Model fitting:
tutorial and introduction to the practice session

EuroSummer School

Observation and data reduction with the Very Large Telescope Interferometer

Goutelas, France
June 4-16, 2006

Isabelle Tallon-Bosc & Michel Tallon

CRAL / Observatoire de Lyon

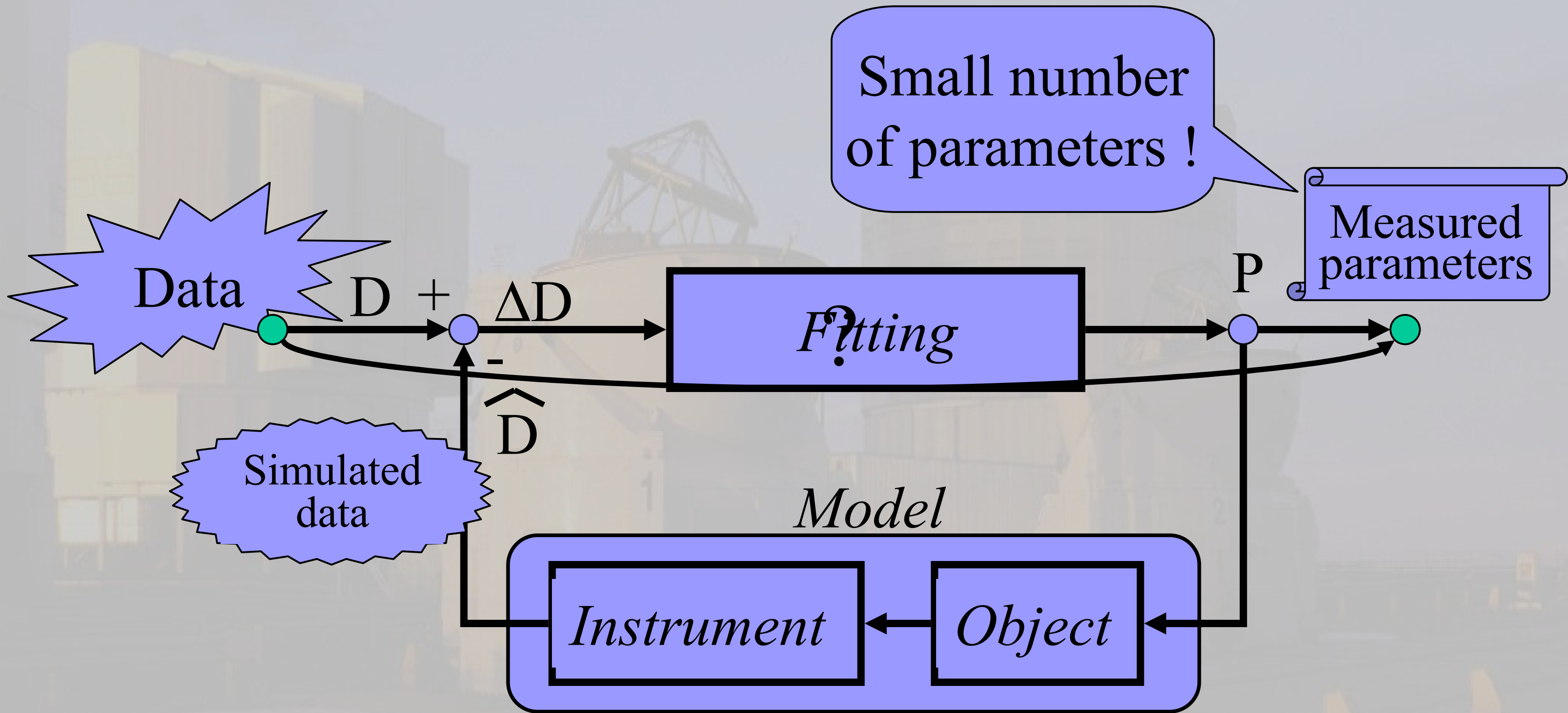
14 June 2006

Why to fit a model?

