Introduction to model-fitting

Michel Tallon, Isabelle Tallon-Bosc, Eric Thiébaut

CRAL, Lyon France

Outline

1. Elements on model-fitting theory
   - understand a few concepts
   - understand the assumptions
   - getting useful hints for the practice

2. Digression on the correlations of data

3. LITpro software
   - short presentation of the main features

4. Adventure of model-fitting
   - examples and hints

Elements on model-fitting theory

- understand the concepts
- understand the assumptions
- getting hints useful for the practice

Model fitting actors

- What we have in hand
  - interferometric data (here OIFITS) ***and* uncertainties on data**

    d

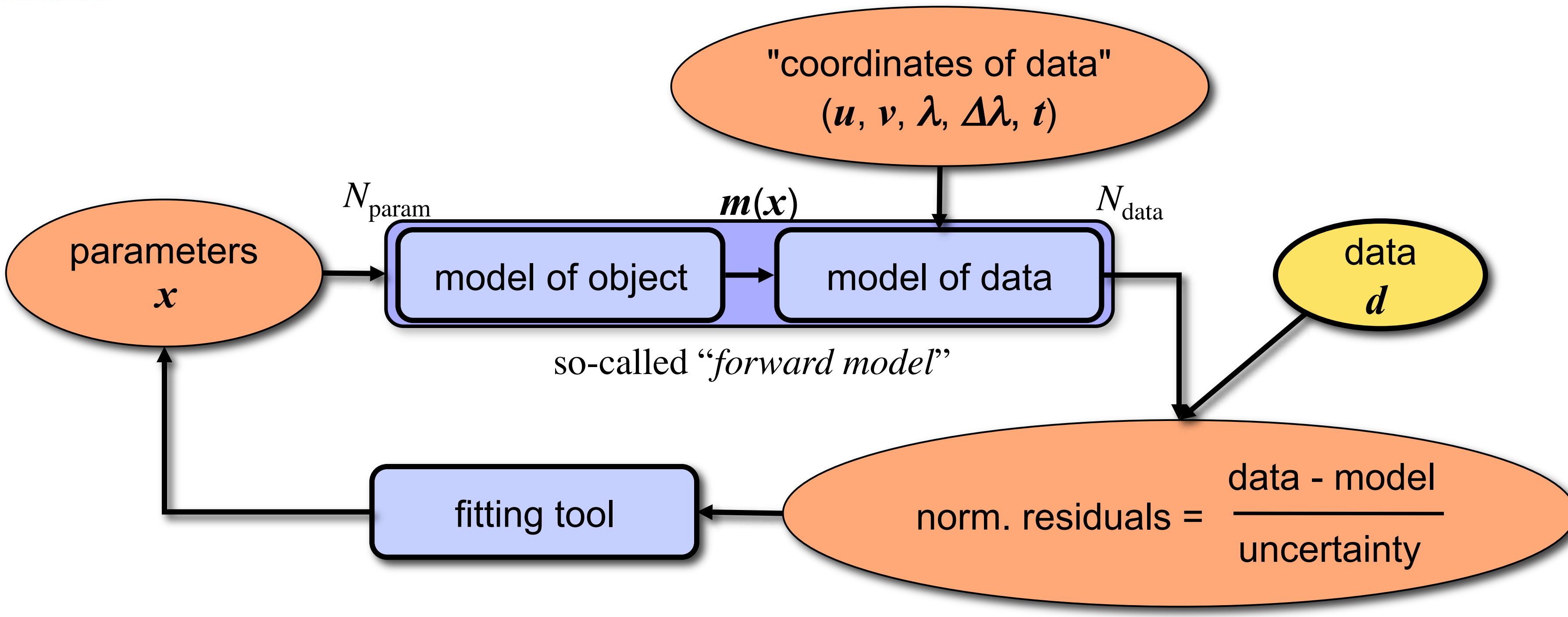
    - OI_VIS2 squared visibility amplitude
    - OI_T3 triple product (amplitude and phase)
    - OI_VIS complex visibility (amplitude and phase)
  - other data :
    - OI_FLUX calibrated or uncalibrated spectrum (OIFITS2)
    - absolute photometry, etc.
  - priors: all possible models of object

***m*(*x*)**

x

- What we want
  - identity the observed object with a model ***m(x)***
  - estimate object parameters ***x***, ***and* uncertainties on the parameters**
  - easy
- What we need
  - tools for model-fitting
  - know what we are doing (*no black magic !*)

Model fitting principle



Criterion for the *best* parameters

- *best* parameters maximize the probability of the data (*knowing the model*)

$$\boldsymbol{x}_{\text{best}} = \arg\max_{\boldsymbol{x}} \text{Pdf}\left(\boldsymbol{d} \,\middle|\, \boldsymbol{m}(\boldsymbol{x})\right)$$

- where

$\boldsymbol{d}$	data (random quantities, known statistics)
$\boldsymbol{x}$	parameters
$\boldsymbol{m}(\boldsymbol{x})$	model (of data): ~ *expected values of data*

- number of parameters < number of data
  - difference from image reconstruction => $\boldsymbol{x}_{\text{best}} = \arg\max_{\boldsymbol{x}} \text{Pdf}\left(\boldsymbol{m}(\boldsymbol{x}) \,\middle|\, \boldsymbol{d}\right)$
- priors are subjective
  - we have strong prior: the model of the object!
  - fundamental difference from image reconstruction

assumption: Gaussian statistics

- data have Gaussian statistics:

$$\mathrm{Pdf}\big(\boldsymbol{d} \,\big|\, \boldsymbol{m}(\boldsymbol{x})\big) = \frac{\exp\left(-\frac{1}{2}\,\boldsymbol{r}^{\mathrm{T}}.\mathbf{C}_{\boldsymbol{r}}^{\;-1}.\boldsymbol{r}\right)}{\sqrt{(2\pi)^{N_{\mathrm{data}}}\,\det\left(\mathbf{C}_{\boldsymbol{r}}\right)}}$$

- where:

$$\boldsymbol{r} = \boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})$$ residuals

$$\mathbf{C}_{\boldsymbol{r}} = \langle \boldsymbol{r}.\boldsymbol{r}^{\mathrm{T}} \rangle - \langle \boldsymbol{r} \rangle.\langle \boldsymbol{r} \rangle^{\mathrm{T}}$$ covariance matrix of residuals

- maximize Pdf ⇔ minimize argument of the Gaussian

$$\boldsymbol{x}_{\mathrm{best}} = \arg\min_{\boldsymbol{x}} \left[\boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})\right]^{\mathrm{T}}.\mathbf{C}_{\boldsymbol{r}}^{-1}.\left[\boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})\right]$$

assumption: data statistically independent

- $\mathbf{C_r}$ is a diagonal matrix:

$$\boldsymbol{x}_{\text{best}} = \arg\min_{\boldsymbol{x}} \left[\boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})\right]^{\mathrm{T}} . \mathbf{C}_{\boldsymbol{r}}^{-1} . \left[\boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})\right]$$

$$= \arg\min_{\boldsymbol{x}} \sum_{i=1}^{N_{\text{data}}} \left(\frac{d_i - m_i(\boldsymbol{x})}{\sigma_i}\right)^2$$

covariances expected…

- thus we need to minimize $\chi^2(\boldsymbol{x})$:

$$\chi^2(\boldsymbol{x}) = \sum_{i=1}^{N_{\text{data}}} \left(\frac{d_i - m_i(\boldsymbol{x})}{\sigma_i}\right)^2 = \sum_{i=1}^{N_{\text{data}}} \frac{r_i^2(\boldsymbol{x})}{\sigma_i^2} = \sum_{i=1}^{N_{\text{data}}} e_i^2(\boldsymbol{x})$$

a.k.a non-linear weighted least squares

where $e_i(\boldsymbol{x})$ normalized residual: random variable with standard normal distribution

=> χ^2 law

- Independency in real world ?
  - calibrator
  - normalization by incoherent flux

χ^2 law: definition

$$\chi^2(\boldsymbol{x}_{\text{best}}) = \sum_{i=1}^{N_{\text{data}}} e_i^2(\boldsymbol{x}_{\text{best}}) \quad \text{with} \quad e_i(\boldsymbol{x}) = \frac{d_i - m_i(\boldsymbol{x})}{\sigma_i}$$

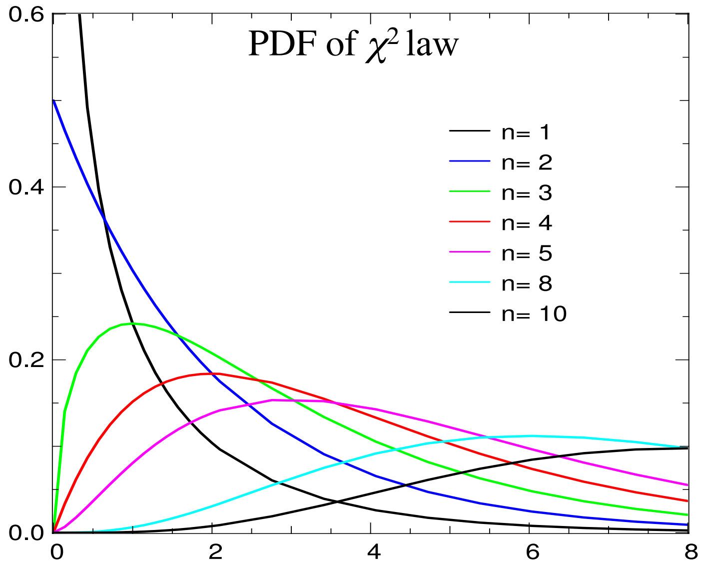
$e_i(\boldsymbol{x}_{\text{best}})$: standard normal distribution $\mathcal{N}(0,1)$

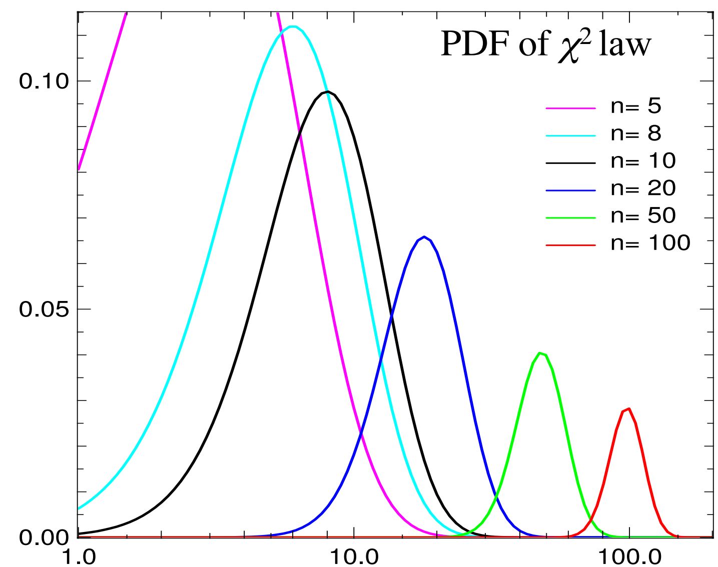
number of degrees of freedom: $N_{\text{free}} = N_{\text{data}} - N_{\text{param}}$

expected value: $\mathrm{E}\{\, \chi^2(\boldsymbol{x}_{\text{best}}) \,\} = N_{\text{free}}$

variance: $\mathrm{Var}\{\, \chi^2(\boldsymbol{x}_{\text{best}}) \,\} = 2\, N_{\text{free}}$

assuming model is good !





