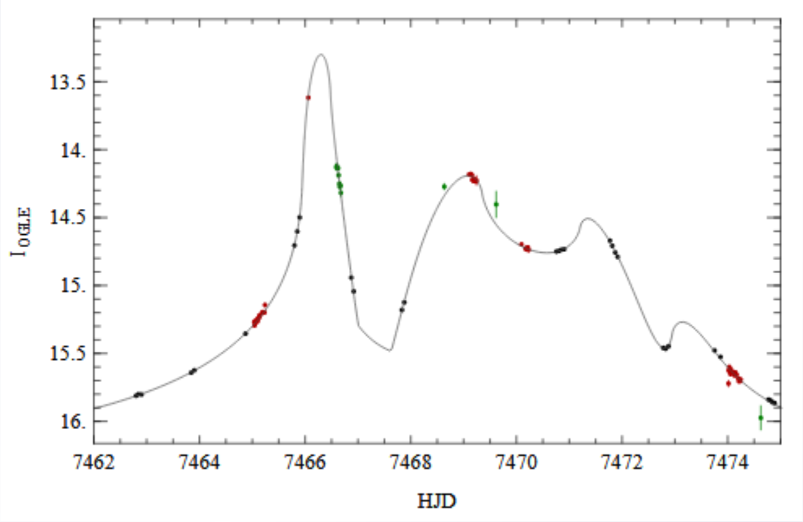
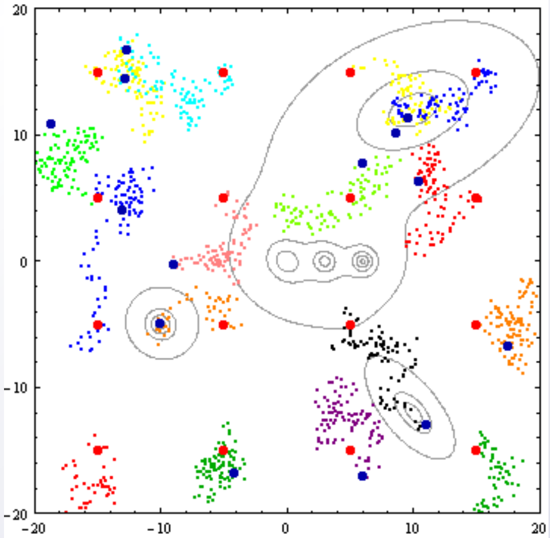
Fitting Light Curves in Microlensing



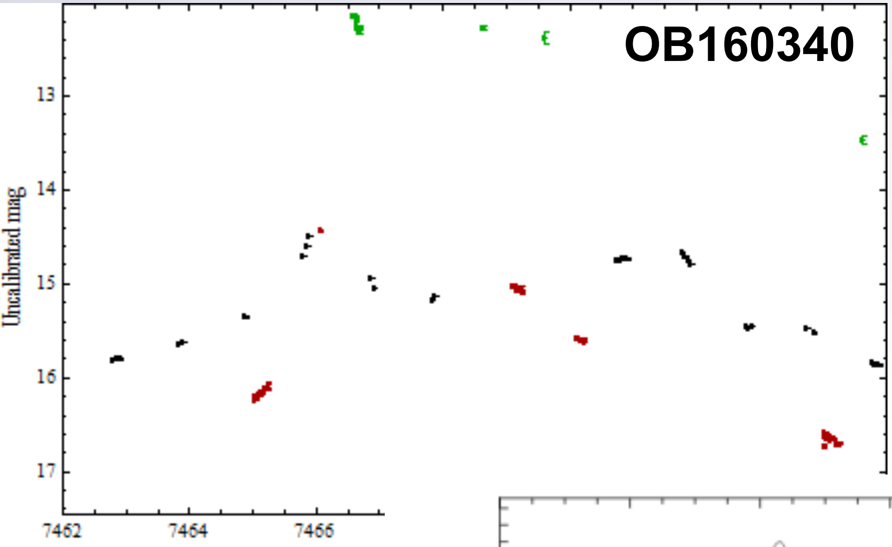


Valerio Bozza

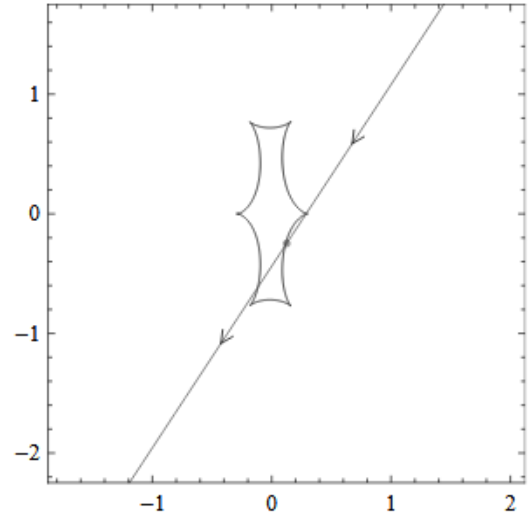
University of Salerno

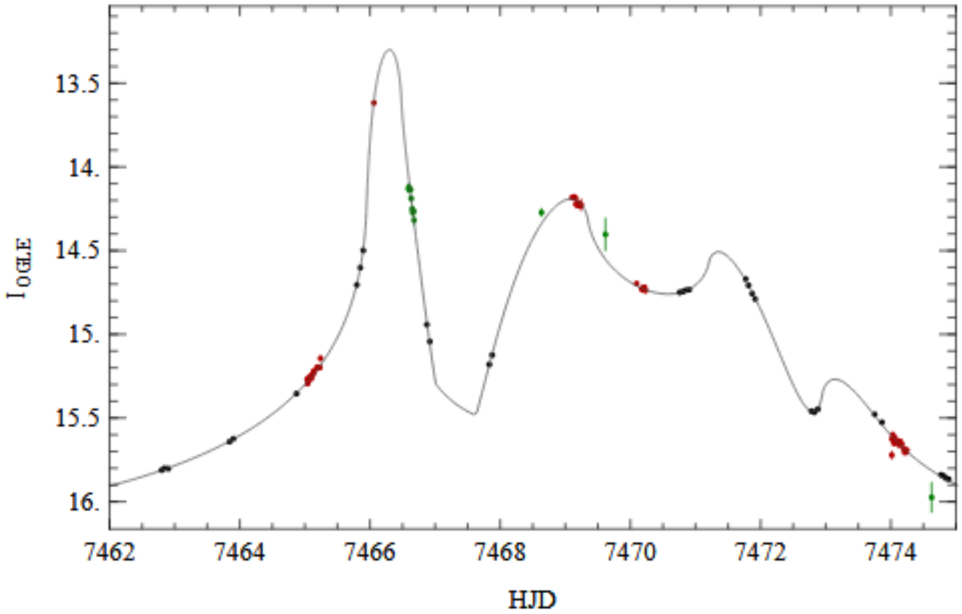
Our goal

- Given the data, we look for a model to explain them



$t_0 = 7467.45$
$t_E = 7.62$
$u_0 = 0.241$
$\alpha = 0.988$
$\rho_* = 0.033$
$s = 0.906$
$q = 0.935$





Summary

- Finding the best model
  - Downhill methods
  - Markov Chain
- Uncertainty assessment
- Degeneracies
- Bayesian analysis
- Initial conditions

Likelihood and χ^2

- Given a model f, the probability that an experiment returns the data y_i with uncertainty σ_i is the **likelihood**:

$$\mathcal{L} = p(y_i \mid f) = \prod_i p_i(y_i \mid f) = N \exp\left[-\frac{1}{2}\sum_i \frac{(y_i - f(t_i))^2}{\sigma_i^2}\right]$$

Independent measures

Gaussian statistical errors

- An estimate of the best model is obtained by maximizing the likelihood, or minimizing the **chi square**:

$$\chi^2 = \sum_i \frac{(y_i - f(t_i))^2}{\sigma_i^2}$$

- The χ^2 is just a function of the model and its parameters.
- Fitting microlensing events is a **minimization problem** for the χ^2.

Fitting microlensing events

- If we are able to calculate the magnification for a **given model** at any times, we can easily evaluate the **corresponding** χ^2.
- Binary microlensing light curves are characterized by a minimum of **7 parameters**.
- In addition, for each dataset we have two calibration parameters: **source and background flux**.

$$y_i = F_* f\left(t_i, \mathbf{p}\right) + F_B$$

- These parameters come linearly and can be found analytically by a **least-squares fit** for any given model.

$$F_* = \frac{\sum \frac{1}{\sigma_i^2} \sum \frac{f_i y_i}{\sigma_i^2} - \sum \frac{f_i}{\sigma_i^2} \sum \frac{y_i}{\sigma_i^2}}{\sum \frac{1}{\sigma_i^2} \sum \frac{f_i^2}{\sigma_i^2} - \left(\sum \frac{f_i}{\sigma_i^2}\right)^2}; \qquad F_B = \frac{\sum \frac{y_i}{\sigma_i^2} \sum \frac{f_i^2}{\sigma_i^2} - \sum \frac{f_i}{\sigma_i^2} \sum \frac{f_i y_i}{\sigma_i^2}}{\sum \frac{1}{\sigma_i^2} \sum \frac{f_i^2}{\sigma_i^2} - \left(\sum \frac{f_i}{\sigma_i^2}\right)^2}$$

- Now we need a minimization algorithm!

Steepest descent

- If χ^2 depends on m parameters $\mathbf{p} = \{p_1, \ldots, p_m\}$, its gradient is

$$\nabla\chi^2(\mathbf{p}) = -2\sum_i \mathbf{J}_i \left[\frac{y_i - f(t_i, \mathbf{p})}{\sigma_i^2}\right]$$

$$\text{where} \quad \mathbf{J}_i = \left(\frac{\partial f_i}{\partial p_1}, \ldots, \frac{\partial f_i}{\partial p_m}\right)$$

- The steepest descent is then implemented by choosing

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \alpha\nabla\chi^2$$

- α is determined by a search along the direction of the gradient.

Gauss-Newton method

- Let us set $\mathbf{p}_{n+1} = \mathbf{p}_n + \boldsymbol{\Delta}$.
- If $\boldsymbol{\Delta}$ is such that $\mathbf{p}_{n+1}$ is a minimum, then

$$0 = \nabla\chi^2\left(\mathbf{p}_n + \boldsymbol{\Delta}\right) = -2\sum_i \mathbf{J}_i\left[\frac{y_i - f\left(t_i, \mathbf{p}_n + \boldsymbol{\Delta}\right)}{\sigma_i^2}\right] \cong$$

$$\cong -2\sum_i \mathbf{J}_i\left[\frac{y_i - f\left(t_i, \mathbf{p}_n\right) - \mathbf{J}_i \cdot \boldsymbol{\Delta}}{\sigma_i^2}\right]$$

- The approximate solution for $\boldsymbol{\Delta}$ is obtained by a linear set of equations

$$\sum_i \mathbf{J}_i\left[\mathbf{J}_i \cdot \boldsymbol{\Delta}\right] = \sum_i \mathbf{J}_i\left[\frac{y_i - f\left(t_i, \mathbf{p}_n\right)}{\sigma_i^2}\right]$$

- Convergence is not guaranteed if we are too far from minimum

Levenberg method

- Interpolates between the two methods, switching from Gauss-Newton to steepest descent when the first fails.
- We modify the normal equations by introducing a **parameter λ**

$$\sum_i \mathbf{J}_i \left[\mathbf{J}_i \cdot \boldsymbol{\Delta} \right] + \lambda \boldsymbol{\Delta} = \sum_i \mathbf{J}_i \left[y_i - f\left(t_i, \mathbf{p}_i \right) \right]$$

- If λ is small, the normal equations work as in Gauss-Newton.
- If λ is large, the new term dominates and **Δ** is rotated toward the steepest descent direction.

Levenberg-Marquardt algorithm

- Steepest descent may be inefficient if there are directions in which χ^2 is **very flat**.
- The final version of the modified normal equations is

$$\sum_i \left[\mathbf{J}_i\left(\mathbf{J}_i \cdot \mathbf{\Delta}\right) + \lambda \left|\mathbf{J}_i\right|^2 \mathbf{\Delta}\right] = \sum_i \mathbf{J}_i\left[y_i - f\left(t_i, \mathbf{p}_i\right)\right]$$

- In Levenberg-Marquardt algorithm, we start from a value of λ close to 1.
- We calculate $\mathbf{\Delta}$; if $\chi^2\left(\mathbf{p}_n + \mathbf{\Delta}\right) < \chi^2\left(\mathbf{p}_n\right)$, we accept the new point $\mathbf{p}_{n+1} = \mathbf{p}_n + \mathbf{\Delta}$ and decrease λ.
- If not, we reject the new point and increase λ.

