

ONERA

THE FRENCH AEROSPACE LAB

DÉPARTEMENT OPTIQUE THÉORIQUE ET
APPLIQUÉE

Technical Report

WISARD software documentation

L. MUGNIER & S. MEIMON May 2007

CONTENTS

ABBREVIATIONS AND ACRONYMS	4
1. INTRODUCTION	5
2. INSTALLATION OF THE WISARD SOFTWARE	6
2.1. Requirements	6
2.2. Unpacking	6
2.3. Dependencies	7
2.4. Installing WISARD	7
2.5. Acknowledgements	7
3. WISARD'S USER MANUAL	8
3.1. General presentation	8
3.2. List of parameters and accepted keywords	8
3.3. Example of use of WISARD	10
4. WISARD DESIGN REPORT	14
4.1. Introduction	14
4.2. Structure of WISARD	14
4.3. Interferometric observables	15
4.3.1. <i>Ideal interferometric data</i>	16
4.3.2. <i>Data model</i>	17
4.3.3. <i>Input Data format in WISARD</i>	17
4.4. From input data to a myopic model	17
4.4.1. <i>Visibility modulus pseudo data</i>	18
4.4.2. <i>Visibility phase pseudo data</i>	18
4.5. From a myopic model to a myopic convexified model	19
4.6. The criterion minimized	20
4.6.1. <i>Expression for the aberration step</i>	20
4.6.2. <i>Expression for the object step</i>	21
APPENDIX I. THE CLOSURE AND BASELINE OPERATORS C AND B	22
APPENDIX II. SQUARE-ROOT OF A GAUSSIAN DISTRIBUTION	23
APPENDIX III. CARTESIAN GAUSSIAN APPROXIMATION TO A POLAR GAUSSIAN DISTRIBUTION	25
III.1. General expression	25

UNCLASSIFIED
(SANS MENTION
DE PROTECTION)

III.2. Gaussian Approximation	25
III.3. The scalar case	26
5. BIBLIOGRAPHY	28

ABBREVIATIONS AND ACRONYMS

CeCILL	Ce[a] C[nrs] I[nria] L[ogiciel] L[ibre] free software license agreement
EII	European Interferometry Initiative
FITS	Flexible Image Transport System
FP 6	Sixth Framework Programme of the European Union
GPL	GNU General Public License
IAU	International Astronomical Union
JRA 4	Join Research Activity 4
NA	Not Applicable
OI FITS	Optical Interferometry FITS
VLT	Very Large Telescope Interferometer
WP	Work Package

1. INTRODUCTION

This report documents the image reconstruction software, called WISARD, developed by ONERA in WP 2.5 of the JRA 4, within the framework of the FP 6. Together with the software archive, it constitutes ONERA's contribution to the outputs of WP 2.5 (Image Reconstruction Tools).

WISARD stands for "Weak-phase Interferometric Sample Alternating Reconstruction Device". It is a software for the reconstruction of images from interferometric data such as the ones that can be recorded by, *e.g.*, the European VLTI.

It is based on (and quite thoroughly described in) the PhD thesis work of S. Meimon [1] and in references [2, 3].

The WISARD software described here is a complete rewrite (from scratch) by S. Meimon and L. Mugnier of earlier programs of the aforementioned thesis. This rewrite has bought us a dramatic gain in speed and in code maintainability as well as the elimination of quite a few bugs. Yet, there may of course remain some. This code and its documentation are distributed in the hope that it will be useful, but on an *as is* basis, without any warranty of any kind. More precisely, WISARD is a free software, licensed under the CeCILL-B license version 1, which can be found at <http://www.cecill.info>.

WISARD is written in the commercial language IDL of ITT Visual Information Solutions: <http://www.ittvis.com/idl/>. It should thus also run with GDL, the GNU Data Language, available at <http://gnudatalanguage.sourceforge.net/>.

Chapter 2 contains the instruction for installing WISARD on your computer.

Chapter 3 documents the use of WISARD and contains an example of reconstruction from interferometric data. The data comes from the Imaging Beauty Contest 2004 organized by Peter Lawson for the IAU [4]. The example batch file is part of the distribution and can be used and modified for self-study.

Chapter 4 contains an in-depth description of the method implemented by the WISARD code. For more details on the method, the interested reader should consult [1, 2, 3]. Ref. [1] is available on the French national multidisciplinary thesis server TEL at the following address: http://tel.archives-ouvertes.fr/docs/00/05/44/98/PDF/these_finale.pdf. The two other references are also available on-line, at <http://laurent.mugnier.free.fr/publis/Meimon-JOSAA-05.pdf> and <http://laurent.mugnier.free.fr/publis/Meimon-OL-05.pdf> respectively.

2. INSTALLATION OF THE WISARD SOFTWARE

2.1. Requirements

The WISARD software should run on all platforms such that (a) IDL or GDL is installed, and (b) the OptimPack optimization package of Éric Thiébaud (see Section 2.3) is installed. This means that WISARD can be installed and run on at least all Unix-like platforms.

2.2. Unpacking

WISARD is distributed in a compressed archive called `Wisard-<date>.tar.gz`. Unpacking it is straightforward and will create a `WISARD/` directory under which is the distribution. From the shell prompt, type:

```
tar xvfz Wisard-<date>.tar.gz
```

The directories created under `WISARD/` and the nature of their contents are the following:

<code>./doc/</code>	This documentation of WISARD
<code>./lib/</code>	WISARD, its sub-routines, and routines used by them:
<code>./lib/wisardlib/</code>	the WISARD main routine and its sub-routines. These routines are under the CeCILL-B license.
<code>./lib/extralib/</code>	Several extra routines: <ul style="list-style-type: none"> – quick-and-dirty code to input (some) OIFITS data (<code>wisard_oifits2data.pro</code>, which needs some work – anyone interested?), plus – routines used by the former code to read the OIFITS format, by J. D. Monnier (see http://www.astro.lsa.umich.edu/~monnier/), plus – some routines from NASA's astro library, used in turn by the former routines, available from http://idlastro.gsfc.nasa.gov/.
<code>./lib/onerolib/</code>	ONERA routines necessary to WISARD. These routines are <i>not</i> part of WISARD but used and distributed by permission of their respective authors. These routines are under the CeCILL-C license.
<code>./lib/optimpacklib/</code>	OptimPack library for IDL (<code>OptimPack_IDL*.so</code>) and its IDL frontend routines (<code>op_*.pro</code>), by É. Thiébaud. This library is <i>not</i> part of WISARD but is used by WISARD and distributed here by special permission of the author.
<code>./optimpack/</code>	OptimPack sources (<code>*.c *.h</code>) and Makefile (see Section 2.3), plus a copy of the IDL frontend routines (<code>op_*.pro</code>). Once again, this library is <i>not</i> part of WISARD but is used by WISARD and distributed here by special permission of the author. These routines are under the GPL.
<code>./pro/</code>	example batch file for WISARD. A good starting point for self-study, cf Section 3.3.
<code>./inputdata/</code>	input data for the example batch file.

2.3. Dependencies

WISARD heavily uses Éric Thiébaud's neat OptimPack software as its minimization engine. So WISARD needs the OptimPack library for IDL (`OptimPack_IDL*.so`) and its IDL frontend (`op_*.pro` files) to be installed. Although OptimPack is *not* part of WISARD, we have included the OptimPack IDL frontend and the pre-compiled OptimPack library for IDL (for Linux and Solaris, both 32 and 64-bit architectures) in the `lib/optimpacklib/` directory of the WISARD distribution, with their author's permission. So if your computer is a Sun running Solaris or a PC running Linux you should have nothing to do.

Should these pre-compiled libraries not work for you, we have also included all the necessary sources (`*.c` `*.h` and `Makefile`) under the `optimpack/` directory. A `make` command from the `optimpack/idl` directory should build the library for your architecture. Then simply copy the resulting `OptimPack_IDL*.so` files in the `lib/optimpacklib/` directory.

2.4. Installing WISARD

Once you have unpacked the archive (Sect. 2.2), and once you have compiled the OptimPack library for IDL if necessary (Sect. 2.3) the only installation to be done is to make all routines under the `lib/` directory known to IDL. Assuming that you unpacked the archive in your home directory, this is achieved by typing the following at the *IDL* prompt:

```
!PATH = expand_path('+~/WISARD/lib') + ':' + !PATH
```

If you have unpacked the archive elsewhere, simply replace `~` with the appropriate directory in the command above. See the example of Section 3.3 for using a relative path instead of an absolute one.

2.5. Acknowledgements

The authors of WISARD hereby express their gratitude to the following people, as WISARD uses some routines of theirs:

- Éric Thiébaud (from CRAL): WISARD heavily uses his (neat) OptimPack software as its minimization engine.
- Frédéric Cassaing and Jean-Marc Conan: WISARD uses a few routines by these ONERA scientists. These routines (and some others by Laurent Mugnier) are distributed along with WISARD, with these authors' permissions, in the `lib/onerilib/` directory.
- John D. Monnier for his OIFITS reading routines, as well as the authors and maintainers of the NASA astronomical library, for their routines used by John's ones. The OIFITS reading routines are available at <http://www.astro.lsa.umich.edu/~monnier/>, while NASA's astronomical library can be found at <http://idlastro.gsfc.nasa.gov/>. For convenience their routines are included in the `lib/extralib/` directory of the WISARD distribution.