χ^2 law: reduced χ^2

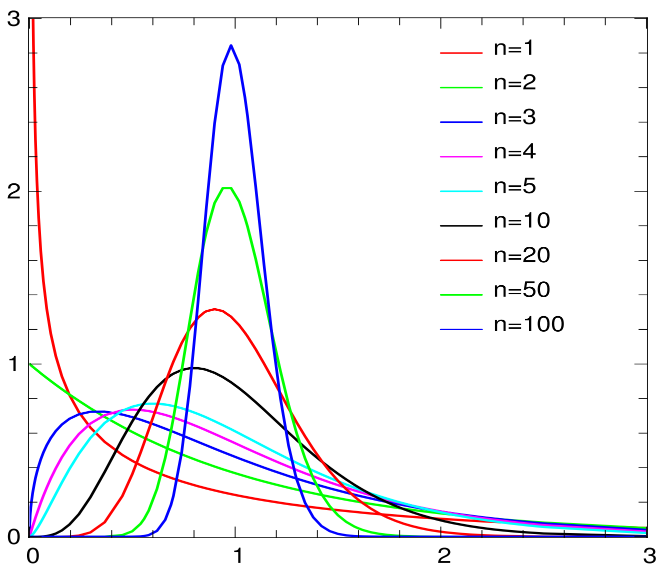
reduced χ^2 : $\quad \chi_r^2 \equiv \dfrac{\chi^2}{N_{\text{free}}}$

number of degrees of freedom: $N_{\text{free}} = N_{\text{data}} - N_{\text{param}}$

expected value: $E\{\chi_r^2(\boldsymbol{x}_{\text{best}})\} = 1$

variance: $\text{Var}\{\chi_r^2(\boldsymbol{x}_{\text{best}})\} = 2 \,/\, N_{\text{free}}$

assuming model is good !



- statistics is very sharp !
  - confidence level not very useful
- in practice, statistics cannot be used to accept or rule out a model
  - modeling errors may be high
  - noise level may be badly estimated
- can be used to compare two models:

$$\frac{\chi^2[\boldsymbol{m}_1]}{N_1} \longleftrightarrow \frac{\chi^2[\boldsymbol{m}_2]}{N_2}$$

keep in mind var. of χ^2

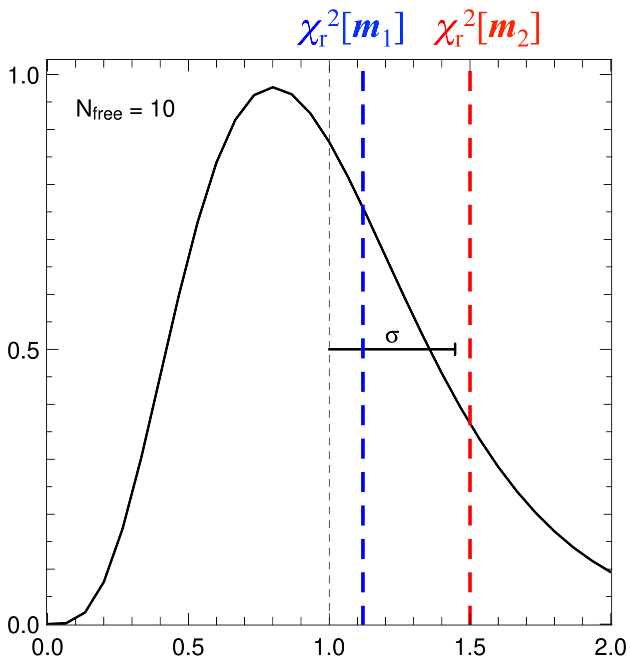
reduced χ^2 : model comparison

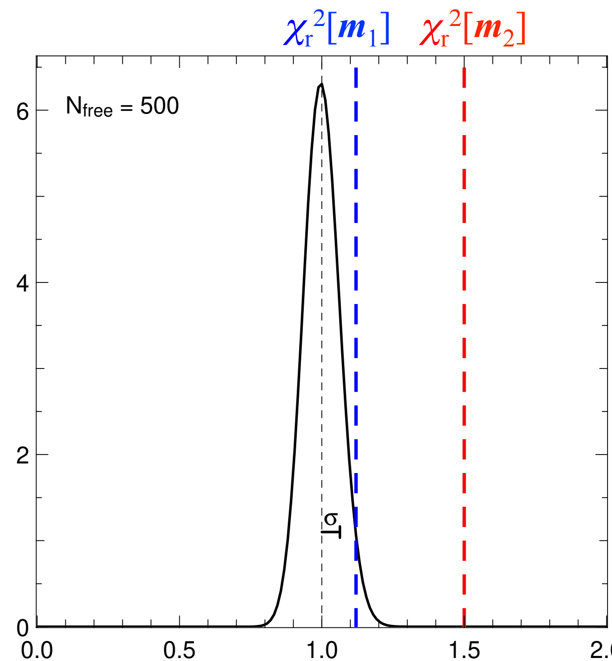
Compare χ_r^2 for different models, relatively to the standard deviation of the χ_r^2 distribution

$N_{\text{free}} = N_{\text{data}} - N_{\text{param}}$

$E\{\chi_r^2(\boldsymbol{x}_{\text{best}})\} = 1$

$\text{Var}\{\chi_r^2(\boldsymbol{x}_{\text{best}})\} = 2 / N_{\text{free}}$





Errors on fitted parameters / 1

- We have seen : $$\boldsymbol{x}_{\text{best}} = \arg\min_{\boldsymbol{x}} \left[\boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})\right]^{\mathrm{T}}.\mathrm{C}_{\boldsymbol{r}}^{-1}.\left[\boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})\right]$$
$$\boldsymbol{r} = \boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})$$
$$\mathbf{C}_{\boldsymbol{r}} = \langle \boldsymbol{r}.\boldsymbol{r}^{\mathrm{T}} \rangle - \langle \boldsymbol{r} \rangle.\langle \boldsymbol{r} \rangle^{\mathrm{T}} = \mathbf{C}_{\boldsymbol{d}}$$
=> $\mathbf{C}_{\boldsymbol{x}}$?

- The diagonal of $\mathbf{C}_{\boldsymbol{x}}$ gives the incertainties of the parameters $\boldsymbol{x}$.
- Off-diagonal terms gives the correlations between the parameters
  - e.g. parameters fully coupled => some features on the object cannot be determined.

Errors on fitted parameters / 2

- We have seen : $\boldsymbol{x}_{\text{best}} = \arg\min_{\boldsymbol{x}} \left[\boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})\right]^{\mathrm{T}} . \mathbf{C}_{\boldsymbol{r}}^{-1} . \left[\boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})\right]$

$$\boldsymbol{r} = \boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})$$

$$\mathbf{C}_{\boldsymbol{r}} = \langle \boldsymbol{r}.\boldsymbol{r}^{\mathrm{T}} \rangle - \langle \boldsymbol{r} \rangle . \langle \boldsymbol{r} \rangle^{\mathrm{T}} = \mathbf{C}_{\boldsymbol{d}}$$

=> $\mathbf{C}_{\boldsymbol{x}}$?

- If a linear model: $\boldsymbol{m}(\boldsymbol{x}) = \mathbf{H}.\boldsymbol{x}$ (parameters $\boldsymbol{x}$ and modeled data $\boldsymbol{m}(\boldsymbol{x})$)

$$\boldsymbol{x}_{\text{best}} = (\mathbf{H}^{\mathrm{T}}.\mathbf{C}_{\boldsymbol{r}}^{-1}.\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}.\mathbf{C}_{\boldsymbol{r}}^{-1}.\boldsymbol{d}$$ (weighted least squares)

=> $\mathbf{C}_{\boldsymbol{x}_{\text{best}}} = (\mathbf{H}^{\mathrm{T}}.\mathbf{C}_{\boldsymbol{r}}^{-1}.\mathbf{H})^{-1}$

- Correlation matrix: $\Gamma_{i,j} = \dfrac{C_{i,j}}{\sigma_i \, \sigma_j}$

Errors on fitted parameters / 2

- But the model $\boldsymbol{m}(\boldsymbol{x})$ is highly non-linear ! => linearisation…

$$\boldsymbol{m}(\boldsymbol{x}) \approx \boldsymbol{m}(\boldsymbol{x}_{\text{best}}) + \left[\frac{\partial \boldsymbol{m}}{\partial \boldsymbol{x}}(\boldsymbol{x}_{\text{best}})\right](\boldsymbol{x} - \boldsymbol{x}_{\text{best}})$$

$$\mathbf{H} = \frac{\partial \boldsymbol{m}}{\partial \boldsymbol{x}}(\boldsymbol{x}_{\text{best}}) \quad , \quad \text{i.e.} \quad H_{i,j} = \frac{\partial m_i}{\partial x_j}(\boldsymbol{x}_{\text{best}})$$

$$\mathbf{C}_{\boldsymbol{x}_{\text{best}}} \approx (\mathbf{H}^{\mathrm{T}}.\mathbf{C}_{\boldsymbol{r}}^{-1}.\mathbf{H})^{-1}$$

- Relation between errors on data and errors on parameters

$$\mathbf{C}_{\boldsymbol{x}_{\text{best}}} \approx \left[\left[\frac{\partial \boldsymbol{m}}{\partial \boldsymbol{x}}(\boldsymbol{x}_{\text{best}})\right]^{\mathrm{T}}.\mathbf{C}_{\boldsymbol{r}}^{-1}.\left[\frac{\partial \boldsymbol{m}}{\partial \boldsymbol{x}}(\boldsymbol{x}_{\text{best}})\right]\right]^{-1}$$

assuming model is good !

- Reminder:
  - assume modeled data are the expected value of data (i.e. the fitted model is good)
  - assume gaussian statistics
  - assume first order expansion is a good approximation
  - this only translates the statistical errors from data to the parameters

Errors on fitted parameters / 3

- General theorem of Cramér-Rao lower bound

$$\mathbf{C}_{\boldsymbol{x}} \geq \left[\nabla_{\boldsymbol{x}} \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x})\right]^{-1}$$

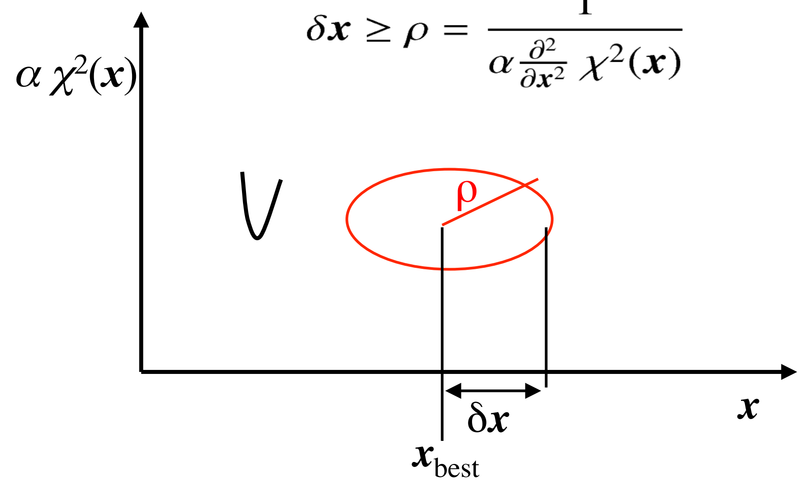
with log-likelihood: $\mathcal{L}(\boldsymbol{x}) = -\log \mathrm{Pdf}\left(\boldsymbol{d} \,\middle|\, \boldsymbol{m}(\boldsymbol{x})\right)$