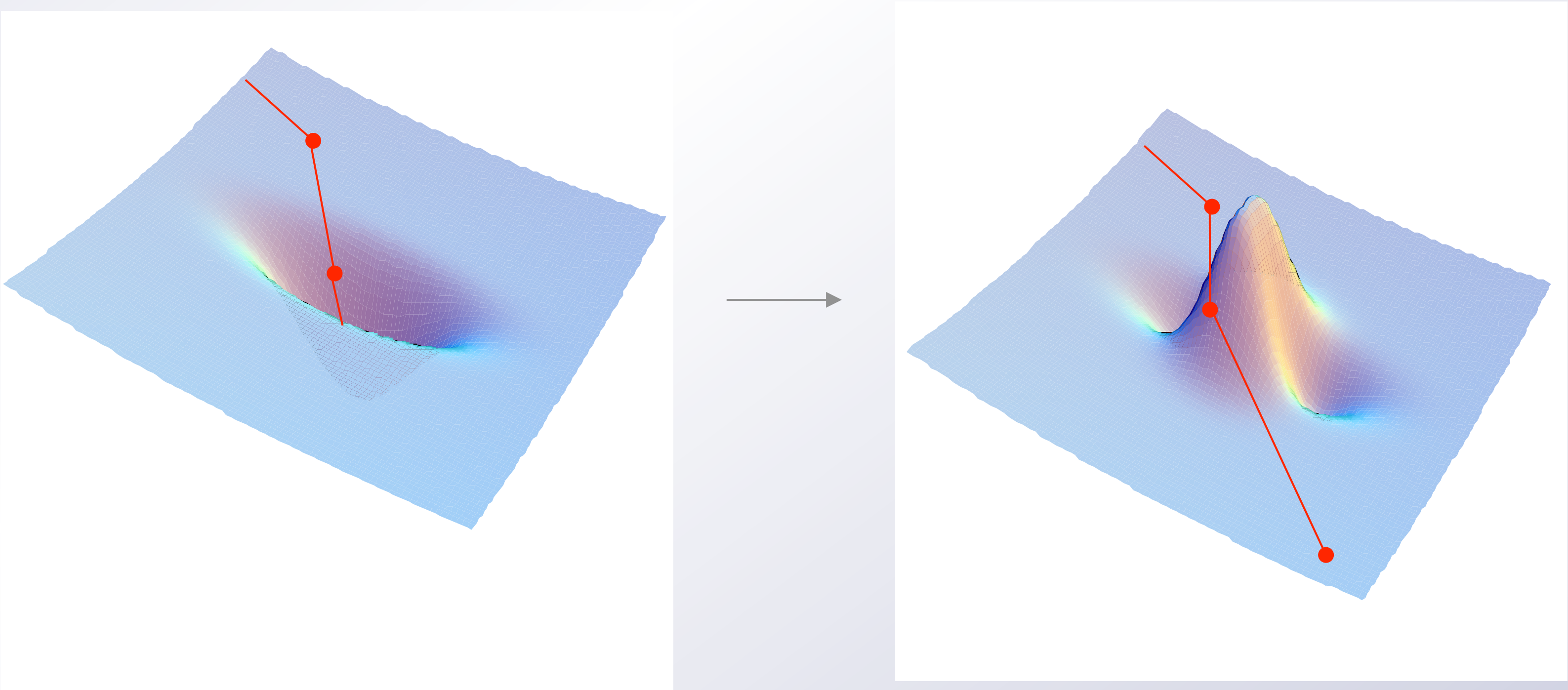
Implementation of Levenberg-Marquardt

- We need to calculate the gradient vector $\mathbf{J}_i = \left(\frac{\partial f_i}{\partial p_1}, \ldots, \frac{\partial f_i}{\partial p_m} \right)$
- The derivatives require the calculation of magnification at two points spaced by dp_i. This is the slowest step.
- The resolution of normal equations can be done by standard Gauss method, Cholesky decomposition…

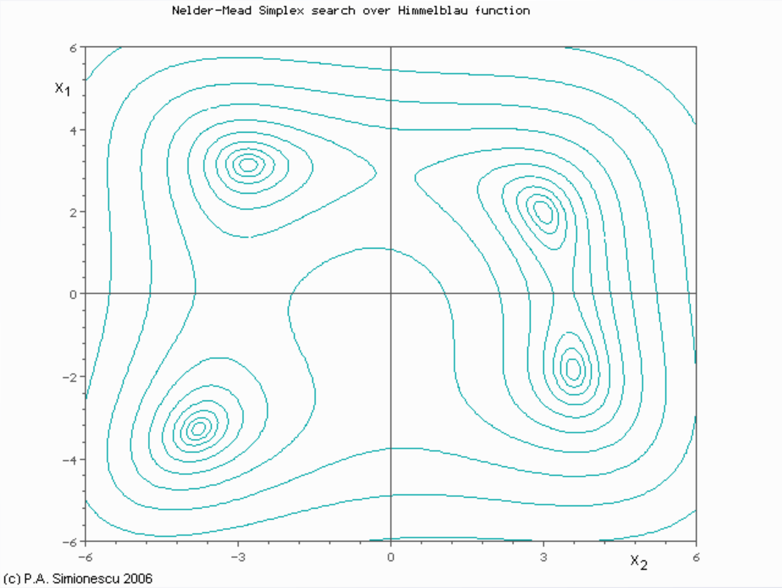
- Levenberg-Marquardt algorithm (nearly) always finds a local minimum.
- It is also very very fast.
- It might get stuck at a local minimum.
- How do we find the best minimum?

Jumping out of minima

- One possibility to enlarge our search is to add a penalty on the χ^2 function.
- Once we find the first minimum, we try to fill it with a bumper and run the fit again.
- If the bumper is small, the fit will still remain in the same dip.
- If the bumper is large enough, the fit will jump out of the hole and discover a different minimum.



Downhill simplex (Nelder-Mead)



- In m dimensions, consider a simplex made of m+1 points $\{\mathbf{x}_1, \ldots, \mathbf{x}_{m+1}\}$
- Let $\mathbf{x}_0$ be the barycenter of the best m points.
- The worst point is replaced by its reflection with respect to $\mathbf{x}_0$:

$$\mathbf{x}_{new} = \mathbf{x}_0 + \gamma\left(\mathbf{x}_0 - \mathbf{x}_{m+1}\right)$$

- There are rules for expansion or contraction by tuning γ.
- No need to calculate gradients.

Differential evolution

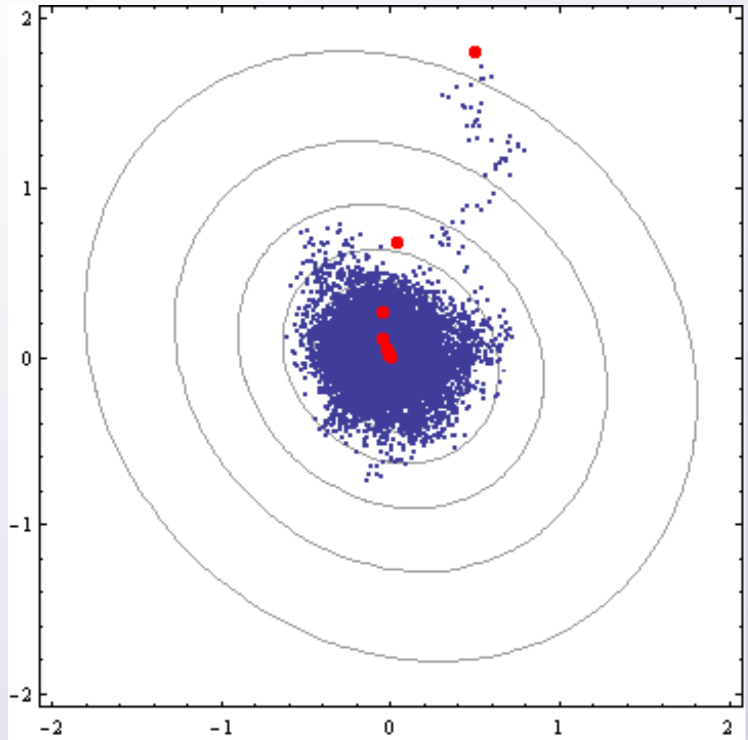
- Start from a population of NP ≥ 4 points ("agents") $\{\mathbf{x}_1, \ldots, \mathbf{x}_{NP}\}$
- For each agent $\mathbf{x}$, pick three more random agents $\boldsymbol{a,b,c}$.
- Generate a new point $\mathbf{y}$ whose components are

  $y_i = a_i + w(b_i - c_i)$ with some probability CR

  $y_i = x_i$ otherwise.
- One random component is always changed.
- If $\chi^2(\mathbf{y}) < \chi^2(\mathbf{x})$ then the new agent replaces the old one.

Markov Chain Monte Carlo

- For a recent review see:
  "Markov Chain Monte Carlo Methods for Bayesian Data Analysis in Astronomy", S. Sharma, arXiv:1706.01629.
- MCMC is **NOT** a minimization algorithm!
- MCMC samples a probability distribution:
  the best model is just a by-product.



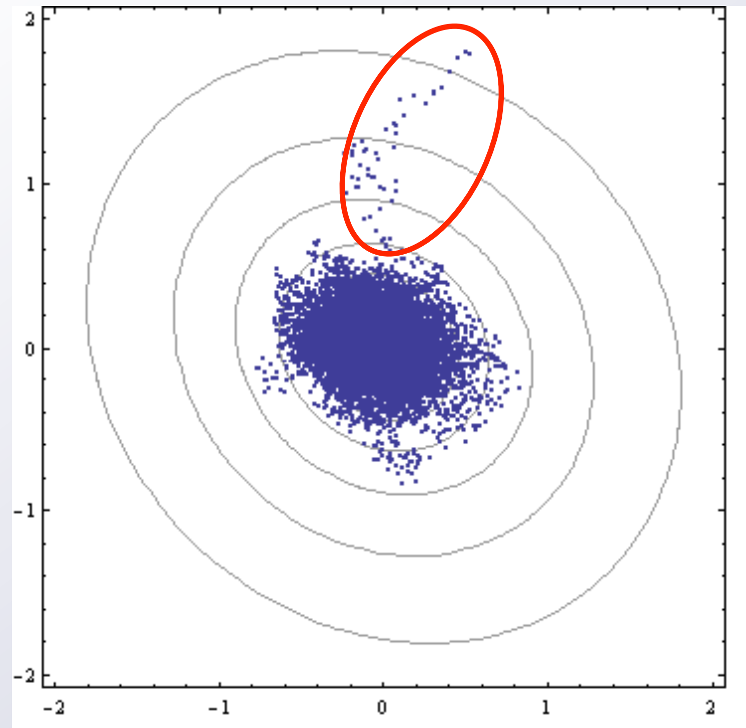
- In this example, after 10000 points, a **Markov chain** finds the best model at accuracy 3×10^{-3}.
- The same accuracy is reached by a **steepest descent** algorithm in 8 steps.

Markov Chain Monte Carlo

- Given the point $\mathbf{x}_n$ in the chain, we randomly draw a candidate new point $\mathbf{y}$ from a **proposal** probability distribution $q(\mathbf{y}|\mathbf{x})$.

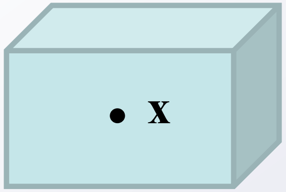
- If $p(\mathbf{y})>p(\mathbf{x})$, we accept the proposal and set $\mathbf{x}_{n+1} = \mathbf{y}$.
- If $p(\mathbf{y})<p(\mathbf{x})$, we accept the proposal with probability $p(\mathbf{y})/p(\mathbf{x})$, otherwise we set $\mathbf{x}_{n+1} = \mathbf{x}_n$. *(Metropolis algorithm)*

- In the limit of large numbers, the chain will become a **representative sampling** of the probability distribution p.
- The "burn-in" must be discarded.
- In our optimization problems, we set $p = \mathcal{L} = \exp(-\chi^2/2)$.



