JMMC-MAN-2300-0003

Revision 1.0

Date: 2 Oct 2009

JMMC

LITpro, a model fitting software

Authors:
Olivier Chesneau et al — FIZEAU

Author: Olivier Chesneau Institute: OCA	Signature: SUBMITTED Date: 2 Oct 2009
Approved by: Isabelle Tallon-Bosc Institute: CRAL	Signature: APPROVED Date: 2009
Released by: Gilles Duvert Institute: LAOG	Signature: RELEASED Date: 2009

Change record

Revision	Date	Authors	Sections/Pages affected
	Remarks		
0.1	01 Nov 2008	Olivier Chesneau	all
	Creation based on some previous documents		
0.2	17 Jul 2009	Guillaume Mella	
	Remove unnecessary parts according to the I.T-B. remarks		
0.3	21 Jul 2009	Olivier Chesneau	
	Final Tuning, polishing		
0.4	21 Aug 2009	Isabelle Tallon-Bosc	
	small corrections; Table 1 compressed; 1 Ref. added; Sec.2.4 rewritten and Sec.3 a little modified		
0.5	14 Sep 2009	Olivier Chesneau	
	Final Tuning, polishing		
0.6	1 Oct 2009	Guillaume Mella	
	Add Isabelle's corrections : biblio and acknowledgements		
1.0	2 Oct 2009	DOCMGR	all
	Release with update of main software url		

Table of contents

List of Tables

List of Figures

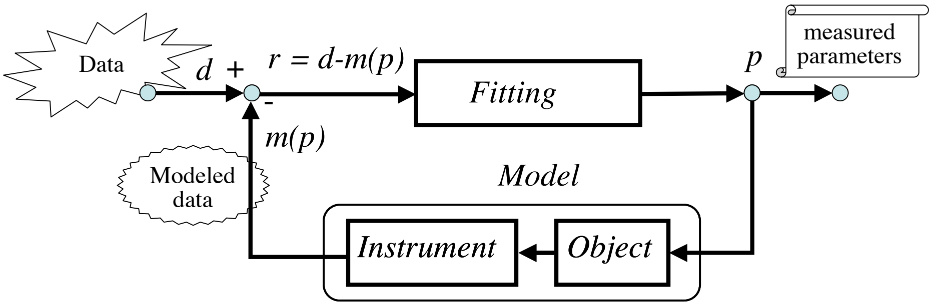

Figure 1: Structure of the Model Fitting Software - diagram taken from [3]

1 Introduction

1.1 Object

This document gives a short overview of the model fitting theory embedded into the `LITpro` software core. The reading of this document can be followed by the reading of the User Manual of the web JAVA Graphical User Interface (GUI)[1] and an extended tutorial [2] focuses on practical aspects of this tool through the analysis of simple examples of a fit, with reference data provided to the user. A reference document is also at disposal for an exhaustive description of the software.

1.2 Reference documents

[1] JMMC-MAN-2300-0001, LITpro GUI - User's Manual (Reference Guide)

[2] JMMC-MAN-2300-0002, Tutorial of LITpro and corresponding Java GUI

[3] JMMC-PUB-2300-0003, paper from the VLTI Euro Summer School of Goutelas, France, 2006, "Model Fitting Tutorial", Tallon-Bosc et al., 2007, New Astronomy Results **51**, 697-705

1.3 Context

The purpose of this section is to present the main components of fitting softwares (Fig. 1). We also outline some details on the capabilities of `LITpro`.

Current optical stellar interferometers do not provide direct images and extracting measurements of geometrical or physical quantities from a limited set of interferometric data is often not an easy task. In contrast to radio interferometry, image reconstruction is still difficult in the optical domain mainly because of the poor uv-coverage and the lack of accurate phase information. Nevertheless, even an imperfect map can be valuable to identify a suitable model of the object, so image reconstruction appears here as very complementary to model fitting. Most of the time, one has to rely on model fitting using simple 'bricks' of spatial distribution of light such as a Uniform Disk for describing a star or a Gaussian distribution to account for smoother structures encountered around dusty sources for instance. Complex radiative transfer models can also directly be fitted on the interferometric data. When fitting, the astronomer wants to estimate the best values of any free parameter of his model, with error bars and some confidence in the results, while avoiding local minima.

The principle of model fitting can be simply described by Fig. 1. By using a model which may include the object as well as the instrument, we can compute modeled data $m(p)$ from the parameters p. The modeled data are then compared with the real data d to get the so-called residuals $r = d - m(p)$. The 'fitting engine' iteratively looks for the set of parameters p which minimizes the residuals.

