Model Fitting tutorial

I. Tallon-Bosc, M. Tallon, E. Thiébaut, C. Béchet

Université de Lyon, Lyon, F-69003, France ; Université Lyon 1, Observatoire de Lyon, 9 avenue Charles André, Saint-Genis Laval, F-69230, France ; CNRS, UMR 5574, Centre de Recherche Astrophysique de Lyon ; Ecole Normale Supérieure de Lyon, Lyon, F-69007, France

Abstract

The purpose of this tutorial is not to describe the theory of model fitting, but to focus on practical aspects and to learn how to work around difficulties. For practice, we use LITpro, a software currently developed within the JMMC[1] research group. LITpro is based on a modified Levenberg-Marquardt algorithm and its architecture allows a flexible implementation of complex models and fits with heterogeneous data. Through the analyzis of an example of a fit on simulated interferometric data, we learn how to deal with degeneracies and how to tackle the problem of local minima, with tools provided by LITpro.

Key words: interferometry, data processing, model fitting, binary stars
PACS: 95.55.Br, 95.75.Kk, 95.75.-z, 97.80.Af

1 Introduction

Although promising devices are being studied for snapshot imaging (Labeyrie, 1996; Vakili et al., 2004), current optical stellar interferometers do not provide direct images. Extracting measurements of geometrical or physical quantities from interferometric data is then not an easy task. We have to rely on model fitting or on image reconstruction. In contrast to radio interferometry, image reconstruction is still difficult in the optical domain mainly because of the poor uv-coverage. 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.

[1] The web-site of the Jean-Marie Mariotti Center is http://www.mariotti.fr

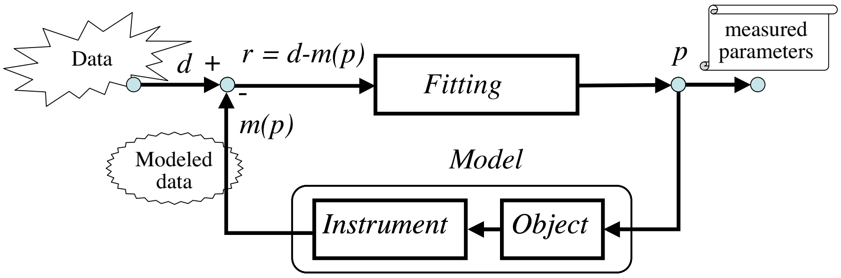

Fig. 1. Structure of the Model Fitting Software

When fitting, the astronomer wants to estimate the 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.

More precisely, the aim is to find the parameters which 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 (Haniff, 1991).

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 the main difficulty of the minimization as we will see in section 3. 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 (see section 3).

The next section describes the software used during the practice session and gives the relevant information to follow the example of a fit detailed in section 3.

2 LITpro: a Model Fitting software

2.1 Context

For this tutorial, we use LITpro (Lyons Interferometric Tool prototype), a new model fitting software under development within JMMC. The purpose of this section is to present the main components of fitting softwares (Fig. 1), using LITpro as an example. We also outline some details on the abilities of LITpro to better understand the fit described in the next section.

The main objective of JMMC is to help astronomers for their interferometric observations, mainly made on the VLTI instruments. Activities concern the search of calibrators, image reconstruction, or tools for preparing observations and analyzing data. Since the beginning of 2004, a group has been in charge of developing a model fitting software for an extended community.

LITpro, written in `Yorick`[2], is a work in progress. It is the successor of LIT, a first prototype, which was succesfully tested on real data on autumn 2005. Among other things, the new architecture greatly simplifies the task of designing and interfacing the models.

A Graphic User Interface (GUI) in JAVA is under development since November 2006. For now, only command line is used.

2.2 Reading of 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 stellar

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

Interferometric 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). 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).

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.

2.3 Modeling of data

2.3.1 Modeling of the object

We need a flexible way of modelling different types of objects. We can first rely on a library of functions which can be combined to describe a complex object. LITpro provides a set of elementary functions including circular (disk, ring, circle, Gaussian), elongated (ellipse, 2G-gaussian, 2D-ring) and some center-to-limb darkening functions.

But this library cannot cover all the needs and users generally want to implement more specific models with their own parameters (e.g. Perrin et al., 2004; Nardetto et al., 2006; Millour, 2006). LITpro allows to easily implement 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 designed for fitting simultaneously interferometric data and spectral energy distribution (or other data), although only monochromatic data are considered in this tutorial.

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.

2.3.2 Modeling of the instrument

The data depend on the object, but also on the instrument used for the observation: object and interferometric measurements are connected and the model fitting software should model this relationship.