Efficient Markov chains

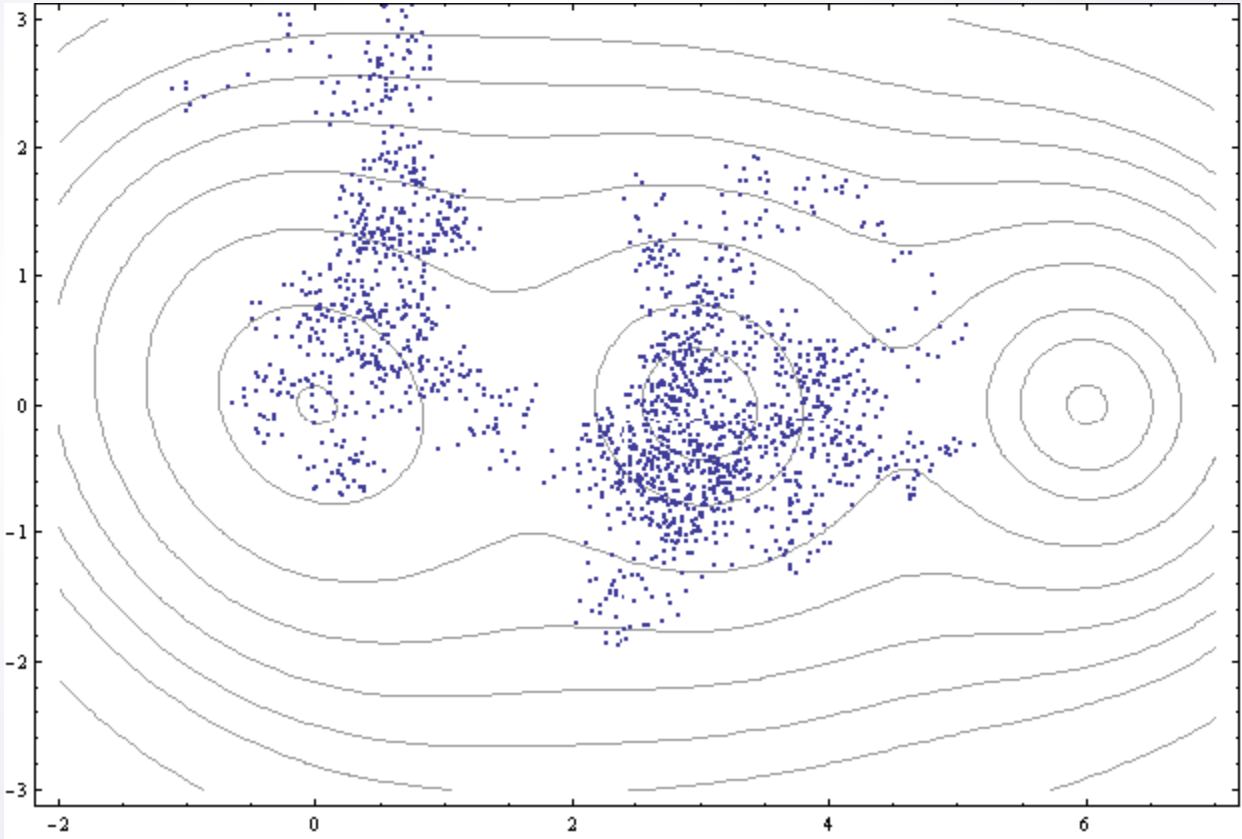
- The proposal probability distribution q($\mathbf{y}|\mathbf{x}$) is crucial to sample the space in the shortest time.
- It is forbidden to change it during the Markov chain.
- We can use a uniform distribution centered on $\mathbf{x}$ within some ranges, a multivariate gaussian or similar.



- The size in each direction can be adapted using the local gradient at the initial conditions.
- A too large q($\mathbf{y}|\mathbf{x}$) will generate very unlikely proposals
- A too small q($\mathbf{y}|\mathbf{x}$) will only sample locally and never reach convergence.
- The **acceptance rate** should be in the range [0.2, 0.6], with a preference for smaller values at large dimensions.
(0.23 is optimal for infinite dimensions)

Convergence

- Markov chains have the ability of jumping out of local minima.
- A Markov chain has converged if, divided into several chunks, each chunk represents a sampling of the same distribution.
- Convergence tests include autocorrelation measures or correlations among several independent chains.



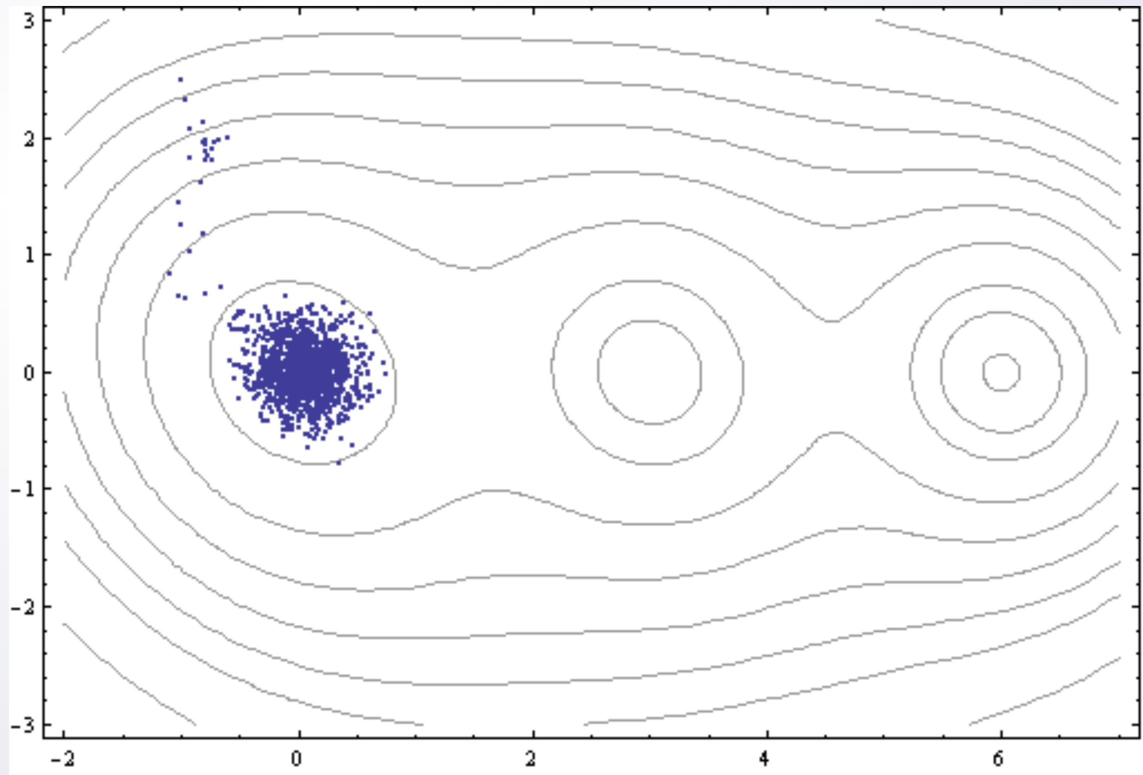
Simulated annealing

- Let us introduce the "**temperature**" T, modifying the probability:

$$p = \exp\left(-\frac{\chi^2}{2T}\right)$$

- At high temperature, all probability ratios tend to 1 and the Markov chain is free to move everywhere.

- The idea (Kirkpatrick et al. 1983) is to start at high temperature to explore the whole parameter space and gradually lower the temperature to pinpoint the best model.



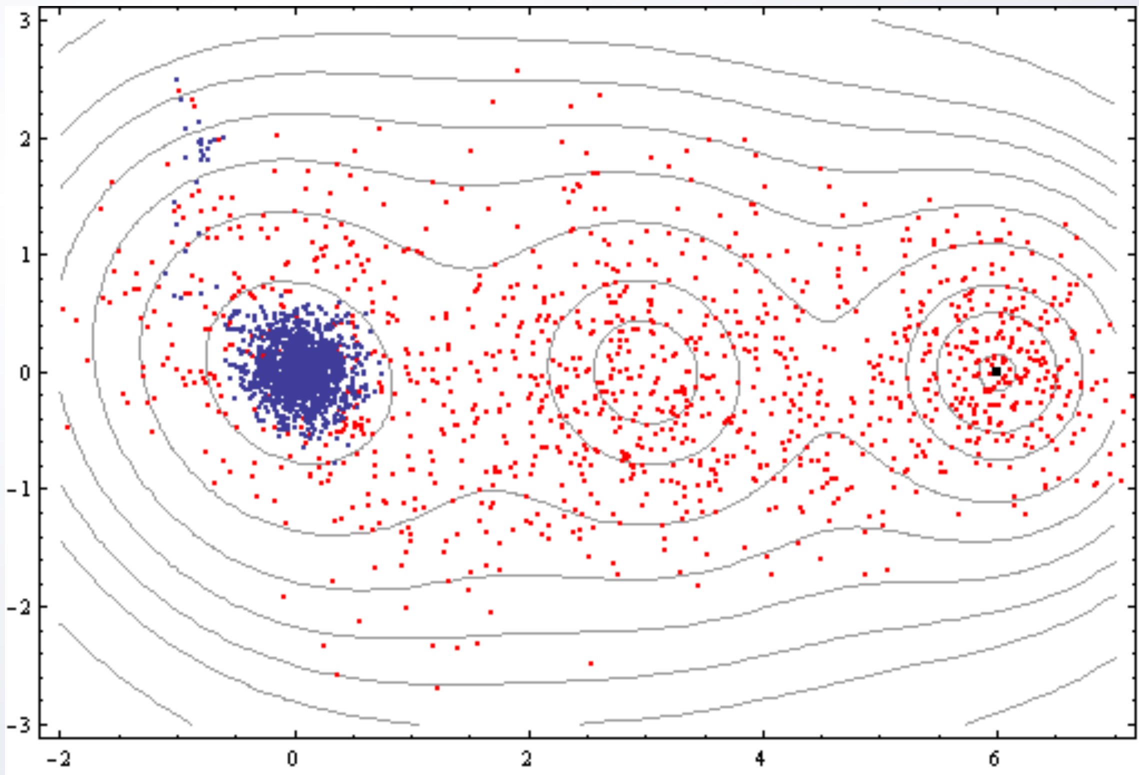
Simulated annealing

- Let us introduce the “**temperature**” T, modifying the probability:

$$p = \exp\left(-\frac{\chi^2}{2T}\right)$$

- At high temperature, all probability ratios tend to 1 and the Markov chain is free to move everywhere.

- The idea (Kirkpatrick et al. 1983) is to start at high temperature to explore the whole parameter space and gradually lower the temperature to pinpoint the best model.



