is also a steady stream of apparently benign errors with messages registering in either the log or the Glish command line window. Moreover, AIPS++ does not seem to gracefully handle user interruption of many of its tasks: the user may end up in an incoherent state, and some processes may continue to run even after AIPS++ is exited. This is unfortunate when such tasks take a long time to finish. Some errors are hard to reproduce. The messages are often cryptic and sometimes indicate the wrong level of severity.

The overall impression is disconcerting for an inexperienced user: it's hard to tell if the software is doing something wrong, if the user made a mistake (*e.g.* typo), or if the only mistake is the report of an error. In a stable package, it is usually safe to assume that the user has just made a typo. Such confidence is impossible for AIPS++, as real bugs and crashes were obvious. One of the testers noted that the number of crashes decreases with one's ability to use the tools, probably because one has learned to avoid forbidden combinations of commands. This is bad news for occasional users, and it is certainly discouraging for people trying to learn how to use the package.

A.6 Support

In general, support has been very good during Phase II. The AIPS++ installation process from binaries is an easy, but long (several hours) process. This process is also sensitive to the user environment (*e.g.*, to which version of the LessTif libraries is used). The AIPS++ tutorial was excellent. The cookbook has been essential and should be completed with a section about line imaging. Bug fixes and improvement implementations have been quick and efficient. Modifications needed for the mosaicing data set have been implemented in collaboration between IRAM and AIPS++. The data sets have been furnished mostly on time. Information about the flagging philosophy used to reduce PdBI data inside GILDAS could have been more detailed.

B Questionnaire of Andrew Baker

B.1 Tester

1. What kind of experience have you with:
 - (a) **data reduction:**
millimeter interferometry (AIPS Classic, DIFMAP, GILDAS, MMA, some MIRIAD); single-dish millimeter bolometry (MOPSI, NIC) and millimeter/centimeter spectroscopy (CLASS); optical and near-IR imaging and spectroscopy (IRAF)
 - (b) **radio interferometry:**
entirely millimeter interferometry (see below)
 - (c) **mm radio interferometry:**
extensive observing and data reduction experience at OVRO; decent amount of experience with reducing PdBI data
 - (d) **data from Plateau de Bure Interferometer PdBI:**
decent amount of experience with single-field (non-mosaic) projects; I usually calibrate data in CLIC, edit in DIFMAP, and map in AIPS Classic
2. What are the reduction packages you are used to?
see above
3. What was your level of experience with AIPS++ before the test?
I participated in a one-day tutorial at ESO in January 2003, led by Steve Myers and Debra Shepherd.

B.2 Data set

1. Project ID: LD42
2. Project kind:
single field mapping of a spatially compact (extragalactic) source with a large velocity width; CO(1-0) line in 3 mm USB, CO(2-1) line in 1 mm LSB; no continuum emission
3. Number of days observed: 1
4. Number of days usable: 1
5. Did you succeed to obtain a believable results inside AIPS++?
No, mainly because I was not able to start on this exercise early enough or devote as much time to it as proved to be required.

B.3 Calibration inside AIPS++ by the iramcalibrator tool

1. Is it easy to know the list of sources, calibrators and the correlator setup?
I looked at the output of msplot in the log window to get this information. The list of sources and calibrators is obvious, although it would be nice if (like CLIC) AIPS++ could also provide a rough indication of flux densities in this source list; such information is useful in choosing which source(s) to use as bandpass, phase, and/or amplitude calibrators. Understanding the correlator setup was trickier; I was initially alarmed to see 32 spectral windows listed for this dataset (a pale yellow “Warning” in the log window that “Specified spectral windows have a differing number of channels. Defaulting the number of channels to the smallest value.” was not terribly reassuring). Eventually I figured out that the data were fine, but that I was simply seeing a more fundamental level of detail than I am used to from CLIC.
2. Is it easy to have an idea of the quality of the observation run (weather conditions, pointing information, Tsys, and so on)?
It was not clear to me how to access the relevant variables (they do not seem to be listed as options in the X vs. Y plot menus of msplot). The closest I could come was to plot amplitude vs. time,

which confirmed that there was no disastrous rise (due to an opacity spike) anywhere in the track, but was not otherwise useful.

3. Had you to flag out some data? Was it easy?

Running the shadow function was trivial and did not appear to flag any data from this dataset. In `msplot`, I flagged interactively in plots of amplitude vs. channel for each of the four sources and each of the sixteen multiple-channel spectral windows. In the vast majority of the windows, one or more of the first three channels needed to be flagged out; the only serious glitch was for source 2/window 4, for which no data were plotted and the log window reported "...subset has F rows." and delivered a "severe" warning that "No data selected - selection criteria is too narrow." Since this was an isolated event, I attribute it to a problem with the data rather than with the software. Flagging everything by hand was rather tedious, but I was reluctant to use the flagger tool blindly, or switch back and forth between flagger and `msplot` (this seemed like begging to lose track of what step I was at), or leave both flagger and `msplot` open at the same time addressing the same data (this sort of situation seemed to cause problems during the tutorial). The ability to do flagging with an intermediate level of interactivity (e.g., determine bad channels for one source/window and apply to the same window for all other sources) in a single tool would be useful. A question for IRAM pundits: is CLIC doing this variety of spectral flagging automatically?

During the interactive flagging in `msplot`, several (apparently non-fatal) error messages appeared in the glish command line window with no corresponding messages in the log window. These fell into several categories:

- - error, <fail>: operand to `.nrow` is not a record
File: `ms.g`, Line 156
- - error, <fail>: operand to `.printworld` is not a record
File: `msplot.g`, Line 1737
- error, non-numeric array index: X
error, non-numeric array index: Y
error, numeric and non-numeric types mixed

4. Did you have any difficulty doing the calibration steps listed below? If so, why?

(a) Phase atmospheric correction.

I think there were no problems, although it would be helpful if the cookbook could illustrate how log window messages should look in the event that non-phase-corrected data are adopted.

(b) Bandpass.

I referred to the notes from my previous CLIC calibration to confirm which source (3C273, the strongest available) should be used as the bandpass calibrator. Inspection of the output plots revealed that the "degphase" parameter needed to be increased from the default (12) to ~ 18 (this is consistent with the CLIC calibration). At 1 mm in both LSB and USB, the "fit" curve fails (by a constant offset of $\sim 15^\circ$ to pass through the data points for baselines 3-4, 3-5, and 4-5; this problem does not appear in the CLIC plots.

(c) Phase calibration.

I referred to the notes from my previous CLIC calibration to confirm which source (1633+382, the stronger of the two options) should be used as the phase calibrator. My initial pass through phase cal was not a success, because I forgot to switch "phasetransfer" and "rawspw" when calibrating the 1 mm data. When I did remember and repeated the calibration, the log window produced a "severe" error ("Caught an exception") because AIPS++ found an existing copy of a `.phase.log` file. I clicked "done" to close "myiramcalibrator", which caused all the GUIs to vanish and the glish command line window to hang (after coughing up one last message: "warning, event glishtk.fail (<task:2>) dropped
LocalExec::SetStatus: abnormal child termination for
/home/ajb/aips2/linux/bin/glishtk"). CTRL-C and a restart were successful.

I then discovered that for the 3mm-LSB data, the default parameters `npointaver = 10` and

phaseswap = 250 gave disastrously poor fits on nearly all baselines. I searched rather randomly for a better combination than (10,250), using the catalog GUI to delete each experimental .phase.log and .gcal file to avoid the problems described above. Eventually I found that (1,180) led to relatively limited phase jumps on only four baselines (2-3, 2-5, 3-4, 4-5). This was not an idle exercise, since (according to the warning messages I was receiving in the log file) the propagation of 3 mm phases to 1 mm data will currently only work using the LSB 3 mm data, even when (as for this dataset) the 3 mm USB data have better quality. The 3 mm USB data presented no problems with default parameters (10,250); the 1 mm data in both sidebands showed phase jumps on baseline 1-3 for both (10,250) and (1,180).

(d) **Flux calibration.**

No problems here, once George Moellenbroek explained how to deal with datasets spanning the 24:00 = 00:00 transition in the time coordinate.

(e) **Amplitude calibration.**

No problems here.

5. How did you check the quality of the calibrations? Was it easy?

I checked the appearance of the standard output plots at each step of the process; this was quite straightforward.

6. Is there a final summary of the whole calibration process? Is it useful?

Provided the calibration fits look reasonable along the way, I do not see the need for a final summary.

7. List of bugs found during the calibration. Which ones have been corrected? Which ones are still pending?

I submitted 3 bug reports through the end of my calibration of the data; these are listed verbatim below. Since I only submitted these hours before writing this report, it is not surprising that they are all still pending.

– **error message regarding leap second table**

When I run `almati2ms` on PdBI data which I converted to ALMATI format with the 30-jan-2003 version of the GILDAS filler, I see error messages which are flagged as “severe” (highlighted yellow) in the log:

Leap second table TALUTC seems out-of-date.

Until table is updated (see `aips++ manager`) times and coordinates derived from UTC could be wrong by 1s or more.

The measurement set nevertheless seems to finish creating correctly, so I have (uncertainly) tagged this as “non-serious”.

– **mycatalog GUI hangs after selection of functions**

In hunting around for catalog functions, I went to utility → catalog → create, then created a tool “mycatalog”, and then found that the GUIs hung as soon as I clicked on a function in the mycatalog window (well, at least if I clicked on function “list” or “gui” – I didn’t think it was worth trying more). This is not so serious, as I can still type “exit” at the glish command line and restart.

– **msplot shows same uv coverage regardless of frequency**

In working with dual-frequency PdBI data, “Plot UV Coverage” in `msplot` shows exactly the same uv coverage for data with a 3mm spectral window selected as for data with a 1mm spectral window selected. Shouldn’t these differ by the ratio of the frequencies (in this case, a factor of 2)? Using `u` and `v` as variables in “Plot X vs Y” has the same problem.

8. List of improvements asked. Have they been implemented?

I asked for one improvement, namely the capability to select time ranges in the `iramcalibrator` “flux” function which crossed 24:00 = 00:00 in time. George Moellenbroek answered within five (!) minutes that this was already implemented and explained the necessary syntax (23:00 29:00, rather than

23:00 05:00).

B.4 Imaging inside AIPS++

1. What degree of mm interferometry expertise do you need to obtain your first unoptimized clean image?

I was not able to produce a clean image, the result of lack of time coupled with lack of familiarity (particularly with spectral line imaging) with AIPS++. One obstacle was my inability to work out on my own how to concatenate the four spectral bands spanning the CO(1-0) line (or those spanning the CO(2-1) line) into a single spectral window. I would normally perform this function with the AIPS Classic task UVGLU, which is not listed in Appendix 2 of the cookbook or in an obvious location in the online documentation. I tried in the end to map just a single spectral window (across part of the line), using the somewhat more informative VLA AIPS++ cookbook as a guide; unfortunately AIPS++ crashed with a segmentation fault shortly after beginning a shallow clean of the 124-channel cube. I have no doubt that advice from the programming team would eventually have allowed me to overcome these hurdles, but going through this exercise at the last minute did not permit extensive consultation.

I can also answer this question in a more general sense: if I had not known in advance how to reduce interferometric data, the difficulty of navigating through the AIPS++ user interface would also have posed a significant obstacle to developing an *understanding* of how to reduce data. I would quantify this impression by saying that the number of mouse clicks required to produce a map seems significantly larger than the number of major conceptual steps involved, an attribute of AIPS++ which I fear would lead interferometry novices to miss the forest for the trees.

2. Did you try several deconvolution methods? Why? If yes, is it easy to swap from one method to another one?
Not applicable.
3. Did you succeed to finely tune the deconvolution? Why? What is easy?
Not applicable. I was working with the Hogbom clean.

B.5 First analysis and production of a publication quality plot inside AIPS++

1. Is the viewer tool easy to use? Why? Did you succeed to make a publication quality plot?
Not applicable. I would note that I usually make publication-quality plots in WIP, so I was a bit worried about FITS-writing capability until I realized that AIPS Classic task FITTP should (in Appendix 2 of the cookbook) be mapped to AIPS++ image.tofits as well as to ms.tofits.
2. Did you succeed to compute the noise on the continuum images or line cube?
Not applicable.
3. Did you succeed to compute the moments of your line data?
Not applicable.
4. For simple sources (i.e. detection), is it easy to make fits in the uv plane?
According to Appendix 2 of the cookbook, the UVFIT functionality is not implemented in AIPS++.

B.6 Comparison of results obtained with AIPS++ and CLIC (Optional)

1. What is the final RMS in the calibrated target image?
Not applicable.
2. What is the peak/noise ratio in the image?
Not applicable.

3. Discuss any differences you found between calibration solutions along the way.
The CLIC passband fits seem somewhat more plausible than the AIPS++ fits, in terms of “fit” curves passing through the data points.
4. Discuss any differences between the CLIC and AIPS++ images.
Not applicable.