The currently available data processing routines generally consider an idealized interferometer which gives measurements on localized points in the (u,v) map. However, on monomode interferometers, like AMBER, the finite diameter of the telescopes induces an apodization of the field of view (Longueteau et al., 2002). The effect of this "antenna diagram" is not easy to model when taking into account partial correction with adaptive optics, and it depends on the kind of measurements done by the interferometer. On multimode interferometers, the Michelson arrangement produces a frequency conversion (Tallon and Tallon-Bosc, 1992) that has complex effects with a finite spectral window (Tallon and Tallon-Bosc, 1993). Furthermore, effects as simple as the sliding of the projected baselines during the exposure time or field apodization from a finite spectral window, could be significant. We plan to implement and study these effects with LITpro.

2.4 Fitting engine

Finally, 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.).

Several fitting algorithms exist, e.g. Simulated Annealing (Ingber, 1989), Pseudo Clean or Levenberg-Marquardt algorithm (Dennis and Schnabel, 1983). LIT-pro uses the Levenberg-Marquardt algorithm, improved by the implementation of trust regions (Moré and 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 are 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 paramaters, covariance and correlation matrices. The latter measures cross dependencies of the parameters and may outline degenerated parameters and limitations of the model.

2.5 Hints for using LITpro

The user sets up a fitting session by editing the so-called *modeling file*, like a form, which will be loaded before launching the fit. In this file, he mainly selects the data files, defines the model, and initializes the parameters.

Various fields can be filled for the parameters, like initial values, bounds, scaling, units. Each parameter can also be fixed (i.e. not fitted). The settings of the parameters can also be changed on-line before or after a fit. On the other hand, any change of the definition of the model can only be done by tuning the *modeling file*.

In the *modeling file*, it is possible to discard any type of data (VIS2, VISamp, VISphi, T3amp or T3phi): it is sometimes useful for difficult cases to start working with a subset of the data. If the model is composite, a same parameter can be used into different elementary modeling functions. The user may also fit simultaneously different models on different data (like for example an object and its calibrators, on raw data, with a proper modeling of the transfer function of the interferometer [3]). After the setup, the *modeling file* is loaded and information is stored into an opaque structure named `world`.

To shorten the learning curve of the software during a practice session, we have prepared specific simple commands, as well as several modeling files for each set of data, so that we can focus our attention on some subtleties of model fitting, with a limited knowledge of this particular software.

[3] This strategy would avoid the correlations introduced in the data by the calibration (Perrin, 2003)

3 Example of a fit with LITpro

We present here an example of a fit with LITpro on simulated data from the interferometry imaging beauty contest[4] (Lawson et al., 2004). These data correspond to the target "Obj1" of the tutorial.

After the installation of the software, the first step is to make the directory of the file `LITpro.i` the current directory (shell command `cd`), then to launch `Yorick`. The first two commands in Yorick are used only once:

```
> #include "LITpro.i"
> lp_practice
```

The first line loads and configures LITpro source code. The second line has been introduced in a later release of the software to recover the functions made available at the time of this practice session. It configures LITpro for practice by defining paths for *modeling files* and data, and by renaming essential functions to make them easier to use. These functions are listed on screen with a short description. Note that the usage of every function can be obtained using the Yorick command "help", e.g. "`help, load_model`". With miscellenea bug corrections, this later release also improves plots and displays more information on the fit (e.g. correlation matrices).

Although designing a model in LITpro is as simple as filling a form, three models were already prepared for target "Obj1": a uniform disk, a binary star, and a binary star with resolved components (two uniform disks). Let us experiment with each of them successively. Note that we concisely present the fit and let the user explore further using all the provided tools.

3.1 Fit with a uniform disk

```
> world = load_model(LIT_MODELS+"Obj1_uniform_disk.i")
```

This command gets the user ready for the fit after loading the definition of the model and the corresponding data, and storing them into an opaque structure, called *world*. The variable `LIT_MODELS` contains the name of the directory, defined by `lp_practice`, where we stored the predefined *modeling files* for the practice session. The uv-coverage of the data can be plotted with command `plot_uvcoverage` (Fig. 2). The user can look at the parameters of the model (the predefined ones in this case) this way:

```
> show_parameters, world
```