3. WISARD'S USER MANUAL

3.1. General presentation

The WISARD reconstruction method is a regularized reconstruction [5, 6], where the solution (reconstructed object) is defined as the one that minimizes a compound criterion (or “metric”) with two terms. One term, called the likelihood term, measures the fit of the reconstruction to the data and the other term (regularization or penalty term) measures the compatibility of the reconstruction with our prior knowledge on the true object. The main routine of WISARD is `wisard.pro`. The input data for `wisard.pro` is typically recorded by an optical interferometer measuring, in time, the simultaneous interferences of at least three telescopes. It consists in squared visibilities and closure phases, the error bars on these quantities, and the spatial frequencies corresponding to each datum. The structure of the data is given in Section 3.2 and more precisions are available in chapter 4.

The prior knowledge is, on the one hand, the hard constraint of positivity of the reconstruction, and, on the other hand, a somewhat fuzzy knowledge that the object to be reconstructed has some kind of smoothness (or piece-wise smoothness, or global smoothness apart from some spikes, etc). In WISARD, several regularization terms are available to embody this prior knowledge on the solution. A smooth solution is obtained by a quadratic regularization, which uses the object's PSD (Power Spectral Density) as input. This regularization is described in [7] and implemented in `j_prior_gauss.pro`.

A piecewise smooth prior is obtained by a linear-quadratic (also called L_1 - L_2) regularization introduced in [8, 9]. This prior is called *edge-preserving* as it allows sharp edges in the object if the data is compatible with them, contrarily to a quadratic regularization.

A variant of this regularization has been designed to grant the solution with some smoothness while allowing spikes; it is a pixel-independent (or white) linear-quadratic regularization called *spike-preserving* in short. Both the edge-preserving and the spike-preserving priors are implemented in `j_prior_l1l2.pro`.

The main parameters for WISARD are the data of course, the Field-Of-View FOV for the reconstruction, the minimum number of pixel NP_MIN the user wants in this FOV, and keywords specific to the desired regularization. For a quadratic regularization, use keyword PSD, and possibly keyword MEAN_O. For an edge-preserving regularization, use keywords DELTA and SCALE. For a spike-preserving regularization, use keywords DELTA, SCALE, and WHITE=1. See 3.2 for the full list of keywords and details on how to set them.

Additionally, some information is displayed while the iterative minimization is in progress, under the following form: ITER/NBITER; Convergence=x. Criterion=jtotal = jdata + jprior where ITER is the number of iterations performed so far, NBITER is the maximum allowed number of iterations, x is the value of the convergence test, jtotal is the current value of the criterion being minimized, jdata is the current value of the likelihood term and jprior is the current value of the likelihood term.

Lastly, we recall that, as for any properly-written IDL program, the on-line documentation on WISARD or any of its sub-routines can be obtained by typing `doc_library, '<routine_name>'` at the IDL prompt (where `<routine_name>` is, for instance, `wisard`).

3.2. List of parameters and accepted keywords

The only positional parameter is `data`. All the other parameters are passed as keywords. The notation /KEYWORD in the following table means KEYWORD=1, as is customary with IDL.

data	(input)	<p>interferometric input data for the reconstruction. <code>data</code> is a vector of structures, one structure per time of measurement, each containing the following fields:</p> <ul style="list-style-type: none"> - VIS2 : the squared visibilities for each baseline; - VIS2ERR : the standard deviation on the squared visibilities; - CLOT : the closure phases for each triplet of telescopes involving telescope 1; - CLOTERR : the standard deviation on the closure phases; - FREQS_U : first coordinate of the spatial frequencies involving telescope 1; - FREQS_V : second coordinate of the spatial frequencies involving telescope 1. <p>The zero frequency must not be present in the data: the data must be normalized, as are OIFITS files, so that the value of the data at this frequency would be 1. In other words, the reconstructed object has a unit sum.</p>
FOV	(input)	<p>Field-Of-View of the reconstructed image, in units consistent with the data. More precisely the unit for FOV must be the inverse of the unit of the arrays of frequencies (<code>FREQS_U</code> and <code>FREQS_V</code>) of the data. Usually <code>FREQS_U</code> and <code>FREQS_V</code> are in $rd^{\ell} - 1$ so FOV should be in rd (radians).</p>
NP_MIN	(input)	<p>MINimum width (Number of Points) of the reconstructed image. The Number of Points of the reconstructed object may be greater, depending on FOV, OVERSAMPLING factor, and frequencies present in the data. See routine <code>WISARD_MAKE_H</code> for details.</p>
OVERSAMPLING	(optional input)	<p>oversampling factor for the reconstructed image. By default, <code>OVERSAMPLING=1</code> and the maximum spatial frequency of the reconstructed object is the maximum frequency of the data. See routine <code>WISARD_MAKE_H</code> for details.</p>
GUESS	(optional input)	<p>initial guess for the reconstructed image (or dirty map if not present). This guess is massaged in the following way: resampled if necessary to the correct size, thresholded to positive values and normalized to a unit sum. This massaged guess is available on exit as a field of <code>AUX_OUTPUT</code>.</p>
NBITER	(optional input)	<p>maximum NumBer of ITERations for the reconstruction, 500 by default. For a better control of the reconstruction, one should rather use <code>THRESHOLD</code> below and not lower this value.</p>
THRESHOLD	(optional input)	<p>convergence THRESHOLD to be used as a stopping criterion for the iterations. By default set to the machine precision in simple precision (approximately $1.19e-07$), although computations are done in double precision. For a (rather) quick-look result, set to a smaller value, e.g., 10^{-6}.</p>
POSITIVITY	(optional input)	<p>POSITIVITY constraint for the reconstruction. It is set to true (1) by default. Set it explicitly to 0 if by misplaced curiosity you do not want to use the positivity constraint in the reconstruction.</p>

MAY 2007

UNCLASSIFIED (SANS MENTION DE PROTECTION)
--

PSD	(optional input)	2-D map of size NP_MIN×NP_MIN containing the PSD for the (quadratic) regularization of the reconstruction. See routine J_PRIOR_GAUSS and the example file for details.
MEAN_O	(optional input)	2-D map of size NP_MIN×NP_MIN containing the MEAN Object to be used for regularization of the reconstruction. MEAN_O is used both for PSD regularization and for white L1-L2 regularization (i.e., when DELTA, SCALE and WHITE are set).
DELTA	(optional input)	scalar factor for L1-L2 regularization, used to set the threshold between quadratic (L2) and linear (L1) regularization. See routine J_PRIOR_L1L2 and the example file for some more details. See [9] for a complete description of the L1-L2 regularization. The PDF of this paper is on-line at: http://laurent.mugnier.free.fr/publis/Mugnier-JOSAA-04.pdf .
SCALE	(optional input)	scalar SCALE factor for L1-L2 regularization. should be of the order of the average object value if WHITE is set, and of the order of the RMS object's gradient value if WHITE is not set. See J_PRIOR_L1L2 and example file for some more details. See the paper cited above for the whole story.
WHITE	(optional input)	flag to switch between edge-preserving regularization (WHITE=0, default) and spike-preserving regularization (WHITE=1). In the latter case the regularization is performed independently on each pixel value, hence the flag name.
LIBRARY	(optional input)	full path of the OptimPack library (see FMIN_OP for details), if necessary. Under Unix systems the OptimPack library should be found automatically.
AUX_OUTPUT	(optional output)	structure containing various optional AUXiliary outputs, for diagnostic purposes. The details of this structure is given in the on-line documentation for the wisard routine.
/DISPLAY	(optional input)	display the reconstructed object and the fit to the visibilities along the way.
/VERSION	(optional input)	prints version number before execution.
/HELP	(optional input)	prints the on-line documentation and exits.
/COPYRIGHT	(optional input)	prints information about copyright and exits

3.3. Example of use of WISARD

An example of use of WISARD for several regularizations types and levels is the `wisard_batch.pro` file, located in the `pro/` directory of the distribution. Studying this example, running it and modifying its parameters should be a good starting point for building your own reconstruction know-how.

The data used here from the Imaging Beauty Contest 2004 organized by Peter Lawson for the IAU [4]. With the software distributed here, you will obtain reconstructions notably better than the ones we obtained with WISARD at the time [10], and at least as good as the best one of [4].

Below are described the essential portions of this file:

First, change directory to the `pro/` directory of the WISARD distribution and run the IDL interpreter.