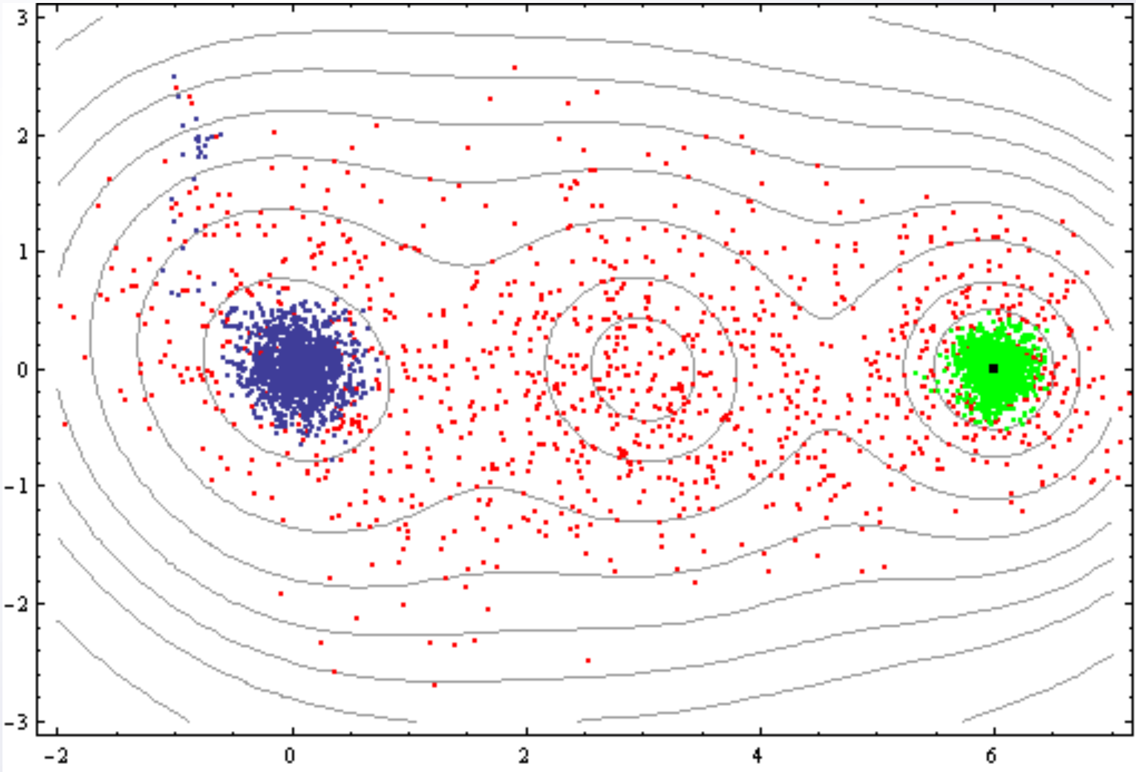
Simulated annealing

- Let us introduce the "**temperature**" T, modifying the probability:

$$p = \exp\left(-\frac{\chi^2}{2T}\right)$$

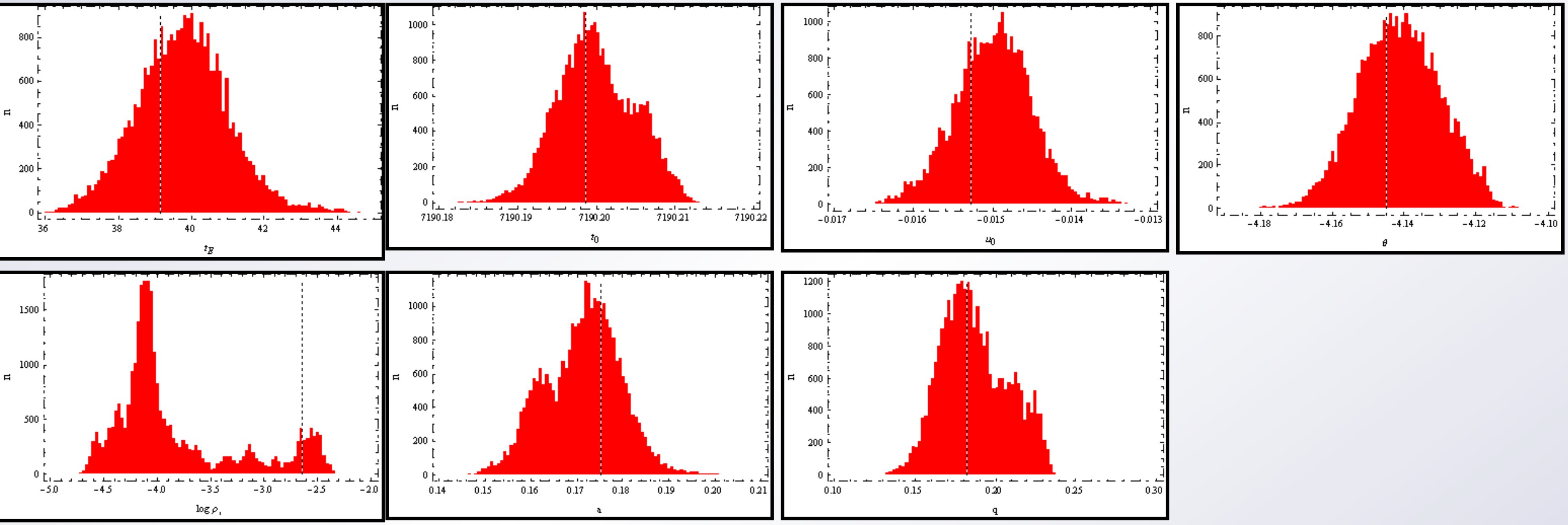
- At high temperature, all probability ratios tend to 1 and the Markov chain is free to move everywhere.

- The idea (Kirkpatrick et al. 1983) is to start at high temperature to explore the whole parameter space and gradually lower the temperature to pinpoint the best model.



Confidence intervals

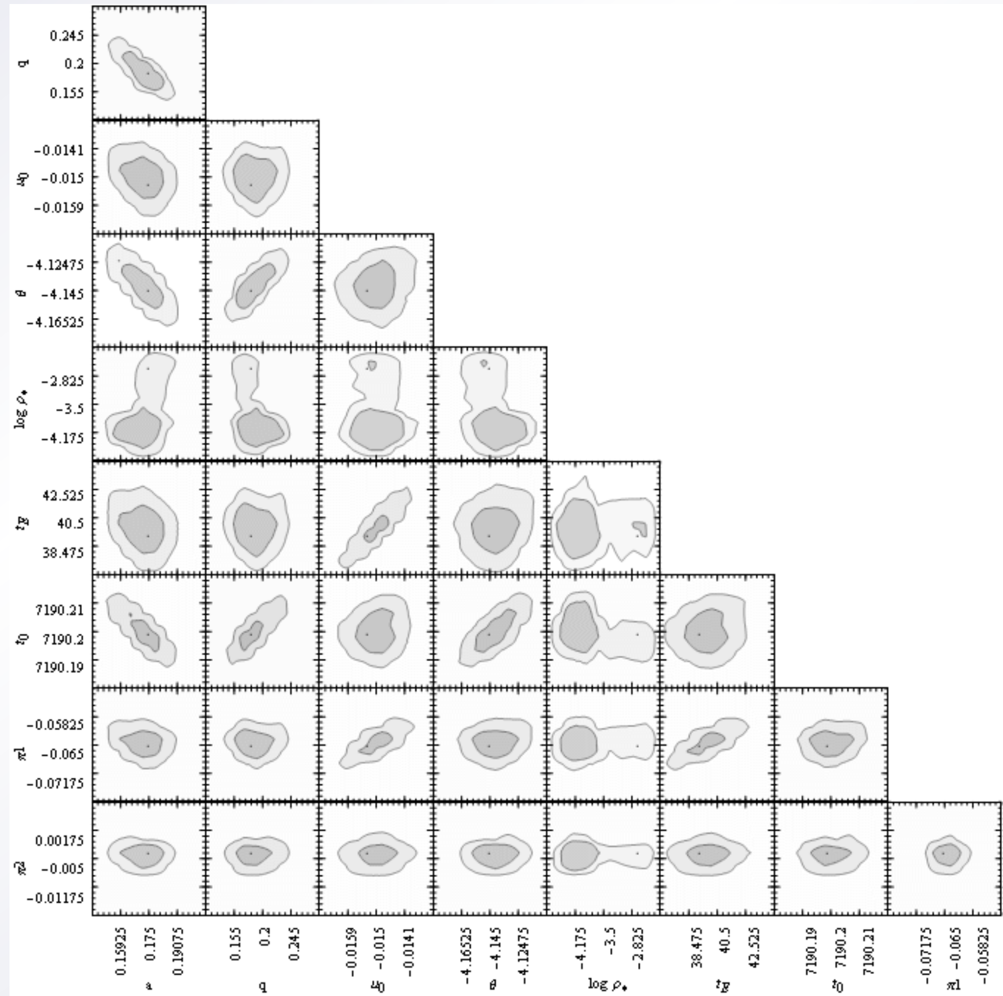
- Once we have sampled our likelihood, we can build **histograms** on any parameters.



- **Confidence intervals** can be obtained:
  1) Sort bins according to their height
  2) Retain higher bins until you reach the desired CL (e.g. 90%)
  3) The CL range is then given by the positions of the two farthest bins on left and right.

Correlation plots

- We can produce density plots on planes defined by any pair of parameters.
- We can define confidence **contours** in the same way.
- This is useful to visualize and detect **degeneracies**.



Fisher and covariance matrices

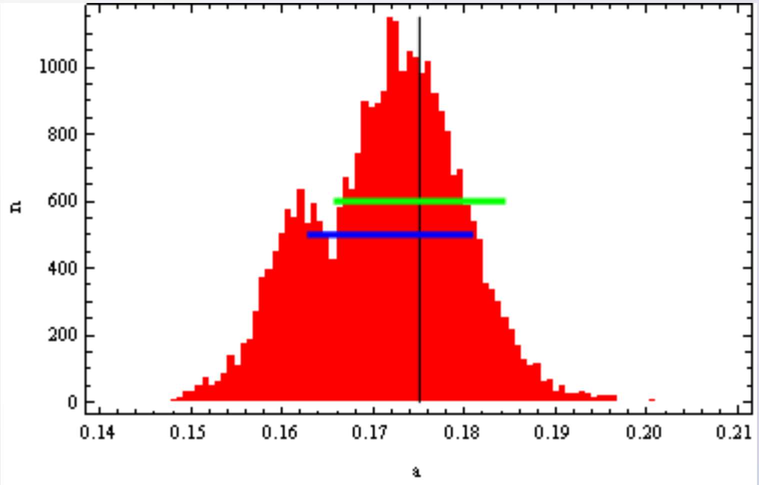
- A common misconception is that MCMC is the only way to obtain the uncertainties in our parameter estimates.
- If you get the best model from other algorithms (e.g. LM), the shape of the minimum is obtained by the **Fisher** matrix

$$F_{mn} = \sum_i \frac{1}{\sigma_i^2} \frac{\partial f(t_i;\mathbf{p})}{\partial p_m} \frac{\partial f(t_i;\mathbf{p})}{\partial p_n}$$

- The **covariance** matrix is just the inverse of the Fisher matrix

$$\mathrm{cov}_{mn} = \left(F^{-1}\right)_{mn}$$

- The variance of each parameter is read along the diagonal of this matrix.



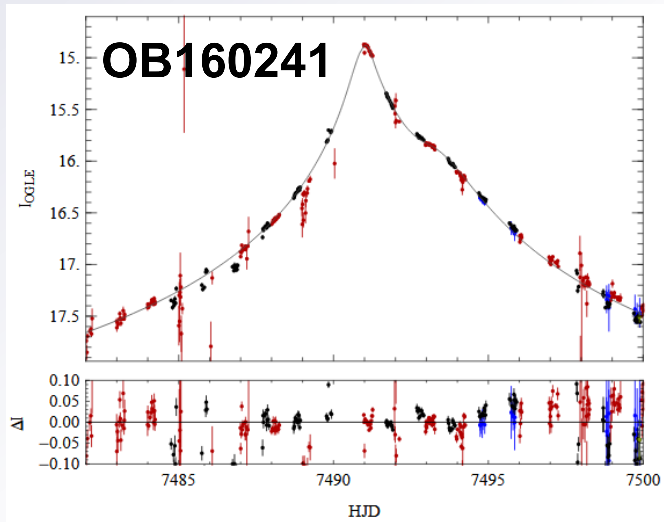
Degeneracies in microlensing

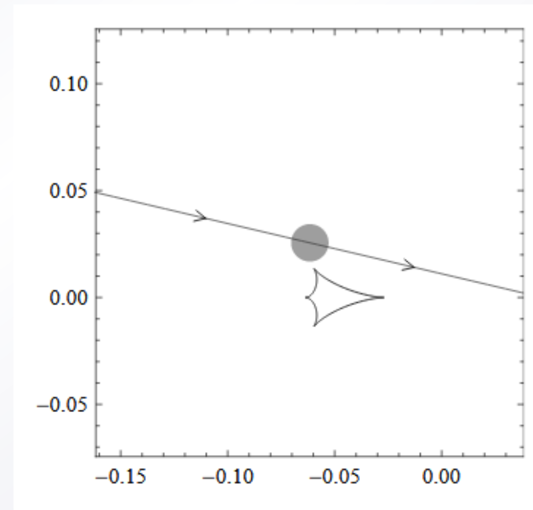
- A degeneracy exists when the same data can be explained by many **different models** with the same likelihood.
- We can have **continuous** degeneracies (e.g. q/s)
- … or **discrete** degeneracies (e.g. wide/close)

- Degeneracies can be “**strong**” i.e. inherent to gravitational lensing physics itself,
- … or “**accidental**” if they arise only because of observational shortcomings (gaps, poor sampling, noise, systematics).