B.7 General

1. What is the degree of flexibility of AIPS++?
I think this question will be best answered by evaluating how well AIPS++ has been able to cope with the variety of PdBI datasets reduced in Phase II. From the point of view of reducing a single dataset, I don’t think “flexibility” is a very useful diagnostic.
2. What is the degree of interactivity of AIPS++?
I would like to have a clear definition of what “interactive” means before I try to answer the question of whether AIPS++ is or isn’t. Is the question *can* I interact with it, do I *have* to interact with it, or do I *want* to interact with it?
3. Does it seem easy to test several hypothesis during the calibration and imaging?
I have not been able to evaluate this aspect of the software.
4. Does AIPS++ seem to be stable enough?
As the perhaps overly detailed account above indicates, I encountered a reasonable number of glitches, hangs, and crashes during the course of (partly) reducing a simple dataset. Not all of these were directly related to the iramcalibrator algorithms which have been the subject of heavy recent development. I was also aware of a steady stream of apparently benign errors (some noted above) with messages registering in either the log or the glish command line window. The overall impression is disconcerting for an inexperienced user: it’s hard to tell if the software has made a mistake, if I’ve made a mistake, or if the only mistake is the report of an error. I guess I would be more confident about eventually mastering the software if the signatures of errors I encountered did not come in such a bewildering variety.
5. Are the tools quick enough for efficient interactive use? If not, which ones should be improved first?
I was able to fill out a substantial fraction of this questionnaire while waiting for msplot to finish plotting source visibility amplitude vs. channel for interactive flagging. This tool’s functionality urgently needs to migrate to a faster vehicle. In other areas, I can only say that speed of execution was not the primary factor limiting my own rate of progress.

B.8 Support

1. Data sets:
 - (a) Had you the data sets in time?
Yes.
 - (b) Had you enough information about the data sets?
Yes. Given that these were my own data, which I had already calibrated in CLIC, this was not a problem.
 - (c) Had you enough information about PdBI calibration methods?
Yes, from past experience.
2. Installation of AIPS++:
 - (a) Was it easy?
Installing Release 1.7 from CD was trivial; however, after running `aupdate` to download the most recent stable build, I was left with a version of AIPS++ which aborted on takeoff. From my

forwarded diagnostic messages, Wes Young deduced that the problem was the old gcc compiler in my SuSE Linux 7.1 installation. After locating another machine in Garching running SuSE Linux 8.1, I tried again, only to find that the installed versions of the lesstif and tcl/tk libraries were too *new* for `aupdate` to handle. Wes kindly provided the older versions of these libraries, and I was able to proceed from there. The longest step in this process was finding the SuSE Linux 8.1 machine (this required waiting until a colleague had departed MPE for a new job); solving the other problems took less than 48 hours total.

(b) **Were the instructions good?**

Yes (once one is using the right combination of operating system and libraries, use of `aupdate` is trivial). I appreciate Wes's generosity in helping me install AIPS++ on what was a foreign system in more ways than one.

3. **Bug corrections and improvements into the `iramcalibrator` tool: was it quick?**

Not applicable, although see above for George's light-speed response to my one implementation request. I started heavily reducing my dataset in AIPS++ too late in the game for my experience to be a fair test of response to bugs.

4. **Documentation:**

(a) **Was the tutorial good?**

Yes, extremely useful. I would not have ventured to experiment in AIPS++ without it.

(b) **Is the `iramcalibrator` cookbook complete?**

More detail on spectral line work would be useful. I appreciate the enormous labor which has already gone into bringing it to its current level.

(c) **Is the general AIPS++ documentation easily understandable?**

I do not find the web-based documentation (at least, the bits which my browser is "pulled" to) to be terribly congenial. I would say the main difficulty is that I am trying to understand how to work with the GUI, but the manual seems mainly to provide information about the (underlying?) Glish.

5. **What are the parts for which you would have like more support?**

spectral line imaging; operation of AIPS++ in GUI mode rather than in Glish mode; some sort of index or organization for the documentation by subject matter rather than by module

B.9 Other remarks

- Time spent: 15 hours total on this exercise, not including the time spent installing the software.
- I was using AIPS++ version "18.499.00 Sat 2003/02/15 01:30:23 GMT" for this exercise.
- Because of a delay in installing a working version of AIPS++ in Garching (until February 19th) and time pressure from other activities, I did not manage to tackle the test dataset until shortly before the end of the testing period. As a result, I do not consider my experience to represent a fair or useful test of AIPS++ user support.

C Questionnaire of Alain Coulais

C.1 Tester

1. What kind of experience have you with:
 - (a) **data reduction:**
Infrared:ISO data (CAM with CIA, ICE and SLICE, PHT with PIA, SWS with SIA, LWS with LIA); mm radio interferometry: IRAM PdBI and 30m; Solar physics: Nançay radioheliograph, Yohkoh satellite in X-rays.
 - (b) **radio interferometry:**
PhD in radio-interferometry and deconvolution methods; experiences with Nançay RadioHeliograph and IRAM PdBI.
 - (c) **mm radio interferometry:**
experiences with IRAM PdBI; writing of a ALMA simulator with F. Viallefond in Gipsy in 1997-98.
 - (d) **data from Plateau de Bure Interferometer PdBI:**
one project in Orion Horsehead processed.
2. What are the reduction packages you are used to?
Infrared: ISO data with IDL(CAM with CIA, ICE and SLICE, PHT with PIA, SWS with SIA, LWS with LIA); mm radio interferometry : IRAM PdBI and 30m with GILDAS and IDL; solar physics : Nançay radioheliograph (IDL, Fortran), Yohkoh satellite in X-rays
3. What was your level of experience with AIPS++ before the test?
I was involved with the AIPS++ Audit during 02' Summer.

C.2 Data set

1. Project ID:
KH44
2. Project kind:
3. Number of days observed:
7 days
4. Number of days usable:
I don't know. Because of problems with preprocessing (FITS conversion) and difficulties in AIPS++, I have spend most of my time on one data set: the 16feb' one.
5. Did you succeed to obtain a believable results inside AIPS++?
I think the outputs that I have obtained are wrong and I don't believe them. I finally succeed to compute channel maps but I have no confidence in them.

C.3 Calibration inside AIPS++ by the iramcalibrater tool

1. Is it easy to know the list of sources, calibrators and the correlator setup?
It is possible to know the name and the relation between name and number. But sometime we use the number (`mplot`) and others times the name (`iramcalibrater`, ...), sometimes is it absolute, other time relative (`imager`). Without extra a priori informations or having a look in the data, it is not possible to distinguish between data and calibrations. I found very difficult to understand the status of the correlator in the current way the informations are provided. If we start from a partially pre-processed data set, it is hard to know or the re-use the useful informations.
2. Is it easy to have an idea of the quality of the observation run (weather conditions, pointing information, Tsys, and so on)?
Not found. I don't know.

3. Had you to flag out some data? Was it easy?
Yes. If you have time to spent to wait for the plots, yes, you can ... But it is so slow that I loose energy to check each configuration. Furthermore, `mplot` did not accept several attempts I would like to do ... I found that after flagging, it is hard to know what are the states of the data. I don't like the flagging strategy. I prefer to remove directly instead of flagging. I found that this way strongly speed up the processing. It is more convenient and faster when flagging iteratively.
4. Did you have any difficulty doing the calibration steps listed below? If so, why?
 - (a) Phase atmospheric correction.
No problem (but no options !)
 - (b) Bandpass.
First times, I was in trouble with `FieldName`, since previously (in `mplot`) we use `FieldID` ... Yes, very strange behavior. Changing the `GIBB` parameter leads to crash.
 - (c) Phase calibration.
No problem, but I did not check the options
 - (d) Flux calibration.
How to know the references ? I did not check the options
 - (e) Amplitude calibration.
No problem (but no options !)
5. How did you check the quality of the calibrations? Was it easy?
On the Postscript plots. I am not sure the quality is OK.
6. Is there a final summary of the whole calibration process? Is it useful?
7. List of bugs found during the calibration. Which ones have been corrected? Which ones are still pending?
I have done the `upgrade` on Friday 21 Feb. I did not find a reproducible bug that I had during Garching' week, for which I have submitted a bug report: this bug should be corrected ! They should be a reply to bug report. I don't know if I did not get it because using a different install. During summer 2002' Audit, I have submitted a reproducible bug in `Glish` (`a:=1:5; a::shape:= [0,5]; a ; no response, 100% CPU ...[0]0x80536c0 fatal internal error (glish), out of memory in append_buf()`) and still no correction (you can try !) ...
8. List of improvements asked. Have they been implemented?
I have asked for several improvements during the week in Garching (cf bugs reports and discussions with experts, about `mplot` and `viewer`). I am sure they are not done.

C.4 Imaging inside AIPS++

1. What degree of mm interferometry expertise do you need to obtain your first unoptimized clean image?
not sure to understand. I have channel maps, but I think they are wrongs. I have had serious troubles with `imager` when using GUI. Scripts are much more reliable.
2. Did you try several deconvolution methods? Why? If yes, is it easy to swap from one method to another one?
not done this time. I was concentrated on my channel maps ! See Audit.
3. Did you succeed to finely tune the deconvolution? Why? What is easy?
not done.

C.5 First analysis and production of a publication quality plot inside AIPS++

1. Is the viewer tool easy to use? Why? Did you succeed to make a publication quality plot?
Please see what we have written for the AIPS++ Audit. The `viewer` tool is very slow, very hard to

configure, not well adapted to mm maps and channels. Please have a test with GILDAS MAPPING and CLIC. It was funny during the Garching week to see that the experts were not able to show me how to preconfigure the `viewer` tool, and how to use the multiplot in `viewer`. It is too complex and too slow.

2. Did you succeed to compute the noise on the continuum images or line cube?
not done
3. Did you succeed to compute the moments of your line data?
not done
4. For simple sources (i.e. detection), is it easy to make fits in the uv plane?
(1) I cannot figure out how to do such a fit in AIPS++ since all the packages in the tool manager appear to me at the same level. (2) I have crashed my whole system (AIPS++ frozen, Linux X server down) when try to read MS set in `viewer`. If it is not OK to do it, the system must inform me !

C.6 Comparison of results obtained with AIPS++ and CLIC (Optional)

1. What is the final RMS in the calibrated target image?
not done (because I think the maps are wrong.)
2. What is the peak/noise ratio in the image?
not done (because I think the maps are wrong.)
3. Discuss any differences you found between calibration solutions along the way.
not done
4. Discuss any differences between the CLIC and AIPS++ images.
not done

C.7 General

1. What is the degree of flexibility of AIPS++?
Rather high in theory but not in practice. Unfortunately, we can do what we should not do. We are supposed to be close to the data, but it is very slow to plot them, and we cannot manipulate them as I like to do in IDL. It is very hard for me to know the state of my data, what has been done or not. I did not understand why it is possible to do the same task several times without warning, or to do tasks in disorder without warning. Using scripts is much more efficient and stable compared to GUI.
2. What is the degree of interactivity of AIPS++?
e.g. : `imager` : all the tools are at the same level and, without a priori informations, it is not possible to figure out what are the steps when going from a MS to images. Without switch on the "all" choice, it is terrific to navigate through all the possibles choices. With my data using GUI, I never succeed to have images; I got messages like `Frequency = 1.68244e+232 GHz, synthesized continuum bandwidth = 0.152 GHz` when using GUI (not with scripts). How is it possible ? I am not confident with tools which send such messages. The processing was adequate through scripts. In several steps, minimal checks are not done. E.g.: I can add several time the same spw in `uvt.ms`.
3. Does it seem easy to test several hypothesis during the calibration and imaging?
To go end to end without crash was my main goal. Even with the tutorial and several partials scripts we use for PhaseIII, I did not succeed to transpose them reliably for my specific case.
4. Does AIPS++ seem to be stable enough?
AIPS++ seems to be much more stable now. During the 2002' summer Audit, it was crashing several times per day. During this Phase2, only 4 or 5 big crashes during a full week (and very few crash in Garching, except all these crashes during tests for Benchmarking with 32 and 64 antennas

probably due to a lack of memory), but now I have strongly increase my ability to use this tool. I would like to mention that I am tired to report bugs or crashes or strange behaviors (`viewer.canvas`, `imager.setdata`, ...).

5. Are the tools quick enough for efficient interactive use? If not, which ones should be improved first?

Absolutely not. The speed in `mplot` and all the `viewer` tools are several magnitudes below Gipsy' and Gildas' equivalent tools. Preliminary phaseIII Benchmark also shows that the core procedures are slow in comparison to GILDAS. Please have a look to the `VIEW`, `MVIEW`, `3VIEW` tools in Gipsy. It is very hard to find/change/restore the options in `viewer`.

C.8 Support

1. Data sets:

- (a) Had you the data sets in time?

A problem was finally detected in the FITS files. I loosed time because of that.

- (b) Had you enough information about the data sets?

I think so. Furthermore, usually, I like to go by myself in the data, to manipulate them and to see where are the problems, how good they are, and so on. I like to do `Histo(data.cube)` in IDL to see immediately the statistical distribution of the data. Typing and seeing the output is immediate in IDL, I know I will need several minutes or more to do it in AIPS++.

- (c) Had you enough information about PdBI calibration methods?

I think so. One problem with my data set is because of the new correlator, then the appearances I am waiting for are different than in the Cookbook.

2. Installation of AIPS++:

- (a) Was it easy?

Yes and no : the directory `data/` was missing after standard installation. Fortunately, M. Caillat knows that usually this dir. is missing ... The aupgrade works fine, but it takes several hours on my station.

- (b) Were the instructions good?

Since `data/` was missing, it is a problem.

3. Bug corrections and improvements into the `iramcalibrater` tool: was it quick?

4. Documentation:

- (a) Was the tutorial good?