MAY 2007

UNCLASSIFIED (SANS MENTION DE PROTECTION)
--

If for instance WISARD has been unpacked in your home directory, it is under the ~/WISARD directory, the commands to be issued at the shell prompt are:

```
cd ~/WISARD/pro
idl
```

Then, at the IDL prompt, add the libraries needed by WISARD to IDL's search PATH:

```
!PATH = expand_path('+../lib') + ':' + !PATH
```

Read the license:

```
dummy = wisard(/copyright)
```

If you accept it, you can go on and pre-process the data file, which is in OIFITS format, into a structure suitable for input into WISARD:

```
datafilename = '../inputdata/dataImagingBeautyContest2004.oifits'
data = wisard_oifits2data(datafilename)
```

Choose the reconstructed Field-Of-View (FOV) and grid size. The default FOV is the inverse of the minimum spatial frequency present in the data, which is often too small (if the data lacks low frequencies). For a 32×32 reconstructed object and a FOV of 18 milli-arcseconds (for instance), type:

```
NP_min = 32L
onemas = 1d-3*(!DPi/180D)/3600D ; one mas in radian
fov = 18.0*onemas
```

Choose the convergence threshold used to stop the iterations. By default one waits until the criterion no longer evolves (the default threshold being given by machine precision and of the order of 10^{-7}). For experimenting with the code, you may choose a larger value, such as 10^{-6} :

```
threshold = 1d-6
```

You may also give an initial guess (if you have performed other reconstructions, or if you want to see how stable the reconstruction is with respect to the initial guess). By default (if the guess is 0 or absent, the so-called *dirty map* is computed by WISARD on the appropriate grid, thresholded to positive values, and used as an initial guess:

```
guess = 0
```

For a first reconstruction, you can use a Gaussian or PSD-based regularization. The PSD must be a 2-D map in Fourier space containing the assumed PSD of the object to be reconstructed. For a disk-like object you should take a PSD with a $1/f^3$ dependence on the spatial frequency (as the square of the FT of a disk has an envelope that goes to zero with this dependence). You should threshold this PSD so that it remains finite at 0 and so that its dynamic range remains reasonable (as a large dynamic range for the PSD may hinder the minimization):

```
distance = double(shift(dist(NP_min), NP_min/2, NP_min/2))
PSD = 1D/((distance^3 > 1D) < 1d6)
```

Run WISARD, displaying both the object being reconstructed (in window 0) and the fit of the reconstructed visibilities to the data (in window 1):

```
x_psd = WISARD(data, $
    FOV = fov, NP_MIN = np_min, $
    GUESS = guess, THRESHOLD = threshold, $
    PSD = psd, $
    AUX_OUTPUT = aux_output_psd, /DISPLAY )
```

MAY 2007

UNCLASSIFIED
(SANS MENTION
DE PROTECTION)

The obtained reconstruction shows the correct overall shape of the reconstructed object, on a dark and quite noiseless background, but without the central peak of the original object (see[4]).

A global factor multiplying the PSD acts as the inverse of a regularization parameter. In other words, for more regularization (to get a smoother object), you should divide the PSD by, *e.g.*, a factor 10 (or more) *i.e.*, assume a weaker object; and for less regularization (to get an object with more details), you should multiply the PSD by, *e.g.*, a factor 10 (or more) *i.e.*, assume a stronger object. If you set $PSD = 100D * psd$ for instance, you will get an under-regularized solution very similar to that obtained without a regularization: quite noisy, but with the emergence of the central peak present in the original object.

Let's now try the white L1-L2 (spike-preserving) regularization. Set $WHITE=1$ and choose $SCALE$ of the order of the object's mean value (this is easy because the reconstructed object is of unit sum). For $DELTA \gg 1$, one obtains a quadratic (L2) regularization and a reconstruction similar to a PSD-based one. For the desired spike-preserving regularization you can set:

```
white = 1B;
scale = 1D/(NP_min)^2; for white L1-L2, scale ~ avg_object_level = 1/NP^2
delta = 1D;
```

Run it:

```
x_l1l2white = WISARD(data, $
                    FOV = fov, NP_MIN = np_min, $
                    GUESS = 0, THRESHOLD = threshold, $
                    SCALE = scale, DELTA = delta, WHITE = white, $
                    AUX_OUTPUT = aux_output_l1l2white, /DISPLAY)
```

The result is much smoother than an under-regularized solution *and* has the central peak of the object as visible as the nose in the middle of one's face (French expression)!

Figure 3.1 shows three reconstructions: an under-regularized PSD-based reconstruction (with the above PSD multiplied by 100), the PSD-based reconstruction run above, and the white L1-L2 reconstruction just obtained.

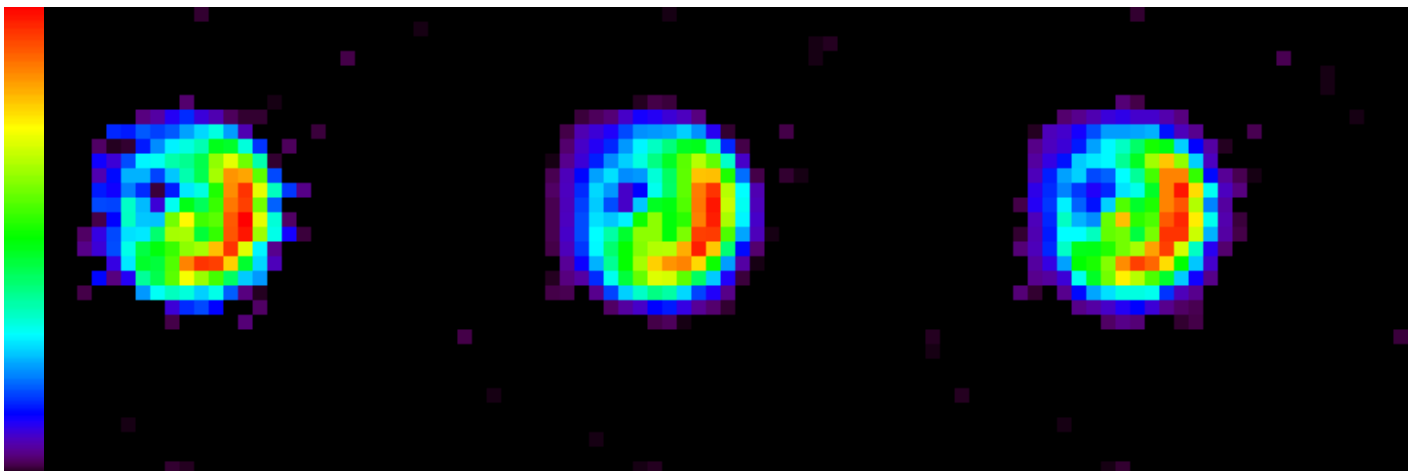


Figure 3.1 – Three reconstructions obtained with WISARD: under-regularized PSD-based (left), correctly regularized PSD-based (center), and white L1-L2 (right). The false-color table is given on the left side.

Figure 3.2 shows the plots of WISARD displayed during the white L1-L2 reconstruction. These plots aim at showing the quality of the fit of the reconstruction to the data: the red crosses show the modulus of

MAY 2007

UNCLASSIFIED
(SANS MENTION
DE PROTECTION)

the reconstructed visibilities (FT of the object at the measured frequencies), while the green squares show the measured visibilities (*i.e.*, the data). The blue line shows the difference of the two, normalized by 10 times the standard deviation of the visibilities. In other words, when the blue line is at a level of 0.1 it means that the fit between the reconstructed and the measured visibilities is at one standard deviation (which means the reconstruction fits the data well!).