Discrete degeneracies

- **Close/Wide** degeneracy in **planets**

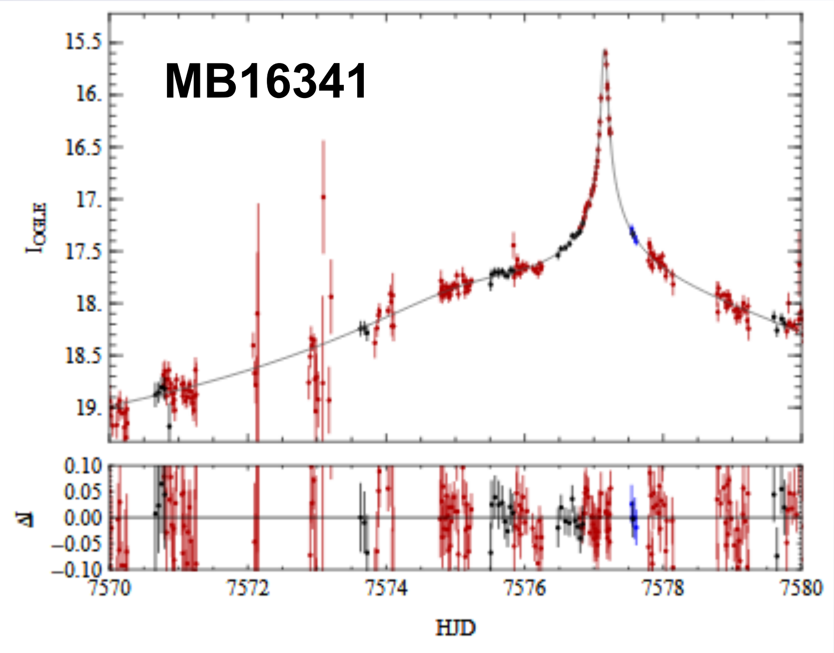


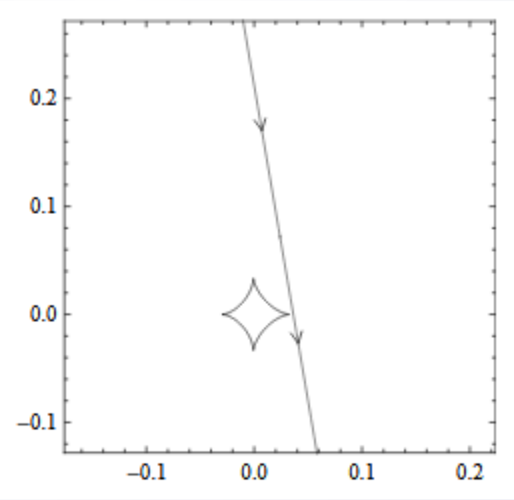


- The central caustic is **invariant** under the transformation $s \leftrightarrow \dfrac{1}{s}$
- All planetary perturbations due to the central caustic suffer from this degeneracy.

Discrete degeneracies

- **Close/Wide** degeneracy in **binaries**

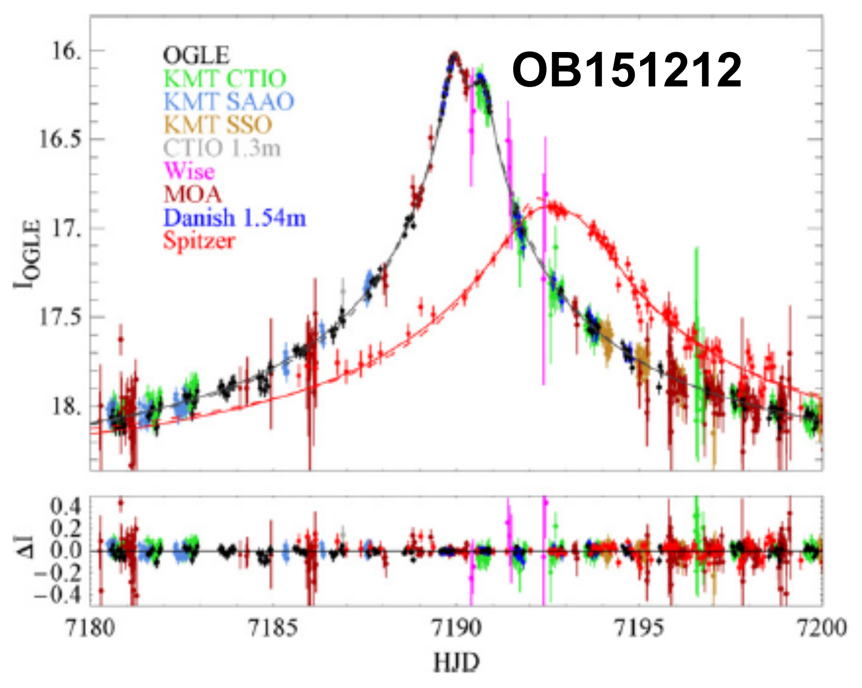




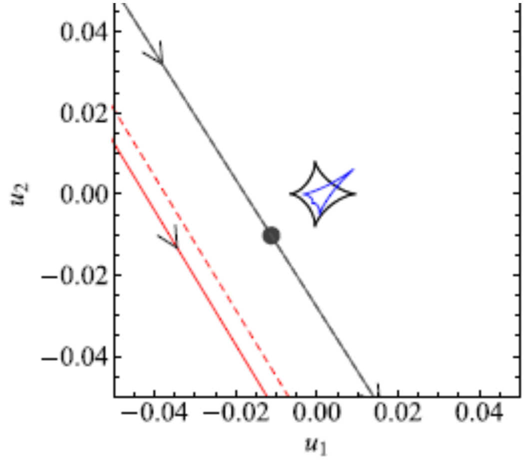
- The **Chang & Refsdal** caustic in the wide regime and the **quadrupole** caustic in the close regime are very similar.
- In addition, the four cusps of a Chang & Refsdal are equivalent (4 possible sub-cases).

Discrete degeneracies

- **Han & Gaudi** (2008) degeneracy
- Light curves with a double peak can be explained by a close approach to a Chang & Refsdal astroidal caustic
- … or by the approach to the back of a central caustic in the planetary regime.

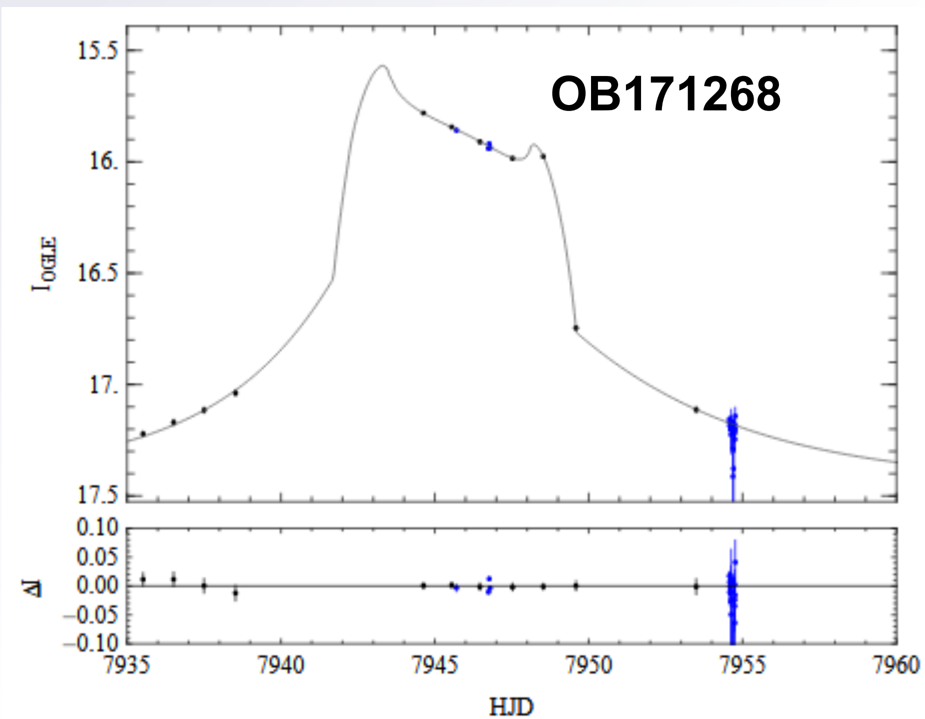


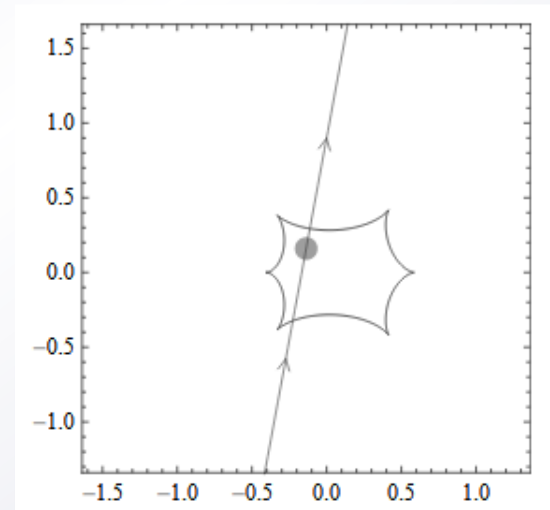
- In either case we have the close/wide sub-cases and all possible cusp approaches for the binary.

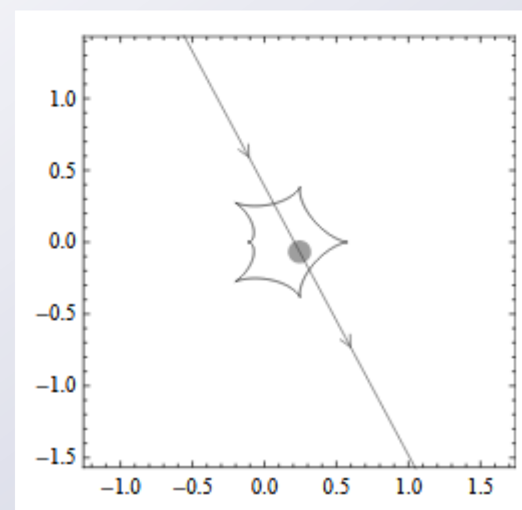


Discrete degeneracies

- **Intermediate** binary degeneracies
  The intermediate binary caustic is very extended. Trajectories crossing different folds may lead to very similar light curves

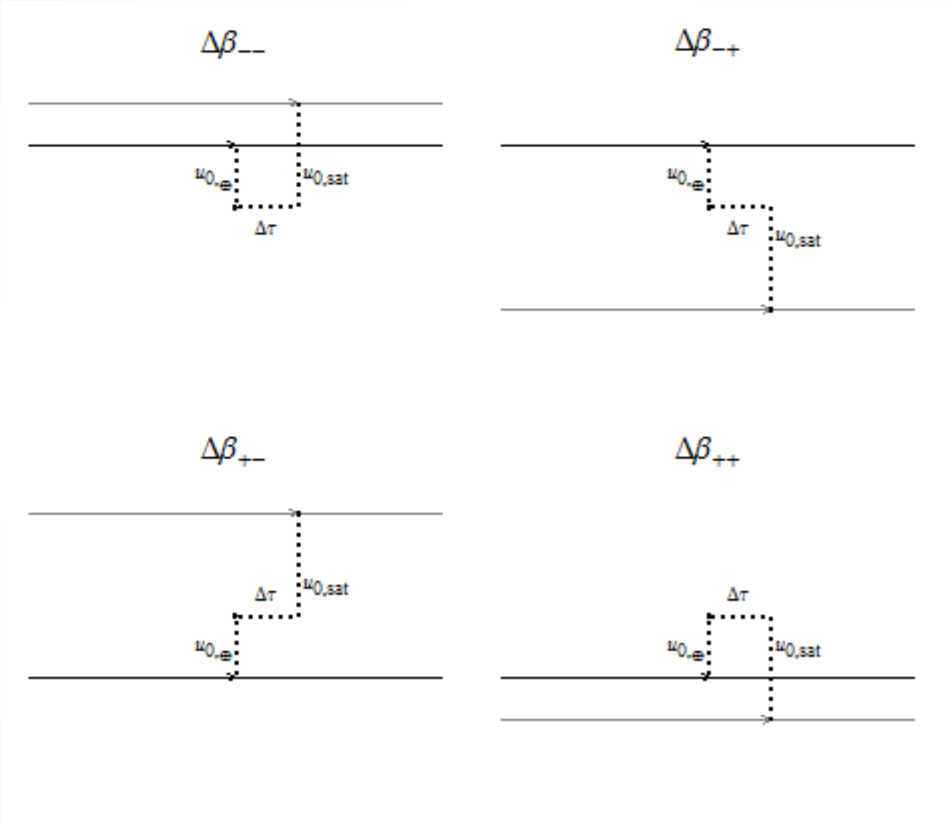






Discrete degeneracies

- **Satellite** degeneracy
  Similarly to what happens for PSPL events, if we have observations from space, we have four options for the signs of $u_{0,\mathrm{Earth}}$ and $u_{0,\mathrm{satellite}}$.