- We come back to χ^2 using Gaussian assumption:

$$\mathcal{L}(\boldsymbol{x}) = \frac{1}{2}\left[\boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})\right]^{\mathrm{T}} . \mathbf{C}_r{}^{-1} . \left[\boldsymbol{d} - \boldsymbol{m}(\boldsymbol{x})\right] + \mathrm{Cte}$$
$$= \frac{1}{2} \chi^2(\boldsymbol{x}) + \mathrm{Cte}$$

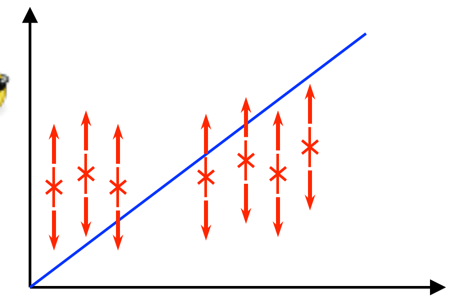
To get the idea, in 1 dimension:

$$\delta \boldsymbol{x} \geq \rho = \frac{1}{\alpha \frac{\partial^2}{\partial \boldsymbol{x}^2} \chi^2(\boldsymbol{x})}$$



Errors on fitted parameters: rescaling

- The model is not so good (~~assumption~~) :
  - χ^2 is bad (>> N_{free})
  - errors on parameters may be good (only statistics) !



$$\chi^2(\boldsymbol{x}_{\text{best}}) = \sum_{i=1}^{N_{\text{data}}} \frac{r_i^2(\boldsymbol{x}_{\text{best}})}{\sigma_i^2} \gg N_{\text{free}}$$

we look for α such that: $$\sum_{i=1}^{N_{\text{data}}} \frac{r_i^2(\boldsymbol{x}_{\text{best}})}{(\alpha\,\sigma_i)^2} = N_{\text{free}}$$

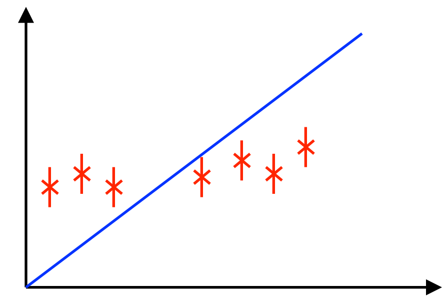
$$\Rightarrow \quad \alpha = \sqrt{\frac{\chi^2(\boldsymbol{x}_{\text{best}})}{N_{\text{free}}}} = \sqrt{\chi_r^2(\boldsymbol{x}_{\text{best}})}$$

$$\Rightarrow \quad \mathbf{C}_{\boldsymbol{x}_{\text{best}}} \approx \alpha^2 \left[\left[\frac{\partial \boldsymbol{m}}{\partial \boldsymbol{x}}(\boldsymbol{x}_{\text{best}})\right]^{\mathrm{T}} . \mathbf{C}_{\boldsymbol{r}}^{-1} . \left[\frac{\partial \boldsymbol{m}}{\partial \boldsymbol{x}}(\boldsymbol{x}_{\text{best}})\right] \right]^{-1}$$

Errors on fitted parameters: rescaling

- The model is not so good (~~assumption~~):
  - χ^2 is bad (>> N_{free})
  - errors on parameters may be good (only statistics) !

$$\chi^2(\boldsymbol{x}_{\text{best}}) = \sum_{i=1}^{N_{\text{data}}} \frac{r_i^2(\boldsymbol{x}_{\text{best}})}{\sigma_i^2} \gg N_{\text{free}}$$

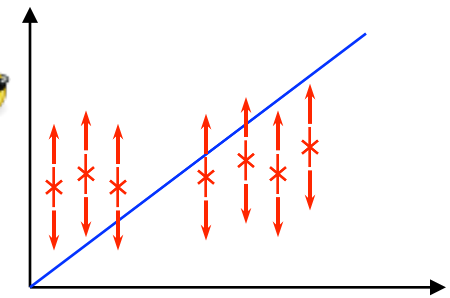


- Errors on parameters could increase when the model does not fit !

- How ?
  - with χ^2 statistics

Rescaling using χ^2 statistics

- The model is not so good (~~assumption~~):
  - χ^2 is bad ($>> N_\text{free}$)
  - errors on parameters may be good (only statistics) !



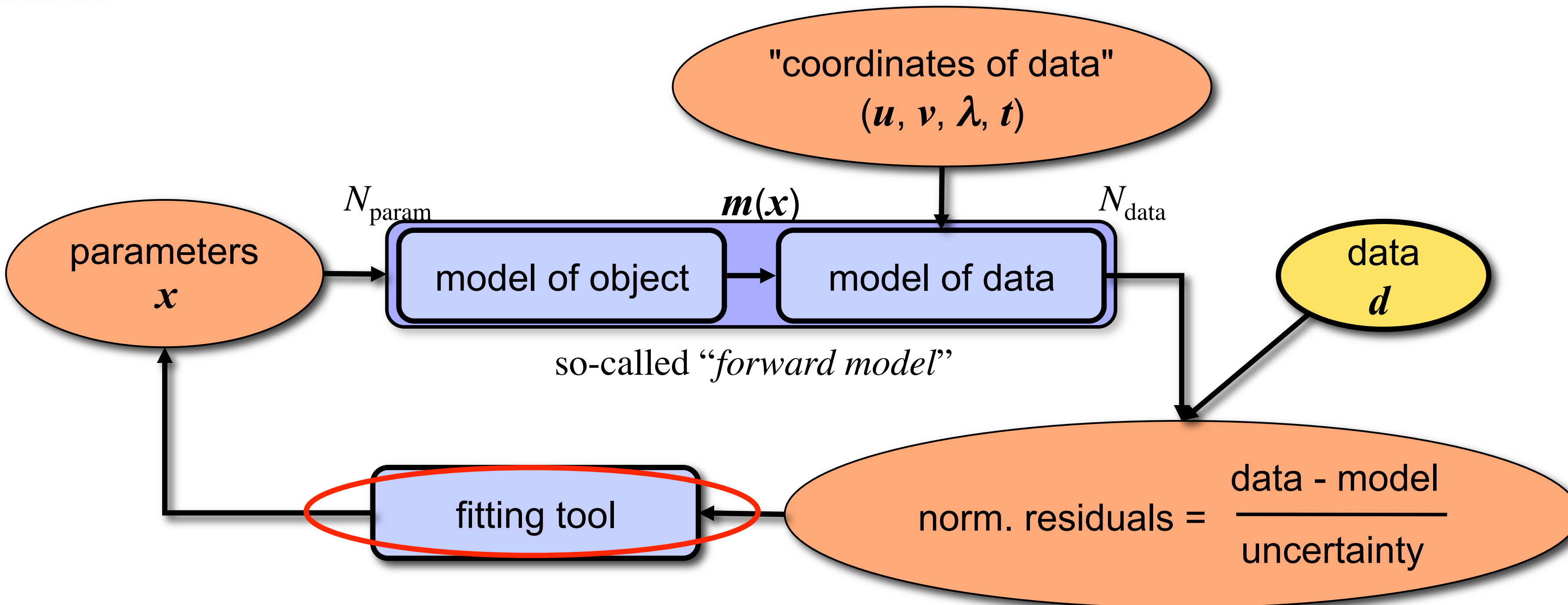
$$\chi^2(\boldsymbol{x}_\text{best}) = \sum_{i=1}^{N_\text{data}} \frac{r_i^2(\boldsymbol{x}_\text{best})}{\sigma_i^2} \gg N_\text{free}$$

we look for α such that: $$\sum_{i=1}^{N_\text{data}} \frac{r_i^2(\boldsymbol{x}_\text{best})}{(\alpha\,\sigma_i)^2} = N_\text{free}$$

$$\Rightarrow \quad \alpha = \sqrt{\frac{\chi^2(\boldsymbol{x}_\text{best})}{N_\text{free}}} = \sqrt{\chi_r^2(\boldsymbol{x}_\text{best})}$$

$$\Rightarrow \quad \mathbf{C}_{\boldsymbol{x}_\text{best}} \approx \alpha^2 \left[\left[\frac{\partial \boldsymbol{m}}{\partial \boldsymbol{x}}(\boldsymbol{x}_\text{best})\right]^\mathrm{T} . \mathbf{C}_{\boldsymbol{r}}^{-1} . \left[\frac{\partial \boldsymbol{m}}{\partial \boldsymbol{x}}(\boldsymbol{x}_\text{best})\right] \right]^{-1}$$

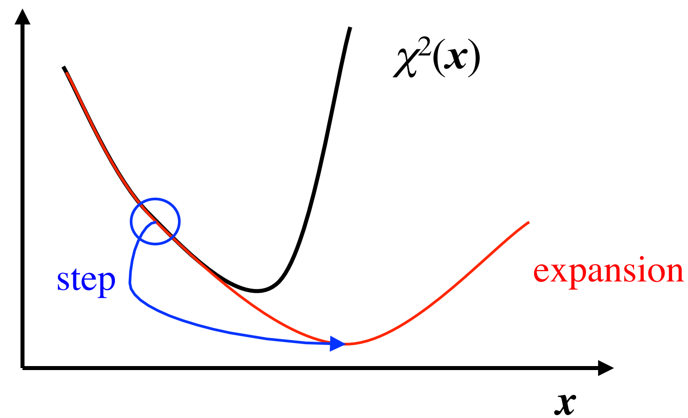
Model fitting principle



Outline of the optimization

- Needs
  - Minimize (iteratively!) $\chi^2(\boldsymbol{x})$ (sum of squares)
  - Non-linear, non-convex
- Local optimization with Newton method
  - step from a local expansion at second order
    - need of gradients (Jacobian matrix)
    - need of second derivatives (Hessian matrix)
  - but step may be too long
    - outside region where quadratic approximation is valid
- Control of the length of the step
  - add a constrain that deforms the cost function
- Levenberg-Marquardt algorithm
  - minimize a sum of squares
  - only need gradients
    - finite differences are ok
  - Hessian is approximated
    - we only keep product of derivatives

Newton step may be too long



=> We are looking for a local minimum

Local optimization with Newton method

- Second order expansion of the "cost function" we want to minimize

$$f(\boldsymbol{x}+\delta\boldsymbol{x}) = f(\boldsymbol{x}) + \delta\boldsymbol{x}^{\mathrm{T}}.\boldsymbol{g}(\boldsymbol{x}) + \frac{1}{2}\delta\boldsymbol{x}^{\mathrm{T}}.\mathbf{H}(\boldsymbol{x}).\delta\boldsymbol{x} + o(\|\delta\boldsymbol{x}\|^2)$$

where

$$\boldsymbol{g}(\boldsymbol{x}) \equiv \nabla f(\boldsymbol{x}) \qquad g_i(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial x_i} \qquad \text{(gradient)}$$

$$\mathbf{H}(\boldsymbol{x}) \equiv \nabla\nabla f(\boldsymbol{x}) \qquad H_{i,j}(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial x_i \partial x_j} \qquad \text{(a.k.a. Hessian matrix)}$$

- Local quadratic approximation around $\boldsymbol{x}$.

$$f(\boldsymbol{x}+\delta\boldsymbol{x}) - f(\boldsymbol{x}) \approx q(\delta\boldsymbol{x}) \equiv \delta\boldsymbol{x}^{\mathrm{T}}.\boldsymbol{g}(\boldsymbol{x}) + \frac{1}{2}\delta\boldsymbol{x}^{\mathrm{T}}.\mathbf{H}(\boldsymbol{x}).\delta\boldsymbol{x}$$