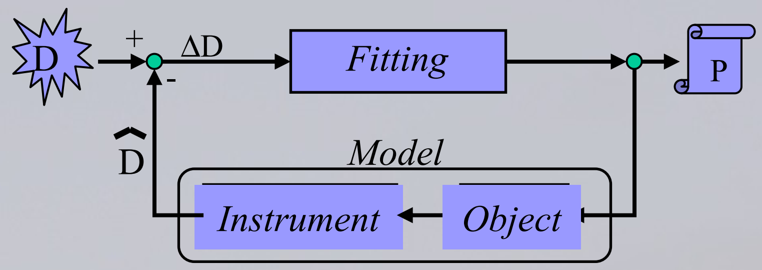
- Extract measurements from the data

  --> parameters, geometrical or physical, of the model of the object
- Measurements possible with an image reconstruction
  - easier than with a model fitting if geometrical parameters
  - more difficult if physical parameters
- Model fitting and Image Reconstruction are complementary
  - e.g. starting from a reconstructed image allows us to build up a better model for the object
  - Difficulties in common (estimation of error bars, local minima)
  - Both approaches are improving (gain of robustness, of speed of convergence)

Principle of the model fitting



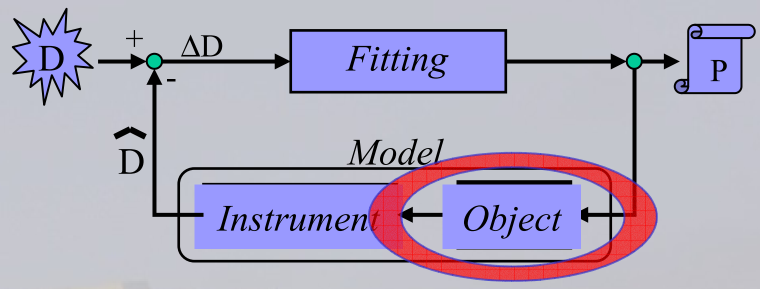
Requirements for a model fitting software



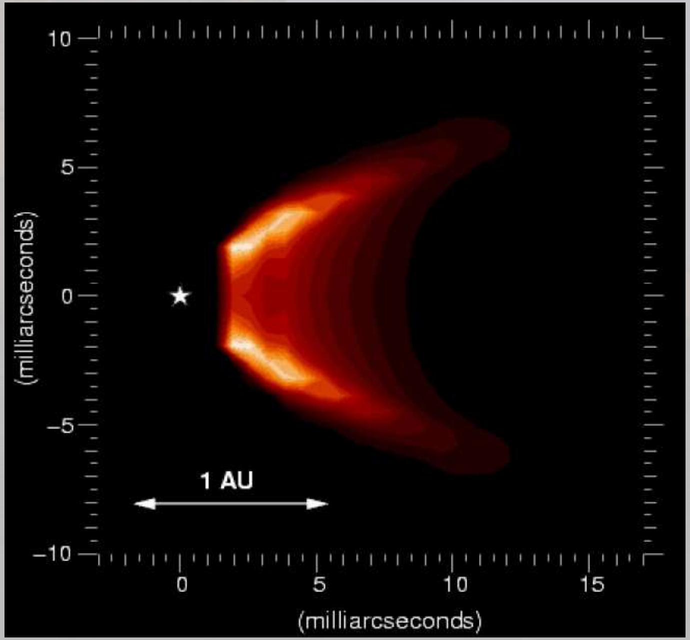
The software, for beeing useful and competitive, should:

- be usable by a large community,
- benefit of different experiences, and common knowledge
- provide models of objects:
  - Easy to tailor, more or less complicated, …
  - allowing to deal with astrophysical numerical models
- take into account instrumental effects (≠ ideal instrument)
- fit the parameters of the model with reliability and robustness

Needs for a MF software: Modelized objects



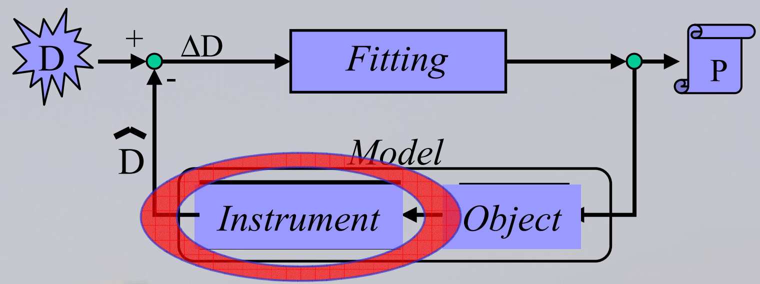
- « Geometrical » objects
  - Library of basic elements (= blocks): disk, ring, gaussian, center-to-limb darkening, etc.
  - Possibility of building blocks
- Possibility to drive an astrophysical model



Model of a micro-jet emitted by a young star (P. Garcia et al.)