This tutorial was mandatory. For me, it would be quite impossible to do something without it and the cookbook. With the help of the cookbook, the tutorial and the scripts we have set up for the benchmark, I know that the problem does not come from me. Usually, when processing data, I can easily transpose the scripts to new cases. It was not the case here. (I have practice the tutorial during the Garching' week, and we (D. Broguiere and me) have written scripts for benchmarking.)

- (b) Is the `iramcalibrater` cookbook complete?

This cookbook is a brilliant document, but it is not complete. Only one case is studied. It is very hard to generalyse. Spectral imaging is not covered in the cookbook. E.g. the options `spwid` and `nchannel` don't have same meaning in `imagers.setdata` and `imager.setimage`. I don't know how to plot a spectral slice through my images. Furthermore, the IRAM new correlator provides much better quality data, then figures should be adjust to the new case. Mosaicing is not adressed.

- (c) Is the general AIPS++ documentation easily understandable?

No. It was noticed during the Audit that a number of examples in AIPS++ documentations are not reproducible or self consistent (e.g. you need to have the data in this dir, you need to

have loaded this module, ... When you start AIPS++ and copy & paste the examples, quite frequently they did not work). Fortunately, the search tool is powerful !

5. What are the parts for which you would have like more support?

C.9 Other remarks

- Time spent: 5.5 full time days.
- (1) We need to have a GUI similar to what F. Viallefond is doing for the simulator in order to help the beginners to process, to avoid to forbidden things, and to have automatic and simple steps when it is easy (e.g. creating ut tables from spw, imaging, ...). I also prefer the @DEFINE in CLIC, where the steps for going through calibration steps are clearly visible. (2) Here we have only 6 antennas and 128 channels. The plots are very slow (please compare to Gipsy, Gildas, IDL, Yoric, ...). It is not possible to live with current AIPS++ speed for 64 antennas and more than 1024 channels. (3) For sure, if I had to process today PdBI (or ALMA !) data, I would prefer by far to use Gildas or my own environment (after doing calibration in Gildas) and would not like to use AIPS++. No gain, only loose (time, energy, confidence ...).

D Questionnaire of Frédéric Gueth

D.1 Tester

1. What kind of experience have you with:

(a) **data reduction:**

I have observed and reduced data from various radio telescope, including the IRAM 30m, the IRAM Plateau de Bure interferometer, the MPIfR Effelsberg 100m dish, the CSO, the CHAMP multi-beam installed at the CSO. I have also participated in data reduction software developments within GILDAS, mostly for the Plateau de Bure interferometer and for CHAMP.

(b) **radio interferometry:**

See next question.

(c) **mm radio interferometry:**

I have been working on the IRAM Plateau de Bure interferometer for years, since the beginning of my PhD in 1993. This includes astronomical work with PdBI data, user support, AoD, some technical studies, lectures at summer schools, and many software developments.

(d) **data from Plateau de Bure Interferometer PdBI:**

See previous question.

2. What are the reduction packages you are used to?

Mostly GILDAS.

3. What was your level of experience with AIPS++ before the test?

I had a few presentations and tutorials during the last years. Then I was involved in the Phase I test, got several tutorials, and exercised myself with AIPS++.

D.2 Data set

1. **Project ID:**

I028 – I wanted initially to also reduce I139 but had no time to do both.

2. **Project kind:**

Mapping, Mosaic, 1 and 3 mm, line and continuum. 1 mm data is of poor quality.

3. **Number of days observed:**

9

4. **Number of days usable:**

8

5. **Did you succeed to obtain a believable results inside AIPS++?**

I managed to go through the calibration successfully. I obtained a continuum image only today, with Kumar's help.

D.3 Calibration inside AIPS++ by the iramcalibrator tool

1. **Is it easy to know the list of sources, calibrators and the correlator setup?**

The Summarize button in the Catalog tool gives access to the list of source and spectral Id. This is fairly easy to do, although it is surprising to have to use the Catalog tool to do that. As for the lists themselves, I have the following comments:

- The list does not give the fluxes of the sources.
- AIPS++ makes no difference between astronomical targets and calibration sources; while it may be argued that this is a satisfactory general scheme for some cases, it is not adapted to the PdBI case.
- The spectral Id list is rather confusing, especially because the LSB and USB bands are listed as separate spectral windows.

- In most applications, one has to select the source and the spectral Id by giving the corresponding number in the list. This is very heavy! A noteworthy exception is the iramcalibrator tool.
2. Is it easy to have an idea of the quality of the observation run (weather conditions, pointing information, Tsys, and so on)?
As far as I know, it is not possible.
 3. Had you to flag out some data? Was it easy?
Yes, I flagged a few points. This was reasonably easy using the `msplot` tool, after having had a tutorial (without the tutorial, I doubt I could have done it). I still don't know how to write a script doing the flagging, which is in my view a key feature that is required for any pipeline processing.
 4. Did you have any difficulty doing the calibration steps listed below? If so, why?
Calibration was done reasonably easily. I noted however the following points:
 - Not possible to know which source is to be used at each step (I had e.g. to check the CLIC report to know the RF bandpass calibrator).
 - Plots are produced as postscript files and are not displayed by the software. Idem for the phase and amp fit rms (ascii files).
 - The flux plot is quite confused, because all antennas are plotted simultaneously.
 - There are no possibilities to add breaks in the fitted curves (this is an improvement that would be desirable, but has low priority).
 - (a) Phase atmospheric correction.
 - (b) Bandpass.
 - (c) Phase calibration.
 - (d) Flux calibration.
 - (e) Amplitude calibration.
 5. How did you check the quality of the calibrations? Was it easy?
I checked the ps files (why is the date not written?) as well as the phase and amp fit rms reports (why are they ascii files and not displayed in the logger window ???). Since these files try to mimic the CLIC results, I managed to understand them, but to judge the calibration results, I'm lacking some easier way than going in another window to open gv and emacs.
 6. Is there a final summary of the whole calibration process? Is it useful?
There are a few postscripts files and an ascii file with the various fit rms. But there is no report.
 7. List of bugs found during the calibration. Which ones have been corrected? Which ones are still pending?
I met a number of problems, but it's hard to distinguish between real bugs and features (e.g. never add a "/" to the directory name in `almatifiller!`). The two major bugs that were found are:
 - The mosaic offsets were lost in the ipb-to-ms file transfer. This was traced back to (a) CLIC exchanging the RA and DEC offsets, and (b) `almatifiller` not reading the offsets at all. The first problem was rapidly fixed by Robert Lucas. The second problem was investigated and fixed by Dominique Broguiere at IRAM.
 - The `vpmanager.setpbgauss` method does not work in the AIPS++ version that I'm using. This function produces the primary beam model to be used by the imaging procedure. When called from `glsh`, I'm getting a bunch of cryptic messages:


```
error, <fail>: operand to .direction is not a record
File: vpmanager.g, Line 318
Stack:  .() is not a function value
<fail>: operand to .direction is not a record
File: vpmanager.g, Line 318
Stack:  .()
```

 When called from the GUI, it seems to be happy, but has done nothing. In both cases, you can save a table (using `vpmanager.saveastable`) and run CLEAN without AIPS++ complaining. Of

course, CLEAN gets into deep troubles (imager crashes but is still running in the background, so one has to exit AIPS++ and kill the imager process). The final word is that I used a table produced and sent to me by Kumar.

8. List of improvements asked. Have they been implemented?

D.4 Imaging inside AIPS++

1. What degree of mm interferometry expertise do you need to obtain your first unoptimized clean image?
I'm afraid a lot. First because I had to fight against the vpmanager. And then, of course, because one has to know which imager functions are to be used among the dozens that are proposed, and how.
2. Did you try several deconvolution methods? Why? If yes, is it easy to swap from one method to another one?
I tried several flavors of CLEAN. This was reasonably easy (just a clic), providing that you think about changing the names of the model file to give a new filename! Otherwise, strange things appear.
3. Did you succeed to finely tune the deconvolution? Why? What is easy?
I didnt try fine tuning the deconvolution.

D.5 First analysis and production of a publication quality plot inside AIPS++

1. Is the viewer tool easy to use? Why? Did you succeed to make a publication quality plot?
The viewer is ok for continuum images, but very difficult to use for data cube. Changing plotting options is rather complex, as it requires very rapidly three different windows. There's little hope to produce a real publication quality plot with this tool: customization of the plot (eg adding text, symbols) is complex; making complex figures (eg two different dataset) seems to me impossible.
2. Did you succeed to compute the noise on the continuum images or line cube?
Not tried, but I think it is doable.
3. Did you succeed to compute the moments of your line data?
Not relevant.
4. For simple sources (i.e. detection), is it easy to make fits in the uv plane?
Not relevant for my project. But I know that UV fits are not implemented in AIPS++. Dominique Broguiere has plugged the UV_FIT GILDAS task into AIPS++, but this was mostly for test purposes.

D.6 Comparison of results obtained with AIPS++ and CLIC (Optional)

1. What is the final RMS in the calibrated target image?
2. What is the peak/noise ratio in the image?
3. Discuss any differences you found between calibration solutions along the way.
4. Discuss any differences between the CLIC and AIPS++ images.

D.7 General

1. What is the degree of flexibility of AIPS++?
AIPS++ looks like a large collection of tools, which of course may be described as a very flexible system. However, it seems to me quite difficult for a newcomer to use this system, precisely because he/she has to navigate through all tools and method to find the appropriate ones. This is difficult because:

- The list of available tools/methods are presented by alphabetic order at every level, hence mixing crucial and non-relevant methods, and giving no indication on the order in which this methods have to be used.
- There are no short cuts to do simple processing step. Typically: imaging a single field is a well-defined problem, which, at zeroth order, requires very little input parameters (most of the parameters can be set to robust default values or guessed from the dataset). But instead of one clearly identified “DO AN IMAGE” button or function, one has to call at least five different methods (to be found in the alphabetic order list). Hence, the logics that is offered to the user is very general and complex (see eg the nchan/start/step parameters in setdata and setimage, that have different meanings).
- The tools that one wants to use are to be found in different packages (eg msplot is not in the same package than iramcalibrator or imager; idem for the viewer). This forces the user to switch between many windows, coming back regularly to the highest level of AIPS++. This, I found, is a major drawback of the toolkit approach.
- The underlying object-oriented programming is systematically presented to the user, while I think it should be definitely hidden. For instance, giving a name to each tool that is created is completely useless and confusing for the user!

2. What is the degree of interactivity of AIPS++?

It seems to me quite unequal. The msplot tool is very interactive and can probably be tuned into a very efficient and convenient tool. The viewer is maybe too interactive, in the sense that it is offering so many options that the user gets lost with two to four windows. For these two tools, an associated scripting language is missing. Other tools are, I think, not interactive enough: CLEAN is close from a black box, in which the user has little access to the intermediate results, and has to run another tool (viewer) to plot the resulting image and/or select a support.

Another point worth mentioning is that the error messages are often very unclear. For instance, adding a “/” at the end of the fits directory in almatifiller results in lots of glish errors, and the following text in the logger:

```
FITS-IDI data source is not readable
Failed to create an ALMA-TI converter
Caught an exception! Event type=create
exception=Tue Mar 4 10:13:37 2003
SEVERE AlmaTI2MS::init()
(file /aips++/stable/code/alma/implement/MeasurementSets/AlmaTI2MS.cc,
 line 810):
FITS-IDI data source is not readable
```

Similarly, a typo in a calibrator source name results in many troubles...

3. Does it seem easy to test several hypothesis during the calibration and imaging?

Yes, but it seems to me difficult to compare the results of the different methods.

4. Does AIPS++ seem to be stable enough?

I experienced several crashes. While I understand that many things are unstable after a furious development cycle, I find it surprising that AIPS++ is so badly protected against typo errors or user mistakes.

5. Are the tools quick enough for efficient interactive use? If not, which ones should be improved first?

My general feeling is that AIPS++ is rather slow. This is due to the intrinsic slowness of some tools (eg mfclean took 800 sec for a continuum image!!), but also to the necessity for the user to navigate through many windows (eg going from the imager to the viewer...).

D.8 Support

1. Data sets:
 - (a) Had you the data sets in time?
Yes.
 - (b) Had you enough information about the data sets?
Yes, it's mine!
 - (c) Had you enough information about PdBI calibration methods?
Yes, I know them.
2. Installation of AIPS++:
 - (a) Was it easy?
This was done by others.
 - (b) Were the instructions good?
I guess yes.
3. Bug corrections and improvements into the iramcalibrater tool: was it quick?
4. Documentation:
 - (a) Was the tutorial good?
 - (b) Is the iramcalibrater cookbook complete?
 - (c) Is the general AIPS++ documentation easily understandable?
5. What are the parts for which you would have like more support?

D.9 Other remarks

Time spent: 4 full time days.

E Questionnaire of Jérôme Pety

E.1 Tester

1. What kind of experience have you with:

(a) **data reduction:**

- From 1994 to 1999, I participated to the reduction of several projects observed at single dish telescopes (IRAM-30m and CSO).
- Since 2000, I have been using and supporting the GILDAS softwares used by IRAM instruments.

(b) **radio interferometry:**

All my experience is with mm radio interferometry (see below).