- Optimal step

$$\delta\boldsymbol{x}_{\text{quad}} = \arg\min_{\delta\boldsymbol{x}} q(\delta\boldsymbol{x}) = -\mathbf{H}(\boldsymbol{x})^{-1}.\boldsymbol{g}(\boldsymbol{x})$$

- + Method to prevent too large steps
  - at each step, reduce the "*trust region*" if quadratic approx is not good

Levenberg-Marquardt method

- Same ideas, but made specific to $\chi^2(\boldsymbol{x})$ function

$$f(\boldsymbol{x}) = \chi^2(\boldsymbol{x}) = \sum_{i=1}^{N_{\text{data}}} e_i^2(\boldsymbol{x}) \qquad \text{with} \qquad e_i(\boldsymbol{x}) = \frac{d_i - m_i(\boldsymbol{x})}{\sigma_i}$$

- Expressions of gradient and Hessian matrix

$$g_k(\boldsymbol{x}) = \frac{\partial f}{\partial x_k}(\boldsymbol{x}) = 2 \sum_{i=1}^{N_{\text{data}}} \frac{\partial e_i(\boldsymbol{x})}{\partial x_k} \, e_i(\boldsymbol{x})$$

$$H_{k,l}(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial x_k \partial x_l} = 2 \sum_{i=1}^{N_{\text{data}}} \frac{\partial e_i(\boldsymbol{x})}{\partial x_k} \frac{\partial e_i(\boldsymbol{x})}{\partial x_l} + 2 \sum_{i=1}^{N_{\text{data}}} \frac{\partial e_i(\boldsymbol{x})}{\partial x_k \partial x_l} \, e_i(\boldsymbol{x})$$

- \+ Approximation of Hessian matrix

$$H_{k,l}(\boldsymbol{x}) \approx 2 \sum_{i=1}^{N_{\text{data}}} \frac{\partial e_i(\boldsymbol{x})}{\partial x_k} \, \frac{\partial e_i(\boldsymbol{x})}{\partial x_l}$$

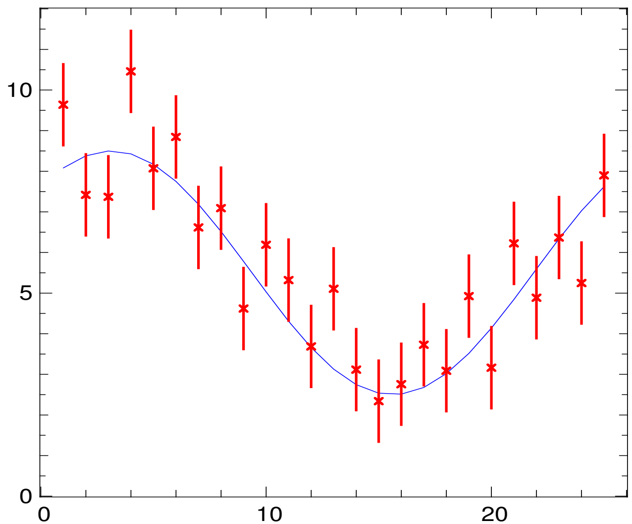
- \+ Method to prevent too large steps…
- \+ Method to take bounds into account…

Summary on theory

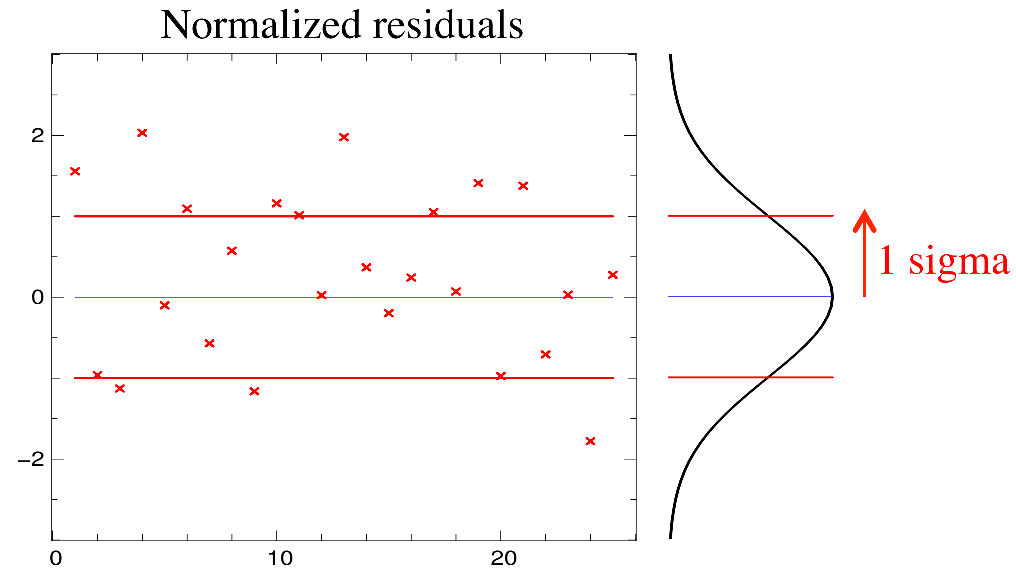
- OI-FITS data
  - with errors on data
- model of object ↔ model of data
- assumption of Gaussian statistics of residuals
- assumption of statistical independency of data
  - not really true in real world
- χ^2 law
  - assume fitted model is good
  - sharp statistics
  - use reduced χ^2 for comparing two models on same data
- errors on parameters
  - estimated from errors on data, rescaled for systematic errors
  - correlations of parameters are estimated (and they must be)
- Optimization
  - Local minimization
  - Need of gradients only (finite differences is ok, **but beware at parameter scales**)

Digression on correlations of data

Appearance of independence



- simulated data
- model is perfect
- model is outside the error bars (1 sigma) for 32% of the data

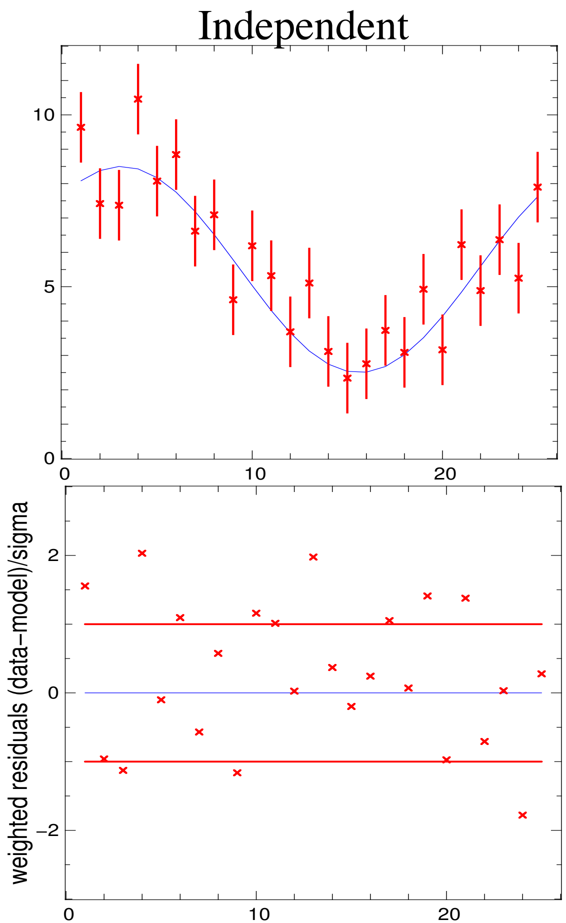


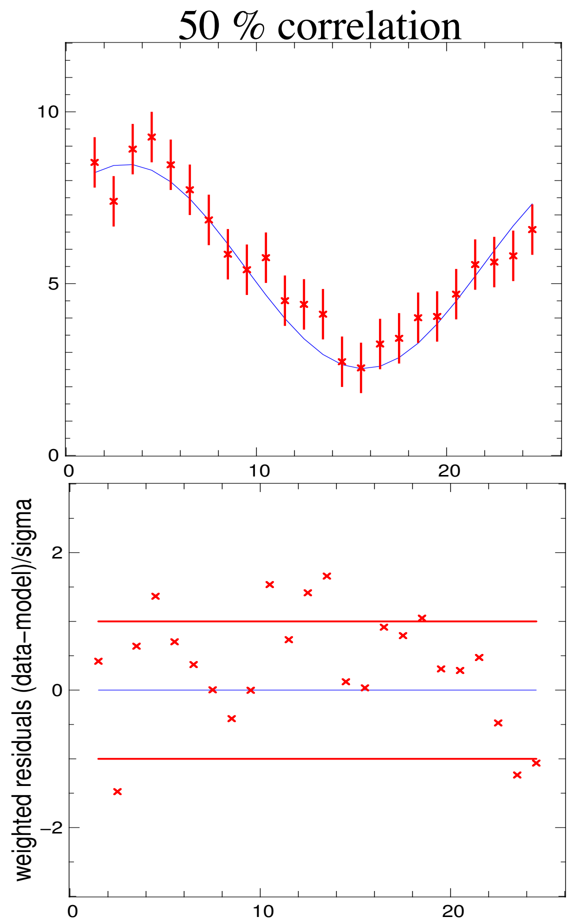
$$e_i(\boldsymbol{x}) = \frac{d_i - m_i(\boldsymbol{x})}{\sigma_i}$$

- easier to compare data with various error bars
- show the true weight of data

Beware : only one realization here !

Data with adjacent correlations: 50%

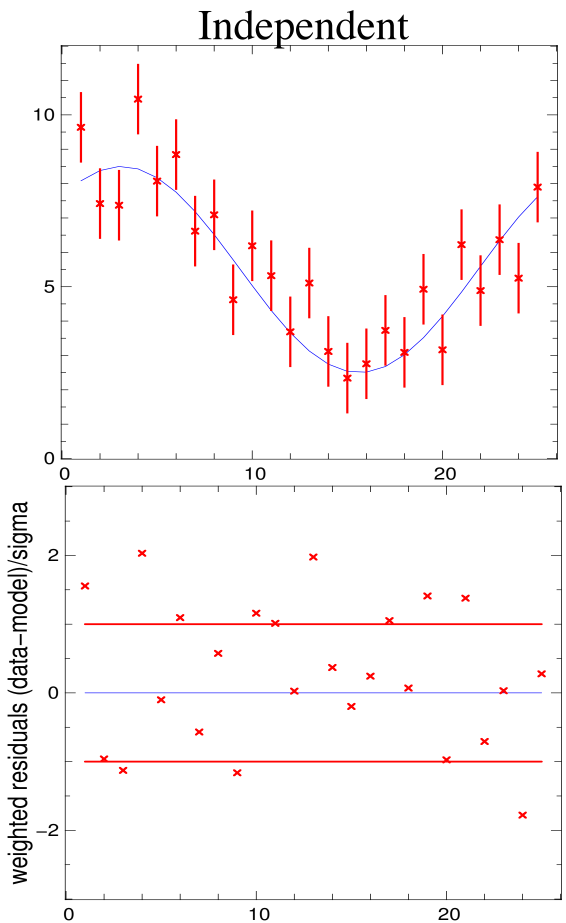


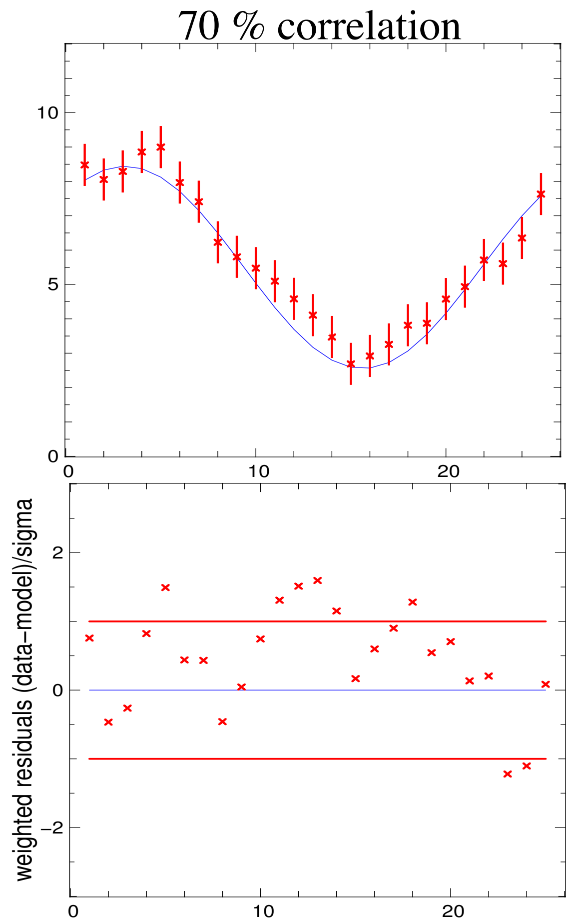


- average of adjacent points
- => 50% correlation coefficient, only between adjacent points.
- Similar effect as spectral correlations in real data
- more alignments of successive points
- less dispersion of residuals

Beware : only one realization !

