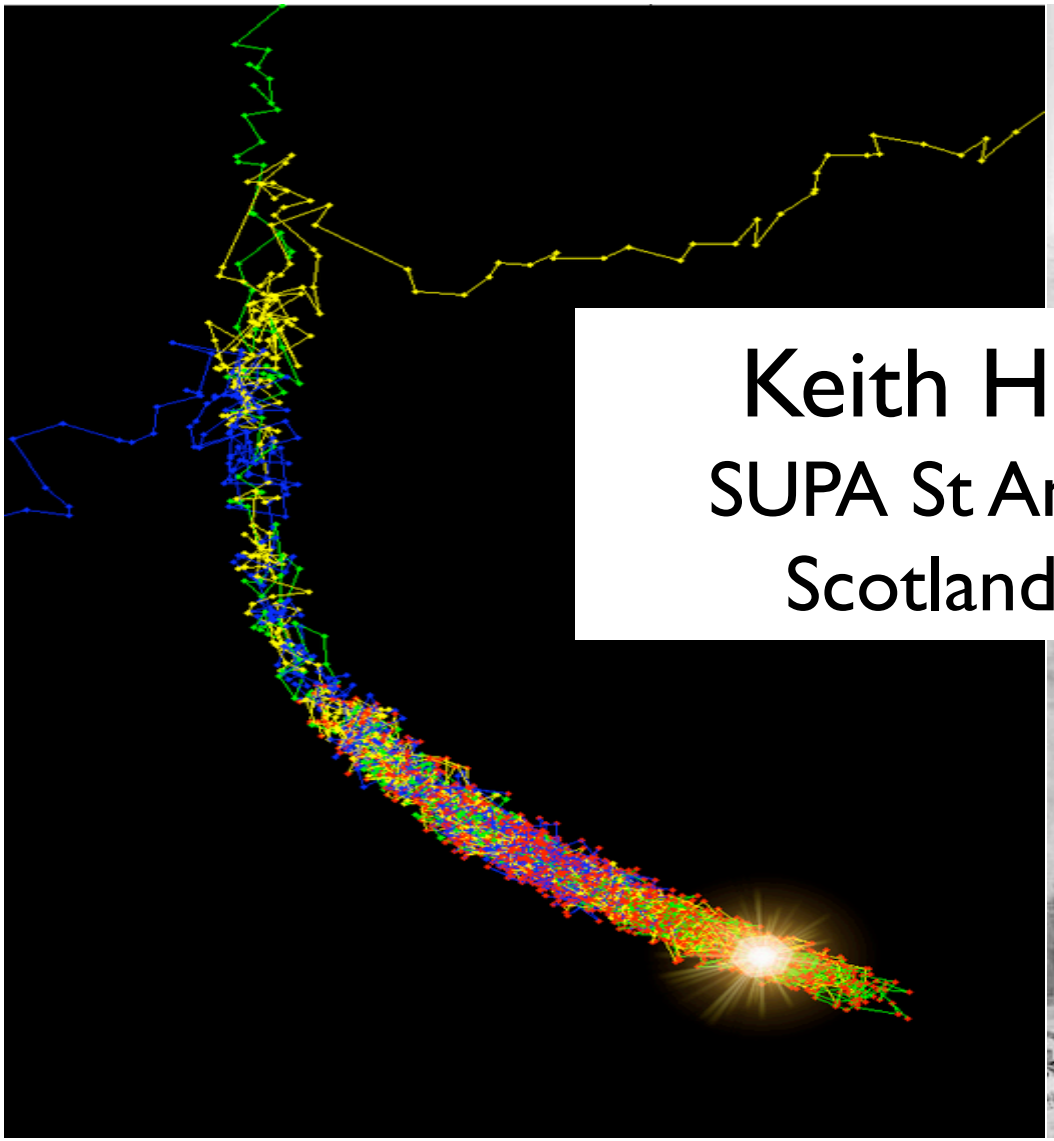
How to Stop a Runaway (Monte Carlo Markov) Chain



Keith Horne

SUPA St Andrews

Scotland, UK

Monte Carlo Markov Chain

(Simple yet powerful Metropolis Algorithm)

Start with a parameter guess : α_k .

Consider a random step: $\Delta\alpha_k$

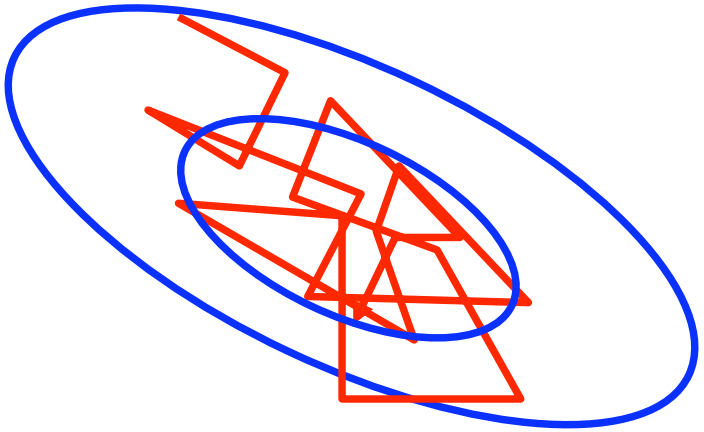
Step "downhill" if χ^2 improves.

Step "uphill" with probability P ~ exp($-\Delta\chi^2 / 2$)