[4] Details are available at http://olbin.jpl.nasa.gov/iau/2004/beauty.html

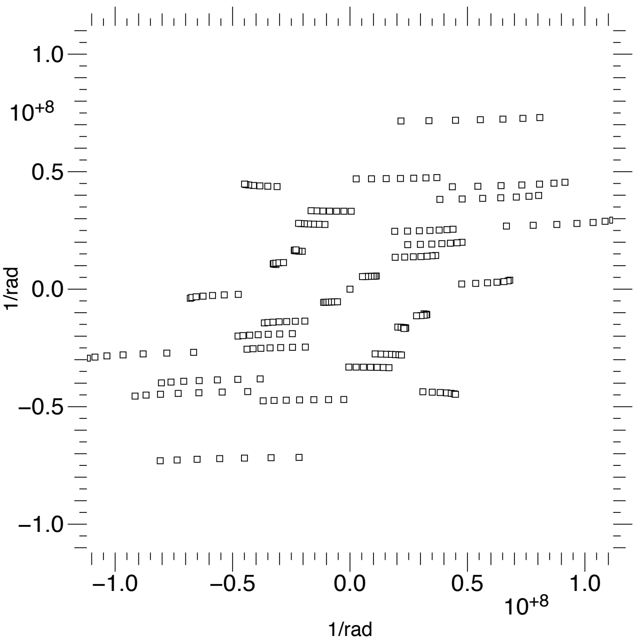

Fig. 2. uv-coverage for target “Obj1”

```
     name       value  prev_val     vmin     vmax   scale  fixed  units
     "i":         0.5       0.5        0        1    AUTO      0
     "r":           0         0        0       20    AUTO      0  mas
     "x":           0         0      -40       40    AUTO      0  mas
     "y":           0         0      -40       40    AUTO      0  mas
```

Here the initial value for the intensity i of the object is set to 0.5, while radius r and coordinates x and y are set to zero. Bounds have been set on each parameter as shown in columns `vmin` and `vmax`. They define an explored field of view of 80 mas in size. Fit is launched by command `go_fit` as follows:

```
> go_fit, world
Number of iterations  40   (Max Number of iterations 200)

Final values for fitted parameters and standard deviation:
     i =          1 +/-  0.0638877
     r =    0.69923 +/- 0.00124607
     x =    13.2999 +/- 1.76058e+07
     y =   -10.8153 +/- 1.69925e+07

Initial Chi2 = 214844    -  Final Chi2 = 65879.6
Initial reduced Chi2 = 891.468   -  Final reduced Chi2 = 273.359

--- Covariance matrix ---
         i         r         x         y
  i    0.0041  -1.4e-13   0.00016   0.00028
  r  -1.4e-13   1.6e-06  -1.5e+02  -5.5e+02
  x   0.00016  -1.5e+02   3.1e+14   1.6e+14
```

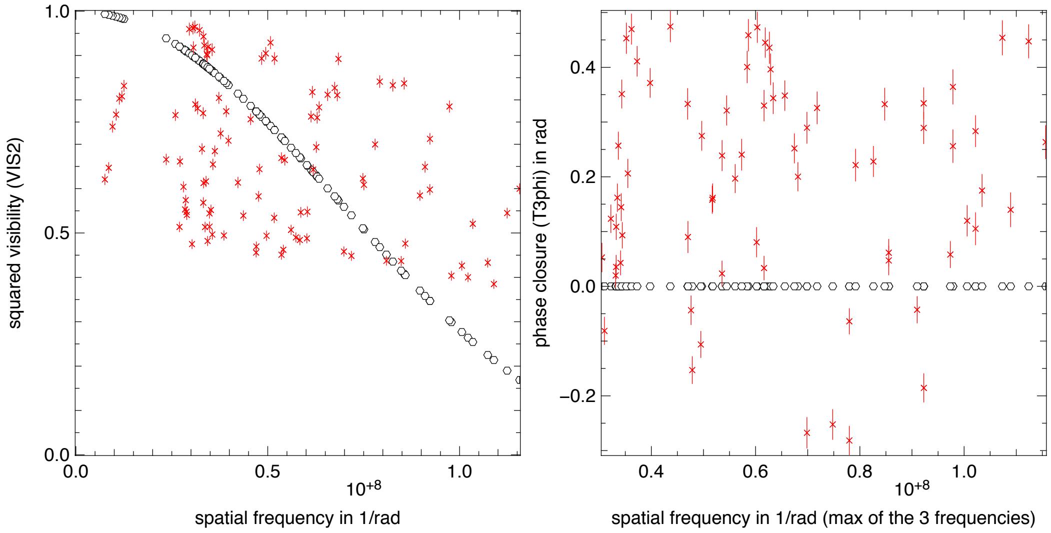

Fig. 3. Squared visibilities (left) and phases of the bispectrum (right) versus spatial frequency. Data (crosses) are plotted with 1-sigma error bars and the model of data (circles) is obtained with a uniform disk of 1.4 mas in diameter. Reduced $\chi^2 \approx 273$ is quite poor.

```
  y   0.00028  -5.5e+02   1.6e+14   2.9e+14

--- Correlation matrix ---
          i          r         x         y
  i         1  -1.7e-09  1.4e-10  2.6e-10
  r  -1.7e-09         1  -0.0068   -0.026
  x   1.4e-10   -0.0068        1     0.55
  y   2.6e-10    -0.026     0.55        1
```