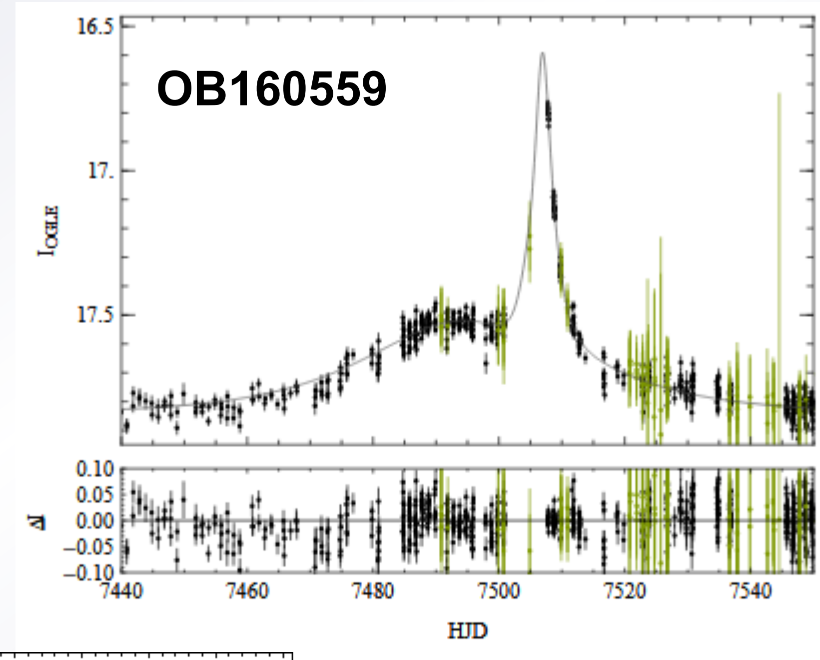


Continuous degeneracies

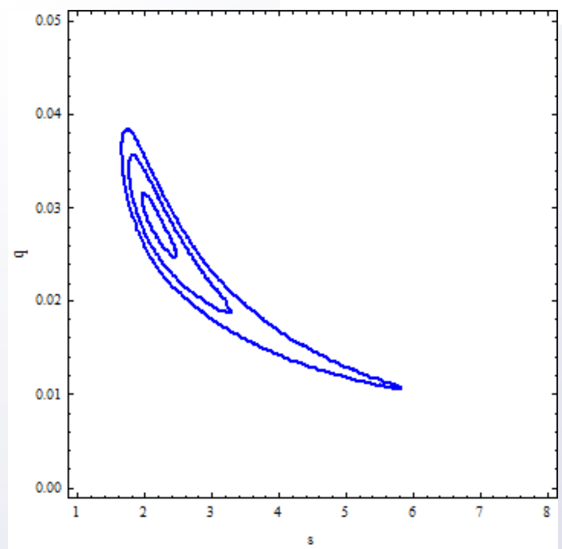
- **s/q** degeneracy
  The size of an astroid caustic depends on the combinations

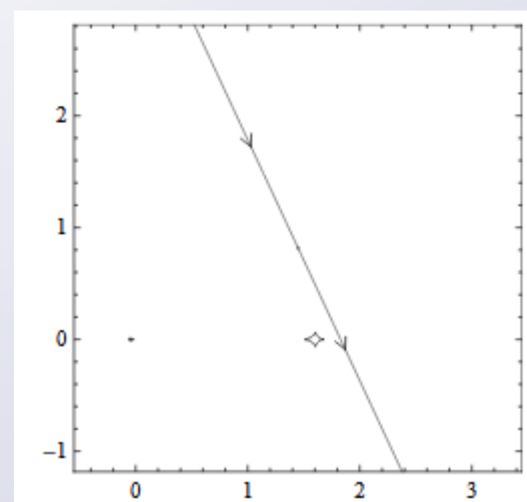
$$\frac{1}{s^2}\sqrt{\frac{q}{(1+q)^3}}$$ Wide regime

$$s^2\frac{q}{(1+q)^2}$$ Close regime



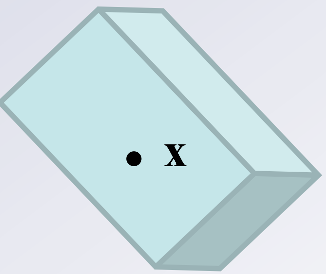
- The mass ratio and separation are highly correlated and poorly known.



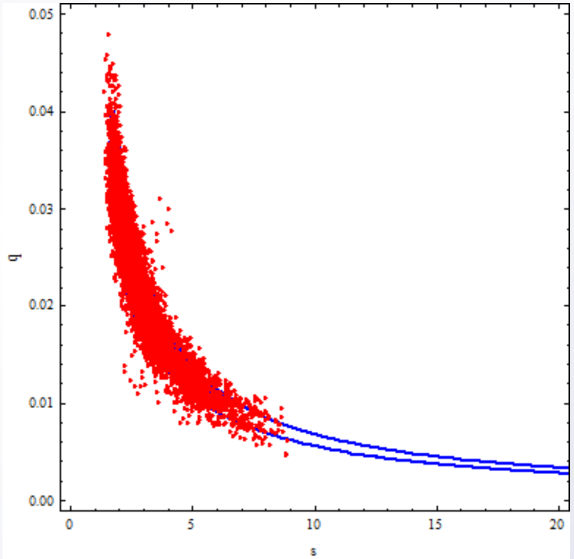


Alternative parameterizations

- The **exploration** of continuous degeneracies is particularly **painful**.



  - We can rotate the box of the proposal distribution (easily achieved if we diagonalize the local Fisher matrix before starting the chain)



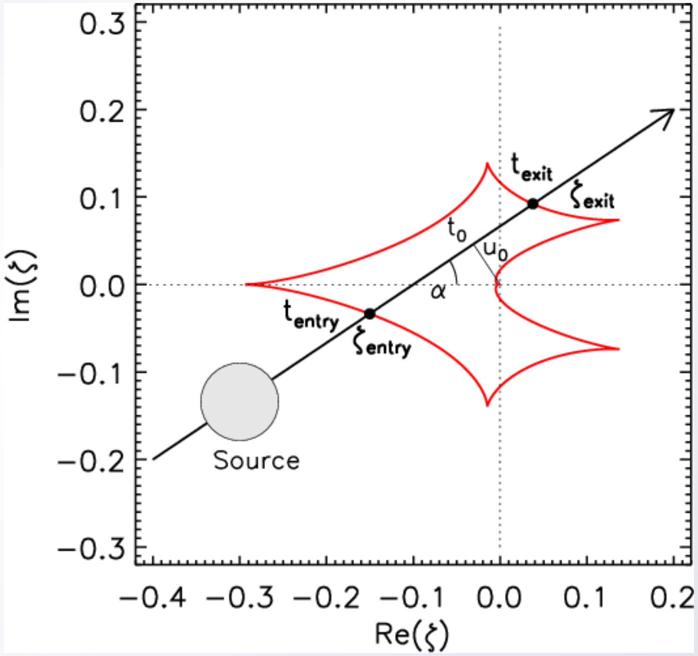
- If the degeneracy is non-linear, choose **new parameters**

$$p_1 = \frac{1}{s^2}\sqrt{\frac{q}{(1+q)^3}}; \quad p_2 = s^2\sqrt{\frac{q}{(1+q)^3}}$$

- Define the origin at the center of the caustic we wish to study.
- Fit the log of some parameters (s, q, ρ, t_E).
- Use other combinations that are clearly established by the data (e.g. source crossing time t_*, time of caustic crossing, …)

Cassan parameters

- Cassan (2008) proposed to use the **curvilinear abscissa** along the caustic.
- u_0, α, t_0, t_E are replaced by t_{entry}, s_{entry} , t_{exit} , s_{exit}.



Walking through degeneracies

- In general, distinct features in the lightcurve occurring at definite times (caustic crossing) **couple** the **Einstein time** to the (s,q) values.
- This mitigates the degeneracy between t_E and u_0 plaguing the PSPL.
- For the same reason, different models may predict very different t_E and thus very different **blending** ratios.
- Typically, planetary models mimicking binary models come at negative blending.
- Other hints may come from unlikely source radii or unlikely Einstein times.
- How do we quantify unlikeliness?

Bayes' theorem

- For all parameters we can define an expected range of possible values.
- A **uniform prior** can be easily implemented by requiring that the proposal point is within the prior.

- However, we may wish to use the information coming from previous studies to decide which model is more likely (stellar luminosity and mass functions, spatial distributions and velocities)
- This information typically comes in the form of (**prior**) **distributions**.

- Bayes theorem states that the **posterior probability** is the product of the likelihood from the data with the prior expectations:

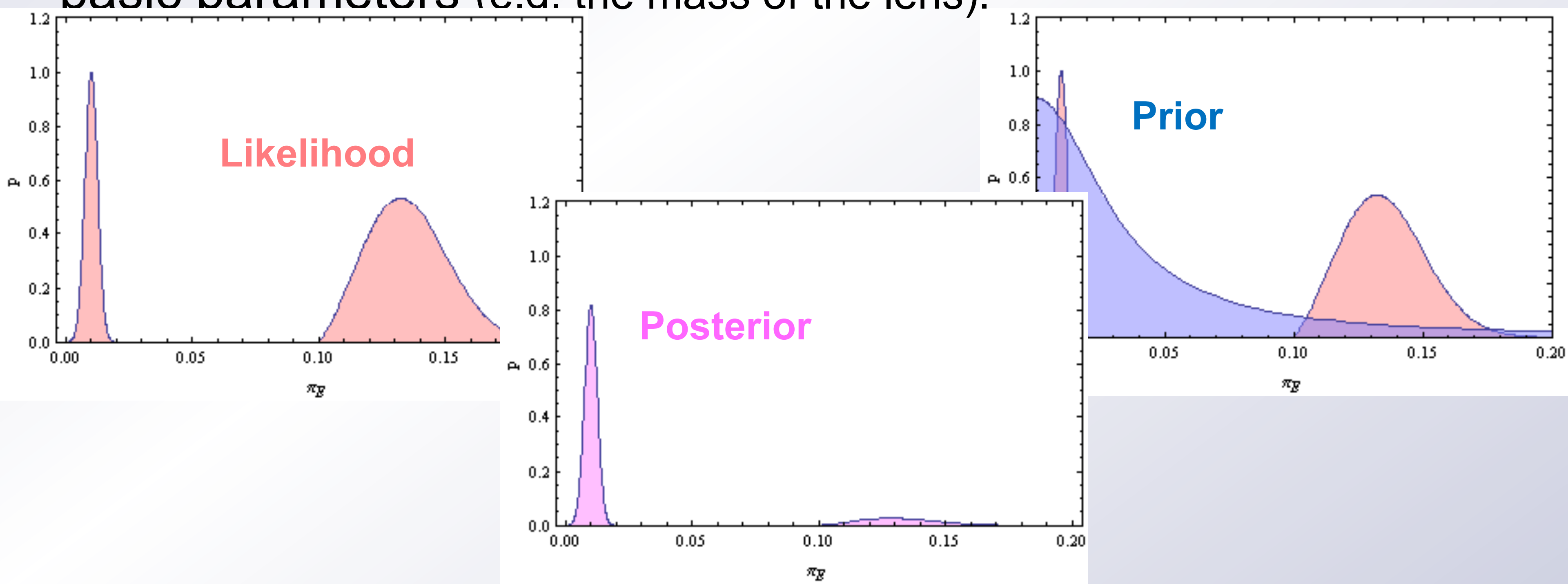
$$p(\mathbf{p} \mid y_i) = \frac{p(y_i \mid \mathbf{p})\,p(\mathbf{p})}{p(y_i)}$$

Bayes in MCMC

$$p(\mathbf{p} \mid y_i) = \frac{p(y_i \mid \mathbf{p})\, p(\mathbf{p})}{p(y_i)}$$

- In our MCMC we just have to sample the product $\exp\left(-\frac{\chi^2}{2}\right) p(\mathbf{p})$

- The **normalization** $p(y_i)$ cancels if we are only interested in relative posterior probabilities (ratios).
- Note that the priors may be distributions on combinations of the basic parameters (e.g. the mass of the lens).