(c) **mm radio interferometry:**

- My first introduction to mm radio interferometry was the second IRAM Millimeter Interferometry School, mid-2000.
- I taught the principles of mm radio interferometry and the details of imaging from uv visibilities during the third IRAM Millimeter Interferometry School, in Sep. 2002.
- Between those two dates, I thoroughly studied the effect of the Atacama Compact Array (ACA) on the wide-field imaging capabilities of ALMA. To do this, I set up (with F. Gueth and S. Guilloteau) an end-to-end simulator of ALMA inside GILDAS.
- I am currently part of the ALMA Science and Computing IPTs.

(d) **data from Plateau de Bure Interferometer PdBI:**

- I am Astronomer On Duty at PdBI, reducing and checking the quality of all data observed during about 3 to 4 weeks a year.
- Each year, I also am the local contact of about 5 projects accepted at PdBI, helping for the setup and the reduction of observations.

2. What are the reduction packages you are used to? GILDAS, IDL

3. What was your level of experience with AIPS++ before the test?

I have followed all phase I part of the AIPS++/IRAM test. As such, I have *i*) followed the tutorials of G. Moellenbrock about the measurement equation and Glish *ii*) played with AIPS++ and *iii*) set up a detailed comparison of the 1mm and 3mm continuum calibration inside AIPS++ and GILDAS softwares. At beginning of phase II, I followed the tutorial of D. Shepherd about flagging, calibrating and imaging PdBI data inside AIPS++.

E.2 Data set

1. Project ID: kg5a and l-1.

2. Project kind:

Detection of the high redshift ($z \simeq 4.2$) quasar PSS2322+1944. The main goal was the detection of the CO (J=4-3) line (3mm line at this redshift). However, the CO (J=10-9) had been searched but not found and the 3mm and 1mm continuum had been measured. I had been part of this project: i.e. I reduced the data with the PI (P. Cox). All results may be found in Cox et al. *A&A*, 387(2):406,2002. This kind of projects currently represent about 25% of the observations made at PdBI.

3. Number of days observed: 4

4. Number of days usable: 3

5. Did you succeed to obtain a believable results inside AIPS++?

- I succeeded to obtain a 1mm continuum image. However there were stripes that I did not succeed to eliminate.

- I did not succeed to obtain the CO (J=4-3) cube at all (main goal of the project). Indeed, the IF bandwidth of PdBI is ~ 1500 km/s large : this makes difficult to correctly center the IF frequency due to the uncertainty of the visible redshifts (about one third of these kind of project needs several tunings). In this case, the first frequency setting allowed us to discover the line wing on one edge of the IF bandwidth and two others settings where needed. Currently, AIPS++ is able to directly handle such a case only in continuum not in line.

More details can be found in the “Others Remarks” section.

E.3 Calibration inside AIPS++ by the iramcalibrator tool

1. Is it easy to know the list of sources, calibrators and the correlator setup?

The Catalog/Summarize function allows you to determine many information about your observations. Here are the relevant ones in my case:

Fields: 5

ID	Name	Right Ascension	Declination	Epoch
1	3C345	16:42:58.81	+39.48.37.00	J2000
2	MWC349	20:32:45.54	+40.39.36.64	J2000
3	3C454.3	22:53:57.75	+16.08.53.56	J2000
4	2230+114	22:32:36.41	+11.43.50.90	J2000
5	PSS2322+19	23:22:07.20	+19.44.23.00	J2000

Data descriptions: 32 (32 spectral windows and 1 polarization setups)

ID	Ref.Freq	#Chans	Resolution	TotalBW	Correlations
1	90029 MHz	1	145990 kHz	145990 kHz	XX
2	93429 MHz	1	-145990kHz	145990 kHz	XX
3	90029.6MHz	128	1250 kHz	160000 kHz	XX
4	93428.4MHz	128	-1250 kHz	160000 kHz	XX
5	90159 MHz	1	152000 kHz	152000 kHz	XX
6	93299 MHz	1	-152000kHz	152000 kHz	XX
7	90159.6MHz	128	1250 kHz	160000 kHz	XX
8	93298.4MHz	128	-1250 kHz	160000 kHz	XX
9	90289 MHz	1	152000 kHz	152000 kHz	XX
10	93169 MHz	1	-152000kHz	152000 kHz	XX
11	90289.6MHz	128	1250 kHz	160000 kHz	XX
12	93168.4MHz	128	-1250 kHz	160000 kHz	XX
13	90419 MHz	1	152000 kHz	152000 kHz	XX
14	93039.1MHz	1	-152000kHz	152000 kHz	XX
15	90419.6MHz	128	1250 kHz	160000 kHz	XX
16	93038.4MHz	128	-1250 kHz	160000 kHz	XX
17	227339 MHz	1	145990 kHz	145990 kHz	XX
18	230718 MHz	1	-145990kHz	145990 kHz	XX
19	227339 MHz	128	1250 kHz	160000 kHz	XX
20	230717 MHz	128	-1250 kHz	160000 kHz	XX
21	227469 MHz	1	152000 kHz	152000 kHz	XX
22	230588 MHz	1	-152000kHz	152000 kHz	XX
23	227469 MHz	128	1250 kHz	160000 kHz	XX
24	230587 MHz	128	-1250 kHz	160000 kHz	XX
25	227599 MHz	1	152000 kHz	152000 kHz	XX
26	230458 MHz	1	-152000kHz	152000 kHz	XX
27	227599 MHz	128	1250 kHz	160000 kHz	XX
28	230457 MHz	128	-1250 kHz	160000 kHz	XX
29	227729 MHz	1	152000 kHz	152000 kHz	XX

30	230328 MHz	1	-152000kHz	152000 kHz	XX
31	227729 MHz	128	1250 kHz	160000 kHz	XX
32	230327 MHz	128	-1250 kHz	160000 kHz	XX

All these informations are valuable. However, this very general presentation scheme has two drawbacks:

- The end user must first translate these information into more practical ones for use in calibration and imaging, i.e. *i)* which source correspond to the bandpass calibrator and to the phase and amplitude calibrators and *ii)* which spectral window correspond to continuum/line, to 1mm/3mm, to the upper/lower sidebands. The end user must then translate those practical information into integer numbers (i.e. Fields ID and Spectral Windows ID) as only those integer numbers are used in all the GUIs of AIPS++. The one worthy exception is the `iramcalibrator` tool for which IRAM has explicitly asked that the user be able to enter directly the source names and the kind of spectral windows (i.e. 1mm-USB).
- These information also are too detailed for daily use. For instance, at PdBI, the only information needed by the user to estimate the quality of the calibration is whether we are dealing with 1mm/3mm and with USB/LSB data. Although it is a bit more complex for imaging, much less information is asked to the user to image his/her data. The reduction software must obviously be able to translate between the user input and the way the data is stored. This may make the information and input schemes less general (i.e. the translation procedures must make use of instrument knowledge) but this will be much more friendly.

2. Is it easy to have an idea of the quality of the observation run (weather conditions, pointing information, Tsys, and so on)?

Currently, these information are either not filled inside AIPS++ or not used. This should quickly be corrected (it is just a question of making a few plots and putting them together into a report as a template is available from PdBI reports). Indeed, these information have two goals: *i)* being able to find potential observing problems which will make the calibration difficult and *ii)* being able to start judging the data quality. For instance, in my data set, the wind was high and the tracking of antenna quite bad on one day over three. As it was a detection project, it has been decided to keep the data set. However, this data set should have been discarded for an imaging project.

3. Had you to flag out some data? Was it easy?

In a try to obtain a correct 1mm continuum image, I tried to flag outlier visibilities. The graphical way is easy to use. But I did not understand how to find the flagging information and to unflag after having leaved the `mplot` plot.

4. Did you have any difficulty doing the calibration steps listed below? If so, why? All the calibration steps run smoothly. However some results were strange. Detailed remarks follow.

(a) Phase atmospheric correction.

It would be good to have a visual indication of the time where the phase atmospheric correction is kept or not as is done in the efficiency plots of the PdBI reports.

(b) Bandpass.

The scale for the amplitude versus IF plots should start from 0. This allows us a much better view of the quality of the fits.

(c) Phase calibration.

When data is missing at the beginning or end of the observations for one antenna of the array because of shadowing, the solution for all the others baselines becomes wrong.

(d) Flux calibration.

It would be easier if efficiency plots were separated per antenna. For instance, on one of the observing dates, one antenna obviously shows some problem. It is impossible to tell which one from this plot.

(e) Amplitude calibration.

- Same remark as for phase calibration.

- Inside CLIC, after each fit, there is the possibility to interactively modify the fits depending of the visual results. For instance you can set a break point to separate the spline fits on each side of this break points. This interactivity (which is an important way to act on the calibration) is currently unavailable inside `iramcalibrater`.
5. How did you check the quality of the calibrations? Was it easy?
Quality of calibrations may be first checked through the overplots of the data points fitted with the fits. However, except for the flux calibration, those plots are output only as Post-Script files. You thus have to independantly use `ghostview` after each calibration. It would be easier to directly output the plots on the screen. Moreover the current plot labels are small and are difficult to see on screens. Finally, there are no caption indicating the day of observation and the band used: it is thus difficult to find which results is attached to which day/band after printing of all the Post-Script files.
 6. Is there a final summary of the whole calibration process? Is it useful?
Quality of calibrations may also be checked through the table of RMS fit per baseline. I would have prefer to see the phase RMS in degree than in radian. More importantly, the 1mm phase RMS should be computed before and after raw phase transfer from 3mm to 1mm. Here I guess the RMS are given before phase transfer.
 7. List of bugs found during the calibration. Which ones have been corrected? Which ones are still pending?
Problem of wrong solution due to shadowing described in the “Phase Calibration” section above. Bug declared too late to be solved.
 8. List of improvements asked. Have they been implemented?
R.Lucas and I asked for a summary of the calibration. It has been implemented in one or two weeks.

E.4 Imaging inside AIPS++

1. What degree of mm interferometry expertise do you need to obtain your first unoptimized clean image?
Here is a Glish script of the operations you need to go through to obtain your first unoptimized clean image:

```
include 'mosaicwizard.g';
imgr:=imager(filename="cont1mm" , compress=F, host='', forcenewserver=T);
imgr.setdata(mode="channel", nchan=108, start=11, step=1,
             spwid=[1,2,3,4], fieldid=1);
imgr.setimage(nx=128, ny=128, cellx='0.42arcsec', celly='0.42arcsec',
             stokes="I", mode="mfs", nchan=1, start=11, step=108,
             spwid=[1,2,3,4], fieldid=1);
imgr.weight(type='uniform');
imgr.makeimage(type="psf", image="cont1mm.beam");
imgr.fitsf(psf="cont1mm.beam");
imgr.clean(algorithm="clark" , niter=100, gain=0.2,
           model="cont1mm.lmv-mod", image="cont1mm.lmv-clean",
           residual="cont1mm.lmv-res" , interactive=F);
imgr.done();
```

Through the GUI, you have to fill out all the corresponding fields. This means that you must specify *i*) the *uv* visibilities you will use, *ii*) the image size, *iii*) the pixel size, *iv*) the kind of weighting to influence the beam shape, *v*) the fact that deconvolution needs a restoration beam found by a Gaussian fit of the PSF and *vi*) the kind of deconvolution algorithm with its associated parameters. For the determination of the image and pixel size, there is an `advise` function. However, this function gives you an upper limit of the pixel size and a lower limit of the image size which will not give you

a good answer as it gives you the pixel size to get one cell across the beam. Nyquist sampling asks for two pixels per beam. And, in fact, deconvolution asks for 3 to 5 pixels per beam. The advice is thus at best misleading. In the GUI, the 6 used functions are lost in the middle of dozens of others. Finally, you need to call the `viewer` tool on the clean image. I would thus say that you must have a pretty good knowledge of interferometry to make your first image.

As a comparison, inside GILDAS, once the data has been calibrated into a *uv* table, only one command (`go image`) performs all those steps. This is not an optimized image and it currently does not work for mosaics but it is enough for a first look of most of the PdBI projects.

2. Did you try several deconvolution methods? Why? If yes, is it easy to swap from one method to another one?

See the “Others Remarks” section.

3. Did you succeed to finely tune the deconvolution? Why? What is easy?

See the “Others Remarks” section.

E.5 First analysis and production of a publication quality plot inside AIPS++

1. Is the viewer tool easy to use? Why? Did you succeed to make a publication quality plot?

When you call the viewer on a data cube, the default is to plot one plane of cube at a time, in black & white, without axis, without contour, without color scale, without header describing the content. To produce a paper quality plot, you have to make a good dozen of mouse clics. Why is the default so rustic?

2. Did you succeed to compute the noise on the continuum images or line cube?

See the “Others Remarks” section.

3. Did you succeed to compute the moments of your line data?

See the “Others Remarks” section.

4. For simple sources (i.e. detection), is it easy to make fits in the *uv* plane?

High-redshift sources are often unresolved in the PdBI compact configuration. In this case, the best way to analyze your data is to fit a point source in the *uv* plane. This allows you to subtract the continuum and then to obtain a spectra with robust error bars (avoiding the artefacts of imaging and deconvolution). However, this method is not currently available inside AIPS++. Work has been done by D. Broguiere of IRAM to adapt the GILDAS tool to AIPS++. But this is not yet robust enough to be used in this phase II test.

E.6 General

1. What is the degree of flexibility of AIPS++?

To me, flexibility is the ability of doing something for which the software has not be initially designed for. The data set I worked on is a typical example. In the time I worked on the phase II test, I did not succeed to obtain the 3mm line data cube. I estimate that at least one more week of full work would be needed for me without help to obtain a satisfactory result. See the “Others Remarks” section for more details.