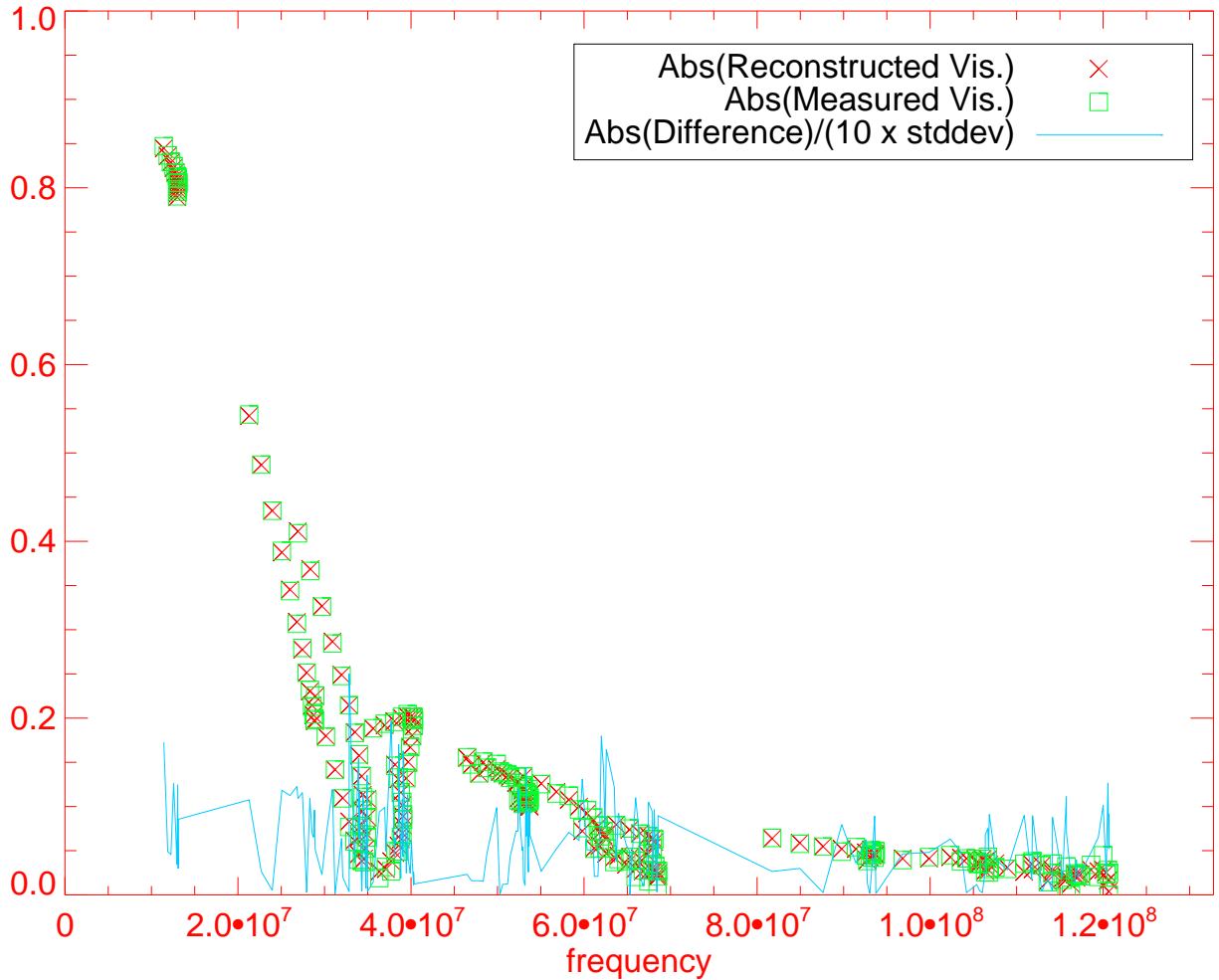


Figure 3.2 – Typical plot displayed by WISARD at convergence (see text).

4. WISARD DESIGN REPORT

4.1. Introduction

Optical interferometry allows one to reach the angular resolution that a hundred meter telescope would provide, using several ten meter telescopes.

The interferograms of current instruments are affected by turbulence, which corrupts the recorded object phases, so one is led to form quantities that are turbulence-independent such as squared visibilities and phase closures.

In order to cope with the missing phase information, we introduce phase calibration parameters to be estimated jointly with the observed object and we propose WISARD (for Weak-phase Interferometric Sample Alternating Reconstruction Device) to perform this estimation. This algorithm combines, within a Bayesian framework, an alternating estimation of the object and phase parameters (in the spirit of self-calibration algorithms proposed by radio-astronomers [11]), a recently developed noise model suited to optical interferometry data [2], and an edge-preserving regularization [9] to deal with the sparsity of the data typical of optical interferometry.

4.2. Structure of WISARD

WISARD is made of the following major blocks:

- a first block `wisard_data2mdata.pro` computes (so-called “myopic”) complex pseudo-data from the input data, and error bars on these pseudo-data. These complex data have the following properties:
 - the phases measurements and error bars are computed from and compatible with the measured phase closures and their error bars;
 - the amplitudes measurements and error bars are computed from and compatible with the measured squared visibilities and their error bars.
- a convexification block `wisard_mdata2cmdata.pro` computes a gaussian approximation of the pseudo visibility data model. This approximation is optimal in the sense of a Kullback Leibler distance (see III);
- a third block `wisard_set_regul.pro` proposes an adapted prior term, to be chosen among a list of regularizations, such as positivity, Power Spectral Density quadratic regularization, linear-quadratic (also called L_1 - L_2) regularization, etc... The algorithmic structure of WISARD make it easy to add other kinds of prior terms to the code;
- a self-calibration block performs the minimization. It alternates
 - a minimization of the criterion w. r. t. the object for given aberrations, which consists essentially in a minimization of a convex criterion under positivity constraint;
 - an optimization of the aberrations for the current object. It is accelerated by performing in parallel several optimizations of a subset of the aberrations, instead of one global optimization.

The pattern of WISARD is described in Fig. 4.1.

MAY 2007

UNCLASSIFIED
(SANS MENTION
DE PROTECTION)

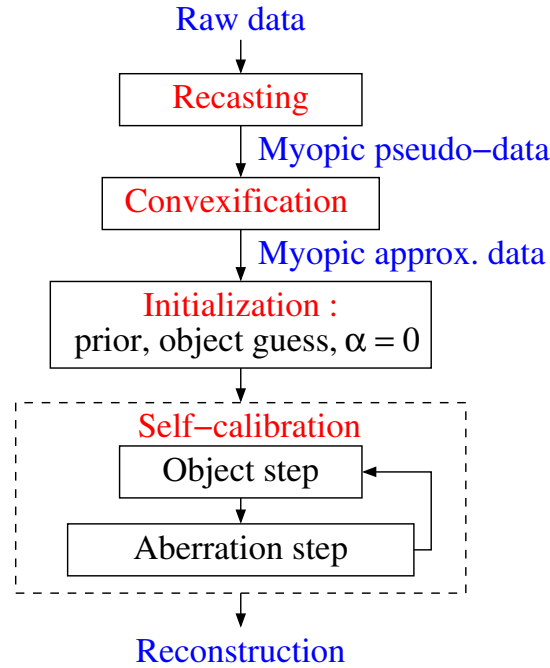


Figure 4.1 – WISARD algorithm loop

4.3. Interferometric observables

Consider a model 2-telescope interferometer, which two identical apertures (T_1, T_2) are located at three-space positions \vec{OT}_1 and \vec{OT}_2 . We denote \mathcal{P} the plane normal to the pointing direction, and \mathbf{r}_1 and \mathbf{r}_2 the projection of \vec{OT}_1 and \vec{OT}_2 onto \mathcal{P} .

The *baseline* \mathbf{u}_{12} is defined by the projection of the displacement $\vec{T_1T_2}$ onto \mathcal{P} :

$$\mathbf{u}_{12} \triangleq \mathbf{r}_2 - \mathbf{r}_1$$

Because of the Earth rotation, the pointing direction changes during an observing night, so the baseline is time dependant too:

$$\mathbf{u}_{12}(t) \triangleq \mathbf{r}_2(t) - \mathbf{r}_1(t) \quad (4.1)$$

Let us consider now a N_t -telescope interferometer. There are as many baselines as ways to choose 2 telescopes among N_t :

$$N_b = \binom{N_t}{2} = \frac{N_t(N_t - 1)}{2} \quad (4.2)$$

For $N_t = 4$, the $N_b = 6$ baselines are:

$$\begin{cases} \mathbf{u}_{12}(t) = \mathbf{r}_2(t) - \mathbf{r}_1(t) \\ \mathbf{u}_{13}(t) = \mathbf{r}_3(t) - \mathbf{r}_1(t) \\ \mathbf{u}_{14}(t) = \mathbf{r}_4(t) - \mathbf{r}_1(t) \\ \mathbf{u}_{23}(t) = \mathbf{r}_3(t) - \mathbf{r}_2(t) \\ \mathbf{u}_{24}(t) = \mathbf{r}_4(t) - \mathbf{r}_2(t) \\ \mathbf{u}_{34}(t) = \mathbf{r}_4(t) - \mathbf{r}_3(t) \end{cases}$$

or in a matrix formulation:

$$\mathbf{u}(t) = \mathbf{B} \cdot \mathbf{r}(t), \quad (4.3)$$

where the $N_b \times N_t$ operator \mathbf{B} is the *baseline operator* (see appendix I page 22).

4.3.1. Ideal interferometric data

We consider a monochromatic source with a wavelength λ , and denote $x(\boldsymbol{\xi})$ the brightness distribution, $\boldsymbol{\xi}$ being angular coordinates. For each baseline, it is possible to access to the following short exposure data:

- a phase $\phi^{\text{data}}(t)$
- a modulus $\mathbf{a}^{\text{data}}(t)$

The complex vector $\mathbf{y}^{\text{data}}(t) = \mathbf{a}^{\text{data}}(t)e^{i\phi^{\text{data}}(t)}$ is called the *complex visibility vector*. According to the Van Cittert-Zernike theorem [12], complex visibilities are *ideally* linked to the Fourier Transform (FT) of $x(\boldsymbol{\xi})$ at the 2D spatial frequency

$$\boldsymbol{\nu}(t) \triangleq \frac{\mathbf{u}(t)}{\lambda}, \quad (4.4)$$

through $y_{ij}^{\text{data}}(t) = \text{FT}[x(\boldsymbol{\xi})](\boldsymbol{\nu}_{ij}(t))$, *i.e.*:

$$\begin{cases} \phi_{ij}^{\text{data}}(t) = \phi_{ij}(\mathbf{x}, t) \triangleq \arg \text{FT}[x(\boldsymbol{\xi})](\boldsymbol{\nu}_{ij}(t)) \\ a_{ij}^{\text{data}}(t) = s_{ij}(\mathbf{x}, t) \triangleq |\text{FT}[x(\boldsymbol{\xi})](\boldsymbol{\nu}_{ij}(t))| \end{cases} \quad (4.5)$$

or in a matrix formulation:

$$\begin{cases} \phi^{\text{data}}(t) = \phi(\mathbf{x}, t) \\ \mathbf{a}^{\text{data}}(t) = \mathbf{a}(\mathbf{x}, t) \end{cases} \quad (4.6)$$

In what follows, we will only consider a discretized version \mathbf{x} of $x(\boldsymbol{\xi})$. The Fourier Transform then corresponds to a matrix product by a Fourier operator \mathbf{H} :

$$\mathbf{y}(\mathbf{x}, t) = \mathbf{H}(t)\mathbf{x} \quad (4.7)$$

For convenience, we also define:

$$\begin{cases} \phi(\mathbf{x}, t) \triangleq \arg(\mathbf{H}(t)\mathbf{x}) \\ \mathbf{a}(\mathbf{x}, t) \triangleq |\mathbf{H}(t)\mathbf{x}| \\ s(\mathbf{x}, t) \triangleq |\mathbf{H}(t)\mathbf{x}|^2 \end{cases} \quad (4.8)$$

The operators $|\cdot|$ and \arg are point-to-point operators.

