The third version of the AMBER data reduction software

Fabien Malbet*[a], Gilles Duvert[a], Florentin Millour[b], Jean-Baptiste Le Bouquin[a], Guillaume Mella[a], Luc Halipré[a], Alain Chelli[a], Evelyne Altariba[a], Gérard Zins[a], Sylvain Lafrasse[a]

[a] Université Joseph Fourier - Grenoble 1 / CNRS, Laboratoire d'Astrophysique de Grenoble (LAOG) UMR 5571, BP 53, 38041 Grenoble Cedex 09, France

[b] Max-Planck Institute für Radioastronomie, Bonn, Germany

The VLTI/AMBER instrument

AMBER is the near infrared (1000–2500nm), 3-telescope, interferometric beam combiner of the VLTI (Petrov et al. 2007). AMBER recombination scheme provides spatially coded fringes on an near-infrared camera. The data processing involves the modeling of the interferogram in the detector plane. Most of the basic data reduction involves the calibration, and the use of a linear relationship between the pixels of the interferogram and the three instantaneous complex visibilities, known as Pixel-to-Visibility Matrix (P2VM, cf. Tatulli et al. 2007).

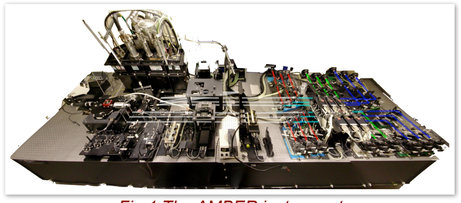


Fig.1 The AMBER instrument

AMBER data reduction software

The data reduction algorithms of AMBER have been incorporated in the instrument software itself (Le Coarer et al. 2004), early in its building, in the form of a C library of functions named `amdlib` (Millour et al. 2004). The AMBER consortium has developed the library for some time and the last version implemented at ESO is `amdlib2`. This library provides the basic functions (computation of the P2VM, real-time measurement of various quantities used in monitoring the observation: atmospheric pistons between telescopes, instantaneous photometries, estimates of V^2, fringe contrast, ...). The same library is also used in the ESO data reduction pipeline for AMBER (Hummel et al. 2008). The C package is complete in the sense that it provides also the basic command-line tools necessary to "reduce" observations, and its final product is uncalibrated OI-FITS files.

Following the delivery of the instrument to ESO, the `amdlib` library maintenance has been taken over by the Jean-Marie Mariotti Expertise Centre (`http://www.jmmc.fr`) and has now been augmented by a Graphical User Interface (GUI) and scripting facility written in Yorick (Munro 1995), with supplementary absolute calibration procedures. Besides, after several years of use of the instrument, that produced numerous scientific results (Malbet et al. 2006), the knowledge of the instrument behaviour ("true instrument") we have acquired during several commissioning runs permitted us to improve considerably the calibration procedure and data reduction scheme. The sum of all this effort is now available with version 3 of the "Amber Data Reduction Package" (hereafter `Amber-DRS`), whose changes with respect to the previous versions are highlighted here.

`amdlib` v3 graphical user interface

The amdlib interface known as **`Amber-DRS`** has been developed in `Yorick` (Munro 1995), an open-source high level interactive scientific language which provides a set of nice interactive graphics plot windows. This interface goes beyond a mere interface of the **`amdlib`** library functions, as we specifically developed:

- a file-browser graphical user interface (GUI), completely written in vanilla yorick,
- advanced scripts to handle many files in batch mode,
- a set of calibration scripts, which are able to provide more advanced calibrations than merely dividing the visibility of one science star file by those of one calibration star file.

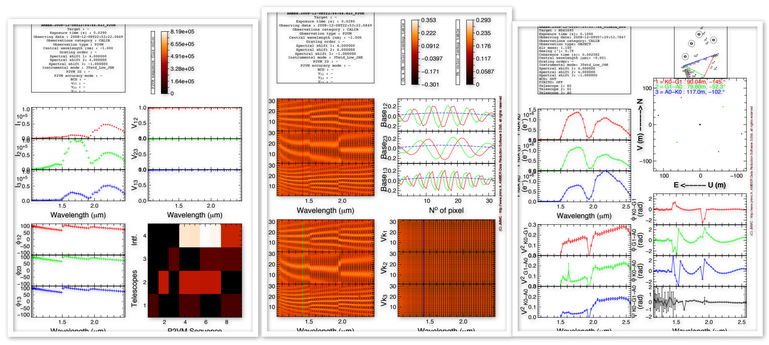


Fig.2: Display of the yorick interface for amdlibShowP2vm, amdlibShowOiData

The file browser GUI is a side-development of **`amdlib`**, which is designed and provided as a separate module called `yoco` (for *yorick contributions*). This module also provides a **`cfitsio`** interface, a specific set of astronomy-related functions, plus several other utilitary scripts made to ease the use of some **`yorick`** functionality. For now this module is only distributed as a part of **`Amber-DRS`**, but it will soon be distributed as a standalone yorick plugin since most of its functions could be used in contexts far beyond those of `Amber-DRS`.