2. What is the degree of interactivity of AIPS++?

- To me, interactivity is asking the good questions at good stopping points and just once. AIPS++ keeps asking the same questions (Field IDs, Spectral window IDs) and does not often try to guess what could be the answers. Granularity inside `iramcalibrator` is not fine enough to reduce PdBI data.
- The numbers of windows you have to simultaneously interact with is also limiting your efficiency. When you start AIPS++, you end up with 3 windows (`manager`, `logger`, `xterm`). You then have at least one window per tools, the needed ones being the `catalog`, the `mplot` (for flagging),

the `iramcalibrater`, the `imager` and the `viewer`. It becomes uneasy when you need to iterate between calibration and imaging...

3. Does it seem easy to test several hypothesis during the calibration and imaging?
 - Due to the lack of flexibility and interactivity of `iramcalibrater`, it is not obvious to test several calibration hypothesis.
 - For the imaging part, I always had to remove the directories created during the cleaning (in particular the model directory when changing the number of imaged channels) to avoid strange behaviors.

4. Does AIPS++ seem to be stable enough?

At first view, AIPS++ does not seem more or less stable than the other packages I am used to. However, it does not seem to gracefully handle user interruption of most of its tasks: You may end up in an incoherent and some processes may continue to run even after exiting AIPS++. This is unfortunate when those tasks take long to finish. Following are two examples of messages I had at the glish prompt in such cases:

```
- LocalExec::SetStatus: abnormal child termination for
/project/ALMA/alma/aips++/linux/bin/imager
Server 'imager' has failed unexpectedly!
You will need to create the relevant tool again.
If that causes unexpected behavior, please restart AIPS++
Please submit a bug-report using bug() if you can reproduce the problem.

- [fatal error, 'segmentation violation' (signal 11), exiting]
Segmentation fault
```

However, I recognize that my lack of knowledge of AIPS++ may make those occurrences more frequent.

5. Are the tools quick enough for efficient interactive use? If not, which ones should be improved first?
 - For comparison, I tried to redo the calibration from scratch inside CLIC and AIPS++. In both case, *i*) I knew exactly what I needed to do and *ii*) I did not try to make any flagging. Starting from the raw data format of PdBI, it took me 16 minutes to interactively calibrate the whole project inside CLIC. Starting from the raw data format of AIPS++ (i.e. the measurement set), it took me 57 minutes to calibrate the same project inside AIPS++. Such a difference is very important when you interactively want to test several hypothesis.
 - My first question is why does AIPS++ take between 30 and 60 seconds to start?

E.7 Support

1. Data sets:

- (a) Had you the data sets in time?
- (b) Had you enough information about the data sets?
- (c) Had you enough information about PdBI calibration methods?

I had already calibrated this data set inside CLIC!

2. Installation of AIPS++:

- (a) Was it easy?

I did the installation of the binaries from the CDROM followed by an internet update of the last stable version. The whole process took several hours. It has been easy. However, the `viewer` was not working with this version due to an incompatible version of the `LessTif` library.
- (b) Were the instructions good? Yes.

3. Bug corrections and improvements into the `iramcalibrator` tool: was it quick?
I submitted a bug report to late to be handle!
4. Documentation:
 - (a) Was the tutorial good?
The tutorial was excellent.
 - (b) Is the `iramcalibrator` cookbook complete?
The cookbook have been an essential work instrument. Many curiosities of AIPS++ are very well explained here. Due to recent changes inside `iramcalibrator`, a few changes are needed in the cookbook. But this is the fate of all cookbooks to be lagging developments. I would have just wish a section about line imaging.
 - (c) Is the general AIPS++ documentation easily understandable?
Documentation is understandable but it is not always easy to find what you are searching for.
5. What are the parts for which you would have like more support?
I had all the support I wanted: the answers were quick and often clear.

E.8 Test charge

1. How can AIPS++ be adapted to reduce data of an instrument for which it was not initially designed?
One FTE year has been required from the NRAO side (i.e. not counting the participation of D. Broguiere of IRAM) to complete phase I of the AIPS++/IRAM test. The data set I worked on shows that `iramcalibrator` probably still has some small problems (i.e. the stripe problem of the 1mm continuum image). I have no doubt that this problem will be quickly solved. However, what surprised me most is that the very general tools of AIPS++ *i)* does not allow me to make a simple *uv* plane fit and *ii)* are not yet able to do something as simple as the imaging of line which has been observed with different frequency tunings.
2. How long is the learning curve for developers who have sufficient experience in the processing of millimeter data, but no experience at all in the AIPS++ programming environment?
3. Can we perform an end-to-end experiment on actual, real-life mm-wave spectroscopic data?
It is certainly possible to perform an end-to-end experiment on actual, real-life mm-wave spectroscopic data. Indeed, just after having followed the `iramcalibrator` tutorial, I succeeded to calibrate and image a new data set in a couple of hours. However, there are important cases for the science currently done with mm interferometers that can not yet be easily treated inside AIPS++.

E.9 Other remarks

- I have spent 6 full days just trying to calibrate and image this project (i.e. not counting the participation to tutorials and the retrieving of data). One of these days has been spent on a stupid mistake. Half a day has been used to fill out this questionnaire. For comparison, such a project is considered simple at IRAM and PI on average need 2 days to calibrate, image and analyze this kind of data.
- Trial made to obtain the 3mm data cube: The project had needed 3 different frequency tunings, each of them using 4 different 160MHz-wide spectral windows. The spectral windows overlap. I had been advised to separately image the 12 spectral windows and then to use the `concatenate` function. However, this did not work because *i)* the velocity axis must be compatible and *ii)* the spectral windows must not overlap. All the 3 frequency settings overlap a lot and I did not want to widely cut to avoid SNR loss, I did not go further.
- For the 1mm continuum image, here is what I tried:

- Standard imaging on two days over three as multi-frequency synthesis seems to work for continuum only when there is an overlap in frequency : stripes appear.
- Flagging of the outliers visibilities directly on data: somehow, this did not help.
- Try several times to flag calibrator data and to redo calibration: stripes were still there.
- At some points, the measurement sets seemed to be in an inconsistent state. So I wrote simple calibration scripts starting from the filling of the ALMATI format and without flagging as I do not know how to flag data inside a script.
- I verified that the results of the calibration inside CLIC without any hand-flagging: image quality was then correct.
- Finally I tried to image the day left. The stripes are less prominent.

F Questionnaire of Debra Shepherd

F.1 Tester

1. What kind of experience have you with:
 - (a) **data reduction:**
I've reduced single dish radio data (using Unipops) as well as interferometric data using AIPS, AIPS++, Miriad, & MMA.
 - (b) **radio interferometry:**
I've been reducing VLA data (7 mm to 6 cm wavelength) in AIPS and AIPS++ for more than 6 years.
 - (c) **mm radio interferometry:**
I have worked extensively with OVRO data, mostly CO or SiO mosaics & continuum. I've also worked with BIMA data (combining 12m+BIMA single field).
 - (d) **data from Plateau de Bure Interferometer PdBI:**
This is the second data set I've reduced from PdBI (all within AIPS++).
2. What are the reduction packages you are used to?
AIPS, AIPS++, Miriad, & MMA
3. What was your level of experience with AIPS++ before the test?
I've been testing AIPS++ for about 2.5 years. I've been reducing line and continuum VLA data in AIPS++ for my science for the past year.

F.2 Data set

1. Project ID:i028
2. Project kind:
CO(J=1-0) line and 3 mm continuum mosaic of a protostellar outflow, HH 288. 1 mm data was also in this dataset but it was not usable due to weather (also, the mosaic spacing was optimized for the primary beam at 3 mm).
3. Number of days observed: 9 days
4. Number of days usable:
8 days (most days did not have usable data either at the beginning or the end of the track when the source was rising or setting because this data was taken under relatively poor, summer conditions.
5. Did you succeed to obtain a believable results inside AIPS++?
Yes. See attached figures

F.3 Calibration inside AIPS++ by the iramcalibrater tool

1. Is it easy to know the list of sources, calibrators and the correlator setup?
Yes, in a summary listing.
2. Is it easy to have an idea of the quality of the observation run (weather conditions, pointing information, Tsys, and so on)?
Seeing when the on-line phase corrections were not good and looking at where the phase and amplitude solutions were poor provided sufficient information to determine the data quality. Tsys and pointing information would have been helpful but it did not appear to be necessary.
3. Had you to flag out some data? Was it easy?
Yes, I flagged extensively (end channels, time ranges where there was poor phase stability, and some correlator glitches in individual channels. It was easy in the AIPS++ task msplot but it was relatively slow.

4. Did you have any difficulty doing the calibration steps listed below? If so, why?
- (a) Phase atmospheric correction. No.
 - (b) Bandpass. No.
 - (c) Phase calibration. No.
 - (d) Flux calibration. No.
 - (e) Amplitude calibration. No.
5. How did you check the quality of the calibrations? Was it easy?
- For the atmospheric phase correction, I wrote a glish script that plotted when the on-line phase correction was not good so I could see what data was being switched. This was easy once I figured out how to write the script. For the bandpass, phase, and amplitude calibration steps, I looked at the postscript plots that were produced by AIPS++ to determine the calibration quality. For the flux calibration, I examined the antenna efficiency and the reported fluxes of the gain calibrators. Based on the normalized efficiency plot, I restricted the time range of the flux determination until I got reasonable values. When the antenna efficiencies were about 20-25% and the reported fluxes were close to those given to me by IRAM, then the flux calibration was adequate.
6. Is there a final summary of the whole calibration process? Is it useful?
- I printed out the plots produced by AIPS++ after each calibration step. I had a script listing of each step during the calibration so I could see exactly what was done. This was an adequate record of the calibration process.
7. List of bugs found during the calibration. Which ones have been corrected? Which ones are still pending?
- (a) Offset directions (for mosaicing) were not properly filled. **CLIC FITS writer & AIPS++ ALMA filler fixed by Dominique.**
 - (b) Antenna positions are not rotated out of the local coord sys. **Not fixed, only a minor problem in a plot, did not affect data.**
 - (c) iramcal functions can't abort due to lack of id(). This was a problem mostly for the flux calibration function. **A 'QUIT' option was added to avoid the problem.**
 - (d) iramcal.flux did not use flag weights properly when calculating antenna efficiency. **fixed.**
 - (e) iramcal.rf typo in name field caused strange behavior. **fixed.**
 - (f) During interactive CLEANing of a mosaic, the first image produced to define the mask region was not correct. **I found a work-around for now, bug report not submitted yet.**
 - (g) During imaging, setimage inputs did not stay fixed when the cursor moved over the entry input. **A minor annoyance, bug report not submitted yet.**
 - (h) Image concatenate did not work on more than 2 channels. **Found last day of testing. Kumar wrote a script for me to work around this problem for now.**
 - (i) MEM crashed when deconvolution was started on this mosaic. **The problem is fixed now.**
 - (j) The documentation describing how to control the multi-field mosaicing options is not adequate. **Identified on last day of testing. Bug report not submitted yet.**
8. List of improvements asked. Have they been implemented?
- (a) Requested that iramcal.phcor should plot alma-flag-corr-row corrections. **not implemented yet, I wrote a small script to do this.**
 - (b) Requested that iramcal should not produce plots with flagged data. **done.**
 - (c) Requested that iramcal functions should be able to choose the reference antenna. **done.**
 - (d) Requested that iramcal.flux should be able to limit time range. **done.**
 - (e) The threshold for which each channel is CLEANed down to is governed by the channel with the brightest emission. This is fine for most VLA data but not for PdBI data where there may not be adequate *uv* coverage. In the HH 288 mosaic the brightest emission was in a channel that could not be adequately cleaned due to significant missing flux. I requested that the logic

for defining the CLEANing threshold in each channel be modified to account for this problem. **Identified on last day of testing, found a work-around by imaging each channel separately. Not fixed yet.**

F.4 Imaging inside AIPS++

1. What degree of mm interferometry expertise do you need to obtain your first unoptimized clean image?

For the single field continuum emission (shown in Figure 1), one would need very little background to produce a good image - just plug in the recommended numbers by following a cookbook and the image should be reasonable. The CO mosaic image (shown in Figure 2) required extensive background in mosaicing extended emission without adequate *uv* coverage. Even with this, the resulting image is not of publication quality. This is a limitation of the data rather than the software. The bugs uncovered during deconvolution and imaging made it difficult to get obtain the final image. I had to develop several work-arounds and I do not think a novice user could have figured them out without extensive help. Most of the problems will be straight forward to fix. Once these problems are corrected, users should be able to produce mosaic images relatively easily.

2. Did you try several deconvolution methods? Why? If yes, is it easy to swap from one method to another one?

Yes, MEM did not work initially (see bug report listed above) so I switched to mfclean. I made a brief comparison using mfmultiscale clean on a channel to see if I could pick up any more extended emission. It did not make a significant improvement in the image because of the missing flux. This type of testing was very easy.

3. Did you succeed to finely tune the deconvolution? Why? What is easy?

I tuned it as far as the data allowed. It was not easy because the documentation for controlling the mosaicing options was not adequate (see bug report above). Once I understood the input options, it was easy.