Data with adjacent correlations: 70%

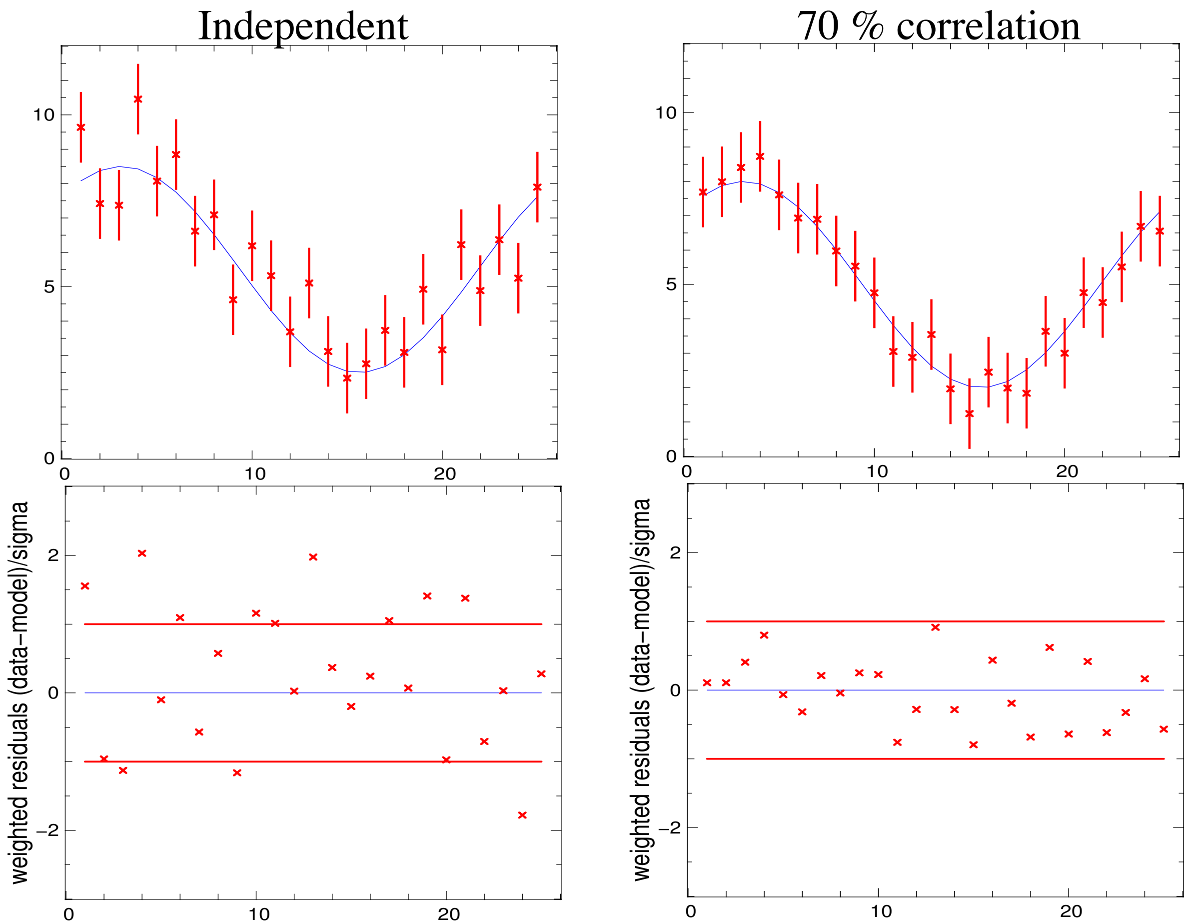




- correlation coefficient:
  - 70% between adjacent points.
  - 25% with next points
- similar effect as (more) spectral correlations in real data
- yet more alignments of successive points
- less dispersion of residuals

Beware : only one realization !

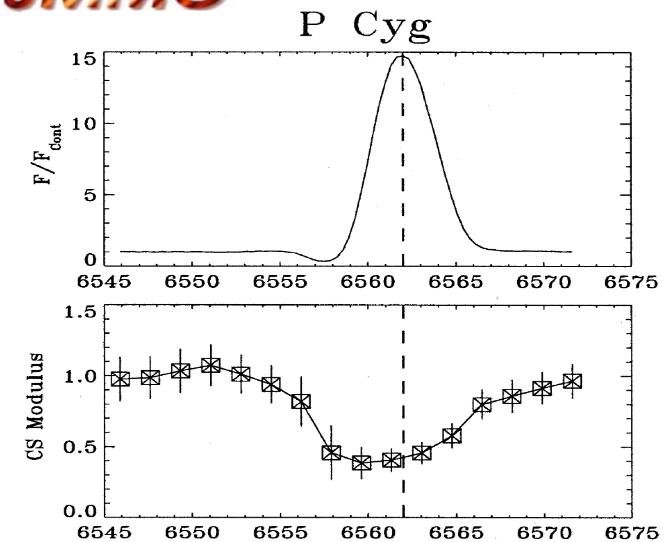
Data with global correlations: 70%



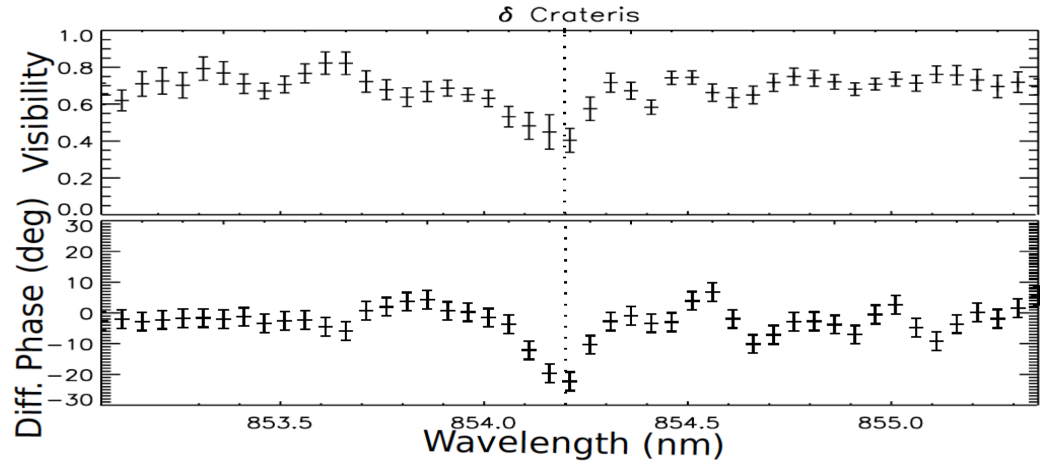
- 70% correlation between any points => more correlations
- Similar effect as noise on normalization (incoherent flux, calibrator)
- Less dispersion of residuals

Beware : only one realization !

Examples on real data

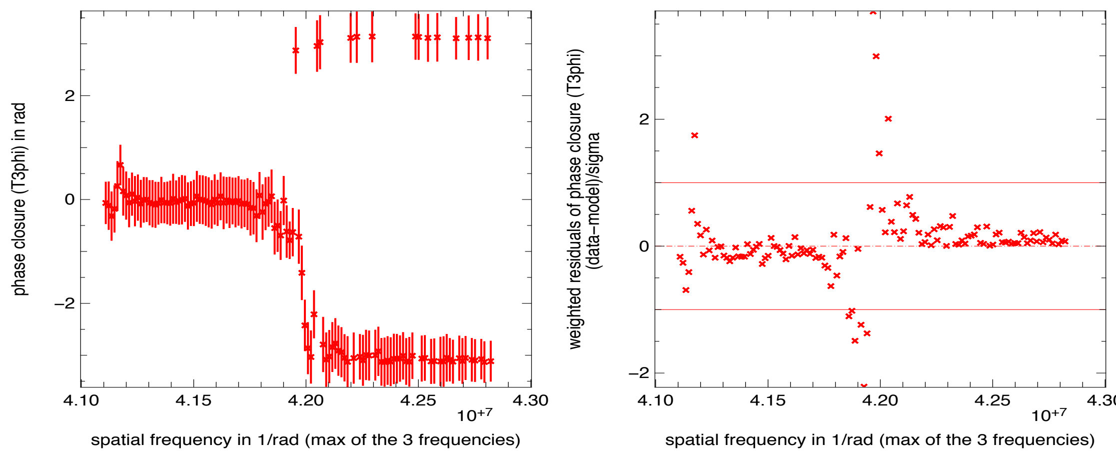


GI2T, Vakili et al 1997



CHARA/VEGA, Berio et al 2011

Amber data,
T3phi on Sirius
Duvert 2013



Summary on correlation

- Several ways to get correlated data
- When assuming independent data, correlations make χ^2 smaller
- Thus don't trust χ^2, confidence level, etc.
  - can be used to compare different models (reduced χ^2) or assess the progress of the fit.
  - difficult to use to accept or rule out a model.

LITpro model fitting software for optical interferometry

CRAL: M.Tallon, I. Tallon-Bosc, F. Soulez

IPAG: G. Mella, H. Beust, L. Bourgès, G. Duvert

CRAL, Lyon France — IPAG, Grenoble, France

What is LITpro ?