In the interferometric case, the model of data is complex and non-linear. Inverting means seeking a set of free parameters that allows us to fit the real data with the simulated ones: this is the function of the fitting process. It minimizes the distance between simulated and real data, the so-called residuals, by using an iterative process, and finally provides the values of the adjusted parameters with various useful informations like standard deviations of the parameters, their correlation and covariance matrices.

More precisely, the aim is to find the parameters that maximize the probability of having observed the data with the current model. Assuming that data are independent random Gaussian variables, this is equivalent to minimizing the so-called chi-square:

$$\chi^2(p) = \sum_i^{N_d} \frac{r_i^2(p)}{\sigma_i^2}, \qquad \text{with} \quad r_i(p) = d_i - m_i(p), \tag{1}$$

where N_d is the number of data and σ_i their standard deviations. This equation actually holds for squared visibilities, and for the amplitude of complex visibilities (or bispectrum). Other expressions of chi-square are used for phases of complex data. One of them is Eq. (1) but with r_i as an angle modulo 2π.

Because the models are non-linear functions of the parameters, local minima of $\chi^2(p)$ generally appear in the space of parameters. This is particularly true for light distributions whose Fourier transform oscillates (generated by sharp border or several point like sources). This is the main difficulty of the minimization. We need some strategy to confirm that the global minimum has been found. We can also use the chi-square statistics to evaluate the quality of the fit:

$$\left\langle \chi^2 \right\rangle = N_f = N_d - N_p, \tag{2}$$

$$Var(\chi^2) = 2N_f, \tag{3}$$

where N_p is the number of parameters and N_f the number of degrees of freedom. Since the chi-square statistics depends on N_f, we generally use the reduced chi-square, χ^2/N_f which averages to unity. Note that its statistics is very sharp for large value of N_f: in practice, on real data, it is difficult to use it for assessing the quality of the fit. But it can be a good means to compare two different models and to assess the progress of the fit.

1.4 Presentation

`LITpro` (Lyons Interferometric Tool prototype) is a software developed within the Jean-Marie Mariotti Center (JMMC) research group. The main objective of the JMMC is to help astronomers to analyze and interpret data obtained with optical interferometers (for instance from recombiners of the Very Large Telescopes Interferometer). Activities concern the search of calibrators, image reconstruction, or tools for preparing observations and analyzing data.

`LITpro` is a software for fitting models on data obtained from various stellar optical interferometers. As a baseline, for modeling the object, it provides a set of elementary geometrical and center-to-limb darkening functions, all combinable together.

`LITpro` is based on a modified Levenberg-Marquardt algorithm and its architecture allows a flexible implementation of complex models and fits with heterogeneous data. It includes a Trust Region Method, minimizing a heterogeneous non-linear and non-convex criterion and allows the user to set boundaries on free parameters. From a robust local minimization algorithm and a starting points strategy, a global optimization solution is effectively achieved. Tools have been developed to help users to find the global minimum.

A Java interface available on the web allows you to use this software on line without having to install it To provide specific model-fitting computation inside the Java GUI, the scientific code is not rewritten in Java but is delegated to the existing LITpro ”expert” layer. This layer has been indeed deployed on a dedicated server, to provide Model-Fitting capabilities through a webservice across the Internet. Internally, `LITpro` uses a hierarchical dynamic structure to store data, and model descriptions. The data format for ensuring the communication between the server and the GUI is based on Extensible Markup Language (XML). This is a natural way to store hierarchical information and is well supported under Java. The core of `LITpro` is written in `Yorick`[1]. There is a `Frequently Asked Questions` menu that opens your default Web browser

---

[1] `Yorick` is a free cross-platform data processing language written by D. Munro and available at `ftp://ftp-icf.llnl.gov/pub/Yorick/`.

Table 1: Some useful web-pages:

The JMMC	`http://www.jmmc.fr`
The `LITpro` model-fitting web-page	`http://www.jmmc.fr/litpro`
The JMMC Public repository of OI-FITS files[1]	`http://apps.jmmc.fr/oidata`
The JMMC OI-FITS validator tool	`http://www.jmmc.fr/oival_page.htm`
The VLTI training schools[2]	`http://www.vlti.org/home.php`
The OLBIN web page[2,3]	`http://olbin.jpl.nasa.gov`

[1] in this web page, one can find the first examples to get some training on `LITpro`.
[2] in this web page, one can find a wealth of pedagogical material about optical interferometry.
[3] go to the software section for information on the OIFITS format.

on the `LITpro` FAQ web page (http://www.jmmc.fr/litpro/faq/) that should be steadily growing depending on users feedback.