Each type of data can be plotted successively:

```
> plot_radial, world, "VIS2", newwin=0
> plot_radial, world, "VISamp", newwin=1
WARNING: Not found any data with type "visamp"
> plot_radial, world, "T3phi", newwin=1
```

The warning reminds the user of the missing VISamp in this data set, as already stated by the command `load_model`. The plot of squared visibilities (Fig. 3) clearly shows that our target is far from a uniform disk of diameter 1.4 mas as derived from the fit. This is confirmed by the very high final reduced chi-square $\chi^2 \approx 273$. The phase of the bispectrum is non-zero (Fig. 3), highlighting some asymmetries in “Obj1”, which cannot be explained by a uniform disk as shown by the plot.

The result of the fit shows huge values for the standard deviations of parameters x and y, denoting that the position of the object is not constrained by the data. Indeed, bispectrum data (as well as squared visibilities) are not sensitive to any translation of the object in the field of view, and we would need complex visibilities to overcome the degeneracy of these parameters. The overall position of “Obj1” should thus be fixed at an arbitrary value.

We can see that the intensity i has increased from its initial value, 0.5, to unity. This behaviour cannot be driven by the data since, in the same way as parameters x and y, reduced interferometric data do not depend on the brightness of the object. In this simple model, we could have set i to a fixed value. But a more general means has been used here. Indeed, fixing a single *intensity* parameter is not possible when the total energy density of a more elaborated model depends on a complex combination of several heterogeneous parameters. In such a case, the standard deviations of the so linked parameters cannot be determined. So we have selected in the *modeling file*, an optional constrain on the total energy density. This selected option adds a term to the expression of χ^2 (Eq. 1) as follows:

$$\chi^2\prime = \chi^2 + N_d \left(\frac{\sum_j \Delta\lambda_j \ m_{j0}}{\sum_j \Delta\lambda_j} - 1 \right)^2 , \tag{4}$$

where the sums are over all the bandwidths $\Delta\lambda_j$, N_d is the number of data, and m_{j0} is the value of the Fourier transform of the object at the zero spatial frequency (which is equal to the total flux) in each bandwidth. The added term has been written so that it holds in case of spectrally resolved data. This option should not be used when no degeneracy appears with the total energy density, because it would enforce wrong priors and bias the estimation of the standard deviations.

3.2 Fit with a binary

In a similar way as before, fitting with the binary star as a model (two Dirac functions) can be done as follows:

```
> world = load_model(LIT_MODELS+"Obj1_binary_punct.i")
> set_parameter, world, "x1", "y1", fixed=1
> show_parameters, world
     name      value  prev_val    vmin    vmax   scale  fixed  units
    "i1":        0.5       0.5       0       1    AUTO      0
    "i2":        0.5       0.5       0       1    AUTO      0
    "x1":          0         0     -40      40    AUTO      1  mas
    "x2":          0         0     -40      40    AUTO      0  mas
    "y1":          0         0     -40      40    AUTO      1  mas
    "y2":          0         0     -40      40    AUTO      0  mas
> go_fit, world
Number of iterations  120   (Max Number of iterations 200)
```

```
Final values for fitted parameters and standard deviation:
     i1 =   0.847126 +/-  0.0541248
     i2 =   0.152842 +/- 0.00977747
     x2 =    -1.9557 +/- 0.00442699
     y2 =   0.987198 +/-   0.005968

Initial Chi2 = 214782    -  Final Chi2 = 55692.7
Initial reduced Chi2 = 891.213   -  Final reduced Chi2 = 231.09

--- Covariance matrix ---
         i1        i2        x2        y2
  i1    0.0029   0.00053  -6.3e-07   4.8e-07
  i2   0.00053   9.6e-05   6.3e-07  -4.8e-07
  x2  -6.3e-07   6.3e-07     2e-05  -1.4e-05
  y2   4.8e-07  -4.8e-07  -1.4e-05   3.6e-05

--- Correlation matrix ---
        i1       i2       x2       y2
  i1        1        1  -0.0026   0.0015
  i2        1        1    0.014  -0.0083
  x2  -0.0026    0.014        1    -0.53
  y2   0.0015  -0.0083    -0.53        1

> plot_radial, world, "VIS2", newwin=2
> plot_radial, world, "T3Phi", newwin=3
```

The indices 1 and 2 identify the parameters of both components. Position (x_1, y_1) of the first one has been fixed (i.e. not fitted): keeping degenerated parameters during the fit is then avoided. The correlation matrix shows that i_1 and i_2 are fully correlated, as expected, since the total energy is still constrained to unity (Eq. 4).