- Parametric model fitting software for interferometry
  - Conceived and developed up-to-now at CRAL in Lyon
  - Graphical User Interface developed at JMMC (Jean-Marie Mariotti Center)
  - Maintained and improved by the "model-fitting" group at JMMC (several labs in France)

- Aim: "exploit the scientific potential of existing interferometers", *e.g.* VLTI

- Use of OIFITS standard for data

- Complementary to image reconstruction
  - Sparse (u,v) coverage
  - Model fitting extracts measured quantities
  - Image reconstruction may help to identify models
  - Fitted model as a first guess for image reconstruction

Leading requirements of LITpro

- Accessible to "general users" + flexible for "advanced users"
  - Opposite needs:
    - General users want simplicity (stepping stone)
    - Advanced users want a powerful tool (pioneering work)
  - Exchanges:
    - general users —(needs)—› advanced users
    - general users ‹—(training)— advanced users
  - Progress must benefit to everybody (share experiences)

- Concentrate on the model of the object
  - Easy implementation of new models.
  - Only need to compute the Fourier transform of the object specific intensity on given coordinates ($\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\lambda}, \boldsymbol{t}$)

Leading requirements ⇒ implementation

- Accessible to astronomers + flexible for advanced users
  - flexible ⇒ high level language (*Yorick*)
    - easy modifications and adds in the software
    - "expert layer"
  - accessible ⇒ GUI
    - new abilities exposed once they are validated in the "expert" layer

- Concentrate on the model of the object
  - From Fourier transform of the object:
    - Modeled data (interferometric, spectroscopic, photometry, …)
    - Images
  - LITpro also provides
    - Modeling builder (with GUI or filling a form)
    - Models of data
    - Fitter "engine"
    - Tools for analysis

Types of data

- OIFITS
  - Squared visibilities (VIS2)
  - Complex visibilities (VISAMP, VISPHI)
  - Bispectrum (T3AMP, T3PHI)

- Others
  - Spectral Energy Distribution (dispersed fringes mode)
  - Photometry (see example)
  - …

Setting up the fitting process / principle

fit altogether

Associate data with a model…
- data files, …
  - + function 1: parameter1 = "p1", …
  - + function 2: parameter1 = "p2", …
  - + …

As many groups as you want …
- other data files, …
  - + function 1: parameter1 = "p3", …
  - + function 3: parameter1 = "p4", …
  - + …

As many groups as you want…
- other data files, …
  - + disk: diameter = "d1", x="x", y="y" …
  - + disk: diameter = "d2", x="x", y="y" …
  - + …

Configuration of parameters
- "p1" —› starting value, bounds, fixed or fitted, units
- "p2" —› starting value, bounds, fixed or fitted, units
- …
- "r1" —› starting value, bounds, fixed or fitted, units
- …

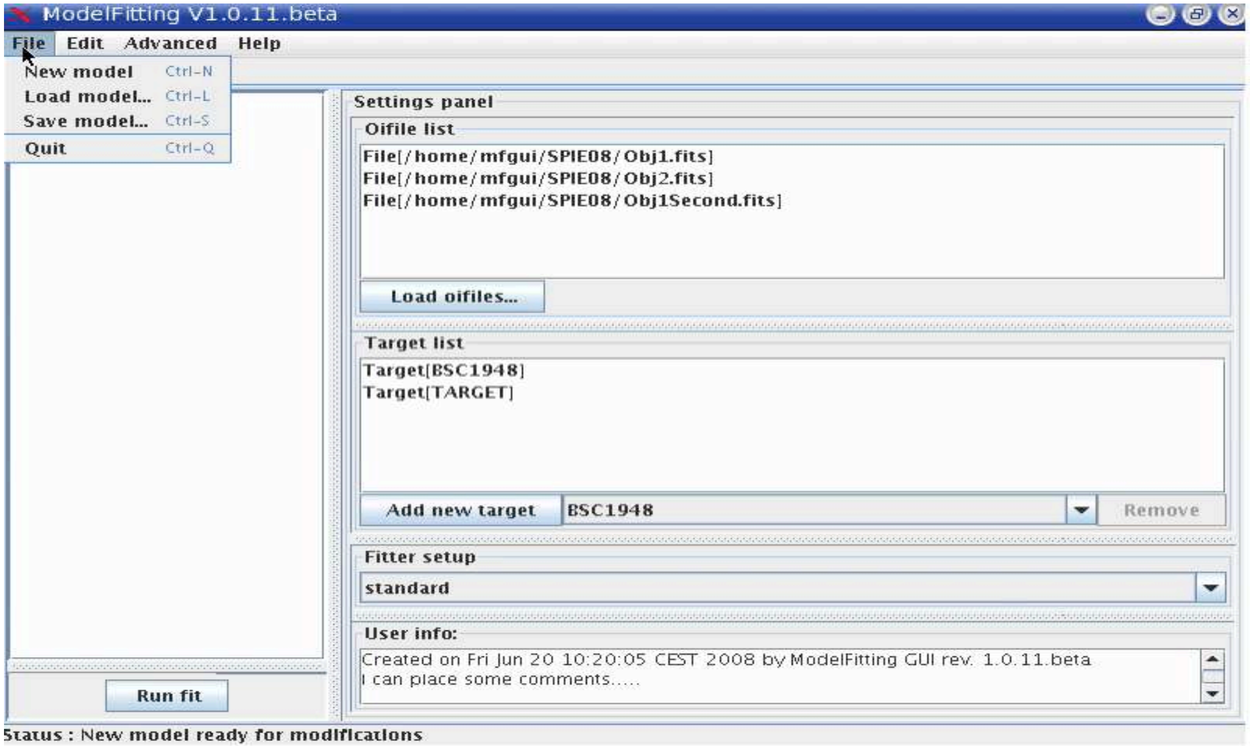
- Through the GUI

Fitting process

- Levenberg-Marquardt algorithm (modified)
  - Combined with a Trust Region method
  - Bounds on the parameters
  - Partial derivatives of the model by finite differences

- Coming improvement…
  - Genetic fitter (global minimum)
  - Interface with OImaging (image reconstruction)
  - Other fitters (for global search)
  - Improved algebra for building object models
  - …

Implementation of the GUI



- Implemented in JAVA
  - Web service
  - Links with other services (JMMC)
    - Virtual Observatory
    - Data explorer
    - User feedback
    - …
- GUI just tells "expert layer" (*Yorick*) what to do
- First public release: October 2009

Work in progress

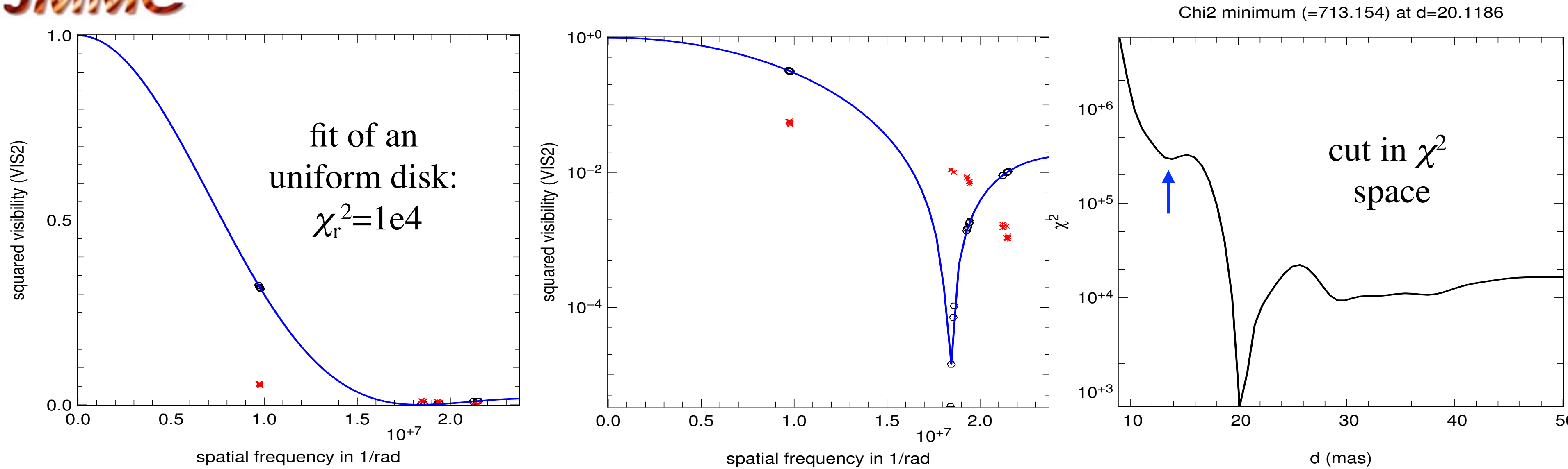
- LITpro
  - First public release Octobre 2009

- High in the list for near future
  - Easy implementation of "user models"
  - Search for global minimum of χ^2
  - Read spectrum in OIFITS2
  - Functions for multichromatic modeling (e.g. dynamics)
  - Cooperation between Image reconstruction and Model fitting

Adventure of model fitting

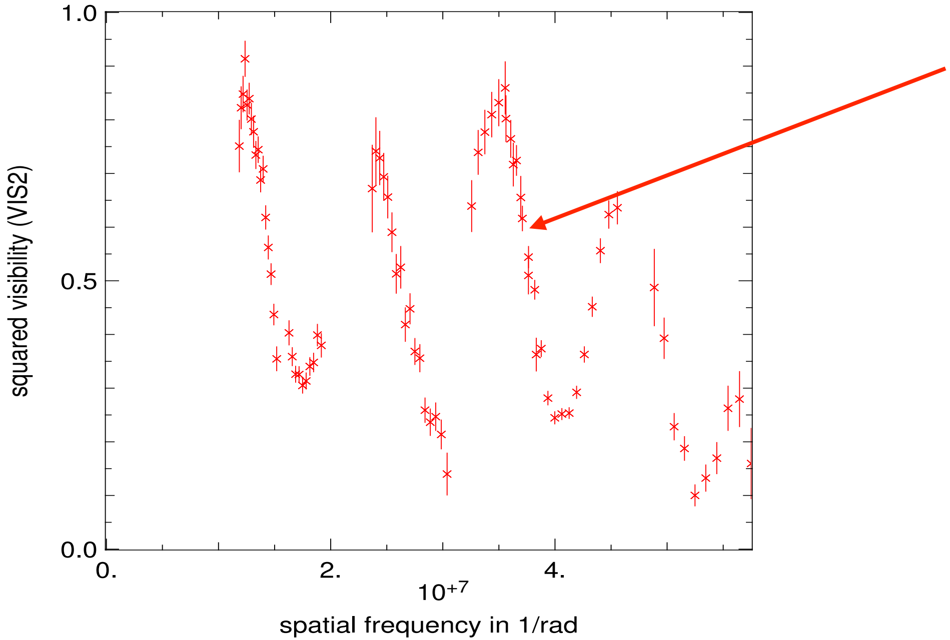
- Local minimum
  - example of an uniform disk
- Observe your data… the Guru way
  - useful for initial guess (local minimum)
- Degeneracies
  - on the total energy
- Example of a "heterogeneous" model-fitting

Beware of local minima !



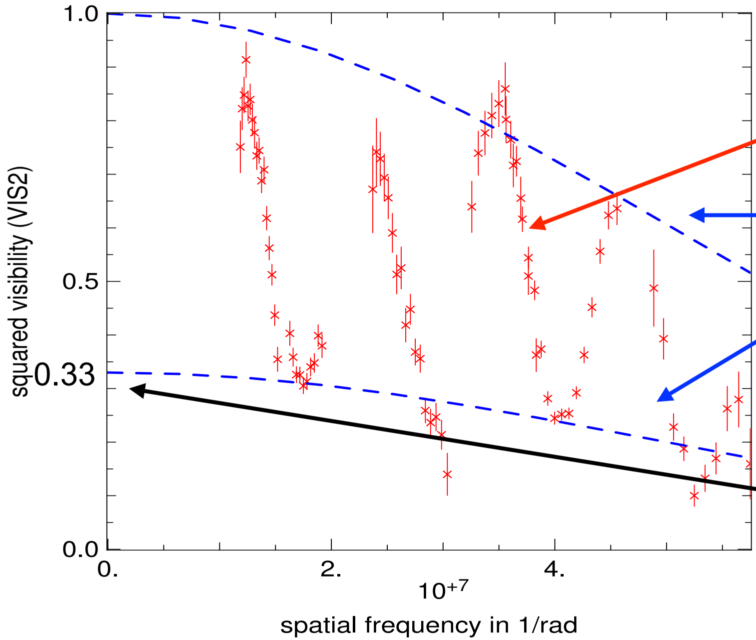
- local minima exists even for a uniform disk, depending on data
- what to do ?
  - change first guess
  - cuts in χ^2 sub-spaces
  - use bounds
  - do not forget the low frequencies (or just confirm what we already know…)

Observe your data !