The batch-mode scripts have all their names appended with "`All`" compared to the standard C-type **`amdlib`** functions. For example, the command-line C-filter proposes ***`amdlibComputeOiData`***, whereas the yorick command-line proposes both ***`amdlibComputeOiData`*** and ***`amdlibComputeAllOiData`***. While the yorick ***`amdlibComputeOiData`*** function provides the same functionality as the C-filter complemented with a convenient GUI to select files graphically, the function ***`amdlibComputeAllOiData`*** computes the raw visibilities for a full night, selecting automatically observation, darks, and P2VM files.

`amdlib` v3 core library improvements

The **`amdlib`** core library, which is responsible for computing the instantaneous correlated fluxes and all the basic interferometric observables (V^2, differential phases and closure phases), has been completely rewritten between versions 2 and 3. It implements most of the algorithms published by Chelli et al. (2009) as well as workarounds of some of the problems audited by the "ATF team" (Malbet et al. 2008):

✓ **Data and Noise Model:** The **`amdlib v3`** library implements the new algorithms based on an improved data and noise model as described in Chelli et al. (2009). In a nutshell, the previous version of amdlib, based upon theoretical assumptions on the instrument, gave biased visibilities (especially at low flux, see Chelli et al. 2009 for details).

✓ **Algorithms**: A number of improvements were also added to the core library.

- The wavelength displacement between the three photometric beams is automatically taken into account.
- In low-resolution mode, the algorithm detects the H-K interband phase shift and compensate the defects of repositioning of the spectrograph prism by displacing the wavelength table of the amount needed to bring the H-K interband to its nominal position.
- **`amdlib v3`** provides on-the-fly bad pixel detection.
- **`amdlib v3`** uses a refined algorithm to compute pistons, and an heuristic scheme to evaluate this piston "goodness of fit". Also, we added a piston closure algorithm to improve piston estimate.
- **`amdlib v3`** uses a goodness of fit test to tag individual visibilities which are not well fitted by the carrying waves of the interferogram.
- similarly, the program tags all visibilities where one of the photometries is below a used-defined value (0 being the default).
- Finally, all bad values of the instantaneous or averaged interferometric observables are tagged in the OI- FITS file using the FLAG columns.

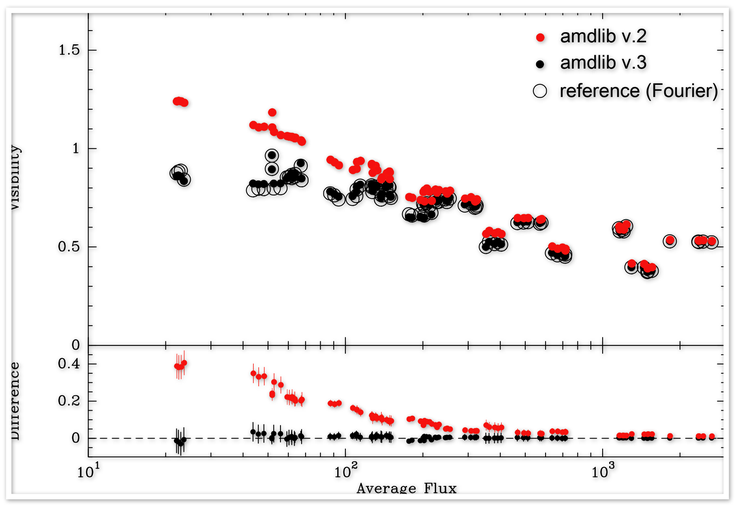


Fig. 3- Top: comparison of visibilities (on various objects with different fluxes and integration times) obtained on 2T experiments with amdlibv2 (red dots) and amdlibv3 (black dots), wrt. the value obtained with a Fourier method (circles), taken as reference (given the recombination scheme of AMBER, only 2T experiments can be reduced both with the P2VM method and with a classical Fourier method). Bottom: visibility difference between each version of amdlib and the classical Fourier, showing the accuracy of the amdlib v3 solution.