F.5 First analysis and production of a publication quality plot inside AIPS++

1. Is the viewer tool easy to use? Why? Did you succeed to make a publication quality plot?

For the continuum emission, the viewer tool was adequate to produce a publication-quality image although the way overlay files are handled are cumbersome. For the line emission, I could display a good movie of the emission but I could not get a single nice display of the channel map with different label types and overlay files for each channel - I ended up making images for each channel and then used latex to display them all on the same page - this was more complicated than it should have been. Of course, I can't do this in Miriad, or AIPS either and I have always made all my publication quality figures using the WIP plotting package.

2. Did you succeed to compute the noise on the continuum images or line cube?

Yes, this was very easy using the statistics tool in the viewer.

3. Did you succeed to compute the moments of your line data?

No, the line image was very poor because of lack of short spacing data, moment maps would be useless.

4. For simple sources (i.e. detection), is it easy to make fits in the uv plane?

To my knowledge, this capability is not available in AIPS++.

F.6 Comparison of results obtained with AIPS++ and CLIC (Optional)

1. What is the final RMS in the calibrated target image?

Continuum: 1.34 mJy/beam. Spectral line: 19 mJy/beam.

2. What is the peak/noise ratio in the image?
Continuum: 14. Spectral line: 40
3. Discuss any differences you found between calibration solutions along the way.
CLIC data reduction not done yet.
4. Discuss any differences between the CLIC and AIPS++ images.
CLIC data reduction not done yet.

F.7 General

1. What is the degree of flexibility of AIPS++?
It seemed flexible enough to mimic IRAM calibration procedures. When I found a problem during the imaging, I was able to find an immediate work-around that was similar to how I do imaging in Miriad.
2. What is the degree of interactivity of AIPS++?
For interactive cleaning, I would say that AIPS++ is highly interactive, more so than any other imaging program I've used. Inside of the iramcalibrator tool, the only programmed interactivity was during the flux calibration - this was adequate to allow me to find reasonable solutions.
3. Does it seem easy to test several hypothesis during the calibration and imaging?
Yes.
4. Does AIPS++ seem to be stable enough?
During this Phase II testing in which new data sets are being processed to discover limitation and bugs that had not been anticipated, I did not expect AIPS++ to be stable - in fact, it was unstable because the programmers responded quickly to bug reports and enhancement requests from the testers.
5. Are the tools quick enough for efficient interactive use? If not, which ones should be improved first?
No, msplot was very slow. I think this is a problem which should be addressed before many users will want to use the AIPS++ software.

F.8 Support

1. Data sets:
 - (a) Had you the data sets in time?
Just barely, I only had 3 days to reduce and image the data because there was a problem with the filler. Since it was more important to fix problems associated with non-mosaic data, the problem of the mosaic filler was not fixed until quite late.
 - (b) Had you enough information about the data sets?
Yes.
 - (c) Had you enough information about PdBI calibration methods?
Yes.
2. Installation of AIPS++:
 - (a) Was it easy?
Not applicable.
 - (b) Were the instructions good?
Not applicable.
3. Bug corrections and improvements into the iramcalibrator tool: was it quick?
Yes.
4. Documentation:

- (a) Was the tutorial good?
Not applicable (since I wrote it).
- (b) Is the iramcalibrator cookbook complete?
Not applicable.
- (c) Is the general AIPS++ documentation easily understandable?
Not always (see bug report listed above).
5. What are the parts for which you would have like more support?
I can't think of any at this time.

F.9 Other remarks

Time spent:

- Data reduction and editing: 8 hours;
- Continuum imaging: 2 hours;
- Spectral line imaging: 8 hours;
- Filling out questionnaire and making nice images: 6 hours;
- Total time: 24 hours.

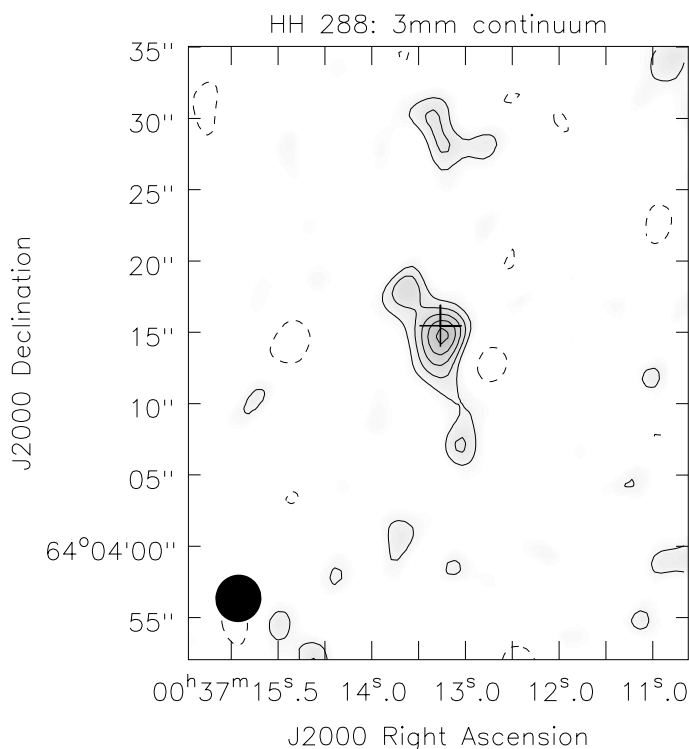


Figure 1: HH 288: 3 mm continuum emission, reduced and imaged in AIPS++. Natural weighting. RMS = 1.34 mJy/beam (expect 1.33 mJy/beam). Peak = 8.422 mJy/beam at $00^{\text{h}}37^{\text{m}}13.251^{\text{s}}$ $64^{\circ}04'14.80''$ (cross marks the position of the millimeter peak reported by Gueth et al. (2001); their positional error = $0.3''$). Total flux = 19.03 mJy (expect 19 ± 1.5 mJy). Contours are plotted at $-2, 2, 3, 4, 6$ & 6σ . The source flux and position compare well with what is reported in Gueth et al. (2001). The source extent and orientation is similar, the image produced by AIPS++ is higher resolution (probably due to weighting). The synthesized beam of $3.3''$ by $3.2''$ at P.A. 165° is shown in the bottom left corner.

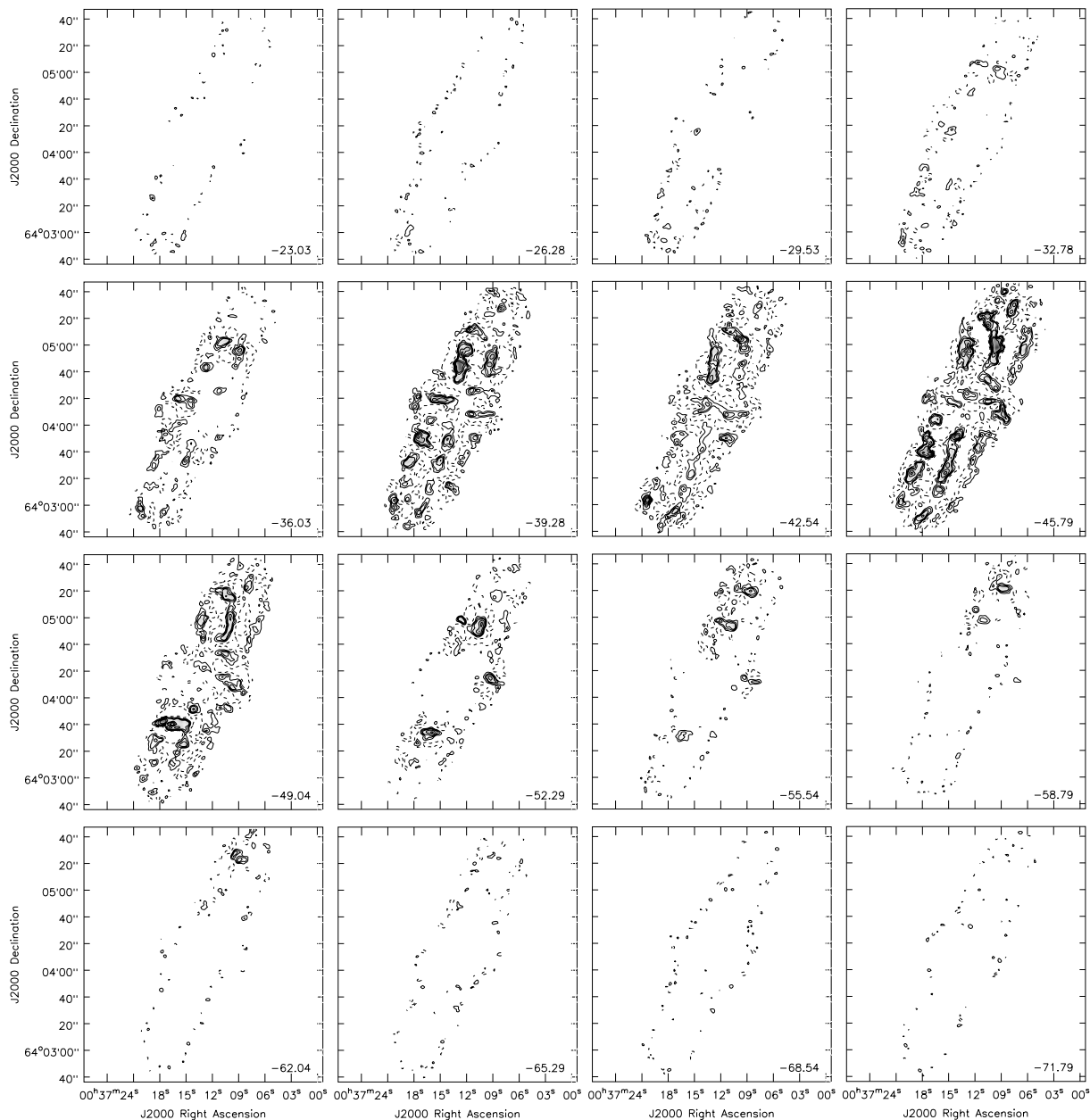


Figure 2: HH 288: CO($J=1-0$) line emission, reduced and imaged in AIPS++. Robust weighting, deconvolved with CLEAN. RMS = 19 mJy/beam. Peak = 770 mJy/beam. Grey scale is displayed from 19 mJy/beam to 770 mJy/beam. Contour levels are plotted at $-3, 3, 6, 9, 15,$ & 21σ . Velocity of each channel (in km/s) is shown in the lower right. The resolution is 3.25 km/s. The synthesized beam (not shown) is $3.1''$ by $2.9''$ at P.A. 93.4° . The Gueth et al. (2001) image of this region is a combined mosaic of PdBI and IRAM 30 data. The PdBI data alone does not have adequate short spacing information to properly image the outflow (as the figure above indicates). None the less, the basic location of the emission peaks are consistent with the brightest peaks in the Gueth et al. image.

G Questionnaire of Leonardo Testi

G.1 Tester

1. What kind of experience have you with:
 - (a) data reduction: good
 - (b) radio interferometry: good
 - (c) mm radio interferometry: good
 - (d) data from Plateau de Bure Interferometer PdBI: fair
2. What are the reduction packages you are used to?
mainly: IRAF, AIPS, GILDAS, mma
3. What was your level of experience with AIPS++ before the test?
None

G.2 Data set

1. Project ID: L02D
2. Project kind:
Mapping: SingleField; 1mm and 3mm; Line and continuum
3. Number of days observed: Four
4. Number of days usable: Three
5. Did you succeed to obtain a believable results inside AIPS++?
No. This is mainly due to lack of experience with AIPS++ and little time available. I am sure that I would have been able to obtain good results if I had more time. Nevertheless, I found a few major obstacles that it would have been difficult to overcome without some more development. These will be detailed below.

G.3 Calibration inside AIPS++ by the iramcalibrater tool

1. Is it easy to know the list of sources, calibrators and the correlator setup?
Yes.
2. Is it easy to have an idea of the quality of the observation run (weather conditions, pointing information, Tsys, and so on)?
I mainly used the phase vs time and amp vs time plots, not the best way to do it. Too slow to plot and re-plot data.
3. Had you to flag out some data? Was it easy?
Yes and No. Easy to flag, I like the interactive approach of msplot and the flagger. However msplot was not useable for my dataset (too slow). If I had to plot, flag and replot each of my 128 channels and 512 channels bands, I would still be here counting the dots as they appear on the screen... I had to decide some broad flagging approach and blindly use the flagger.
4. Did you have any difficulty doing the calibration steps listed below? If so, why?
No. Even though the lack of any help from the package is a bit discouraging. The approach of presenting everything you have in alphabetical order is a bit too dumb to be helpful (specific examples below).
 - (a) Phase atmospheric correction.
ok
 - (b) Bandpass.
ok, I requested a modification to fit a higher (well, arbitrary) polinomia, which was implemented quickly

- (c) Phase calibration.
ok, after fixing a few bugs
 - (d) Flux calibration.
ok
 - (e) Amplitude calibration.
ok
5. How did you check the quality of the calibrations? Was it easy?
Looking at the plots. It was easy
 6. Is there a final summary of the whole calibration process? Is it useful?
Yes, the plots, very useful
 7. List of bugs found during the calibration. Which ones have been corrected? Which ones are still pending?
 - Arbitrary polynomial for RF. Done
 - Parameters for changing phase transfer behaviour. Done
 - Parameters to identify phase wraps. Done
 8. List of improvements asked. Have they been implemented?