The plots of VIS2 or T3phi (Fig. 4) show that the ability of this model seems more appropriate to fit our data. Nevertheless the final reduced chi-square $\chi^2 \approx 231$ is only marginally improved. At low frequencies, the model for VIS2 is systematically higher than the data. Furthermore, the model for T3phi shows some symmetry at these scales. This denotes that the model is not enough resolved compared to the data. When starting from separation 0, the fit could have been trapped in a close local minimum at coordinates (-2, 1) mas. At high frequencies, model for VIS2 has values close to one whereas the real visibilities are systematically lower: we need to decrease the modeled visibilities, for instance by giving some extension to the components.

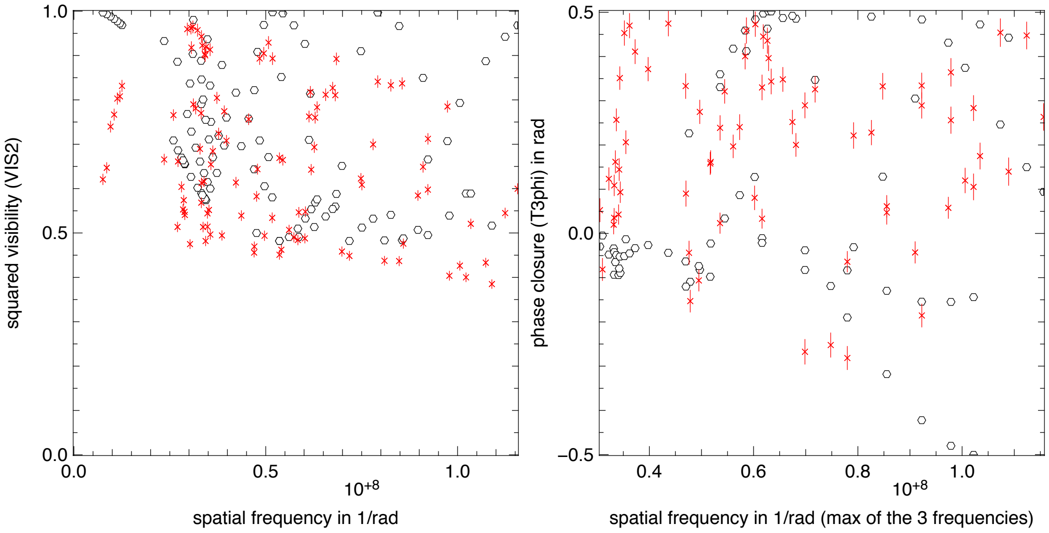

Fig. 4. Squared visibilities (left) and phases of the bispectrum (right) versus the spatial frequency. The data (crosses) seem to be better fitted with a model of data (crosses) obtained with a binary. Reduced $\chi^2 \approx 231$ is slightly improved but still very high.

3.3 Exploring local minima

Before attempting to fit the last model, let us look at the problem of local minima. With local minima, 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 means to solve this problem, other than to try various starting values. Here, we proposed a simple tool, "chi2_map", for gridding the initial values of parameters x_2 and y_2 over a full map while the first component is fixed at the center: for each initial (x_2, y_2), all the parameters (x_2, y_2, i_1, i_2) are fitted and the final χ^2 is plotted at coordinates of initial (x_2, y_2). This map is displayed on left panel of Fig. 5. It shows areas of constant χ^2: starting the fit from any point in one of these areas, converges to the same solution and yields the same χ^2 as the one at the local minimum (hence the constant values on theses areas). The deepest area identifies the best local minimum.

This tool is now disused owing to the high computation load for making a full map. It is now replaced by "sniffer_map", where parameters x_2 and y_2 are fixed. For each position (x_2, y_2) shown in the map, the components are fixed at positions (0,0) and (x_2, y_2) respectively. Only i_1, i_2 are fitted (here still with normalized total energy). This map is displayed on the right panel of Fig. 5. Here again, the map shows many local minima. The smallest χ^2 in this map gives the best place where to put the secondary component, and hence the best starting value for (x_2, y_2). The deepest one is at the position $(-6.5, 20)$.