- Geometrical distorsion of the models
  - rotation, scaling, elongation, …

Needs for a MF software: Instrumental effects

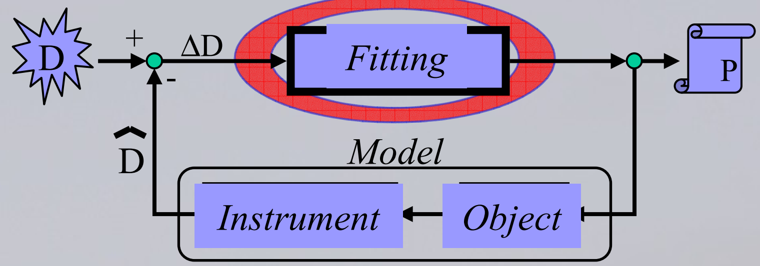


Examples of instrumental effects we should modelize:

- Modal filtering (monomode fibers)
- Halo in partial corrected images with AO
- Finite spectral bandwidth
- Displacement of baselines during data acquisition
- […]

(-> FOV limitation)

Needs for a MF software: a fitting « engine »



- The suitable fitting engine deals with:
  - a non linear and non convex inverse problem
  - several local minima => global optimization
  - a small number of parameters,
  - bounded parameters (e.g. positivity constraint)
- It calculates the gradients from finite differences
- It provides analysis of the results (covariance matrix,…)

Possible engines:

- Levenberg-Marquardt
- Simulated Annealing

with use of the trust regions

Model fitting software: an example

- Context: JMMC / JRA4
  - Objective: efficient public tools for interferometric data processing
  - Model fitting, Image reconstruction, Calibration
  - Model fitting group exists since 2004
- Model fitting : two developpements
  - MCS version, professional, simplified
  - Prototype in high level language (yorick) for R&D
    - First version (C. Bechet, E. Thiebaut): LIT
    - Second version, more easily enrichable: LITpro

The one you will use !

since the MCS not yet available

LITpro: why?

- Complete rewriting of the previous LIT software:
  - using LIT allowed us to have practice
  - easier implementation of users models

  ==> design of a new architecture

- LITpro is a « young » algorithm

  ⇒ Bugs remain to be discovered …

  may be by you, sorry!

LITpro: how it proceeds

- Select the data
  - OI-DATA
- Select the model(s)
  - description with a script file
- Prepare the fit
  - loading a model configures the software to get it ready for fitting
- Tune the parameters (name, value, bounds, fixed/free)
- Fit
- Analyze the fit (plots,…)
- And try again until satisfaction

LITpro: reminder of the OI-DATA standard

- Based on fits file format standard
- Set of tables linked each other.
  - TARGET : informations on the object (name, coordinates, etc.)
  - ARRAY: description of the telescope configuration (optional)
  - WAVELENGTH : list of observed wavelengths and bandwidths
  - DATA Tables (measurements + error bars)
    - VIS2: squared visibilities
    - VIS: complex visibility (VISamp / VISphi)
    - T3: bispectrum (T3amp / T3phi). T3phi is phase closure

LITpro: constraints added to OI-DATA

- Clean the tables:
  - Remove duplicated targets,
  - Split the data tables in several data blocks (DB), one per target (need to link one model to a target).
- Group of spatial frequencies for reducing computation load
  - duplicates are removed,
  - all spatial frequencies put on the right side of the uv plane.
- ucoord, vcoord (# baselines) -> ufreq, vfreq (# baselines x # wavelengths).
  - all arrays have the same size.
- LITpro must accept other types of data (spectrum, etc.)

LITpro: definition of the model

- General rules:
  - Define a group GRP = set of data blocks (joined by a criterion, for ex., the same target TGT)
  - Link a model to a group
    - Model = linear combination of functions
    - User functions or basic functions of the library: circular (disk, circle, ring, gaussian), elongated (ellipse, 2D-gaussian, 2D-ring), different center-to-limb darkening functions
  - Global parameters for all the groups (you may use one same parameter for every group)
  - Fit all the groups together, if #GRP>1
  - Configuration written in a file --> define a model = load a « script » file

- Examples:
  - Ex1: 1 GRP, 1 TGT, 1 model
  - Ex2: 2 GRP (TGT + calibrator, sharing common parameters)