G.4 Imaging inside AIPS++

1. What degree of mm interferometry expertise do you need to obtain your first unoptimized clean image?
A lot! Unless you blindly plug in the numbers on the cookbook without worrying what you are doing (then the software could do it automatically)
2. Did you try several deconvolution methods? Why? If yes, is it easy to swap from one method to another one?
No, I did not managed to get one image out of imager with clean, I certainly did not tried anything else
3. Did you succeed to finely tune the deconvolution? Why? What is easy?
I did not succeeded to deconvolve at all

G.5 First analysis and production of a publication quality plot inside AIPS++

1. Is the viewer tool easy to use? Why? Did you succeed to make a publication quality plot?
Never reached that far
2. Did you succeed to compute the noise on the continuum images or line cube?
Never reached that far
3. Did you succeed to compute the moments of your line data?
Never reached that far
4. For simple sources (i.e. detection), is it easy to make fits in the uv plane?
Never reached that far

G.6 Comparison of results obtained with AIPS++ and CLIC (Optional)

1. What is the final RMS in the calibrated target image?
Never reached that far
2. What is the peak/noise ratio in the image?
Never reached that far

3. Discuss any differences you found between calibration solutions along the way.
Never reached that far
4. Discuss any differences between the CLIC and AIPS++ images.
Never reached that far

G.7 General

1. What is the degree of flexibility of AIPS++?
Too much and too little at the same time. You start the thing to reduce your interferometry dataset and everything is offered to you from sigle dish to modeling (in alphabetical order!). You go to your package because you think that this is where you think your tasks should be, you fill in the data and then you are stuck: you have to hunt into another package to find how to plot, flag etc. Then you go back calibrate, and you have to hunt for your imager... etc... So, in principle, it offers everything, however, at the same time, it is very rigid: to deconvolve a damn thing with clean, you have to go through a nightmare, setting parameters and running at least four different functions to be chosen out of a pool of a few dozens (of course listed in alphabetical order). If you set something wrong it runs for a while and then exits with a severe error, in a few cases it can hang and one has to restart over, rerunningeach and every single function every time (of course the first few times I had assumed that it was not necessary to go back and click again the go button for the same paramenters in the setimage function. Of course I got it wrong so many times that I never managed to deconvolve an image (I feel like a damn idiot). I never understood advice: why is advice supposed to exist at all? Why one does not have the parameters filled in as defaults in the setimage? (and of course, why one should change the parameters offered by advice, what kind of useful advice is if one should not follow it?
2. What is the degree of interactivity of AIPS++?
Too much and too little at the same time. It is nice to be able to change every parameter, but not having to do it every time! If I want a dirty image of a dataset, I should be able just to click a button or type "make a dirty image" whithout having to go through everything! We are designing the ALMA system in such a way that everybody can use it, e.g a biologist that wants to seach for a stupid molecule in a given source, and then we ask him/her to set the appropriate pixelsize for Natural weighting of the visibilities and the correct number of pixels to map the primary beam? The default running mode should be that one does not need of even to know that those parameters exist.
3. Does it seem easy to test several hypothesis during the calibration and imaging?
4. Does AIPS++ seem to be stable enough?
I never had major problems, except with my wrongly set imager parameters.
5. Are the tools quick enough for efficient interactive use? If not, which ones should be improved first?
No. I do not think it is important to optimize the iramcalibrater. Of the other tools, certainly msplot and imager are slower than corresponding GILDAS routines

G.8 Support

1. Data sets:
 - (a) Had you the data sets in time?
Yes
 - (b) Had you enough information about the data sets?
Yes
 - (c) Had you enough information about PdBI calibration methods?
Yes
2. Installation of AIPS++:

- (a) Was it easy?
Yes
- (b) Were the instructions good?
Yes
- 3. Bug corrections and improvements into the iramcalibrator tool: was it quick?
Yes
- 4. Documentation:
 - (a) Was the tutorial good?
Yes
 - (b) Is the iramcalibrator cookbook complete?
Yes
 - (c) Is the general AIPS++ documentation easily understandable?
Did not read much of it (because of lack of time), that is where all my problems with imager come from.
- 5. What are the parts for which you would have like more support?

G.9 Other remarks

- Time spent: 4 full time days (from installation to questionnaire).
- I could not find a way to mask out channels from the “continuum” bands and output the data from different spectral bands in a single channel equivalent (u,v) dataset for continuum map. My specific problem is that I detect several lines also in the continuum bands, additionally, I have continuum bands with different channel widths, and I would have liked to merge everything out for the continuum image (as done in GILDAS). This, of course, speeds up the imaging part...

H Questionnaire of Christine Wilson

H.1 Tester

1. What kind of experience have you with:

- (a) data reduction: 15 years
- (b) radio interferometry: 15 years, but mostly mm interferometry
- (c) mm radio interferometry: 15 years
- (d) data from Plateau de Bure Interferometer PdBI: none

2. What are the reduction packages you are used to?

Most recently I have used mma to calibrate OVRO data and miriad to image OVRO data. I recently learned how to use AIPS to calibrate VLA data (and relearned how to use AIPS to image). Ten years ago I used miriad to calibrate and image bima data and AIPS to image OVRO data.

3. What was your level of experience with AIPS++ before the test?

I had no experience of any kind with AIPS++ before the test.

H.2 Data set

1. Project ID: h121

2. Project kind:

Mapping: SingleField

1mm and 3mm

Line and continuum

3. Number of days observed: 5

4. Number of days usable: 2

(other 3 were very short and I ran out of time, so didn't reduce them)

5. Did you succeed to obtain a believable results inside AIPS++?

H.3 Calibration inside AIPS++ by the iramcalibrator tool

1. Is it easy to know the list of sources, calibrators and the correlator setup?

yes, I used catalog

2. Is it easy to have an idea of the quality of the observation run (weather conditions, pointing information, Tsys, and so on)?

I didn't know how to check this with AIPS++ (the information didn't seem to be in the AIPS++ IRAM cookbook), so I didn't.

3. Had you to flag out some data? Was it easy?

Yes, I had to flag some data.

It was quite easy to flag the end and gibbs channels using the flagger tool. However, I got myself into difficulty there in the setchan step when I wanted to flag one channel and the gui didn't put it in [], so then it flagged all my channels. But I was able to figure out how to unflag them.

I also found it easy to flag data in msplot. In particular, a first quick look at the data (amp vs. uvdistance, amp vs. channel, phase vs. time) was easy and sometimes bad data were obvious and therefore easy to flag.

What I did not always find easy was to identify which were the bad data by comparing a plot in msplot with one of the plots from rf (the visnorm plot) or phase. It would have been easier if phase

could have plotted data with the x axis in UT time instead of seconds, so that the scales in the postscript plots and the msplot window were the same. This was the most difficult part of the flagging process and one that I dreaded having to repeat if I'd made a mistake. However, repeating the flagging wasn't too bad as long as I kept very careful notes in a text file as to what exactly I had flagged

I also found the final step of flagging the actual source data took quite some time, mainly because msplot was very slow when I was plotting lots of source data with lots of channels. Even for my cleanest day (where I just had one little time section of source data to flag), the flagging took **, with most of it being spent doing plots of amplitude versus time and amplitude versus channel with msplot to look for spuriously high points.

4. Did you have any difficulty doing the calibration steps listed below? If so, why?
 - (a) **Phase atmospheric correction.**
No problems.
 - (b) **Bandpass.**
See the comments on data flagging listed above. In addition, I sometimes found it hard to get what looked like a really good fit to the phase versus time (visnorm) if the data had a large time gap in it. But I generally did the best I could and then ignored fits that I would have considered non-optimal, since I don't think the phase fit mattered a lot at this stage.
 - (c) **Phase calibration.**
See the comments on data flagging listed above. Otherwise this step was fairly straightforward and the fits were good, even with time gaps in the data.
 - (d) **Flux calibration.**
Ah, now, there's a story ... I somehow managed to get a data set completely hung up when I ran flux on it, so hung up that no OTHER function would run on it. I had to run initial to fix it.
I'm still not entirely sure whether the efficiencies that are being reported in the logger are right. For my data they looked kind of low, i.e. too close to perfect, but maybe I just had a really good data set.
 - (e) **Amplitude calibration.**
No problems
5. How did you check the quality of the calibrations? Was it easy?
I only checked the calibrations by looking at how well the curves fit the data. If there were other ways I was supposed to check it, I wasn't aware of them.
6. Is there a final summary of the whole calibration process? Is it useful?
I did not have a final summary of the calibration process, but that could be that I didn't know what I was supposed to look for. I think I started my calibration with an older version of rf, phase, etc.
7. List of bugs found during the calibration. Which ones have been corrected? Which ones are still pending?
I'm sorry, I've lost track. I don't think any of the bugs I found in iramcalibrator are still pending. (My bugs seem to be things that totally hold me up, and so have to be fixed ...)
8. List of improvements asked. Have they been implemented?
I'd like to be sure that the efficiencies and their uncertainties are correct as reported.

H.4 Imaging inside AIPS++

1. What degree of mm interferometry expertise do you need to obtain your first unoptimized clean image?
For the continuum images, it is easy, just follow the cookbook. But some knowledge is very useful when surprises turn up, like when I detected my source in 3 mm continuum but not 1 mm continuum.

I suspect that either (1) CO 1-0 is contributing strongly to the 3 mm image or (2) something went wrong in my calibration (or imaging) of the 1 mm image. Also, I managed to get confused about which spectral windows to select for data that had more than one continuum spectral window. For 3 mm I just used one of the two (I think); for 1 mm I discovered I could use both (I think), but I'm still confused because I didn't detect the source...

For the spectral line images, you need some basic understanding of how to select channels and how channels relate to velocity. It would help if there were examples in the cookbook. For example, I NEVER managed to select my data by velocity ranges, I always had to do it by channels.

2. Did you try several deconvolution methods? Why? If yes, is it easy to swap from one method to another one?

Yes, I tried both uniform and natural on my 1 mm continuum image when I didn't detect it in the uniform image. (I still didn't detect it with natural weighting.) It was easy to swap, I just had to run the weight function again. One thing I'm not sure about is whether it is necessary to make a new psf image if you change the weight function or not (the cookbook wasn't clear on that).

3. Did you succeed to finely tune the deconvolution? Why? What is easy?

I succeeded in fine tuning the convolution for the 3 mm continuum image. I followed the cookbook to put a clean box around the source and cleaned down to a 1 sigma threshold (having measured sigma in regions away from the source).

I did NOT succeed in fine tuning the convolution for the 3 mm spectral line image. I was able to clean down to a noise threshold but I could not successfully clean with a multi-channel clean box. I have no idea why. I had some trouble at first making the clean box until I looked at the AIPS++ help and realized how to easily use the same box for each channel. I understand how to make different boxes for each channel, I think, but I was running out of time to complete the tests. And in any case, even if I used my simple clean box, clean would not complete successfully.

Perhaps because I could not figure out how to use a clean box, my cleaned line data were not very nice (there were serious negative residuals in some of the channels, for example).

I just had time at the end to do a first shallow clean of the 1mm line data and I DID see the CO 2-1 line from the source. So that proves that my 1 mm calibration can't have been completely wrong ...

H.5 First analysis and production of a publication quality plot inside AIPS++

1. Is the viewer tool easy to use? Why? Did you succeed to make a publication quality plot?

It was reasonably easy to use. I made simple publication quality plot for the 3 mm continuum image (contours on grey-scale, beam, axis labels, but no source label).

I did not try making a good plot for the 3 mm channel maps since my cleaned images were not very nice (see previous section).

2. Did you succeed to compute the noise on the continuum images or line cube?

Yes to both questions, using the viewer.

3. Did you succeed to compute the moments of your line data?

I did not have time to try this.

4. For simple sources (i.e. detection), is it easy to make fits in the uv plane?

I did not have time to try this (and in any case I did not have a simple source).

H.6 Comparison of results obtained with AIPS++ and CLIC (Optional)

1. What is the final RMS in the calibrated target image?
2. What is the peak/noise ratio in the image?
3. Discuss any differences you found between calibration solutions along the way.
4. Discuss any differences between the CLIC and AIPS++ images.

I could not do even a simple comparison of my results with the published images because I did not have the full data set and so my angular resolution was a factor of two worse than what was published. Also, it is not obvious to me that the contours in the figure caption were labeled correctly (the caption suggests there should be 12 contours to the peak of the CO J=1-0 image and I can only see 6 or so).

H.7 General

1. What is the degree of flexibility of AIPS++?

I don't think I can really answer that. This test took more time than I had available, so I had no time to explore alternative data reduction paths.

2. What is the degree of interactivity of AIPS++?

I like the interactivity. The GUIs are slow, for sure, but I used them all through this test because other than being slow, they were a very useful memory aid to what I was supposed to be doing.

3. Does it seem easy to test several hypothesis during the calibration and imaging?

Yes, for imaging. I found the calibration rather painfully awkward, in particular when I was trying to get rid of bad data in the rf, phase, and flux steps. It was frustrating enough to work through that I had no wish to try alternatives if I had something that looked like it was working well enough.