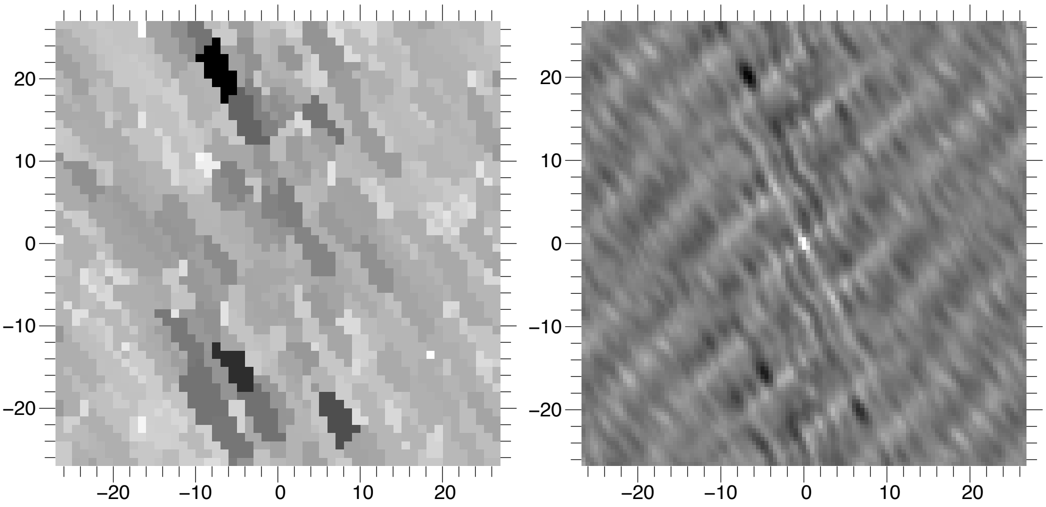

Fig. 5. χ^2 maps provided by the disused “chi2_map” (left) and the newer “sniffer_map” (right). One component is fixed at the center, while the other component is initialized before the fit (left) or kept fixed during the fit (right) at the corresponding position in the map. Relative intensities are fitted in any case. The map on the right shows a minimum at position $(-6.5, 20)$. We can notice another minimum at the symmetric position $(6.5, -20)$, where χ^2 is however higher.

This initial position can now be set using command `set_parameter` after loading model `"Obj1_binary_punct.i"`. After 121 iterations, the final reduced chi-square is $\chi^2 \approx 11.6$, which is a significant improvement from the former value. `go_fit` yields the following values and standard deviations for the fitted parameters:

```
Final values for fitted parameters and standard deviation:
     i1 =   0.833749 +/-  0.0532687
     i2 =   0.166242 +/-  0.0106306
     x2 =   -6.68312 +/- 0.00387742
     y2 =    20.0899 +/-  0.0055778
```

3.4 Fit with a binary with resolved components

As we have already seen from Fig. 4, the VIS2 data are damped towards high frequencies, in contrast to our model of binary. The Fourier transform of the model should be multiplied by a ”weighting envelope” in the uv map. This corresponds to convolving the object with some function in the image plane to somehow enlarge the Dirac functions modeling the components. The last model provided for “Obj1” was built with a set a two uniform disks. From the previous model, two new parameters are then added, r_1 and r_2, standing for the radius of each component, with initial values set to zero and bounded to 20 mas. From what was learnt up to now, we can easily conduct the fit this way:

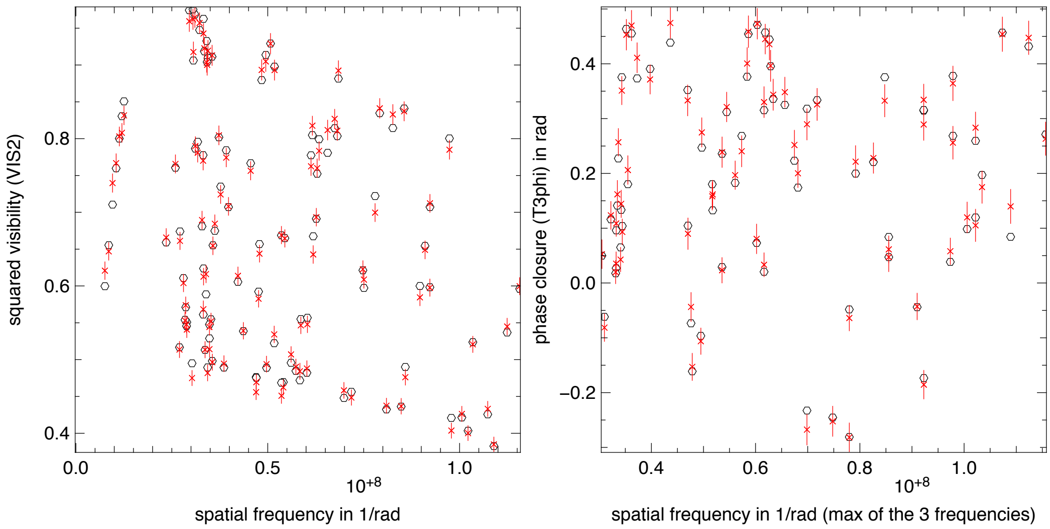

Fig. 6. Squared visibilities (left) and phases of the bispectrum (right) versus the spatial frequency, for the data (crosses) and the model from a binary of two uniform disks (circles). Damping the model with uniform disks has dramatically improved the value of the reduced χ^2 down to 0.76.

```
> world = load_model(LIT_MODELS+"Obj1_binary_disk.i")
> set_parameter, world, "x1", "y1", fixed=1
> set_parameter, world, "x2", value=-6.5
> set_parameter, world, "y2", value=20
> go_fit, world
```