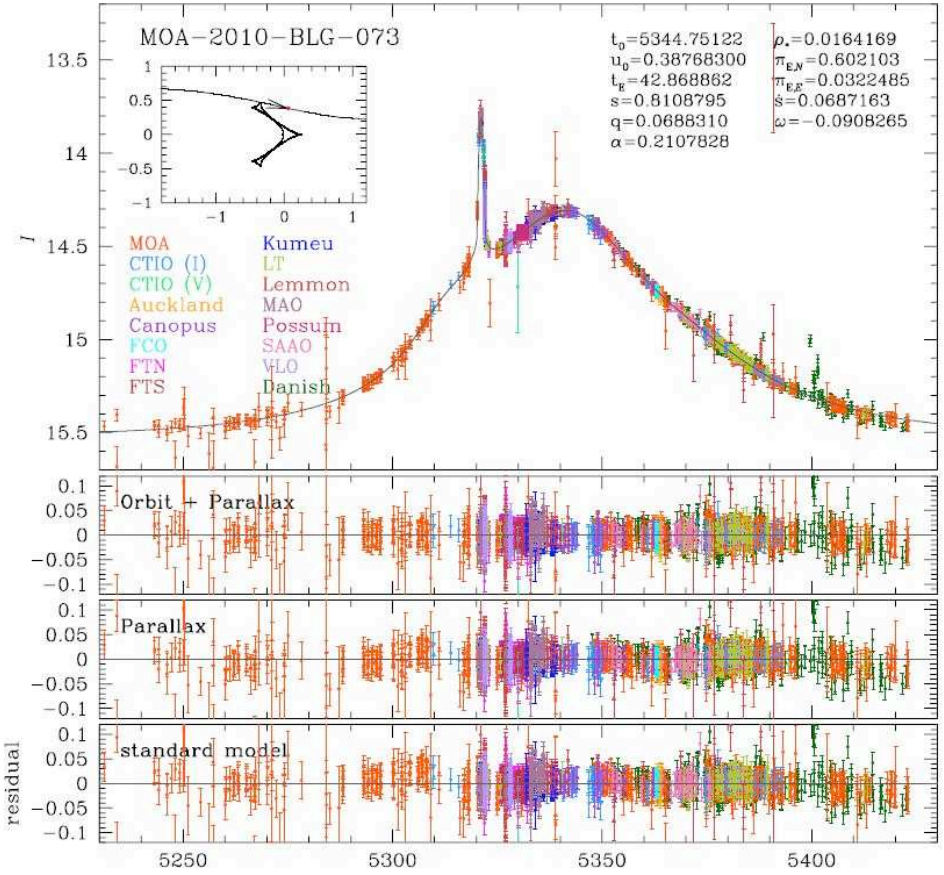
Iterate

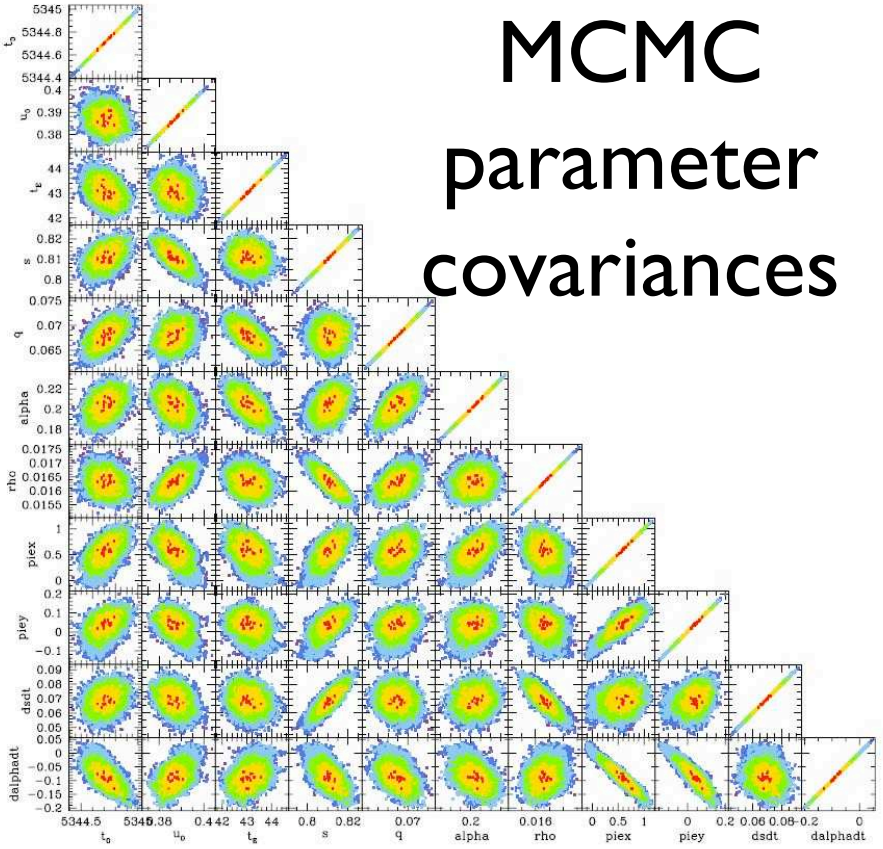
Decide to stop.

The *N* -point chain
maps out P(model | data)



MCMC fit to KB-10-073





MCMC parameter covariances

Light curve fit (from Chengho Han)

Binary lens + finite source + parallax + orbital motion

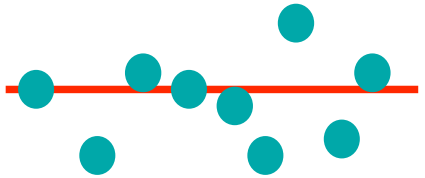
MCMC refinements

- "Gibbs Sampling" – step one parameter at a time.
- Take steps proportional to standard deviation (of recent points) along each parameter axis.
- Adjust step lengths for *P(accept)* ~ 30 %.
- Compute covariance matrix (from recent points in the chain) and step along its eigenvectors (equivalent orthogonal parameterisation).
- "Temperature" $P(accept) \sim \exp\left(-\chi^2 / 2T\right)$
- Include priors and promote small error bars:

$$-2\ln\left[\mathrm{P}\left(\mathrm{model} \mid \mathrm{data}\right)\right] = \chi^2 + 2\sum_{i=1}^{N} \ln\sigma_i - 2\ln\left(\mathrm{Prior}\right)$$

Parameters altering Error Bars

N data points D_i with unknown error bars:



2 Parameters: $\mu \quad \sigma$

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{D_i - \mu}{\sigma} \right)^2$$

$$P(\mu,\sigma \mid D) = \frac{\exp\left(-\chi^2/2\right)}{\left(\sqrt{2\pi}\,\sigma\right)^N} \times P(\mu,\sigma)$$

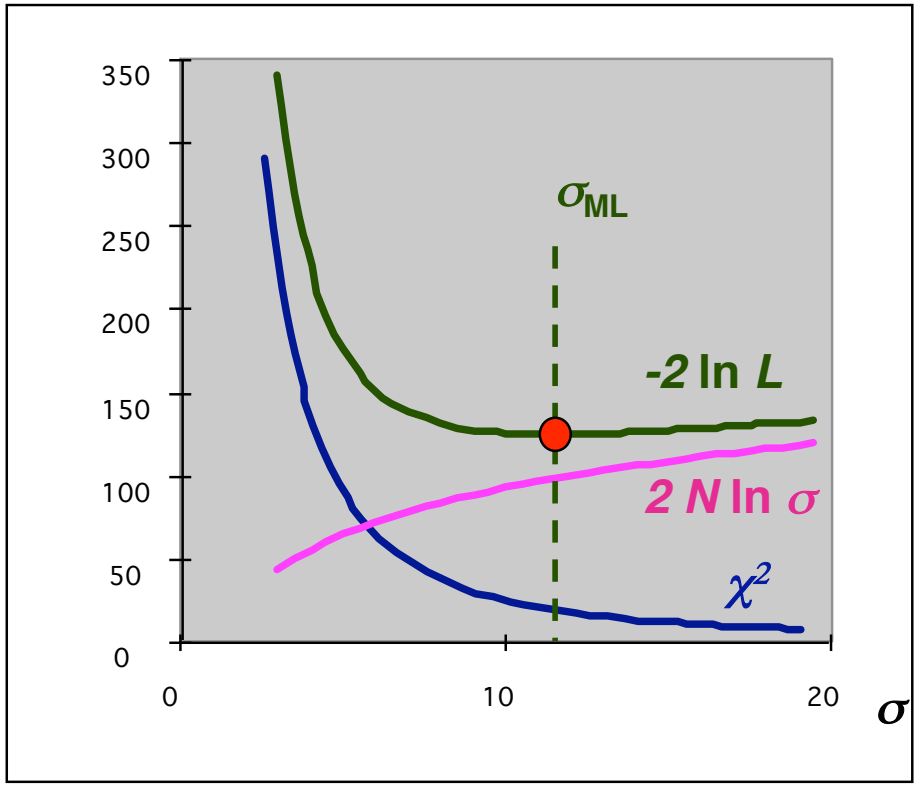
To find μ, minimise χ^2.