4. Does AIPS++ seem to be stable enough?

I don't know how to answer this one. I certainly got frustrated during these tests because when something failed, I didn't know whether it was my error or a problem in AIPS++. In a stable package, it is usually safe to assume that you've just made a typo or something. However, I would say I ran into enough failures that were obviously not my fault (like when flux hung up my data set) or where I couldn't find anything I'd done wrong (like when uvt seemed to corrupt one of my output files) to make me suspicious of AIPS++ itself whenever something went wrong. Just to make things more complicated, sometime BOTH factors are at work. For example, when I tried to use velocity in imager to setdata and set image, I found that I had been doing something wrong at first (not specifying nchan in addition to the velocity specifications). However, even when I did figure that out using the AIPS++ help, I still couldn't successfully make a psf image, so clearly something else was still wrong.

To give a counter example, last September I relearned AIPS imaging and learned (for the first time) AIPS calibration of VLA data. Now, I'll certainly admit I was swearing at AIPS more than once during that week, but at least I could be sure that it was either my errors or some inflexibility in AIPS that I was getting frustrated with, not buggy code.

So I guess I would say I'm not sure AIPS++ is stable enough yet. I personally don't feel I understand it well enough to encourage a new graduate student to use it. (I also don't feel very confident about AIPS, but we have to use something for our VLA data calibration, at least.) On the other hand, I would say that I'm impressed that I was able to learn enough from Debra's morning tutorial to participate in this test and that I've made it as far as I have.

5. Are the tools quick enough for efficient interactive use? If not, which ones should be improved first?

Again, I can't really answer this. I have the impression that I had significant waiting times during the calibration stages. I would often start a task (any task) and then go to get a coffee down the hall, or go check my mail, or something. Imaging was faster; despite the problems I had with the multi-channel imaging and the 1 mm continuum, when I was actually running a task or function, it went fairly quickly.

H.8 Support

1. Data sets:

- (a) Had you the data sets in time?

Yes

- (b) Had you enough information about the data sets?

I had enough information about the 5 data sets I was given; however, I discovered on the last day that I had not been given all the data that were used in the published version of this project (Lindqvist et al. 1999). I wanted to do a "blind" reduction of the data to see how well my results agreed with the paper, and so only discovered that data were missing when I looked at the paper for help when I unexpectedly did not detect the source at 1mm after having detected it in 3 mm.

- (c) Had you enough information about PdBI calibration methods?

At a basic level, yes. What I don't feel I had was detailed information on how the data were usually flagged i.e. I only flagged data that appeared bad in visibility plots and shadowed data. I did not know how to look for data with bad Tsys, for example. My knowledge was a bit incomplete because I was trained in Garching at the end of January and some of the information about IRAM calibration was still being passed to the NRAO people at that time.

2. Installation of AIPS++:

- (a) Was it easy?

Installing the CD was very easy. I had some trouble with aupdate. It was not trouble with the instructions, but that things were missing and so my new installation didn't work properly. First there was a configuration file that was not set properly for Solaris and after that was fixed, it still didn't work. Then it was discovered that some libraries were missing and I had to download them and do aupdate again. Basically I started my first aupdate at about 9 am EST and finally had a full working version of AIPS++ by 1:30 pm EST the next day.

- (b) Were the instructions good?

The instructions were clear but the syntax was not always correct for a solaris user. Some of them I knew enough to fix the syntax; for one I needed my system administrator's help (but maybe I shouldn't have; I just didn't know what my cdrom was called).

3. Bug corrections and improvements into the iramcalibrator tool: was it quick?

I would say generally yes. For example, I discovered a bug in flux which hung up my data set in a semi-permanent way at the end of an afternoon; by the next morning Kumar had emailed me how to recover from it (using initcal); an updated version of flux with several things fixed was available by the following morning.

I had a frustrating bug or feature that one of my data sets seems to have gotten corrupted when I wrote it out using uvt (strangely, it was the first of a set of four that was corrupted). I couldn't see any reason why the data set would have had problems, I could find no errors in my logger or glish window, and it didn't seem to repeat.

4. Documentation:

- (a) Was the tutorial good?

The tutorial was excellent.

- (b) Is the iramcalibrator cookbook complete?

The cookbook was excellent. I would suggest adding some detailed notes on how to image spectral line data in addition to the continuum notes, as some aspects are different and I ran into some trouble there.

(c) **Is the general AIPS++ documentation easily understandable?**

I only looked at the general AIPS++ documentation at the end when I was doing imaging etc. It seemed pretty clear when I read it (at least, I could see that I had done something wrong before) but in the two cases I looked at, I didn't succeed in completing the command.

The first case was in imager, where I wanted to select data by velocity. I used the help to understand how to use the parameters, but I never succeeded in creating a psf image when I'd selected by velocity. I don't know whether that is due to (a) pilot error on my part (b) a general error in AIPS++ or (c) some error in the AIPS++-IRAM interface (for example, maybe the Vlsr wasn't being read from the header correctly). It'd be interesting to know if other people managed to select by velocity (rather than channel) successfully.

The second case was also in imager, when I was trying to make a region mask for multi-channel data. The help gave me a clue as to how to create a multi-channel mask, and I seemed to create the mask itself just fine (i.e. when I looked at the mask, there were selected regions in some channels and none in others). However, I never successfully ran clean with that channel mask, and I don't know what was wrong.

5. **What are the parts for which you would have like more support?** Spectral line imaging. There is nothing in the IRAM cookbook on this and as I said above, I ran into some errors that I couldn't solve.

H.9 Other remarks

I thought some times might be helpful ...

- 30 hours elapsed time to install a working version of AIPS++ with all the stuff from aupdate; however, actual working time for me (as opposed to aupdate just chugging away or down time at night) was perhaps 4 hours
- 4 hours + 4 hours + 2 hours + 8 hours + 1 hours = 19 hours to calibrate data from 09may98 (includes the flux bug along the way)
- 2 hours of runtime for another aupdate
- 3 hours for calibration of 05may98. Notice how much faster this was; it was a slightly cleaner data set and I discovered few or no new bugs.
- 5 hours to extract source data with uvt and flag it with flagger and msplot for both days
- 6 hours to do imaging and quick test of publication quality plot (although note that I did not figure out how to select channel data by velocity OR how to use clean boxes on multi-channel data)
- 2.5 hours to fill out this questionnaire!

So total working time for me about 40 hours spread over 2.5 weeks. It would have been nice to have more time to sort out the imaging difficulties I had. All times include time spent keeping a log of what I was doing, emailing back and forth with bugs and problems, etc.

I Defect and enhancement summary

Bug reports within the AIPS++ system were submitted by A. Baker, A. Coulais, F. Gueth, J. Pety, D. Shepherd, L. Testi, and C. Wilson. In addition, many bugs and enhancement requests were made off-line by all testers and fixed immediately by the AIPS++ team if the problem: 1) prevented the tester from processing their data; or 2) if there was a significant short-coming in the options that prevented good calibration. Several imaging problems were identified on the last day of testing and could not be addressed in time for this test. Below is a summary of all bugs and enhancements (those submitted officially and those worked off-line) that were found during the ALMA Phase II testing.

For all items listed below that have not been fixed, bug reports or enhancement requests have been submitted within the AIPS++ software.

I.1 Problems fixed during calibration & imaging

1. Offset directions (for mosaicing) were not properly filled. **CLIC FITS writer & AIPS++ ALMA filler fixed by Dominique.**
2. iramcal functions can't abort due to lack of id(). This was a problem mostly for the flux calibration function. **A 'QUIT' option was added to avoid the problem.**
3. iramcal.flux did not use flag weights properly when calculating antenna efficiency. **fixed.**
4. iramcal.rf typo in name field caused strange behavior. **fixed.**
5. MEM crashed when deconvolution was started on the HH 288 mosaic. **fixed.**
6. When running almati2ms on PdBI data which was converted to ALMATI format with the 30-jan-2003 version of the GILDAS filler, There is a "severe" error reported saying that the Leap second table is out of date. **fixed.**
7. iramcal.flux doesn't work correctly when the continuum sub band channel number differs from 64 (what was in Phase I data). **done: new parameter added (numchancont) to allow user to specify number of channels in the band.**
8. iramcal.phase doesn't always find good solutions on data for which the phase is bouncing between 180 and -180 degrees. **done: 2 parameters added (npointaver & phasewrap) to help control the solutions.**
9. iramcal.flux 'fixed' parameter doesn't work in GUI mode. **fixed.**
10. iramcal can't choose 3mm USB for data reduction. **fixed.**
11. For data with rapid phase decorrelation, the rf.visnorm=T option does not work as well as expected. **fixed.**
12. Phase transfer from 3mm to 1mm does not work correctly for data that has phase wrapping (e.g. 25mar97 GG Tau data). **fixed.**
13. iramcal should not produce plots with flagged data. **fixed.**
14. iramcal.phcor gave an error when running from glish. **fixed.**
15. iramcal.flux doesn't account for flag weights properly when calculating antenna efficiency. **fixed.**
16. iramcal.flux doesn't work when the spectral window changed in an observation. **fixed.**

I.2 Problems found and NOT yet fixed

1. The Antenna efficiencies that are reported during iramcal.flux have incorrect error estimates. **Bug report submitted after testing. Not fixed.**
2. When data is missing at the beginning or end of the observation for one antenna (due to, e.g., shadowing), the phase calibration solutions become wrong for all antennas. **Bug identified late in testing. Not fixed.**

3. Image concatenate did not work on more than 2 channels because the reference frequency (and hence velocity) was not filled. This also affected velocity selection parameters in imager. **Bug report submitted after testing. Kumar wrote a script to do the image concat to work around this problem for now.**
4. During interactive CLEANing of a mosaic, the first image produced to define the mask region was not correct. **Work-around found. Bug report submitted after testing. Not fixed.**
5. msplot shows same uv coverage regardless of frequency. **Bug report submitted after testing. Not fixed.**
6. Creating a user-defined catalog GUI (rather than the default catalog) hangs after selection of functions. **Bug report submitted after testing. Not fixed.**
7. The vpmanager.setpbgauss function to define the primary beam pattern for PdBI (required for mosaicing) did not work all the time (AIPS++ installed at IRAM had this failure, AIPS++ installed at NRAO did not have a problem). **Bug report submitted last day of testing. Not fixed.**
8. Using imager.setdata/setimage to define 4 images, each composed of 32 channels, only generated a single image plane. Where as, defining 8 images (16 channels each) worked as expected. **Bug report submitted near the end of testing. Not fixed.**
9. During baseline calibration (iramcal.rf), if an antenna (or just the first antenna?) is missing in the solution, the resulting plot that is produced has BoundingBox errors and cannot be displayed. **Not fixed.**
10. Using viewer in fitspf crashes aips++. **Bug report submitted last day of testing. Not fixed.**
11. In the almatifiller, putting a "/" at the end of the fitsdir directory path causes a SEVERE error - AIPS++ should handle this even more gracefully and simply let the user know that the format specification is wrong or it should accept both formats (with or without the "/" at the end of a directory path) **Not fixed.**
12. Omitting the specification of a mask region on the first channel in a cube causes AIPS++ to crash. **Bug report submitted after testing completed. Not fixed.**
13. ALMA: tool not written to scripter. **Not fixed.**
14. Flagger chokes when flagging iram dataset. **Not fixed.**
15. Antenna positions are not rotated out of the local coord sys. **Not fixed, a minor problem in a plot, did not affect data.**
16. During imaging, setimage inputs did not stay fixed when the cursor moved over the entry input. **A minor annoyance. Bug report submitted after testing.**
17. The documentation describing how to control the multi-field mosaicing options is not adequate. **Bug report submitted after testing. Not fixed.**
18. almatifiller returns F, not T, when finished. **Minor annoyance. Not fixed.**
19. Clean says it did not reach threshold when it did. **Minor annoyance. Not fixed.**
20. During the Summer 2002 Audit of AIPS++, it was found that setting the vector shape in glish to have an axis=0 (e.g. a::shape=[0,5];) hangs AIPS++ when you request a report of the vector values. **Not fixed.**

I.3 Enhancements made during Phase II testing

1. Requested that iramcal should not produce plots with flagged data. **done.**
2. Requested that iramcal functions should be able to choose the reference antenna. **done.**
3. Requested that iramcal.flux should be able to limit time range. **done.**

4. The phase transfer from 3mm to 1mm should be able to be derived from multiple spectral windows. **done.**
5. Requested that iramcal.rf should have an input for the polynomial degree you want to fit to phase and amplitude bandpasses. **done.**
6. The plot produced by iramcal.flux should be saved as a ps file. **done.**

I.4 Enhancements requested but not yet implemented

1. Requested that iramcal.phcor should plot alma-flag-corr-row corrections. **Not implemented yet.**
2. The threshold for which each channel is CLEANed down to is governed by the channel with the brightest emission. This is fine for most VLA data but not for PdBI data where there may not be adequate *uv* coverage. In the HH 288 mosaic the brightest emission was in a channel that could not be adequately cleaned due to significant missing flux. I requested that the logic for defining the CLEANing threshold in each channel be modified to account for this problem. **Identified on last day of testing, found a work-around by imaging each channel separately. Not implemented yet.**
3. imager.weight is designed for single fields. However, to weight a mosaic, one must select each field and set the weights individually. The imager.weight function should have an option to say that you want to weight a mosaic so it will weight each field correctly. **Identified on last day of testing. Not implemented yet.**
4. msplot locate should suppress 'no data found' warnings. **Minor annoyance. Not implemented yet.**
5. iramcal should provide a summary of the total calibration process. **Not implemented yet.**