MAY 2007

**UNCLASSIFIED
(SANS MENTION
DE PROTECTION)**

4.3.2. Data model

The long exposure observables considered in this report are squared moduli $\mathbf{s}^{\text{data}}(t)$ and closure phases $\beta^{\text{data}}(t)$. These observable are affected by a noise, and we suppose in this report that only second degree statistics are provided, and that the closure phase measurement set and the squared modulus measurement set are two uncorrelated sets of variables.

The noise distribution we can make out of these statistics is then gaussian:

$$\begin{cases} \mathbf{s}^{\text{data}}(t) = \mathbf{a}^2(\mathbf{x}, t) + \mathbf{s}^{\text{noise}}(t), & \mathbf{s}^{\text{noise}}(t) \sim \mathcal{N}(0, \mathbf{R}_{\mathbf{s}(t)}) \\ \beta^{\text{data}}(t) = \mathbf{C}\phi(\mathbf{x}, t) + \beta^{\text{noise}}(t), & \beta^{\text{noise}}(t) \sim \mathcal{N}(0, \mathbf{R}_{\beta(t)}) \end{cases} \quad (4.9)$$

The algebraic structure of the *closure* operator \mathbf{C} is described in appendix I page 22.

4.3.3. Input Data format in WISARD

The data format is a vector of structures, one structure for each time of measurement, with the following fields:

- VIS2 : the squared visibilities for each baseline
- VIS2ERR : the standard deviation on the squared visibilities
- CLOT : the closure phases for each triplet of telescopes involving telescope 1
- CLOTERR : the standard deviation on the closure phases
- FREQS_U : first coordinate of the spatial frequencies involving telescope 1
- FREQS_V : second coordinate of the spatial frequencies involving telescope 1

For example, at each moment t , for a 4-telescope array, the fields contain:

field	#	notation
VIS2 :	6	$s_{12}^{\text{data}}(t), s_{13}^{\text{data}}(t), s_{14}^{\text{data}}(t), s_{23}^{\text{data}}(t), s_{24}^{\text{data}}(t), s_{34}^{\text{data}}(t)$
VIS2ERR :	6	$\sigma_{s_{12}(t)}, \sigma_{s_{13}(t)}, \sigma_{s_{14}(t)}, \sigma_{s_{23}(t)}, \sigma_{s_{24}(t)}, \sigma_{s_{34}(t)}$
CLOT :	3	$\beta_{123}^{\text{data}}(t), \beta_{124}^{\text{data}}(t), \beta_{134}^{\text{data}}(t)$
CLOTERR :	3	$\sigma_{\beta_{123}(t)}, \sigma_{\beta_{124}(t)}, \sigma_{\beta_{134}(t)}$
FREQS_U :	3	$u_{12}(t), u_{13}(t), u_{14}(t)$
FREQS_V :	3	$v_{12}(t), v_{13}(t), v_{14}(t)$

The covariance matrices $\mathbf{R}_{\mathbf{s}(t)}$ and $\mathbf{R}_{\beta(t)}$ are implicetly supposed diagonal.

4.4. From input data to a myopic model

This section describes the code `wisard_data2mdata.pro`.

We want to recast the data model in a myopic one, with visibility phase and modulus pseudo-data. The corresponding format is described below. The myopic data format is a vector of structures, one structure for each time of measurement, with the following fields:

MAY 2007

**UNCLASSIFIED
(SANS MENTION
DE PROTECTION)**

- VISAMP : the visibility moduli for each baseline
- VISAMPERR : the standard deviation on the visibility moduli
- VISPHI : the visibility phases for each baseline
- VISPHIERR : the standard deviation on the visibility phases
- FREQS_U : first coordinate of every spatial frequency
- FREQS_V : second coordinate of every spatial frequency

For example, at each moment t , for a 4-telescope array, the fields contain:

field	#	notation
VISAMP :	6	$a_{12}(t)^{\text{data}}, a_{13}(t)^{\text{data}}, a_{14}(t)^{\text{data}}, a_{23}(t)^{\text{data}}, a_{24}(t)^{\text{data}}, a_{34}(t)^{\text{data}}$
VISAMPERR :	6	$\sigma_{a_{12}(t)}, \sigma_{a_{13}(t)}, \sigma_{a_{14}(t)}, \sigma_{a_{23}(t)}, \sigma_{a_{24}(t)}, \sigma_{a_{34}(t)}$
VISPHI :	6	$\phi_{12}(t)^{\text{data}}, \phi_{13}(t)^{\text{data}}, \phi_{14}(t)^{\text{data}}, \phi_{23}(t)^{\text{data}}, \phi_{24}(t)^{\text{data}}, \phi_{34}(t)^{\text{data}}$
VISPHIERR :	6	$\sigma_{\phi_{12}(t)}, \sigma_{\phi_{13}(t)}, \sigma_{\phi_{14}(t)}, \sigma_{\phi_{23}(t)}, \sigma_{\phi_{24}(t)}, \sigma_{\phi_{34}(t)}$
FREQS_U :	6	$u_{12}(t), u_{13}(t), u_{14}(t), u_{23}(t), u_{24}(t), u_{34}(t)$
FREQS_V :	6	$v_{12}(t), v_{13}(t), v_{14}(t), v_{23}(t), v_{24}(t), v_{34}(t)$

4.4.1. Visibility modulus pseudo data

Each $\{a_{ij}^{\text{data}}(t), \sigma_{a_{ij}(t)}\}$ is computed from input data $\{s_{ij}^{\text{data}}(t), \sigma_{s_{ij}(t)}\}$ according to:

- in every case, $a_{ij}^{\text{data}}(t) = \sqrt{|s_{ij}^{\text{data}}(t)|}$;
- if $s_{ij}^{\text{data}}(t) > \sigma_{s_{ij}(t)}$ then $\sigma_{a_{ij}(t)} = \frac{\sigma_{s_{ij}(t)}}{2\sqrt{|s_{ij}^{\text{data}}(t)|}}$;
- else $\sigma_{a_{ij}(t)} = \frac{\sqrt{\sigma_{s_{ij}(t)}}}{2}$.

See appendix II for justification of these formulae.

4.4.2. Visibility phase pseudo data

For each instant t , the vector $\phi^{\text{data}}(t)$ - containing the $\phi_{ij}^{\text{data}}(t)$ - and the vector $\sigma_{\phi}(t)$ - containing the $\sigma_{\phi_{ij}(t)}$ - is computed from the vector $\beta^{\text{data}}(t)$ - containing the input data $\beta_{1ij}^{\text{data}}(t)$ - and the vector $\sigma_{\beta}(t)$ -containing the input data $\sigma_{\beta_{1ij}(t)}$ - according to:

- $\phi^{\text{data}}(t) = C^{\dagger} \beta^{\text{data}}(t)$;
- $\sigma_{\phi}(t) = 3 \cdot C^{\dagger} \text{Diag}\{\sigma_{\beta}(t)\} C^{\dagger, T} \otimes \text{Id}$.

MAY 2007

**UNCLASSIFIED
(SANS MENTION
DE PROTECTION)**

where \otimes denotes a term-to-term product, and C^\dagger is defined in appendix I. Let R be a diagonal matrix, the vector of the diagonal components being r . For any matrix M with compatible dimensions, we have:

$$\begin{aligned} \{MRM^T \otimes \mathbf{Id}\}_{ij} &= \delta_{ij} \sum_k \sum_l R_{ik} \Delta_{kl} M_{jl} \\ &= \delta_{ij} \sum_k \sum_l R_{ik} \Delta_{kl} M_{il} \\ &= \delta_{ij} \sum_k M_{ik}^2 r_k \end{aligned}$$

So we can write:

$$\sigma_\phi(t) = 3 (C^\dagger \otimes C^\dagger) \sigma_\beta(t)$$

This is how the computation of $\sigma_\phi(t)$ is coded in `wisard_data2mdata.pro`.

4.5. From a myopic model to a myopic convexified model

This section describes the code `wisard_mdata2cmdata.pro`.