Night transfer function and AMBER data absolute calibratiion

The yorick interface of **`amdlib v3`** allows the user to compute the transfer function throughout the night with identification of the different set-ups and calibrators. Then the user must choose the right time interval and the righ wavelength range to produce calibrated data. This is done by using first **`amdlibSearchAllDiameters`** and the yorick script **`amdlibCalibrateAllOiData`**.

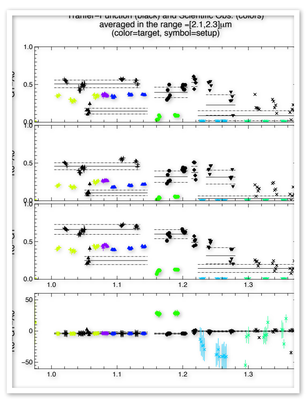
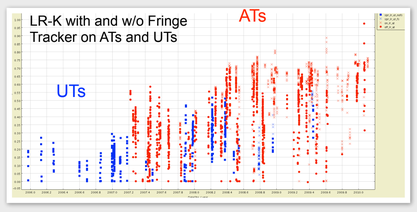


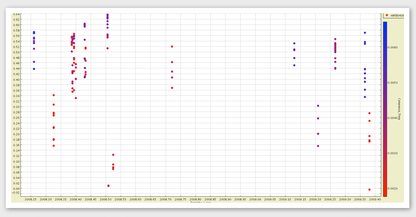
Fig.4: Example of a night calibration for night 2008-01-01. The coloured markers are SCIENCE objects and the black markers are calibrators. The solid and dashed lines correspond to the best estimate of the transfer function.

Application to long-term AMBER characterization

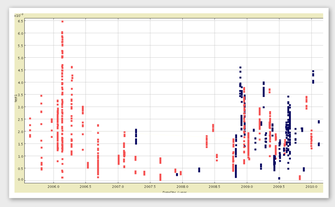
We have used the yorick interface of the Amber-DRS to process all calibrators observed in service mode since the beginning of AMBER in May 2004 and have a complete survey of the transfer function.



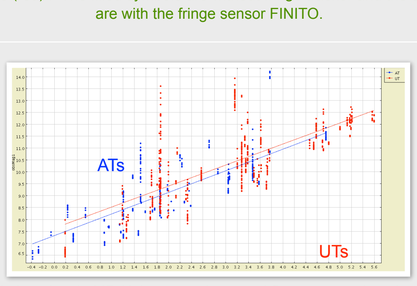
Time series of the transfer function with time from 2004 to 2010 on UTs (blue) and ATs (red) . The filled symbols are without fringe tracker and the crosses are with the fringe sensor FINITO.



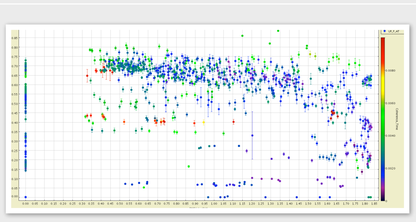
Variation of the transfer function with the coherence time for the calibrator HIP 86406 in low resolution K without fringe tracker on the ATs.



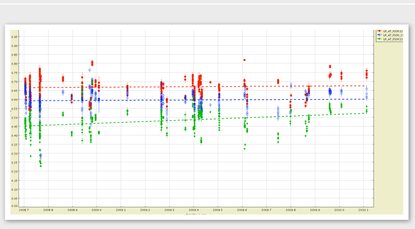
Flux measured in function of time in AMBER. The magnitude is computed the same way as in the bottom left panel.



Correlation plots in MR-K between the actual flux measured on the y-axis and the catalog K magnitude on the x-axis.



AMBER transfer function in presence of the fringe tracker FINITO . The plot represents the transfer function in function of the the residual RMS phase of FINITO.



Time series on a good RMS sequence in LR on ATs. Red symbols are baseline 12, blue ones are baselines 23 and the remaining green symbols are baselines 13.

Conclusion

The new version of the AMBER DRS will be made public as soon as the Preliminary Aceeptance Chile (PAC) of AMBER will be passed and it will be distributed by the JMMC at the address:

`http://www.jmmc.fr`. We would like to thanks all beta-testers and ESO staff for their precious help.

References

Chelli et al. 2009, A&A 502, 705
Hummel et al. 2008, SPIE 7013, E136
Le Coarer et al. 2004, SPIE 5492, 1423
Malbet et al. 2006, SPIE 6268, 2
Malbet et al. 2008, ESO Doc No. VLT-TRE-AMB-15830-7120, issue 1.2 - arXiv: 0808.1315
Millour et al. 2004, SPIE 5491, 1222
Munro 1995, Comput. Phys. 9(6), 609
Petrov et al. 2007, A&A 464, 1
Tatulli et al. 2007, A&A 464, 29