To find σ, minimising χ^2 fails !

$$\chi^2 \to 0 \quad \text{as } \sigma \to \infty$$

Maximum Likelihood promotes small σ, minimises:

$$-2\ln L = \chi^2 + 2N\ln\sigma$$



MCMC with Refinements

Start with a parameter guess : α_k , $\sigma_k = \sigma(\alpha_k)$.

Consider a random step (match $cov(\alpha_k, \alpha_j)$) .

$\Delta\alpha_k \sim f_k \sigma_k N(0, 1)$

Compute $-2 \ln(L \times P) = \chi^2 + 2 \Sigma_i \ln(\sigma_i) - 2 \ln(P)$

Step "downhill" always.

Step "uphill" with P(accept) = $(L \times P)_{new} / (L \times P)_{old}$

Iterate

Compute $cov(\alpha_k, \alpha_j)$ over recent steps.

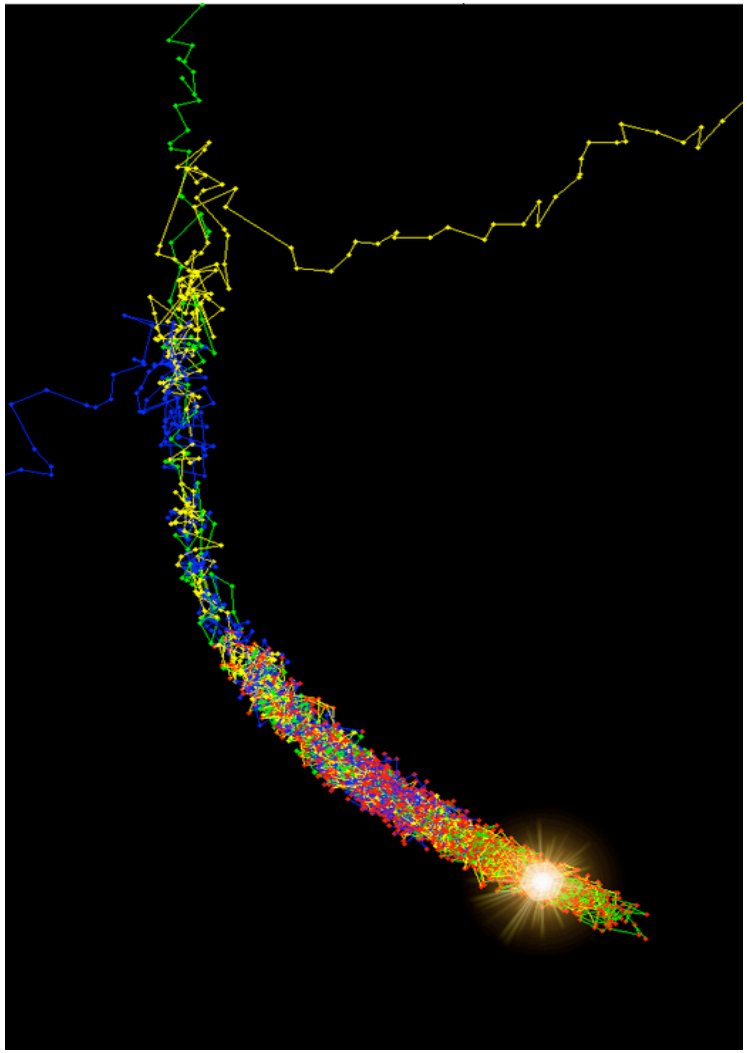
Adjust f_k to maintain P(accept) ~ 30%

Iterate

Decide to stop.

The *N* -point chain maps out P(model | data)

MCMC simple and powerful
But can be very slow:



Our χ^2 evaluations are expensive.

Burn-in stage - must reject.
Several chains to test for convergence.

Uncertainties (in parameter values and error bars) scale as $N^{-1/2}$ after N steps.
For ~1% accuracy, need $N > 10^4$ independent samples.

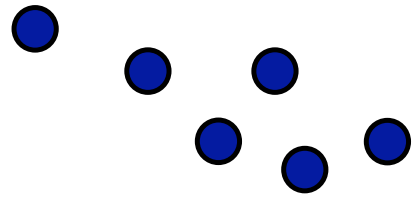
Correlations: "Thin the chain"
to retain uncorrelated samples.

Parameter grids:
One chain for each grid cell.

MCMC Parameters and Error Bars

- **Simple** : e.g. the Sample Mean and Sample Variance over *N* points:

$$\bar{\alpha} \equiv \frac{1}{N}\sum_{i=1}^{N}\alpha_i \qquad \bar{\sigma}^2 \equiv \frac{1}{N-1}\sum_{i=1}^{N}\left(\alpha_i - \bar{\alpha}\right)^2$$



  - **Easy to compute statistics** of the parameter estimates:
  - Mean, Median, Mode, RMS, Covariances, …
  - Statistics for **any functions of parameters.**
  - Accurate estimates require long chains.

For ~1% accuracy, need $N > 10^4$

MCMC Parameters and Error Bars

- **Simple method** : Sample Mean and RMS over *N* points:
  - The $\chi^2(\alpha_i)$ values are not used.
  - Information (hard-won) is wasted.