The convexified myopic data format is a vector of structures, one structure for each time of measurement, with the following fields:

- VIS : the complex visibilities for each baseline
- W_RAD : radial weight on the visibility moduli, to be used in criterion (see sect. 4.6).
- W_TAN : tangential weight on the visibility moduli, to be used in criterion (see sect. 4.6).
- FREQS_U : first coordinate of every spatial frequency
- FREQS_V : second coordinate of every spatial frequency

For example, at each moment t , for a 4-telescope array, the fields contain:

field	#	notation
VIS :	6	$y_{12}^{\text{data}}(t), y_{13}^{\text{data}}(t), y_{14}^{\text{data}}(t), y_{23}^{\text{data}}(t), y_{24}^{\text{data}}(t), y_{34}^{\text{data}}(t)$
W_RAD :	6	$w_{\text{rad},12}(t), w_{\text{rad},13}(t), w_{\text{rad},14}(t), w_{\text{rad},23}(t), w_{\text{rad},24}(t), w_{\text{rad},34}(t)$
W_TAN :	6	$w_{\text{tan},12}(t), w_{\text{tan},13}(t), w_{\text{tan},14}(t), w_{\text{tan},23}(t), w_{\text{tan},24}(t), w_{\text{tan},34}(t)$
FREQS_U :	6	$u_{12}(t), u_{13}(t), u_{14}(t), u_{23}(t), u_{24}(t), u_{34}(t)$
FREQS_V :	6	$v_{12}(t), v_{13}(t), v_{14}(t), v_{23}(t), v_{24}(t), v_{34}(t)$

Each $\{y_{ij}^{\text{data}}(t), w_{\text{rad},ij}(t), w_{\text{tan},ij}(t)\}$ is computed from myopic data $\{\phi_{ij}^{\text{data}}(t), \sigma_{\phi_{ij}}(t), a_{ij}^{\text{data}}(t), \sigma_{a_{ij}}(t)\}$

according to:

$$\begin{aligned}
 y_{ij}^{\text{data}}(t) &\triangleq [a_{ij}^{\text{data}}(t) - \bar{y}_{ij}^{\text{data}}(t)] \exp i\phi_{ij}^{\text{data}}(t) & (4.10) \\
 \bar{y}_{ij}^{\text{data}}(t) &= a_{ij}^{\text{data}}(t) \left[e^{-\frac{\sigma_{\phi_{ij}}^2(t)}{2}} - 1 \right] \\
 \Rightarrow y_{ij}^{\text{data}}(t) &= a_{ij}^{\text{data}}(t) \exp \phi_{ij}^{\text{data}}(t) \left[2 - e^{-\frac{\sigma_{\phi_{ij}}^2(t)}{2}} \right] \\
 w_{\text{rad},ij}(t) &= [\sigma_{\text{rad},ij}^2(t)]^{-1} \\
 &= \left[\frac{1 + e^{-2\sigma_{\phi_{ij}}^2(t)}}{2} \cdot \sigma_{a_{ij}}^2(t) + \frac{(1 - e^{-\sigma_{\phi_{ij}}^2(t)})^2}{2} \cdot a_{ij}^{\text{data}^2}(t) \right]^{-1} \\
 w_{\text{tan},ij}(t) &= [\sigma_{\text{tan},ij}^2(t)]^{-1} \\
 &= \left[\frac{1 - e^{-2\sigma_{\phi_{ij}}^2(t)}}{2} \cdot \sigma_{a_{ij}}^2(t) + \frac{1 - e^{-\sigma_{\phi_{ij}}^2(t)}}{2} \cdot a_{ij}^{\text{data}}(t)^2 \right]^{-1}
 \end{aligned}$$

These equations are explained in appendix III.

4.6. The criterion minimized

We define:

$$\begin{aligned}
 z_{\text{rad},ij}(t) &\triangleq \Re e \left\{ z e^{-i\phi_{ij}^{\text{data}}(t)} \right\}, \quad \forall z \in \mathbb{C} \text{ (Cf. eq. 4.10)} \\
 z_{\text{tan},ij}(t) &\triangleq \Im m \left\{ z e^{-i\phi_{ij}^{\text{data}}(t)} \right\}, \quad \forall z \in \mathbb{C} \text{ (Cf. eq. 4.10)} & (4.11) \\
 \mathbf{y}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) &\triangleq \mathbf{y}(t)^{\text{data}} - \mathbf{y}(\mathbf{x}, t) e^{i\bar{B}\boldsymbol{\alpha}(t)}
 \end{aligned}$$

and propose to use the following data-likelihood criterion:

$$\begin{aligned}
 \mathcal{J}^{\text{data}}(\mathbf{x}, \boldsymbol{\alpha}) &\triangleq \sum_t \mathcal{J}^{\text{data}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) \triangleq \sum_t \sum_{ij} \mathcal{J}_{ij}^{\text{data}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) & (4.12) \\
 \mathcal{J}_{ij}^{\text{data}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) &\triangleq \frac{1}{2} w_{\text{rad},ij}(t) [\mathbf{y}_{\text{rad},ij}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t)]^2 + \frac{1}{2} w_{\text{tan},ij}(t) [\mathbf{y}_{\text{tan},ij}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t)]^2
 \end{aligned}$$

4.6.1. Expression for the aberration step

With Eqs. 4.11, we have:

$$\mathbf{y}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) = \mathbf{y}(t)^{\text{data}} - \mathbf{y}(\mathbf{x}, t) e^{i\bar{B}\boldsymbol{\alpha}(t)}$$

Before starting the aberration step, we compute all the elements useful for the minimization independent of $\boldsymbol{\alpha}$, i.e. $\mathbf{y}(\mathbf{x}, t)$, $\boldsymbol{\alpha}(\mathbf{x}, t)$ and $\phi(\mathbf{x}, t)$. To compute the gradient in $\boldsymbol{\alpha}$ we also compute two other quantities, called wx1 and wx2 in the codes. All these quantities are arranged in the structure `weights_x`, which does not depend on the $\boldsymbol{\alpha}$.

4.6.2. Expression for the object step

With Eqs. 4.11 and 4.8, we gather:

$$\begin{aligned} \mathbf{y}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) &= \mathbf{y}(t)^{\text{data}} - \mathbf{H}(t) \mathbf{x} \text{Diag} \left\{ e^{i\bar{B}\boldsymbol{\alpha}(t)} \right\} \\ &= \mathbf{y}(t)^{\text{data}} - \underbrace{\text{Diag} \left\{ e^{i\bar{B}\boldsymbol{\alpha}(t)} \right\} \mathbf{H}(t)}_{PH(t)} \mathbf{x} \end{aligned}$$

Moreover, with Eqs. 4.12 and 4.11, we have:

$$\begin{aligned} \mathcal{J}_{ij}^{\text{data}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) &\triangleq \frac{1}{2} w_{\text{rad},ij(t)} \left[\mathbf{y}_{\text{rad},ij}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) \right]^2 + \frac{1}{2} w_{\text{tan},ij(t)} \left[\mathbf{y}_{\text{tan},ij}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) \right]^2 \\ &= \frac{1}{2} w_{\text{rad},ij(t)} \Re e^2 \left\{ \mathbf{y}_{ij}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) e^{-i\phi_{ij}^{\text{data}}(t)} \right\} + \frac{1}{2} w_{\text{tan},ij(t)} \Im m^2 \left\{ \mathbf{y}_{ij}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) e^{-i\phi_{ij}^{\text{data}}(t)} \right\} \end{aligned}$$

We gather :

$$\begin{aligned} \mathcal{J}_{ij}^{\text{data}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) &= \frac{w_{11,ij(t)}}{2} \cdot \Re e^2 \left\{ \mathbf{y}_{ij}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) \right\} \\ &+ w_{12,ij(t)} \Im m \left\{ \mathbf{y}_{ij}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) \right\} \Re e \left\{ \mathbf{y}_{ij}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) \right\} \\ &+ \frac{w_{22,ij(t)}}{2} \cdot \Im m^2 \left\{ \mathbf{y}_{ij}^{\text{res}}(\mathbf{x}, \boldsymbol{\alpha}(t), t) \right\} \end{aligned} \quad (4.13)$$

with (for clarity, we omit here the ij and t indexes)

$$\begin{aligned} w_{11} &= w_{\text{rad}} \cos^2 \phi^{\text{data}} + w_{\text{tan}} \sin^2 \phi^{\text{data}} \\ w_{12} &= (w_{\text{rad}} - w_{\text{tan}}) \sin \phi_{ij}^{\text{data}}(t) \cos \phi^{\text{data}} \\ w_{22} &= w_{\text{rad}} \sin^2 \phi^{\text{data}} + w_{\text{tan}} \cos^2 \phi^{\text{data}} \end{aligned} \quad (4.14)$$

It is from this expression that the criterion and its gradient w. r. t. the object are computed in WISARD.

APPENDIX I THE CLOSURE AND BASELINE OPERATORS C AND B

Let N_t be the number of telescopes of the interferometric array. We have the following definitions:

$$\mathbf{B}_2 \triangleq \begin{bmatrix} -1 & 1 \end{bmatrix} \quad (\text{I.1})$$

$$\mathbf{B}_{N_t} \triangleq \left[\begin{array}{c|c} -\mathbf{1}_{n-1} & \mathbf{Id} \\ \hline \mathbf{O} & \mathbf{B}_{n-1} \end{array} \right] \quad (\text{I.2})$$

$$\bar{\mathbf{B}}_{N_t} \triangleq \left[\begin{array}{c} \mathbf{Id} \\ \hline \mathbf{B}_{n-1} \end{array} \right] \quad (\text{I.3})$$

$$\mathbf{C}_{N_t} \triangleq \left[-\mathbf{B}_{n-1} \mid \mathbf{Id} \right] \quad (\text{I.4})$$

$$\mathbf{C}^\dagger \triangleq \mathbf{C}^T [\mathbf{C}\mathbf{C}^T]^{-1} \quad (\text{I.5})$$

for $N_t \geq 3$.