Priors in microlensing

- Microlensing events are normally occurring on source stars in the bulge lensed by stars in the disk/bulge.
- These objects follow some spatial distributions, mass, luminosity and velocity functions.
- In order to use Bayesian approach in microlensing we need a **Galactic model**.
- "Stochastic distributions of lens and source properties for observed galactic microlensing events", Dominik (2006).
- "A synthetic view on structure and evolution of the Milky Way", Robin et al. (2003) (Besançon model)
- "Stellar Contribution to the Galactic Bulge Microlensing Optical Depth", Han & Gould (2003)
- Another combination of models is in Bennett et al. (2008)
- **Blending** light gives a further constraint (see Beaulieu's talk).

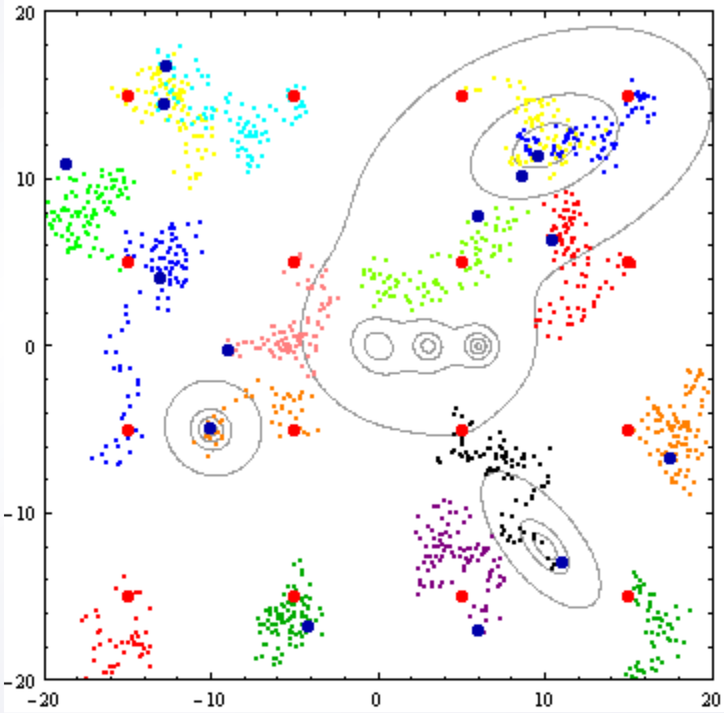
Initial conditions

- Microlensing **parameter space** is huge and full with local χ^2 minima.
- If we start from an arbitrary initial condition we would seldom end in the global minimum.

- We need to **explore** all the relevant parameter space and make sure we find the true best model(s).
- Two ways:
  - Grid search
  - Template library

Grid search

- We may define a **grid** in the parameter space and start fits from all points.

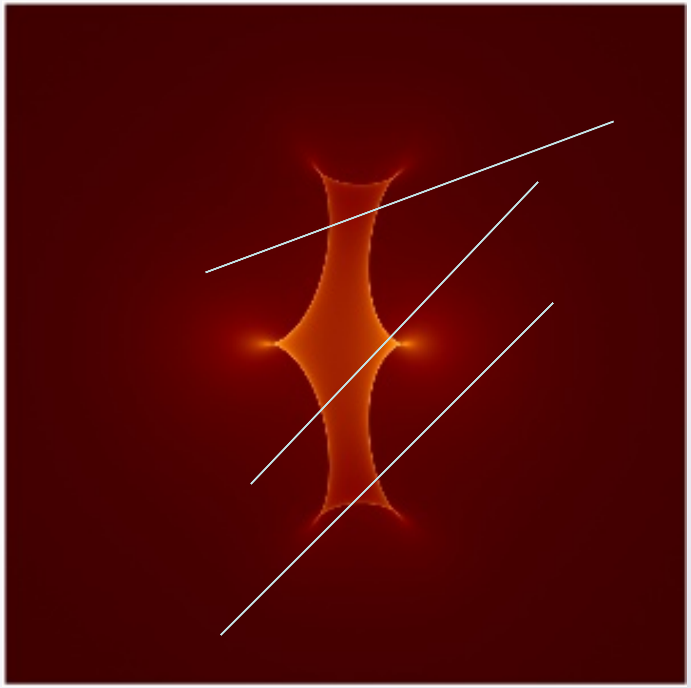


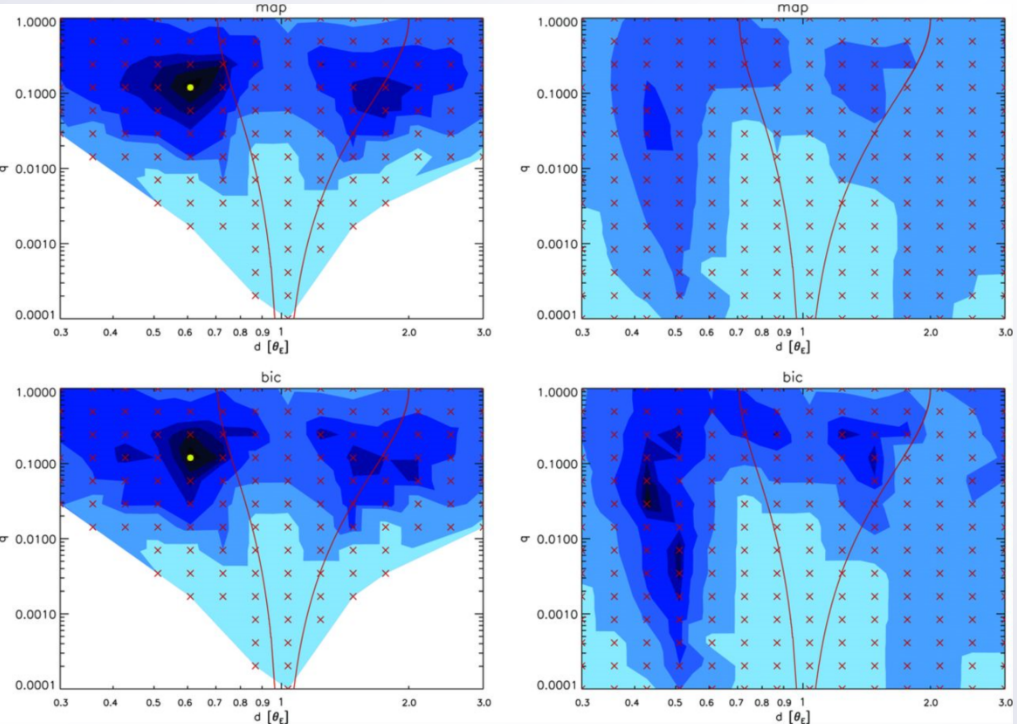
- Many fits will just never converge
- Many fits will end up in the same minima.
- Many minima will be missed.

- A **too coarse** grid may miss possible candidate models.
- A **too dense** grid has many redundant or useless fits.

Two-steps grid search

- Inverse-ray-shooting codes may keep **(s,q) fixed** in a first search, so as to use the same magnification map.
- Once these preliminary models are found, we can run a **full fit** including (s,q).

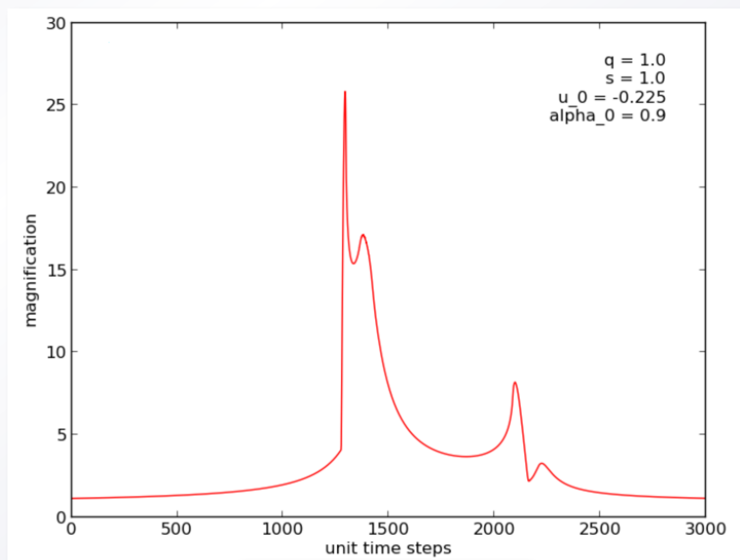




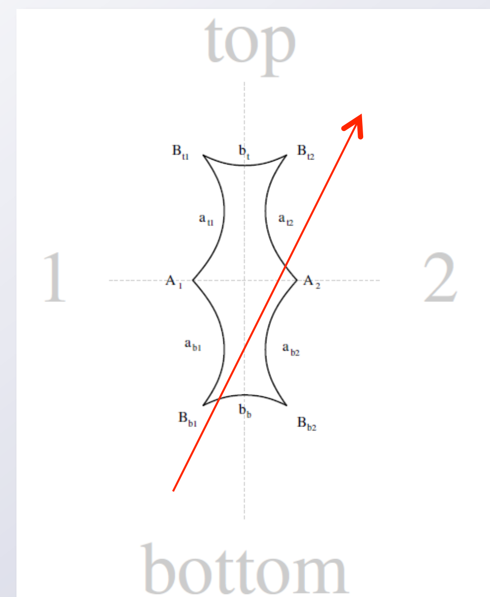
- Codes for a full Bayesian approach along these lines are available (ML/MAP/BIC) (Kains et al. 2012)

Template libraries

- It would be much more efficient to start the fit from an initial condition that resembles our data.
- We need to build a **library of light curves** covering all possible morphologies (Di Stefano & Mao 1996, Night et al. 2005).
- The most systematic attempt has found **73** different morphologies out of **232** regions in the parameter space (Liebig et al. 2015)
- It was limited to equal-mass binaries!