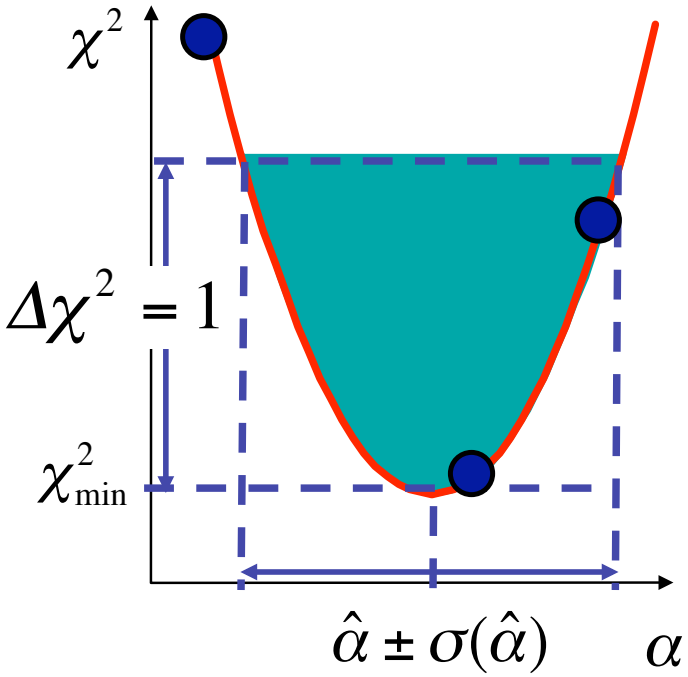
Need $N > 10^4$ for ~1% accuracy

- **Can we do better ?** :
- Fit the $\chi^2(\alpha_i)$ values

$$\chi^2(\alpha) = \chi^2(\hat{\alpha}) + \left(\frac{\alpha - \hat{\alpha}}{\sigma(\hat{\alpha})}\right)^2 + O(\Delta\alpha^3)$$

- Fit with **probability weights:** $w \sim exp(-\chi^2/2)$

to focus fit near the minimum.



K=3 parameters, "machine precision" with only *N* >3

Multi-Parameter Models

For M parameters: α_k , fit *M*-dimensional paraboloid to $\chi^2(\alpha_k)$

$$\chi^2 = \chi^2(\hat{\alpha}) + \sum_{j\,k} \Delta\alpha_j \, H_{j\,k} \Delta\alpha_k + O(\Delta\alpha^3)$$

$$\mathrm{cov}\left(\alpha_j, \alpha_k\right) = \left(H^{-1}\right)_{j\,k}$$

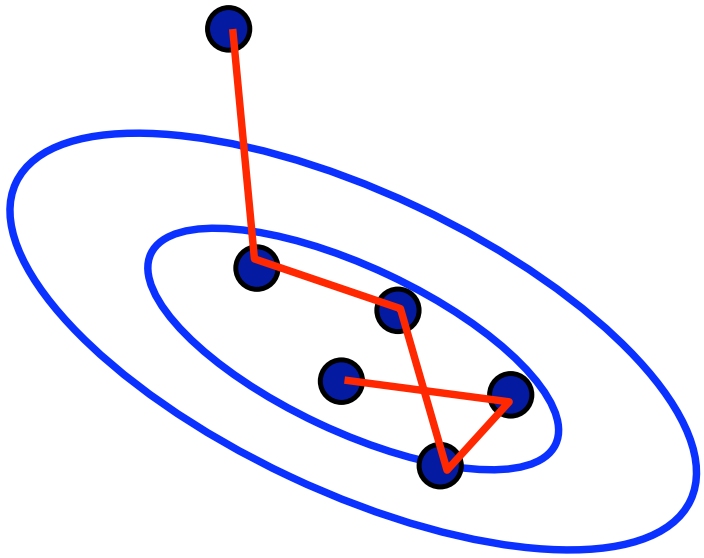
Fit *K* parameters: χ^2_{min}

+ *M* parameters: α_k

+ *M(M+1)/2* Hessian: $H_{i\,j}$

Total: *K = 1 + (M+3) (M/2)*

Check: for *M = (1,2,3,…)*
need *N > K = (3,6,10,…)*



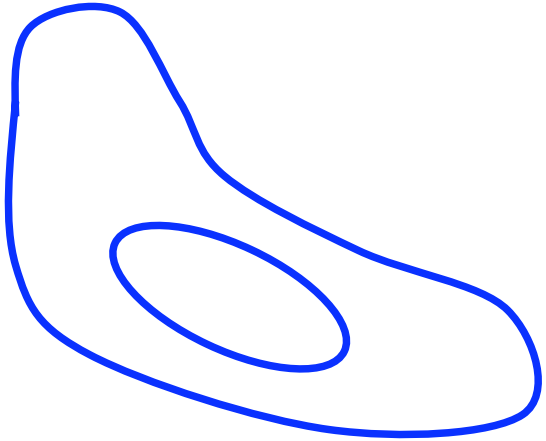
Can use rejected steps.
Can even use burn-in

For Non-Linear Parameters: Include Cubic Terms

For M parameters: α_k , include **cubic terms** in the fit to $\chi^2(\alpha_k)$

$$\chi^2 = \chi^2(\hat{\alpha}) + \sum_{j\,k} H_{j\,k}\,\Delta\alpha_j\,\Delta\alpha_k$$

$$+ \sum_{j\,k\,l} C_{j\,k\,l}\,\Delta\alpha_j\,\Delta\alpha_k\,\Delta\alpha_l \quad + \quad O(\Delta\alpha^4)$$



Fit *K* parameters:

K = 1 χ^2_{min}

+ *M* parameters: α_k

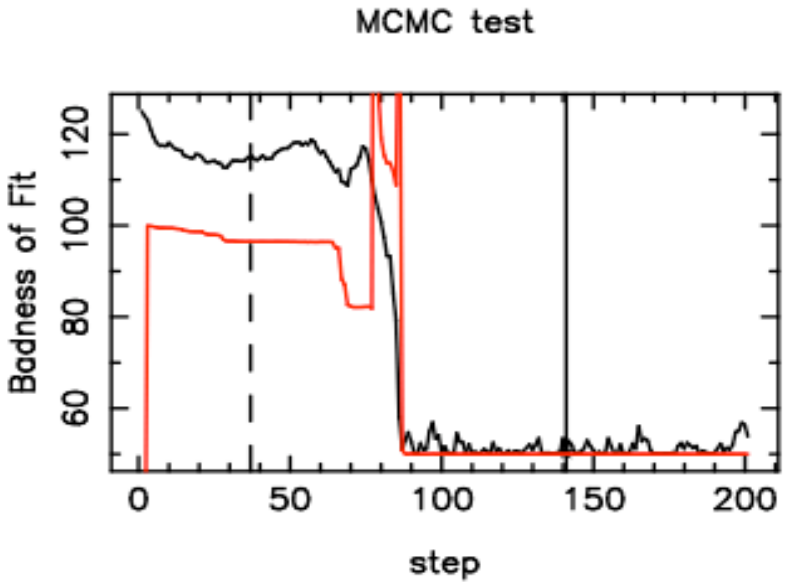
+ *M(M+1)/2* Hessian: $H_{j\,k}$

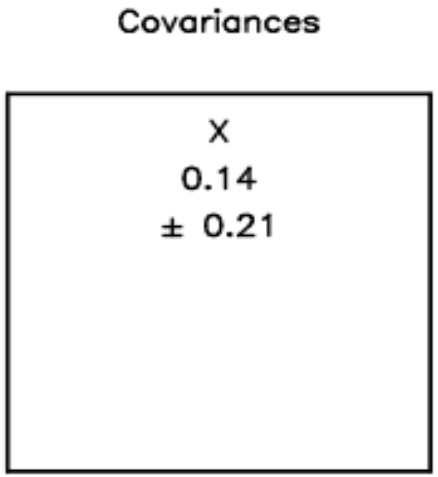
+ *M(M+1)(M+2)/6* cubics: $C_{j\,k\,l}$

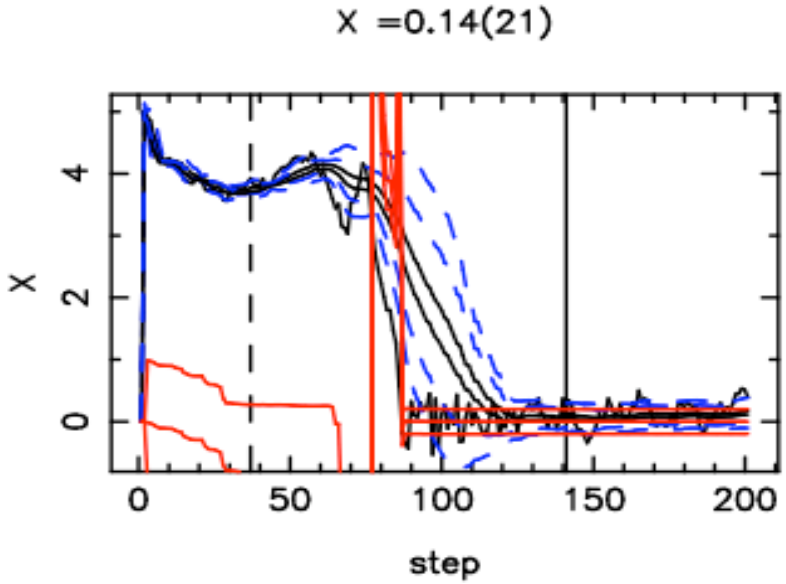
Perhaps use e.g. BIC
(promote simple models)
to use < *K* terms