2 Description

2.1 Reading data

Since `LITpro` must be usable by a large community, it is necessary to read data stored in the format which is now widely used in the Optical Long Baseline INterferometric (OLBIN) community : the OI Exchange Format.

We will not describe in detail the OI Exchange Format (Pauls et al. 2005), based on FITS (Flexible Image Transport System). Some useful information can be found in the OLBIN web page. We emphasize that having the data in this format, as close as possible to its requirements is important. Let us only remind here that an OI-data file is a set of linked tables (FITS bintable): the TARGET table collects information on the object (name, coordinates, etc.), the ARRAY table describes the configuration of the telescopes, the WAVELENGTH table gives the list of observed wavelengths and bandwidths, and other tables give the measurements and the error bars of squared visibilities (VIS2 tables), amplitude and phase of complex visibilities (VIS tables) or amplitude and phase of bispectrum (T3 tables).

As the OI FITS format is quite recent, some variations of format among different groups can occur that may affect the reading of the data. A 'format checker' therefore runs when a new data set is included in the software, and check the various tables and formats in the data file. These tests are currently limited, and we cannot guarantee that all OI-FITS files are readable by the system. We therefore invite the user to have a look on the FAQ web-page.

LITpro reads this format and transforms it in order to simplify and speed up the forthcoming process. For example, it cleans the tables by removing duplicated targets and by splitting the data tables in several data blocks, one per target, so that linking a different model to each target is easier. Also, the baseline coordinates are converted into spatial frequencies.

We finally note that the user also can select the types of data to be fitted: VIS2, VISAMP, VISPHI, T3AMP, T3PHI, SED. This is important to perform partial fitting using some sub-sets of observables affected by different kind of errors, or sensitive to different aspects of the source.

2.2 Modeling data

We need a flexible way of modeling different types of objects. We can first rely on a library of functions **which can be combined to describe a complex object**. In this first release, `LITpro` provides a set of elementary functions including:

- circular ones (i.e. with radial symmetry)

  Uniform and Limb-darkened disks with various center-to-limb darkening functions,

  Circle (infinitely thin ring) and rings,
  Gaussian distributions,

- elongated (or flattened) ones (i.e. with a one axis symmetry)
  Uniform ellipse,
  2D-ring (with finite width),
  2D-Gaussian,
- an unresolved point

The possibilities offered by these simple geometric models are already large for performing an astrophysical interpretation. Stellar photospheres can be described by uniform and Limb-darkened disks, stars with dusty winds by a Gaussian or a Circle if the inner rim is bright. Circumstellar disks can be often described at first order by a 2D-Gaussian or a Uniform ellipse (if truncated), or by a 2D-ring (if the inner rim is bright, depending on the inclination),... Complex sources are generated by combining several of these elemental bricks and we note that the binary and other multiple sources are not proposed directly in this list. The user builds a specific model from a sub-set of these individual bricks. For instance, a binary consisting of two unresolved stars will be modeled by two points. Then, the mutual relationships between these elements must be taken into account by the user, i.e. the separation and the fluxes of the sources normalized to the total flux. The user has to consider whether there is a range of separation constrained by other means (for instance, 2-20 milliarcsecond), or whether there is a acceptable range of flux difference between the sources. Based on the individual characteristics of the elements and the spatial position and flux relationships between them, the generated model can then be compared to the data.

This library allows the user to build a specific flux distribution on the sky based on some bricks. When the number of visibility points is limited, a limit is rapidly reached on the number of models that can account for the data. This strategy cannot cover all the needs and users generally want to implement more specific models with their own parameters, for instant flux distributions generated by radiative transfer code with a limited number of free parameters. `LITpro` is developed to allow an easy implementation of user functions by only requiring them to compute the Fourier transform of the object at given spatial frequencies, wavelengths and time. From this, `LITpro` computes all the necessary quantities as needed (visibilities, spectral energy distribution, partial derivatives of the model, map of the object model). `LITpro` is also designed for fitting simultaneously interferometric data and spectral energy distribution (or other data). It is also of a great interest that the model fitting software can be coupled with astrophysical numerical models. This is foreseen in `LITpro`: either directly by getting the required object map from a given set of free parameters, or from a set of maps pre-computed on a sparse grid of astrophysical parameters: an interpolating algorithm can fit across the maps using some geometrical operations, like a rotation, a scaling or an elongation.

In the frame of the present manual and the tutorial, only the fitting strategy based on the use of simple geometrical models is proposed.