Modulation = binary (or >2 components)

Observe your data !



$$V^2_{\min} = 0.33 \Rightarrow r = 0.27$$

Modulation = binary (or >2 components)

Attenuation = components resolved

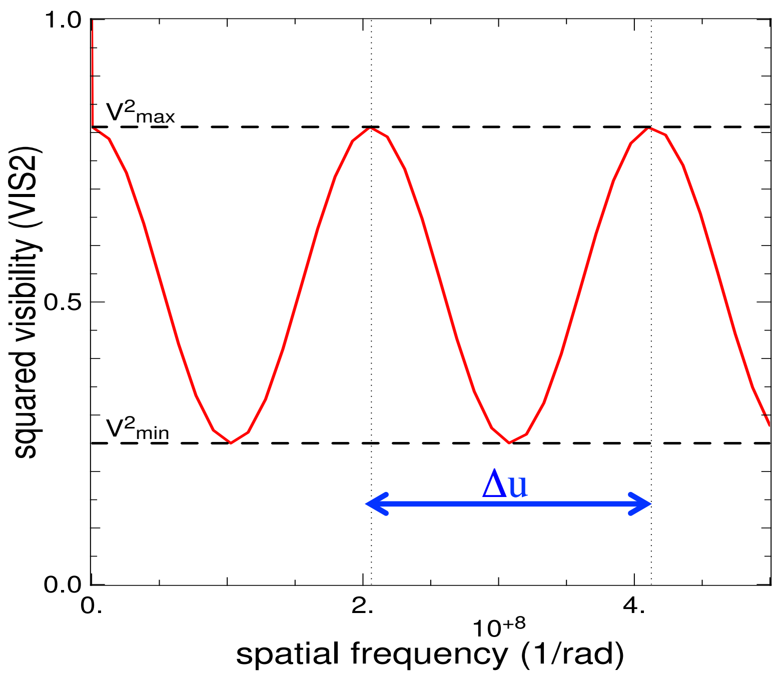
- binary convolved by an extended function
- ⇩
- Fourier transform multiplied by a window

Minimum of modulation gives intensity ratio of the components:

$$r = \frac{1 - \sqrt{V^2_{\min}}}{1 + \sqrt{V^2_{\min}}}$$

— Starting from a good first guess may be decisive —

Binary with what ?



… with background

$$\Phi_{\mathrm{background}} = 1 - \sqrt{V^2_{\mathrm{max}}}$$ (flux in background)

$$\Phi_{\mathrm{main}} = \frac{1}{2}\left(\sqrt{V^2_{\mathrm{max}}} + \sqrt{V^2_{\mathrm{min}}}\right)$$ (flux in main component)

$$\Phi_{\mathrm{secondary}} = \frac{1}{2}\left(\sqrt{V^2_{\mathrm{max}}} - \sqrt{V^2_{\mathrm{min}}}\right)$$ (flux in secondary component)

$$r = \frac{\sqrt{V^2_{max}} - \sqrt{V^2_{min}}}{\sqrt{V^2_{max}} + \sqrt{V^2_{min}}}$$ (flux ratio of the components)

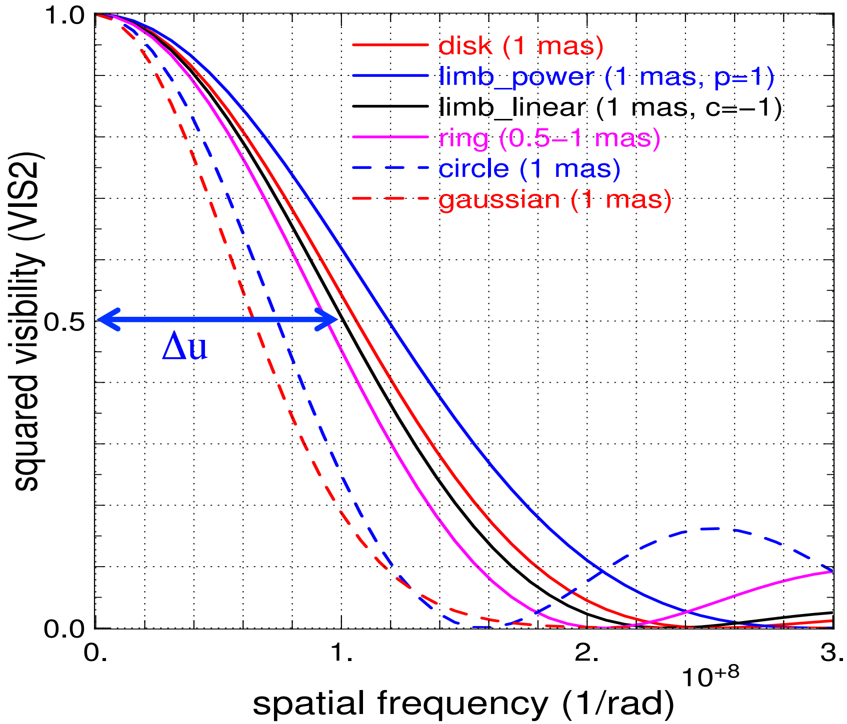
Separation of the components: $\rho = 1/\Delta u$

Here:

$\Delta u \sim 2\ \ 10^{+8}\ \ 1/\mathrm{rad}$

$\rho \sim 5\ \ 10^{-9}\ \mathrm{rad} \sim 1\ \mathrm{mas}$

Size of various object shapes



- Δu : width at half maximum (rad^{-1})

- typical FWHM of the object :

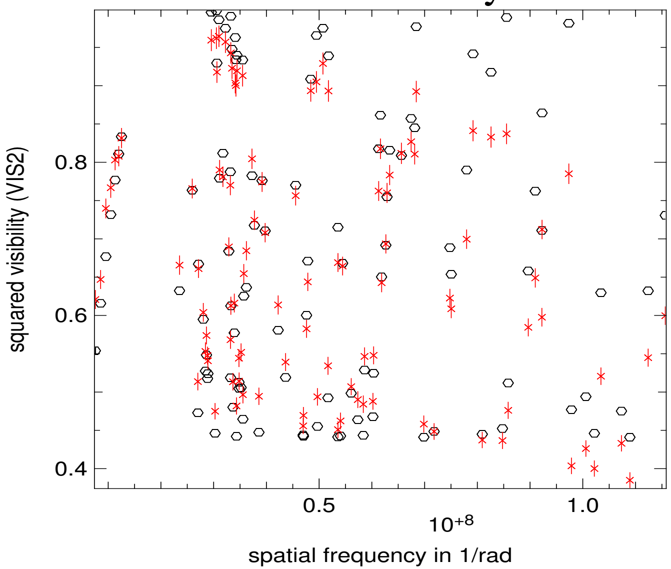
  fwhm [mas] ~ $10^8 / \Delta u$

- gaussian is the smallest :

  fwhm [mas] ~ $0.6 \times 10^8 / \Delta u$

Degeneracy on total energy

fit of a binary



Model of the binary

- main component at (0,0) with intensity i1
- secondary at (x,y) with intensity i2

```
Final values for fitted parameters and standard deviation:
      i1 =      0.20152 +/- 9.95e+04
      i2 =       0.9982 +/- 4.93e+05
       x =      -6.6657 +/-  0.00441  mas
       y =        20.08 +/-  0.00631  mas
.
        Chi2: initial= 7.376e+04 - final=  1983 - sigma= 14.2127
reduced Chi2: initial=     730.3 - final= 19.63 - sigma= 0.14072
Number of degrees of freedom = 101
.
--- Correlation matrix ---
                i1        i2         x         y
  i1             1         1    0.0011   -0.0015
  i2             1         1    0.0011   -0.0015
   x        0.0011    0.0011         1     -0.44
   y       -0.0015   -0.0015     -0.44         1
```

- this degeneracy does not change χ^2
- huge errors because of no curvature of $\chi^2(\boldsymbol{x}_{\text{best}})$ for i1+i2
- this prevents reading the values of i1 and i2

Degeneracy on total energy: a solution

- FAQ:
  - We could construct a normalized model !
  - Yes, but we want to combine all sorts of functions…
  - We could combine normalized functions !
  - Not always possible ! Ex: disk with constant amplitude (spot on a star)

- *When total energy is not fixed by the data*, we add this constraint:

$$\chi^2_\star(\boldsymbol{x}) = \chi^2(\boldsymbol{x}) + N_d \left(\frac{\sum_i \Delta\lambda_i \, \widetilde{O}(\boldsymbol{x}, \boldsymbol{u}, \lambda_i, \ldots)\Big|_{\boldsymbol{u}=0}}{\sum_i \Delta\lambda_i} - 1 \right)^2$$

This drives total energy to unity

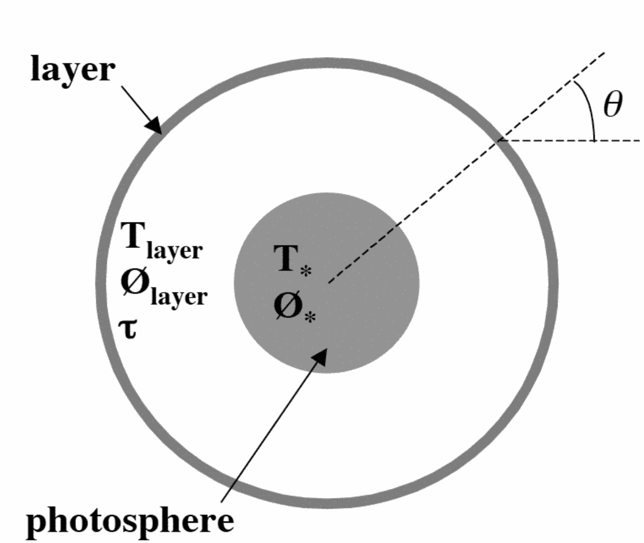
- But the added term MUST BE ZERO at the end of the fit !
  - If not: χ^2 is changed and quantities are wrong !