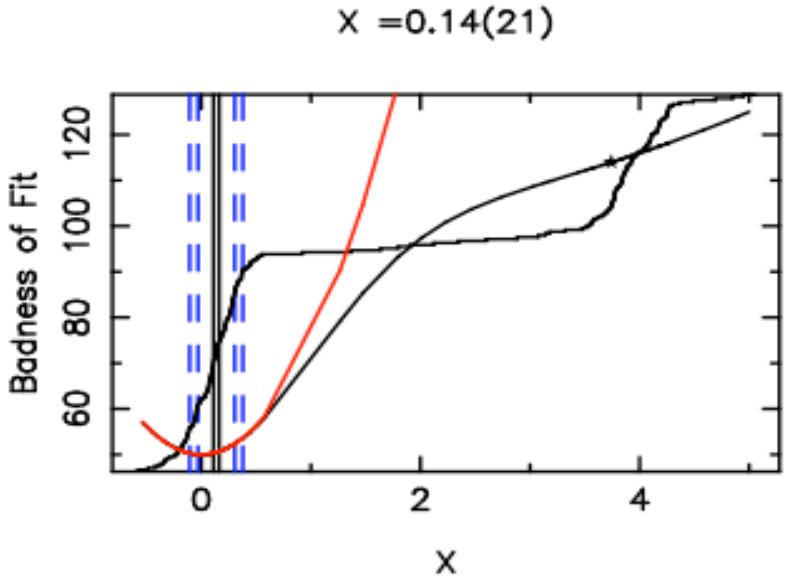
$$\text{BIC} = \chi^2 + K\,\ln(N)$$

One-Parameter Test

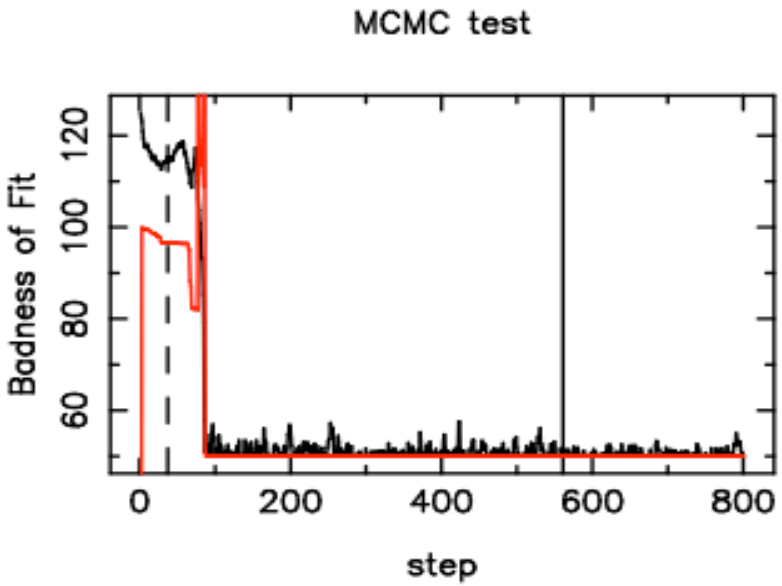


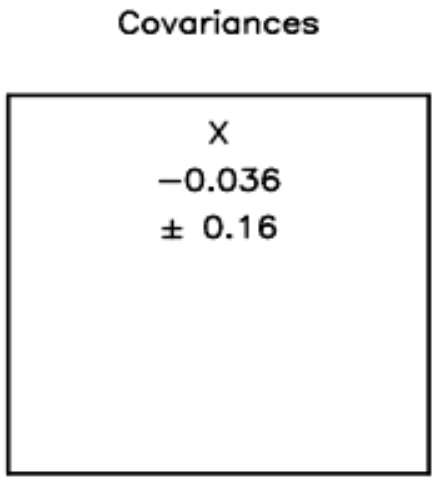


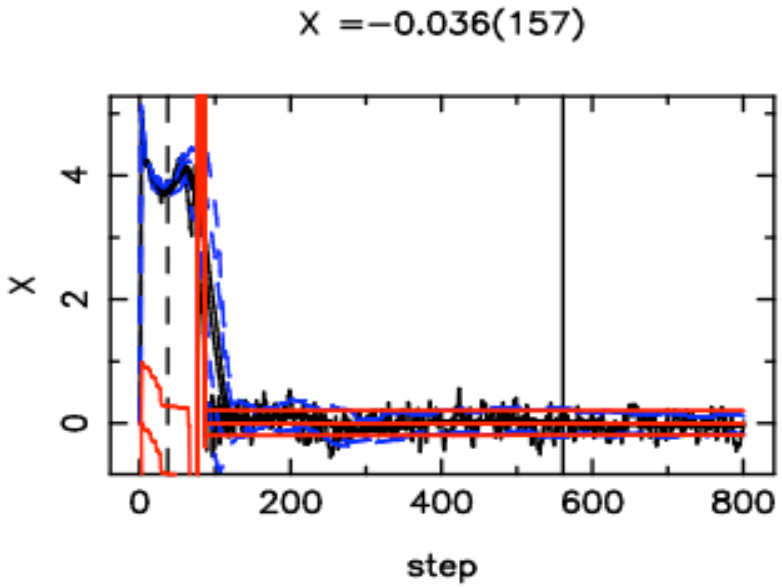


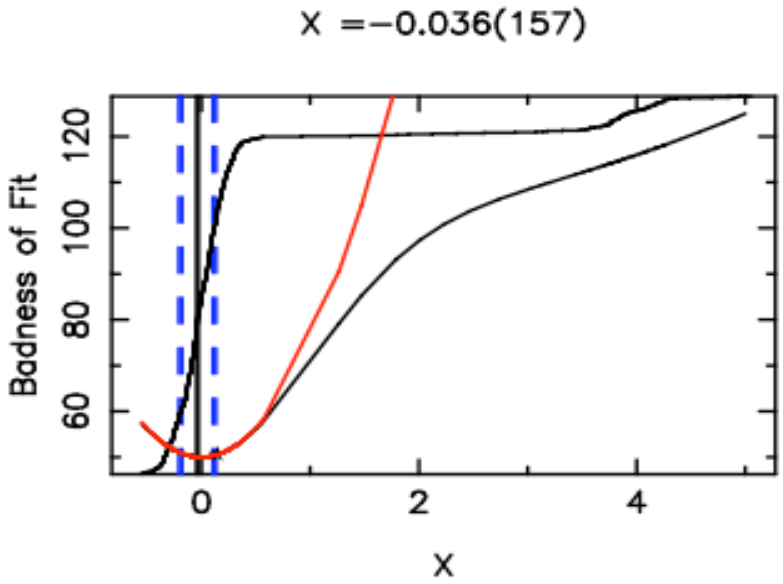


One-Parameter Test

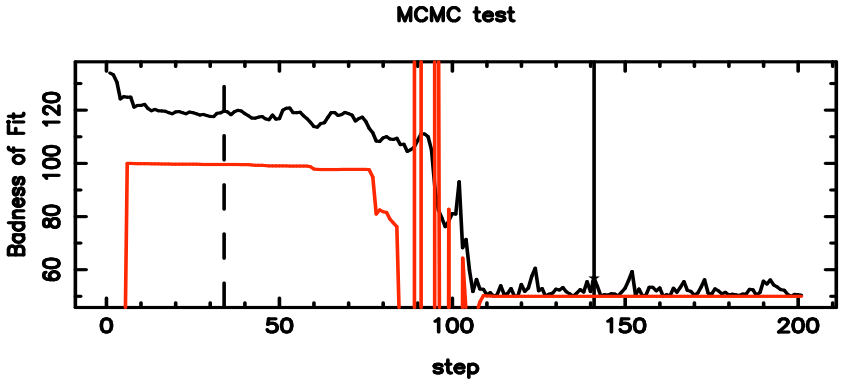


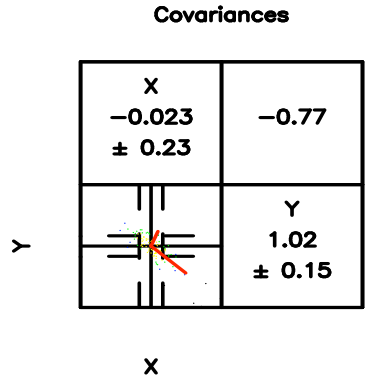


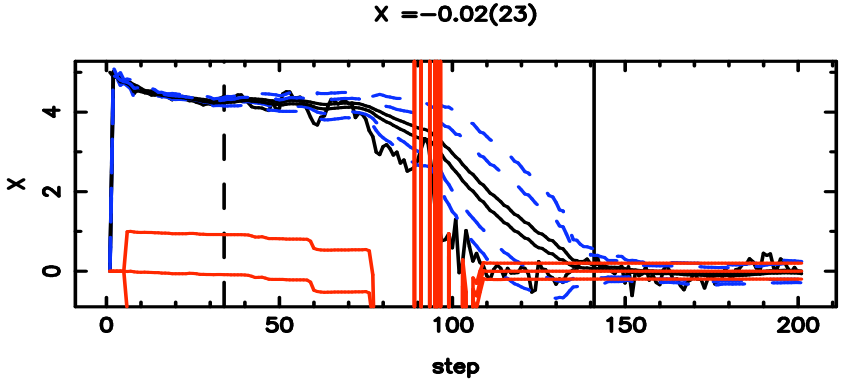


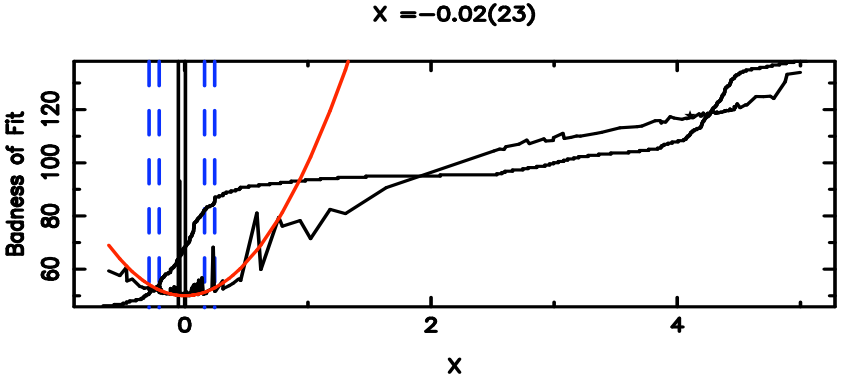


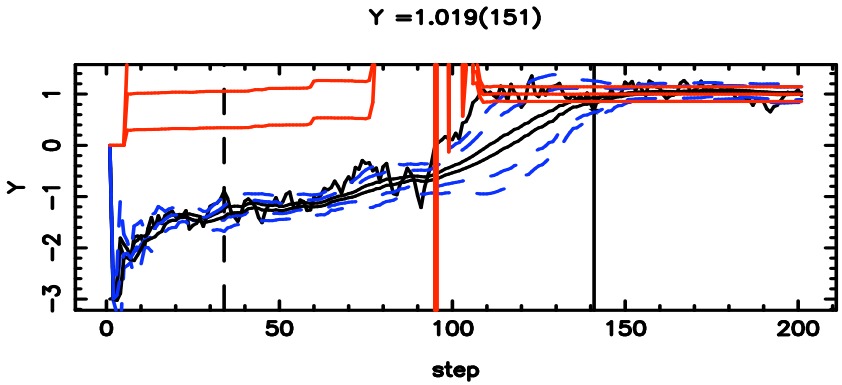
Two-Parameter Test

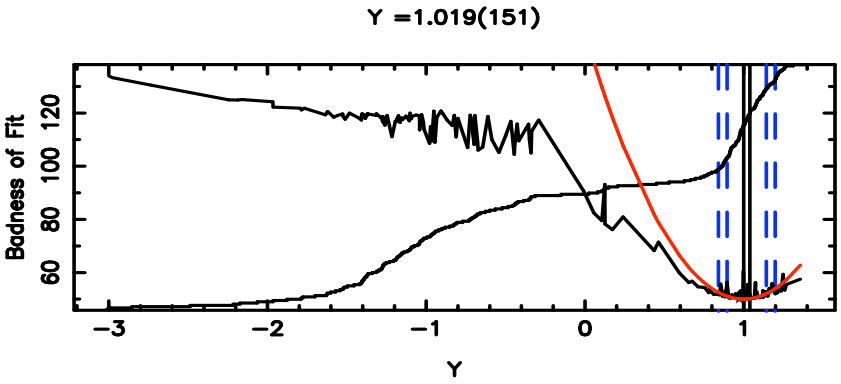




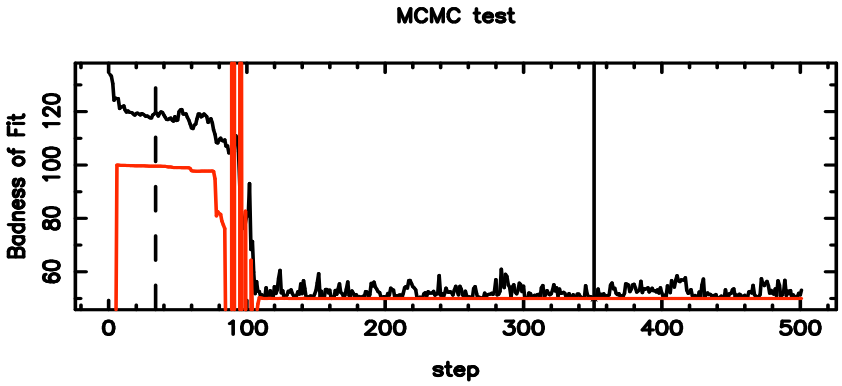