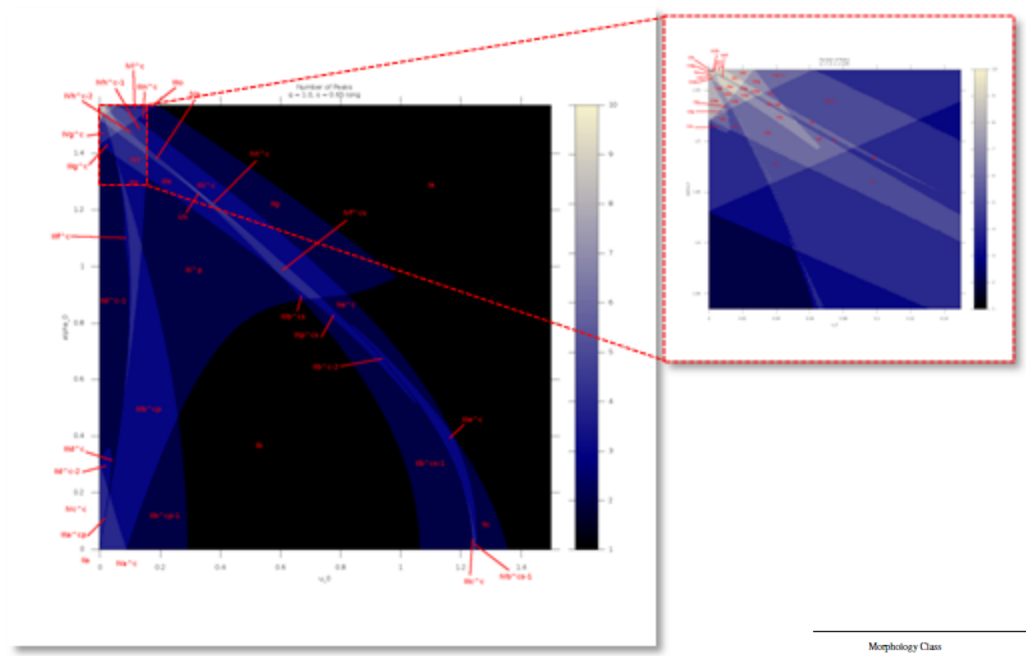
$\mathrm{F}\text{-}\bar{\mathrm{F}}\text{-}\mathrm{F}\ \bar{\mathrm{C}}$



$[b_b\ a_{b1}\ a_{t2}]\ A_2$

Template libraries

- Light curves are classified according to the number and nature of their peaks (fold crossing, cusp approach, …)

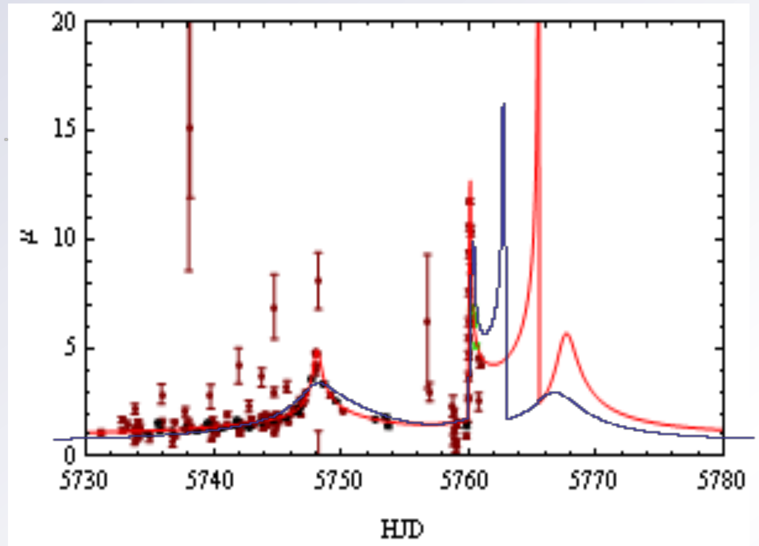


- Regions in the parameter space are identified after a scansion. (Liebig et al. 2015)

	Morphology Class	Close	Intermediate	Wide
I	C	outside caustics between caustics	outside caustics	outside caustics between caustics
II	F-F	$[a_{tp1}a_{tp2}]$, $[a_{tx1}a_{tx2}]$, $[a_{bp1}a_{tp2}]$, $[a_{tx1}b_t]$, $[a_{bp1}a_{tp1}]$	$[a_{t1}a_{t2}]$, $[a_{b1}a_{t2}]$, $[a_{t1}b_t]$, $[a_{b1}b_t]$, $[b_bb_t]$	$[a_{b1}b_{t1}]$, $[b_{b1}b_{t1}]$
	Ĉ Ĉ	$B_{t1}B_{t2}$, A_1B_{t1}, A_1C_{tp}, A_1C_{tx}, A_1B_{t2}, $C_{tx}B_{t2}$	$B_{t1}B_{t2}$, A_1B_{t1}	$B_{t1}B_{t2}$, A_1B_{t1}, $B_{b1}D_1$, $B_{b1}B_{t2}$
	C-C	$[A_1A_2]$	$[A_1A_2]$	-
	C-F	$[a_{tx1}B_{t2}]$, $[A_1a_{tp2}]$	$[a_{t1}B_{t2}]$, $[A_1a_{t2}]$, $[B_{b1}b_t]$, $[A_1a_{t2}]$	$[B_{b1}b_{t1}]$
III	Ĉ F-F	$A_1[a_{tp1}a_{tp2}]$, $A_1[a_{bp1}a_{tp2}]$, $[a_{tx1}b_t]B_{t2}$, $A_1[a_{tx1}b_t]$, $[a_{bp1}a_{tp2}]B_{t2}$, $[a_{bp1}a_{tp1}]B_{t1}$, $[a_{tx1}a_{tx2}]B_{t2}$, $[a_{tx1}a_{tp2}]B_{t2}$, $[a_{bp1}a_{tp2}]C_{tp}$, $[a_{bp1}a_{tp1}]C_{tp}$, $[a_{bp1}a_{tp1}]B_{t2}$	$A_1[a_{t1}a_{t2}]$, $A_1[a_{b1}a_{t2}]$, $[a_{t1}b_t]B_{t2}$, $A_1[a_{t1}b_t]$, $[a_{b1}a_{t2}]B_{t2}$, $[a_{b1}b_t]B_{t2}$, $B_{b1}[a_{b1}b_t]$, $[a_{b1}a_{t1}]B_{t1}$, $[a_{t1}a_{t2}]B_{t2}$, $[a_{b1}b_t]B_{t2}$, $B_{b1}[b_bb_t]$	$[a_{t1}b_{t1}]B_{t2}$, $A_1[a_{t1}b_{t1}]$, $B_{b1}[a_{b1}b_{t1}]$, $[a_{b1}a_{t1}]B_{t1}$, $[a_{b1}b_{t1}]B_{t2}$, $B_{b1}[b_{b1}b_{t1}]$, $A_1[b_{b1}b_{t1}]$, $[a_{b1}b_{b1}]B_{t2}$
	F-F-F	$[a_{tx1}b_ta_{tx2}]$, $[a_{bp1}a_{tp1}a_{tp2}]$	$[a_{t1}b_ta_{t2}]$, $[a_{b1}a_{t1}a_{t2}]$, $[a_{b1}a_{t1}b_t]$, $[b_ba_{b1}b_t]$, $[a_{b1}a_{t2}b_t]$, $[a_{b1}b_ta_{t2}]$	$[a_{b1}a_{t1}b_{t1}]$, $[a_{b1}b_{b1}b_{t1}]$
	Ĉ F-C	$[A_1a_{tp2}]A_2$, $C_{bp}[abp1C_{tp}]$	$[A_1a_{t2}]A_2$, $A_1[a_{t1}B_{t2}]$, $[A_1b_t]B_{t2}$	$[A_1b_{t1}]B_{t2}$
	Ĉ Ĉ Ĉ	$B_{b1}A_1B_{t1}$, $A_1C_{tx}B_{t2}$	$B_{b1}A_1B_{t1}$	$B_{b1}A_1B_{t1}$, $B_{b1}D_1B_{t2}$, $A_1B_{t1}B_{t2}$, $B_{b1}A_1B_{t2}$
	C-F-F	-	$[A_1b_ta_{t2}]$	-
	Ĉ C-F	$A_1[C_{tx}b_t]$	-	-
IV	Ĉ F-F Ĉ	$A_1[a_{tp1}a_{tp2}]A_2$, $A_1[a_{bp1}a_{tp2}]A_2$, $A_1[a_{tx1}a_{tx2}]B_{t2}$, $B_{b1}[a_{bp1}a_{tp2}]B_{t2}$, $B_{b1}[a_{bp1}b_{tp1}]B_{t1}$, $C_{bp}[a_{bp1}a_{tp2}]C_{tp}$	$A_1[a_{t1}a_{t2}]A_2$, $A_1[a_{b1}a_{t2}]A_2$, $A_1[a_{t1}b_t]B_{t2}$, $A_1[a_{t1}a_{t2}]B_{t2}$, $B_{b1}[a_{b1}a_{t2}]B_{t2}$, $B_{b1}[a_{b1}a_{t1}]B_{t1}$, $A_1[a_{b1}b_t]B_{t2}$, $B_{b1}[a_{b1}b_t]B_{t2}$	$A_1[a_{t1}b_{t1}]B_{t2}$, $B_{b1}[b_{b1}b_{t1}]B_{t2}$, $B_{b1}[a_{b1}a_{t1}]B_{t1}$, $A_1[a_{b1}b_{t1}]B_{t2}$
	F-F F-F	$[a_{tx1}b_t][b_ta_{tx2}]$, $[a_{bp1}a_{tp1}][a_{tp1}a_{tp2}]$, $[a_{bp1}a_{tp1}][a_{tx1}b_t]$	$[a_{t1}b_t][b_ta_{t2}]$, $[a_{b1}a_{t1}][a_{t1}a_{t2}]$, $[a_{b1}a_{t1}][a_{t1}b_t]$, $[b_ba_{b1}][a_{b1}b_t]$, $[a_{b1}a_{t2}][a_{t2}b_t]$, $[a_{b1}b_t][b_ta_{t2}]$	$[a_{t1}b_{t1}][b_{t2}a_{t2}]$, $[a_{b1}a_{t1}][a_{t1}b_{t1}]$, $[a_{b1}b_{b1}][b_{b1}b_{t1}]$, $[a_{b1}b_{b1}][b_{t2}a_{t2}]$
	Ĉ F-F-F	$[a_{bp1}a_{tp1}a_{tp2}]B_{t2}$, $A_1[a_{tx1}a_{tx2}b_t]$	$[a_{b1}a_{t1}a_{t2}]B_{t2}$, $[a_{b1}a_{t1}b_t]B_{t2}$, $B_{b1}[a_{b1}a_{t1}b_t]$, $A_1[a_{t1}b_ta_{t2}]$, $A_1[a_{b1}b_ta_{t2}]$, $B_{b1}[a_{b1}a_{t2}b_t]$	$B_{b1}[a_{b1}a_{t1}b_{t1}]$, $[a_{b1}b_{b1}b_{t1}]B_{t2}$
	F-F-F-F	-	$[b_ba_{b1}a_{t1}b_t]$, $[a_{b1}a_{t1}a_{t2}b_t]$, $[b_ba_{b1}a_{t2}b_t]$	-
	C-F F-F	-	$[a_{b1}a_{t1}][a_{t1}B_{t2}]$, $[A_1b_t][b_ta_{t2}]$	$[A_1b_{t1}][b_{t2}a_{t2}]$
	F-F Ĉ Ĉ	$[a_{bp1}a_{tp1}]C_{tp}B_{t2}$, $[a_{bp1}a_{tp1}]C_{tx}B_{t2}$, $A_1C_{tx}[a_{tx1}b_t]$, $A_1C_{tx}[a_{tx2}b_t]$, $[a_{bp1}a_{tp1}]C_{tx}B_{t1}$	-	$[a_{b1}b_{b1}]D_1B_{t2}$
	C-C C-C	-	-	$[A_1D_1][D_2A_2]$
	F-F C-F	$[a_{bp1}a_{tp1}][C_{tx}b_t]$	-	-
	C-F Ĉ Ĉ	-	-	$[A_1b_{t1}]D_1B_{t2}$
	Ĉ F-C Ĉ	$C_{bp}[a_{bp1}C_{tp}]B_{t2}$	-	-

Matching a template to data

- **Peaks in the dataset** must be identified and ranked according to their prominence.



- The two most prominent peaks must be **matched** to the most prominent peaks in the template.
- We get (s,q,u_0,α,ρ) from the template.
- (t_0,t_E) are obtained by the peak matching.

- If there is only one peak, the **anomaly time** can be taken as the position of the second peak.

RTModel

- RTModel (http://www.fisica.unisa.it/GravitationAstrophysics/RTModel.htm) is an **automatic platform** for real-time modeling.
- It takes data and anomaly alerts from ARTEMiS (http://www.artemis-uk.org/).
- It uses matching from a library of 244 templates.
- For each initial condition, the Levenberg-Marquardt fit is repeated five times using the bumpers method.
- The calculation of the magnification is done by VBBinaryLensing.
- All models found are ranked by their χ^2.
- Duplicates are removed if they fall within the same covariance ellipsoid.
- Models are posted on a public webpage automatically.
- RTModel runs on a 8-core workstation taking 2 hours per event.

Outlook

- Higher order effects (parallax, orbital motion) may dramatically increase the number of light curve morphologies.

- Grid searches in too many dimensions are unfeasible.
- Template libraries require a long construction.

- Similar issues hold for triple and multiple lenses.

- In view of WFIRST, we need to improve our automatic modeling capabilities.
  pyLIMA, MulensModel