- Other degeneracies in practice
  - translation of the map (unless phase reference)
  - symmetries if no phase
  - …

Degeneracy on total energy: solved

```
Final values for fitted parameters and standard deviation:
      i1 =     0.83203 +/-   0.0812
      i2 =     0.16797 +/-   0.0164
       x =     -6.6657 +/-  0.00441  mas
       y =       20.08 +/-  0.00631  mas
.
        Chi2: initial= 7.376e+04 - final=  1983 - sigma= 14.2127
reduced Chi2: initial=     730.3 - final= 19.63 - sigma= 0.14072
Number of degrees of freedom = 101
.
--- Correlation matrix ---
           i1        i2         x         y
  i1        1         1   0.00021   0.00058
  i2        1         1   -0.0011   -0.0029
   x  0.00021   -0.0011         1     -0.44
   y  0.00058   -0.0029     -0.44         1
```

Example: chromatic model + heterogeneous data / 1



Perrin et al, A&A 426, 279, 2004

$$I(\lambda,\theta) = B(\lambda, T_\star)\exp(-\tau(\lambda)/\cos(\theta)) + B(\lambda, T_{\rm layer})\,[1-\exp(-\tau(\lambda)/\cos(\theta))]$$

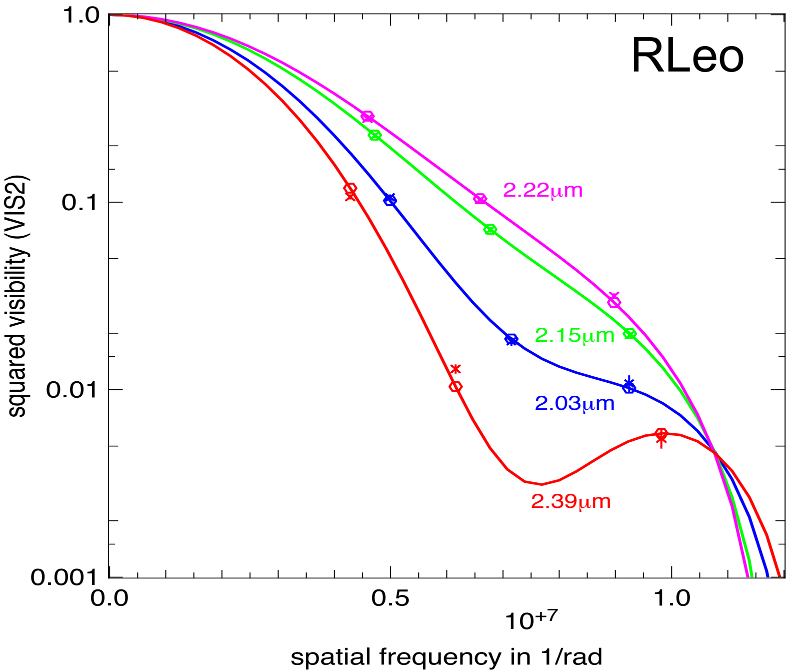
for $\sin(\theta) \leq \varnothing_\star/\varnothing_{\rm layer}$ and:

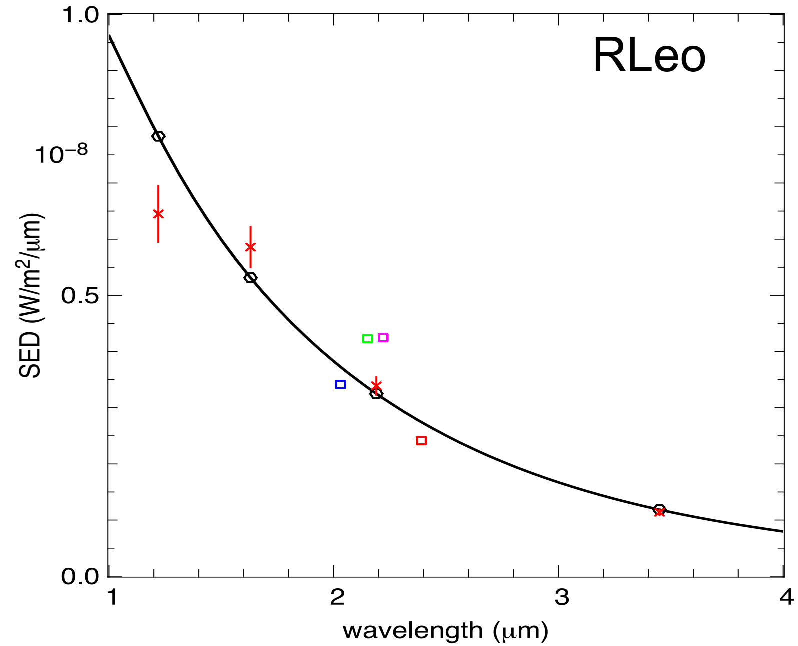
$$I(\lambda,\theta) = B(\lambda, T_{\rm layer})\,[1-\exp(-2\tau(\lambda)/\cos(\theta))]$$

- Why this example in particular ?
  - Fitting procedure is difficult
    - Need to improve procedures for "general users" (accessible ?)
    - How LITpro performs ?
  - Fitting interferometric + photometric data
    - Assess how it can help the fitting process

CENTRE DE RECHERCHE ASTROPHYSIQUE DE LYON

Example: chromatic model + heterogeneous data / 2

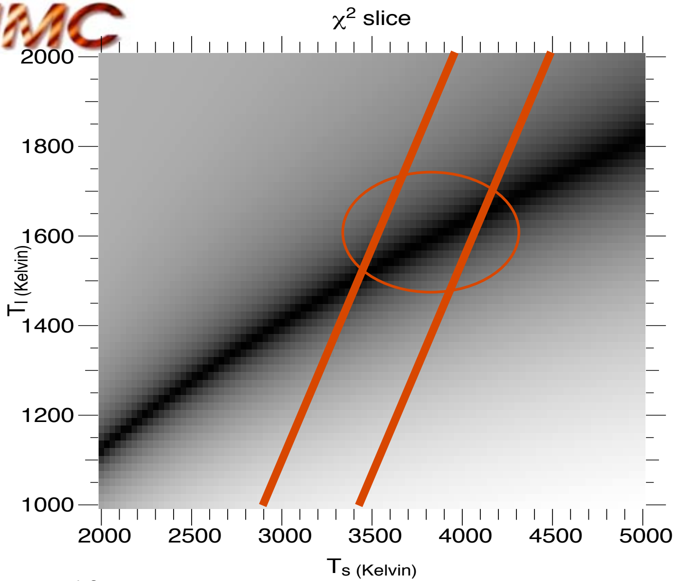


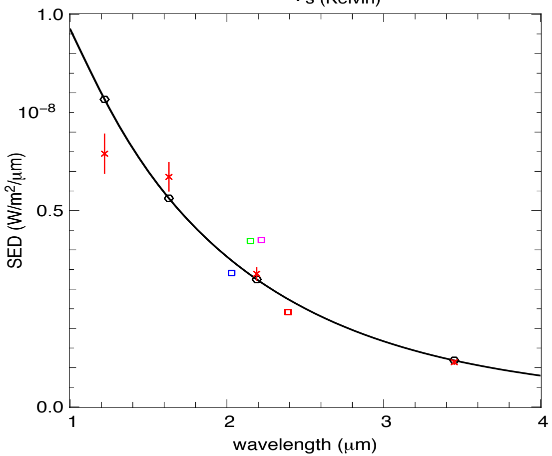


Perrin et al, A&A 426, 279, 2004

- squared visibilities : 4 sub-bands in K band (IOTA)
- magnitudes : J, H, K, L bands (Whitelock et al 2000)
- Difficulty: T_* and T_L coupled (no constrain on total flux from interferometric data)

Perrin et al. fitting procedure





1) (R_*, R_L) from gridding
   - fit all other parameters from fixed sampled values (R_*, R_L)
   - arbitrary initial values of other parameters
   - => (R_*, R_L) fixed

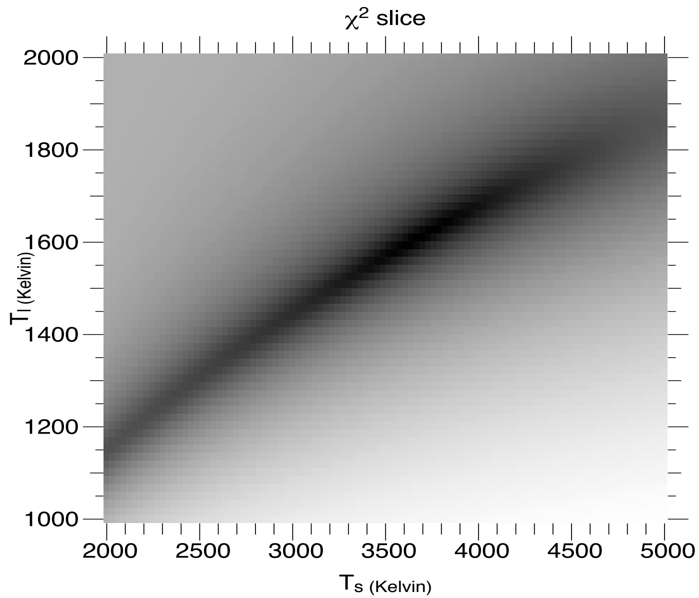
2) (T_*, T_L) from gridding + intersection with K photometry
   - No use of the other bandwidths

3) Fit the 4 optical depths from fixed other parameters

4) Compare/check photometry with other bandwidths: J, H, L.

Simultaneous fitting of all the data

(interfermetric data + photometric data)



1) Overall size of the object ?
   - Radius of uniform disk: 18 mas

2) Overall temperature for this disk size ?
   - For an uniform disk: 1540K

3) Fit from this initial values
   - R_*=18 mas, T_*=T_L=1540 K
   - Initial values of optical depths set to zero => uniform disk

May be useful (and reassuring) to use physical arguments for the first guess…

Comparison of results

Parameter	Perrin et al.	Simultaneous fit	Fit with relative photometry
$R_\star$ (mas)	10.94 ± 0.85	11 ± 0.13	11 ± 0.19
R_L (mas)	25.00 ± 0.17	25.4 ± 0.16	25.4 ± 0.18
$T_\star$ (K)	3856 ± 119	3694 ± 113	3778 ± 163
T_L (K)	1598 ± 24	1613 ± 35	1681 ± 174
$\tau_{2.03}$	1.19 ± 0.01	1 ± 0.14	0.9 ± 0.35
$\tau_{2.15}$	0.51 ± 0.01	0.42 ± 0.08	0.36 ± 0.17
$\tau_{2.22}$	0.33 ± 0.01	0.27 ± 0.05	0.23 ± 0.11
$\tau_{2.39}$	1.37 ± 0.01	1.2 ± 0.13	1.08 ± 0.32
γ	–	–	0.9 ± 0.2

Fit with only relative photometry, like the SED given by an optical interferometer

Correlation matrix

	R_l	Rs_ratio	T_l	T_s	tau1	tau2	tau3	tau4
R_l	1	-0.66	-0.36	0.14	0.21	0.17	0.16	0.13
Rs_ratio	-0.66	1	0.71	-0.6	-0.67	-0.67	-0.66	-0.62
T_l	-0.36	0.71	1	-0.74	-0.94	-0.93	-0.93	-0.92
T_s	0.14	-0.6	-0.74	1	0.91	0.91	0.92	0.92
tau1	0.21	-0.67	-0.94	0.91	1	0.99	0.99	0.99
tau2	0.17	-0.67	-0.93	0.91	0.99	1	0.99	0.99
tau3	0.16	-0.66	-0.93	0.92	0.99	0.99	1	0.99
tau4	0.13	-0.62	-0.92	0.92	0.99	0.99	0.99	1

Conclusions on the adventure

- Local minima even with uniform disk
  - cuts in χ^2 space
  - change first guess
  - check χ_r^2 if variations are significant
- Model-fitting algorithm has no brain (yet!)
  - use yours: look carefully at the data: (u,v) coverage, baselines
- Degeneracies may appear
  - check covariances of parameters
  - check ON/OFF normalization of total energy
- Quality of the fit / model
  - χ^2
  - understand errors *and correlations* on parameters
  - various plots

Ready for the practice?

Your road map: 5 (+1) exercises

1. Fit of a simple model on one file (Arcturus)
   – explore the software
   – easy fits, easy problem
2. Fit with parameter sharing on several files (Arcturus)
   – more evolved model
3. Fit with degeneracies (binary)
   – explain them !
4. Fit on AMBER data
   – you are alone (almost)
5. Fit of a star + environment with chromatic artefacts
6. … additional optional exercises