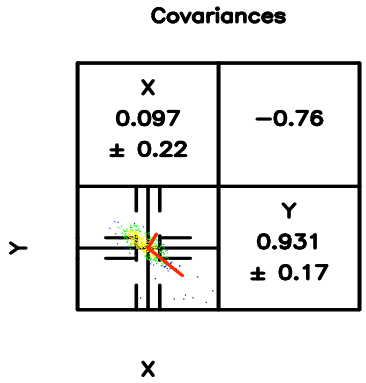


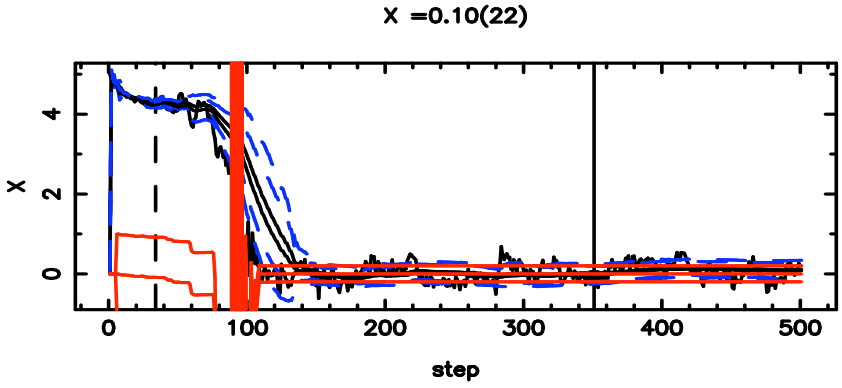


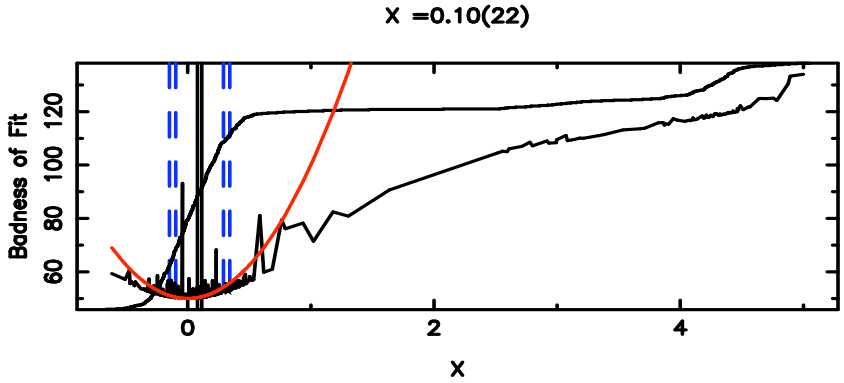


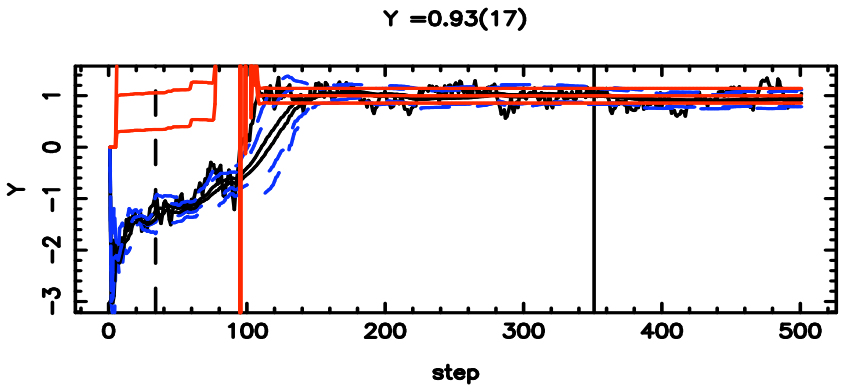
Two-Parameter Test

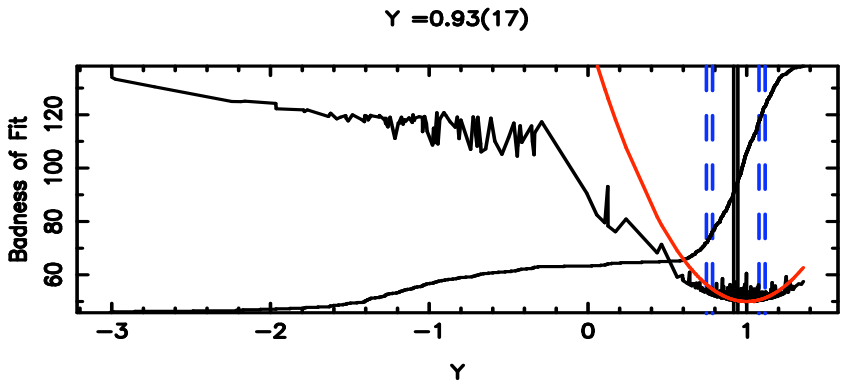




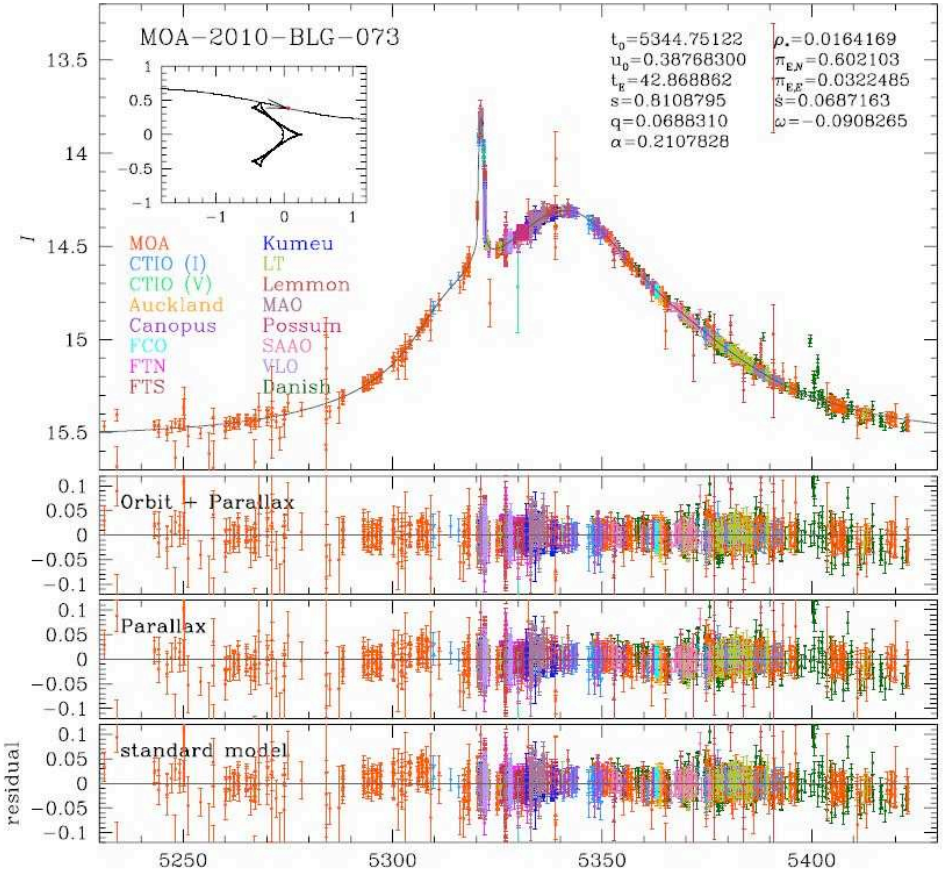


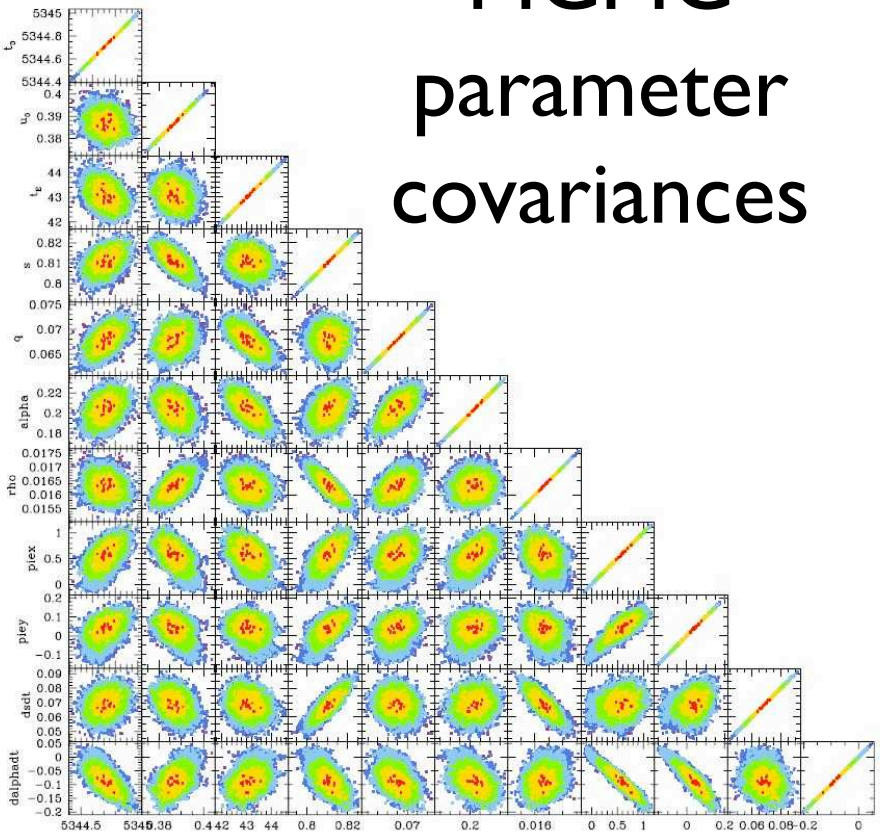






Not yet tested, but may speed up our microlens fits





MCMC parameter covariances

Light curve fit (from Chengho Han)

Binary lens + finite source + parallax + orbital motion

Summary

- **MCMC is** a **simple yet powerful** algorithm for mapping out P(model | data) in a high-dimensional parameter space.
- **MCMC can be slow -- requires a long chain** (after burn-in) for accurate parameter and error bar estimates.
- Typically $>10^4$ evaluations of χ^2 for 1% accuracy

- **To save CPU time, stop early** (if appropriate).
- **Fit** of **a quadratic** (or cubic or higher-order) function **to the χ^2 values**.
- Use all the χ^2 evaluations, including the rejected points.
- **Accurate parameter estimates and error bars** more rapidly, without a long chain.

Thanks to the Organisers

- For a successful, informative and enjoyable conference
- In a beautiful venue – Salerno.