It is easy to see that $\mathbf{C}\mathbf{B} = \mathbf{0}$.

MAY 2007

**UNCLASSIFIED
(SANS MENTION
DE PROTECTION)**

APPENDIX II SQUARE-ROOT OF A GAUSSIAN DISTRIBUTION

Let us suppose we measure the squared value s of a positive value a , with an additive noise system:

$$s^{\text{data}} = a^2 + s^{\text{noise}},$$

s^{noise} being 0 mean gaussian with the variance σ_s^2 . Let \hat{a} be the estimator of a from s^{data} defined by $\hat{a} = \begin{cases} \sqrt{s^{\text{data}}}, & \text{if } s^{\text{data}} > 0 \\ 0 & \text{else} \end{cases}$.

Although \hat{a} is not gaussian vector, we will approximate its distribution by a gaussian one. In many cases, this approximation is valid, as shown in fig. II.1.

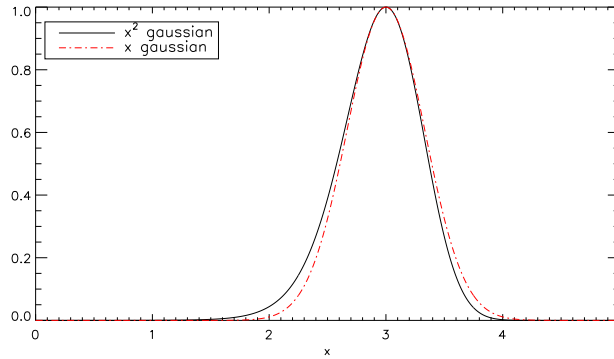


Figure II.1 – Comparison between a gaussian distribution of x and a gaussian distribution of x^2

However, if σ_s is greater than a few a^2 , this estimator is biased. We have studied the behavior of the mean $\langle \hat{a} \rangle$ and standard deviation $\sigma_{\hat{a}}$ of this estimator in function of σ_s/a^2 (see figs. II.2 and II.3).

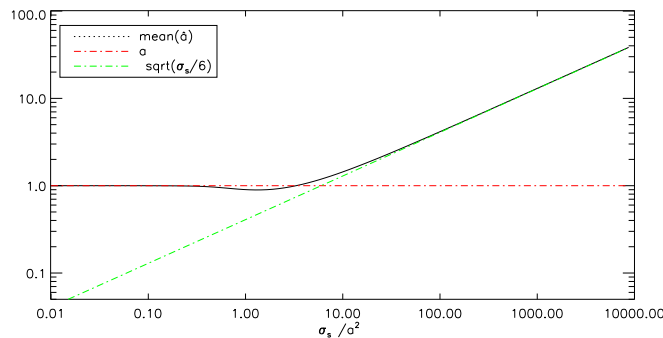


Figure II.2 – Mean of the estimator \hat{a} in function of σ_s/a^2

We then propose the data model for a

$$a^{\text{data}} = a + a^{\text{noise}}$$

MAY 2007

UNCLASSIFIED
(SANS MENTION
DE PROTECTION)

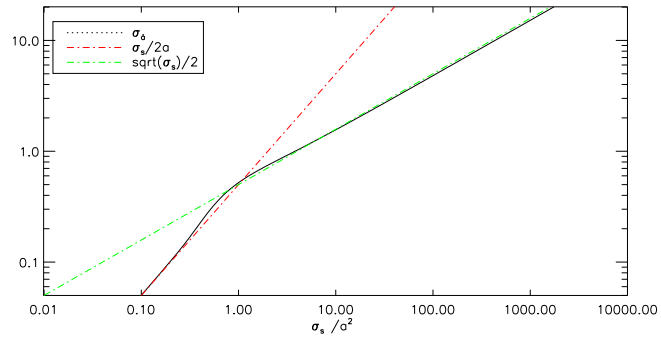


Figure II.3 – Standard deviation of the estimator \hat{a} in function of σ_s/a^2

with :

$$a^{\text{data}} = \begin{cases} 0 & \text{if } s^{\text{data}} \leq 0 \\ \sqrt{\sigma_s/6}, & \text{if } 0 \leq s^{\text{data}} \leq \sigma_s/6 \\ \sqrt{s^{\text{data}}}, & \text{if } s^{\text{data}} \geq \sigma_s/6 \end{cases}$$

$$\sigma_a = \begin{cases} \sqrt{\sigma_s}/2 & \text{if } s^{\text{data}} \leq \sigma_s \\ \frac{\sigma_s}{2\sqrt{s^{\text{data}}}}, & \text{if } s^{\text{data}} \geq \sigma_s \end{cases}$$

We also decide to discard the data such that $s^{\text{data}} + \sigma_s \leq 0$.

APPENDIX III

CARTESIAN GAUSSIAN APPROXIMATION TO A POLAR GAUSSIAN DISTRIBUTION

III.1. General expression

With consider a polar distribution of a gaussian vector \mathbf{y} of modulus \mathbf{a} and phase ϕ :

$$\phi^{\text{data}} = \bar{\phi} + \phi^{\text{noise}} \quad (\text{III.1})$$

$$\mathbf{a}^{\text{data}} = \bar{\mathbf{a}} + \mathbf{a}^{\text{noise}} \quad (\text{III.2})$$

where ϕ^{noise} and $\mathbf{a}^{\text{noise}}$ are 0 mean real gaussian vectors, of covariance matrices \mathbf{R}_a and \mathbf{R}_ϕ (the vectors ϕ^{noise} and $\mathbf{a}^{\text{noise}}$ are supposed uncorrelated).

With the definitions

$$\left\{ \begin{array}{l} \bar{\mathbf{y}} \triangleq \bar{\mathbf{a}} \exp i\bar{\phi} \\ \mathbf{y}^{\text{noise}} \triangleq \mathbf{y}^{\text{data}} - \bar{\mathbf{y}} \\ \mathbf{y}_{\text{rad}}^{\text{n}} \triangleq \Re \left\{ \mathbf{y}^{\text{noise}} e^{-i\bar{\phi}} \right\} \\ \mathbf{y}_{\text{tan}}^{\text{n}} \triangleq \Im \left\{ \mathbf{y}^{\text{noise}} e^{-i\bar{\phi}} \right\} \\ \bar{\mathbf{y}}^{\text{noise}} \triangleq \begin{bmatrix} \mathbf{y}_{\text{rad}}^{\text{n}} \\ \mathbf{y}_{\text{tan}}^{\text{n}} \end{bmatrix} \end{array} \right. \quad (\text{III.3})$$

we gather:

$$\left\{ \begin{array}{l} \mathbf{y}_{\text{rad}}^{\text{n}} = [\bar{\mathbf{a}} + \mathbf{a}^{\text{noise}}] \cos \phi^{\text{noise}} - \bar{\mathbf{a}} \\ \mathbf{y}_{\text{tan}}^{\text{n}} = [\bar{\mathbf{a}} + \mathbf{a}^{\text{noise}}] \sin \phi^{\text{noise}} \end{array} \right. \quad (\text{III.4})$$

A complex vector is gaussian if and only if each of its components is gaussian. A complex is gaussian if and only if, in any cartesian basis, its two components are gaussian. So \mathbf{y} is gaussian if and only if $\bar{\mathbf{y}}^{\text{noise}}$ is gaussian, which is not the case [2]. In what follows, we show how to optimally approximate the distribution of $\bar{\mathbf{y}}^{\text{noise}}$ by a gaussian distribution.

III.2. Gaussian Approximation

We characterize our Cartesian additive gaussian approximation, *i.e.*, its mean $\langle \bar{\mathbf{y}}^{\text{noise}} \rangle$ and covariance $\mathbf{R}_{\bar{\mathbf{y}}^{\text{noise}}}$, by minimizing the Kullback-Leibler distance between the two noise distributions, which gives [2]:

$$\left\{ \begin{array}{l} \langle \bar{\mathbf{y}}^{\text{noise}} \rangle = E \left\{ \begin{bmatrix} \mathbf{y}_{\text{rad}}^{\text{n}} \\ \mathbf{y}_{\text{tan}}^{\text{n}} \end{bmatrix} \right\} = \begin{bmatrix} \bar{\mathbf{y}}_{\text{rad}}^{\text{n}} \\ \bar{\mathbf{y}}_{\text{tan}}^{\text{n}} \end{bmatrix} \\ \mathbf{R}_{\bar{\mathbf{y}}^{\text{noise}}} = E \left\{ \begin{bmatrix} \bar{\mathbf{y}}_{\text{rad}}^{\text{n}} - \mathbf{y}_{\text{rad}}^{\text{n}} \\ \bar{\mathbf{y}}_{\text{tan}}^{\text{n}} - \mathbf{y}_{\text{tan}}^{\text{n}} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{y}}_{\text{rad}}^{\text{n}} - \mathbf{y}_{\text{rad}}^{\text{n}} \\ \bar{\mathbf{y}}_{\text{tan}}^{\text{n}} - \mathbf{y}_{\text{tan}}^{\text{n}} \end{bmatrix}^{\text{T}} \right\} \end{array} \right. \quad (\text{III.5})$$

and we define

$$\mathbf{R}_{\bar{\mathbf{y}}^{\text{noise}}} \triangleq \begin{bmatrix} \mathbf{R}_{\text{rad,rad}} & \mathbf{R}_{\text{rad,tan}} \\ \mathbf{R}_{\text{rad,tan}}^T & \mathbf{R}_{\text{tan,tan}} \end{bmatrix}$$