The final reduced chi-square is $\chi^2 \approx 0.76$, which might be too much perfect, probably because of the use of simulated data. The plots of VIS2 or T3phi illustrate this satisfying result (Fig. 6). `go_fit` yields the following values and standard deviations for the fitted parameters:

```
Final values for fitted parameters and standard deviation:
    i1 =   0.853174 +/-  0.0545116
    i2 =   0.146826 +/- 0.00940554
    x2 =    -6.7288 +/- 0.00487464
    y2 =     20.135 +/- 0.00673896
    r1 =   0.304875 +/- 0.00320666
    r2 =   0.291736 +/-  0.0236471
```

This result matches perfectly the values of the binary used to compute the simulated data for the interferometry imaging beauty contest: 2 components of diameter 0.6 mas with flux ratio of 5.75 (which corresponds to $i_1/i_2 = 0.85185/0.14815$) and separation $(\rho, \theta) = (21.2\text{ mas}, 341.6°)$, which corresponds to $(x_2, y_2) = (-6.69176, 20.1162)$ mas (obtained with the function `rho_PA_to_xy`).

An image of the solution can be plotted with `plot_image` (Fig. 7):

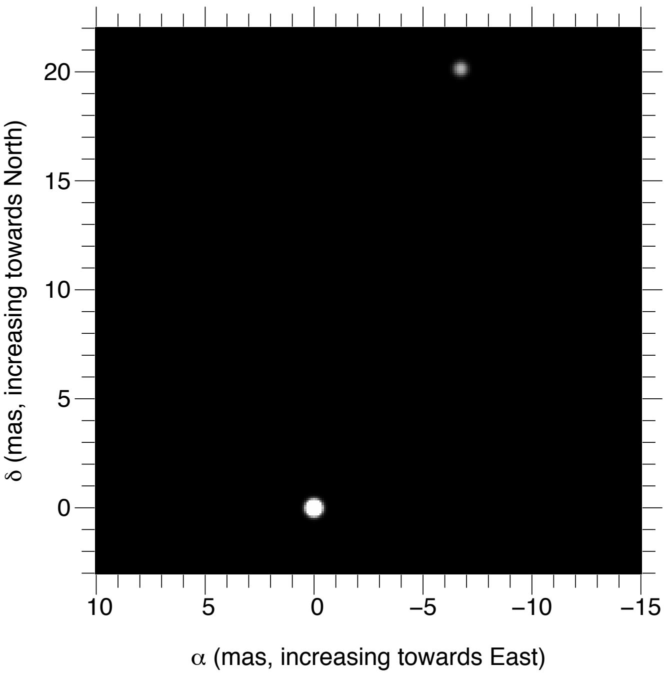

Fig. 7. Image of the solution, given by the last fit

```
> plot_image, world, xmin=-15, xmax=10, ymin=-3, ymax=22, pixscale=0.1;
> palette, "stern.gp"
```

Since modeling functions in LITpro are only requested to compute the Fourier transform of the object, `plot_image` derives its map using an FFT. This method allows the object map, understood as a "density" function, to be correctly interpolated, with a maximum resolution consistent with the chosen pixel size (keyword `pixscale`).

3.5 Deeper in local minima

We have learnt that "Obj1" data do not allow the determination of the central position of the object in the field of view. So we could be surprised by the asymmetries appearing in the χ^2 maps of Fig. 5. Indeed, if we consider a point (x, y) in the map, we know that the symmetrical point $(-x, -y)$ should have exactly the same χ^2 by swapping the indices of the two components: this situation corresponds to the same object, only translated by vector $(-x, -y)$, since the first component is still fixed at central position.

But for each point in the map, the fit begins with $(i_1, i_2) = (1, 0)$ as an initial guess. At the symmetric point, the intensities of the starting point should be swapped in order to get exactly the same situation and to end with the translated object. Without this swap, we do not start from the same initial guess. By reaching a different χ^2, we thus know that local minima exist along the intensity axis, even if we only fit the intensities.