2.3 Fitting engine

The model fitting software must have a competitive and reliable fitting engine. With an iterative procedure, this core of the software provides a numerical solution to the minimization problem of the non-linear residuals. Given a model, an optimal set of values for the parameters is computed. Usually the model does not linearly depend on parameters. Considering all the possible models a user may want to use, nothing can ensure that the chi-square would be a convex function of the parameters. Then, the fitting engine performs an optimization of a non convex inverse problem. As far as possible, the global minimum is searched. Unlike image reconstruction, model fitting deals with a small number of parameters. The user may need to bound those parameters (e.g. positive value, $0 - 2\pi$ domain, etc.). But first of all he must set initial values to those parameters. And it is important to keep in mind that the fit depends of this departure and that it is necessary to perform the fitting process with different starting values, even if some tools exit for reducing this difficulty (see following section).

This means that when using `LITpro`, it is important to keep in mind that the fitting engine needs *a seed, i.e. a set of parameters as starting points.* We stress that in many cases the quality of seed, i.e. the closeness of the initial guess versus the 'reality' of the data may affect the course of the fitting procedure despite some efforts in overcoming this issue (see following section). This also implies that a good healthy check is to perform the fitting work with well-chose different seeds.

`LITpro` uses the Levenberg-Marquardt algorithm, improved by the implementation of trust regions (More & Sorensen 1983). It computes the first partial derivatives of the model by finite differences, so that the user is not requested to provide any function for their computation. Nevertheless, scaling factors may be necessary to let the fitting engine know the order of magnitude of each parameter, so that it can estimate suitable lengths for the estimation of the first finite differences. The *scales* can be automatically estimated (e.g. from the bounds of the parameters when they are known), or adjusted by the user.

After the optimization, the software provides the user with all the necessary quantities to assess the quality of the fit: final chi-squares, errors bars on the estimated parameters, covariance and correlation matrices. The latter measures cross dependencies of the parameters and may outline degenerated parameters and limitations of the model.

2.4 Exploring local minima

An intrinsic difficulty of the model fitting is the existence of local minima, where the fitting engine may be trapped far from the global minimum: the fitting algorithm can indeed converge towards a solution which is not the best one. The initial guess is then critical.

There is currently no general way of solving this problem, other than to try various starting values. Here we propose two different tools for helping the preparation of the fit.

One is called "sniffer_map" and is used if a binarity is suspected. Where is the best position (X,Y) of the secondary component in the field of view when the primary one is supposed to be centered ? To answer this question, "sniffer_map" scans all the possible positions of the coordinates X and Y from their minimal values to their maximal ones, and fit the intensity ratio between the components of the binary model made of 2 single points, one at the origin (0,0) and one at coordinates (X,Y). The result is a map of chi-squares of those fits: the deepest minimum will give the best place where to put the secondary component and hence the best starting values for the fitting of the data with the simple binary model.

"sniffer_map" is currently limited to this action but we foresee to improve its utility, by replacing the single centered point by the current model: detection of a single point near for example a resolved brighter disk will be possible.

Another tool is the analysis of cuts in the chi-square space with "chi2_slice". Cuts versus any parameter of the current model can be plotted with this tool, either versus one parameter p (curve $chi\_square(p)$, called $1Dchi2$) or versus two parameters (map $chi\_square(p1, p2)$, called $2Dchi2$). The such found minima may give the initial values of the parameters that will lead to a better fit.

3 Conclusion

Model fitting is not an easy task, and the `LITpro` tool should be used with some care. It is important to understand how the fitting engine behaves, how to analyze the results of the fit, and that the different types of data may contain various degeneracies which must be taken into account in the design of the models. Furthermore it is useful to have some notion on the relationship between a shape in the image plane and its Fourier transform in the *uv* plane, since model fitting operates in the Fourier plane, i.e. the space of data.

Furthermore, the model fitting must deal with the existence of local minima, often preventing the knowledge of the global minimum. `LITpro` proposes different tools for starting the fit from the best place, like cuts in the chi-square space or directly a map of the chi-square for the fit with the simple binary system.

Other tools must be developped. We indeed constantly improve the capabilities of this software, both the engine and the GUI, to provide the user with a more robust and user-friendly tool.

We now invite you to perform the first tests with the data sets at your disposal, with the help of the User Manual and the Tutorial.

References

[1] Pauls, T. A. and Young, J. S. and Cotton, W. D. and Monnier, John D., in Data exchange standard for optical (visible/IR) interferometry, Publ. Astron. Soc. Pac., 117, 837, 1255-1262, 2005, JF

[2] Moré, J. J. and Sorensen, D. C., in Computing a trust region step, SIAM J. Sci. Stat. Comput., 4, 3, 553-572, 1983, . . . J+

Acknowledgements

The software presented in this article has been implemented in Yorick, a free data processing language written by D. Munro (http://sourceforge.net/projects/yorick/).