For a 0 mean gaussian vector ϕ^{noise} of covariance matrix \mathbf{R}_ϕ ,

$$\begin{aligned} E \{ \sin \phi_i^{\text{noise}} \} &= 0 \\ E \{ \cos \phi_i^{\text{noise}} \} &= \exp -\frac{\mathbf{R}_{\phi_{ii}}}{2} \\ E \{ \sin \phi_i^{\text{noise}} \sin \phi_j^{\text{noise}} \} &= \sinh \mathbf{R}_{\phi_{ij}} \cdot \exp -\frac{\mathbf{R}_{\phi_{ii}} + \mathbf{R}_{\phi_{jj}}}{2} \\ E \{ \cos \phi_i^{\text{noise}} \cos \phi_j^{\text{noise}} \} &= \cosh \mathbf{R}_{\phi_{ij}} \cdot \exp -\frac{\mathbf{R}_{\phi_{ii}} + \mathbf{R}_{\phi_{jj}}}{2} \\ E \{ \cos \phi_i^{\text{noise}} \sin \phi_j^{\text{noise}} \} &= 0 \end{aligned} \quad (\text{III.6})$$

By combining Eqs. III.5, III.3, III.4 and III.6, we obtain:

$$\begin{aligned} E \{ \mathbf{y}_{\text{radi}}^{\text{n}} \} &= \bar{a}_i \left[e^{-\frac{\mathbf{R}_{\phi_{ii}}}{2}} - 1 \right] \\ E \{ \mathbf{y}_{\text{tani}}^{\text{n}} \} &= 0 \\ [\mathbf{R}_{\text{rad,rad}}]_{ij} &= \left[\bar{a}_i \bar{a}_j \left(\cosh \mathbf{R}_{\phi_{ij}} - 1 \right) + \mathbf{R}_{a_{ij}} \cosh \mathbf{R}_{\phi_{ij}} \right] \cdot e^{-\frac{\mathbf{R}_{\phi_{ii}} + \mathbf{R}_{\phi_{jj}}}{2}} \\ [\mathbf{R}_{\text{rad,tan}}]_{ij} &= 0 \\ [\mathbf{R}_{\text{tan,tan}}]_{ij} &= \left(\bar{a}_i \bar{a}_j + \mathbf{R}_{a_{ij}} \right) \sinh \mathbf{R}_{\phi_{ij}} \cdot e^{-\frac{\mathbf{R}_{\phi_{ii}} + \mathbf{R}_{\phi_{jj}}}{2}} \end{aligned} \quad (\text{III.7})$$

III.3. The scalar case

Now, we make the additionnal assumption that both ϕ^{noise} and $\mathbf{a}^{\text{noise}}$ are decorrelated, i.e.

$$\begin{cases} \mathbf{R}_a = \text{Diag} \{ \sigma_{a,i}^2 \} \\ \mathbf{R}_\phi = \text{Diag} \{ \sigma_{\phi,i}^2 \} \end{cases}$$

We obtain:

$$\begin{cases} \mathbf{R}_{\text{rad,rad}} = \text{Diag} \{ \sigma_{\text{rad},i}^2 \} \\ \mathbf{R}_{\text{tan,tan}} = \text{Diag} \{ \sigma_{\text{tan},i}^2 \} \\ \mathbf{R}_{\text{rad,tan}} = 0 \end{cases}$$

with

$$\begin{aligned} \sigma_{\text{rad},i}^2 &= \frac{\bar{a}_i^2}{2} \left(1 - e^{-\sigma_{\phi,i}^2} \right)^2 + \frac{\sigma_{a,i}^2}{2} \left(1 + e^{-2\sigma_{\phi,i}^2} \right) \\ \sigma_{\text{tan},i}^2 &= \frac{\bar{a}_i^2}{2} \left(1 - e^{-2\sigma_{\phi,i}^2} \right) + \frac{\sigma_{a,i}^2}{2} \left(1 - e^{-2\sigma_{\phi,i}^2} \right) \end{aligned} \quad (\text{III.8})$$

In this case, we can plot for one complex visibility the true noise distribution - i.e. a gaussian noise in phase and modulus - and our gaussian approximation (see fig. III.1)

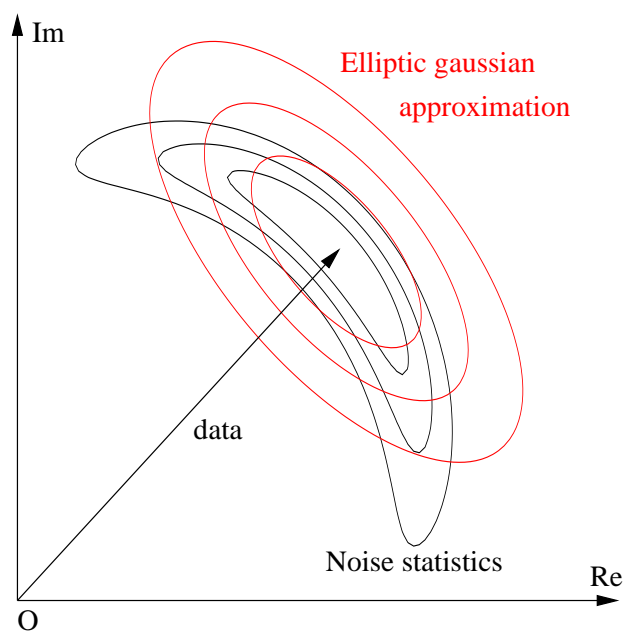


Figure III.1 – Polar gaussian distribution contour plot and its cartesian gaussian approximation

5. BIBLIOGRAPHY

- [1] S. Meimon,
Reconstruction d'images astronomiques en interférométrie optique,
PhD thesis, Université Paris Sud (2005).
1
- [2] S. Meimon, L. M. Mugnier and G. Le Besnerais,
A convex approximation of the likelihood in optical interferometry,
J. Opt. Soc. Am. A (November 2005).
1, 4.1, III.1, III.2
- [3] S. Meimon, L. M. Mugnier and G. Le Besnerais,
Reconstruction method for weak-phase optical interferometry,
Opt. Lett., 30 (14), pp. 1809–1811 (July 2005).
1
- [4] P. R. Lawson, W. D. Cotton, C. A. Hummel, J. D. Monnier, M. Zhao, J. S. Young, H. Thorsteinsson, S. C. Meimon, L. Mugnier, G. Le Besnerais, E. Thiébaud and P. G. Tuthill,
An interferometric imaging beauty contest,
In *New frontiers in stellar interferometry*, edited by W. A. Traub, vol. 5491, pp. 886–899. Proc. Soc. Photo-Opt. Instrum. Eng. (2004),
Conference date: June 2004, Glasgow, UK.
1, 3.3
- [5] G. Demoment,
Image Reconstruction and Restoration: Overview of Common Estimation Structures and Problems,
IEEE Trans. Acoust. Speech Signal Process., 37 (12), pp. 2024–2036 (December 1989).
3.1
- [6] J. Idier, editor,
Approche bayésienne pour les problèmes inverses,
Hermès, Paris (2001).
3.1
- [7] J.-M. Conan, L. M. Mugnier, T. Fusco, V. Michau and G. Rousset,
Myopic Deconvolution of Adaptive Optics Images by use of Object and Point Spread Function Power Spectra,
Appl. Opt., 37 (21), pp. 4614–4622 (July 1998).
3.1
- [8] L. M. Mugnier, C. Robert, J.-M. Conan, V. Michau and S. Salem,
Myopic deconvolution from wavefront sensing,
J. Opt. Soc. Am. A, 18, pp. 862–872 (April 2001).
3.1

- [9] L. M. Mugnier, T. Fusco and J.-M. Conan,
MISTRAL: a Myopic Edge-Preserving Image Restoration Method, with Application to Astronomical Adaptive-Optics-Corrected Long-Exposure Images.,
J. Opt. Soc. Am. A, 21 (10), pp. 1841–1854 (October 2004).
3.1, 3.2, 4.1
- [10] S. C. Meimon, L. M. Mugnier and G. Le Besnerais,
A novel method of reconstruction for weak-phase optical interferometry,
In *New frontiers in stellar interferometry*, edited by W. A. Traub, vol. 5491, pp. 909–919. Proc. Soc. Photo-Opt. Instrum. Eng. (2004),
Conference date: June 2004, Glasgow, UK.
3.3
- [11] T. J. Cornwell and P. N. Wilkinson,
A new method for making maps with unstable radio interferometers,
Mon. Not. R. Astr. Soc., 196, pp. 1067–1086 (1981).
4.1
- [12] J. W. Goodman,
Statistical optics,
John Wiley & Sons, New York (1985).
4.3.1