This can be experimented by starting a fit from different symmetrical positions with the modeling file “`Obj1_binary_disk.i`”. For instance, starting from $(x_2, y_2, i_1, i_2) = (-6.5, 20, 1, 0)$ yields the best fit we obtained previously. Starting from the symmetric position $(x_2, y_2, i_1, i_2) = (6.5, -20, 1, 0)$ yields a different final reduced chi-square $\chi^2 \approx 89$. In this local minimum, the intensities of the components are almost unchanged (not swapped), as well as the position (x_2, y_2) (Fig. 5). Swapping now i_1 and i_2 in the starting conditions yields exactly the same reduced chi-square as the best fit, with swapped indices of the components. This solution is the same of the first one, only translated.

4 Conclusion

Through the example of a fit made with LITpro, we tried to show that model fitting is not so easy. It is helpful 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. For instance, considering that the squared visibilities data take values between 0.5 and 1 (Fig. 4), we can easily compute by hand an estimate of the intensities of the components of the binary: $(i_1, i_2) \approx (0.85, 0.15)$, and notice that these values are quite stable in the results of all the fits.

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. A classical tool here is to analyze cuts in the chi-square space. Another means has been introduced with the `sniffer_map` function, where a fit on a subset of the parameters is applied at each point in the map. The maps of chi-square allow the localization of the best local minimum.

The example of the fit was applied on simulated data. During the practice session, the students could undertake another fit on real data (“Obj2”) and apply a similar approach. For more practice, the last revision of LITpro and other data will be made available in a next future. We indeed continue to improve the capabilities of this software, in order to provide a first public release in 2007, with a powerful GUI.

Acknowledgements

The software presented in this article has been implemented in Yorick, a free data processing language written by D. Munro.

References

Dennis, J. E., Schnabel, R. B., 1983. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, Englewood Cliffs, NJ.

Haniff, C. A., 1991. Least-squares Fourier phase estimation from the modulo 2π bispectrum phase. J. Opt. Soc. Am. A 8 (3232), 134–140.

Ingber, L., 1989. Very fast simulated re-annealing. Math. Comput. Modelling 12 (8), 967–973.

Labeyrie, A., 1996. Resolved imaging of extra-solar planets with future 10-100km optical interferometric arrays. Astron. Astrophys. Supplement 118, 517–524.

Lawson, P. R., Cotton, W. D., Hummel, C. A., Monnier, J. D., Zhao, M., Young, J. S., Thorsteinsson, H., Meimon, S. C., Mugnier, L., Le Besnerais, G., Thiébaut, E., Tuthill, P. G., 2004. An interferometry imaging beauty contest. In: Traub, W. A. (Ed.), New Frontiers in Stellar Interferometry. Vol. 5491 of SPIE Conference. Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, p. 886.

Longueteau, E., Delage, L., Reynaud, F., 2002. Influence of the antenna diagram on a stellar interferometer that is suffering from telescope-pointing errors. Appl. Opt. 41 (28), 5835–5844.

Millour, F., 2006. Interférométrie différentielle avec AMBER. Ph.D. thesis, Grenoble University, France.

Moré, J. J., Sorensen, D. C., 1983. Computing a trust region step. SIAM J. Sci. Stat. Comput. 4 (3), 553–572.

Nardetto, N., Fokin, A., Mourard, D., Mathias, P., 2006. Probing the dynamical structure of delta Cephei atmosphere. Astron. Astrophys. 454 (1), 327–332.

Pauls, T. A., Young, J. S., Cotton, W. D., Monnier, J. D., 2005. Data exchange standard for optical (visible/IR) interferometry. Publ. Astron. Soc. Pac. 117 (837), 1255–1262.

Perrin, G., 2003. The calibration of interferometric visibilities obtained with single-mode optical interferometers - Computation of error bars and correlations. Astron. Astrophys. 400 (3), 1173–1181.

Perrin, G., Ridgway, S. T., Mennesson, B., Cotton, W. D., Woillez, J., Verhoelst, T., Schuller, P., Coudé du Foresto, V., Traub, W. A., Millan-Gabet, R., Lacasse, M. G., 2004. Unveiling Mira stars behind the molecules. Confirmation of the molecular layer model with narrow band near-infrared interferometry. Astron. Astrophys. 426 (1), 279–296.

Tallon, M., Tallon-Bosc, I., 1992. The object-image relationship in Michelson stellar interferometry. Astron. Astrophys. 253 (2), 641–645.

Tallon, M., Tallon-Bosc, I., 1993. Beam combination for wide field imaging. In: Robertson, J. G., Tango, W. J. (Eds.), Very High Angular Resolution Imaging. Vol. 158 of IAU Symp. Kluwer, Dordrecht, The Netherlands, pp. 83–90.

Vakili, F., Aristidi, E., Abe, L., Lopez, B., 2004. Interferometric remapped array nulling. Astron. Astrophys. 421 (1), 147–156.