Bayesian analysis in nuclear physics

Ken Hanson

T-16, Nuclear Physics; Theoretical Division
Los Alamos National Laboratory

Tutorials presented at LANSCE
Los Alamos Neutron Scattering Center
July 25 – August 1, 2005

Los Alamos
NATIONAL LABORATORY

This presentation available at
http://www.lanl.gov/home/kmh/

LA-UR-05-5680

Goals of tutorials

My aim is to

- present overview of Bayesian and probabilistic modeling
- cover basic Bayesian methodology relevant to nuclear physics, especially cross section evaluation
- point way to how to do it
- convince you that
  - Bayesian analysis is a reasonable approach to coping with measurement uncertainty

- Many thanks to my T-16 colleagues
  - Gerry Hale, Toshihiko Kawano, Patrick Talou

Outline – four tutorials

1. **Bayesian approach**

   probability – quantifies our degree of uncertainty

   Bayes law and prior probabilities

2. **Bayesian modeling**

   Peelle's pertinent puzzle

   Monte Carlo techniques; quasi-Monte Carlo

   Bayesian update of cross sections using Jezebel criticality expt.

3. **Bayesian data analysis**

   linear fits to data with Bayesian interpretation

   uncertainty in experimental measurements; systematic errors

   treatment of outliers, discrepant data

4. **Bayesian calculations**

   Markov chain Monte Carlo technique

   analysis of Rossi traces; alpha curve

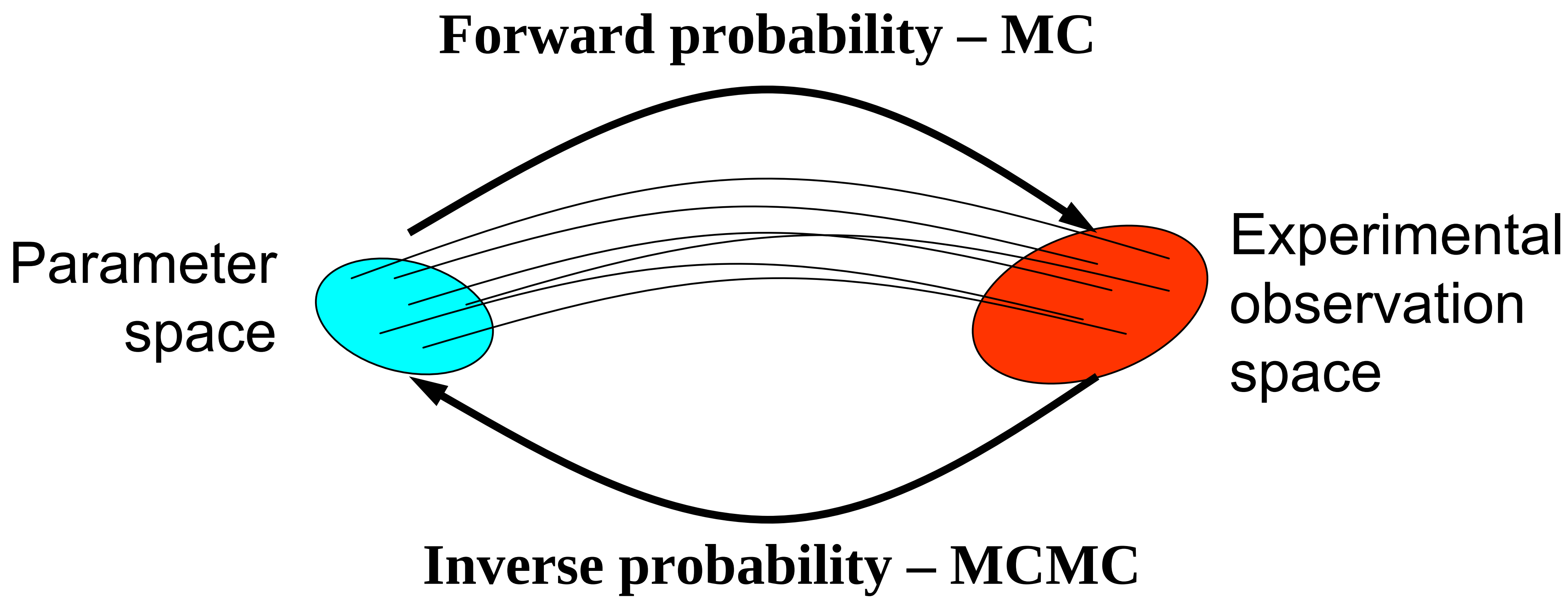
   background estimation in spectral data

Slides and bibliography

- These slides can be obtained by going to my public web page:
  http://public.lanl.gov/kmh/talks/
  - link to **tutorial slides**
  - short **bibliography** relevant to topics covered in tutorial
  - other presentations, which contain more detail about material presented here
- Noteworthy books:
  - D. Sivia, *Data Analysis: A Bayesian Tutorial* (1996); lucid pedagogical development of the Bayesian approach with an experimental physics slant
  - D. L. Smith, *Probability, Statistics, and Data Uncertainties in Nuclear Science and Technology* (1991); lots of good advice relevant to cross-section evaluation
  - G. D'Agostini, *Bayesian Reasoning in Data Analysis: A Critical Review*, (World Scientific, New Jersey, 2003); Bayesian philosophy
  - A. Gelman et al., *Bayesian Data Analysis* (1995); statisticians' view
  - W. R. Gilks et al., *Markov Chain Monte Carlo in Practice* (1996); basic MCMC text

Tutorial 4
Bayesian calculations

Forward and inverse probability



- Forward probability - determine uncertainties in observables resulting from model parameter uncertainties; use Monte Carlo
- Inverse probability - infer model parameter uncertainties from uncertainties in observables; use Markov chain Monte Carlo

MCMC - problem statement

- Parameter space of n dimensions represented by vector $\mathbf{x}$
- Given an “arbitrary” **target** probability density function (pdf), $q(\mathbf{x})$, draw a set of samples $\{\mathbf{x}_\mathrm{k}\}$ from it
- Only requirement typically is that, given $\mathbf{x}$, one be able to evaluate $Cq(\mathbf{x})$, where C is an unknown constant, that is, $q(\mathbf{x})$ need not be normalized
- Although focus here is on continuous variables, MCMC applies to discrete variables as well

- It all started with seminal paper:
  - N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equations of state calculations by fast computing machine,” *J. Chem. Phys.* **21**, pp. 1087–1091 (1953)
    - MANIAC: 5 KB RAM, 100 KHz, 1 KHz multiply, 50 KB disc

Uses of MCMC

- Permits evaluation of the expectation values of functions of **x**, e.g.,

$$\langle f(\mathbf{x})\rangle = \int f(\mathbf{x})\, q(\mathbf{x})\, d\mathbf{x} \cong (1/\mathrm{K})\, \Sigma_{\mathrm{k}}\, f(\mathbf{x}_{\mathrm{k}})$$

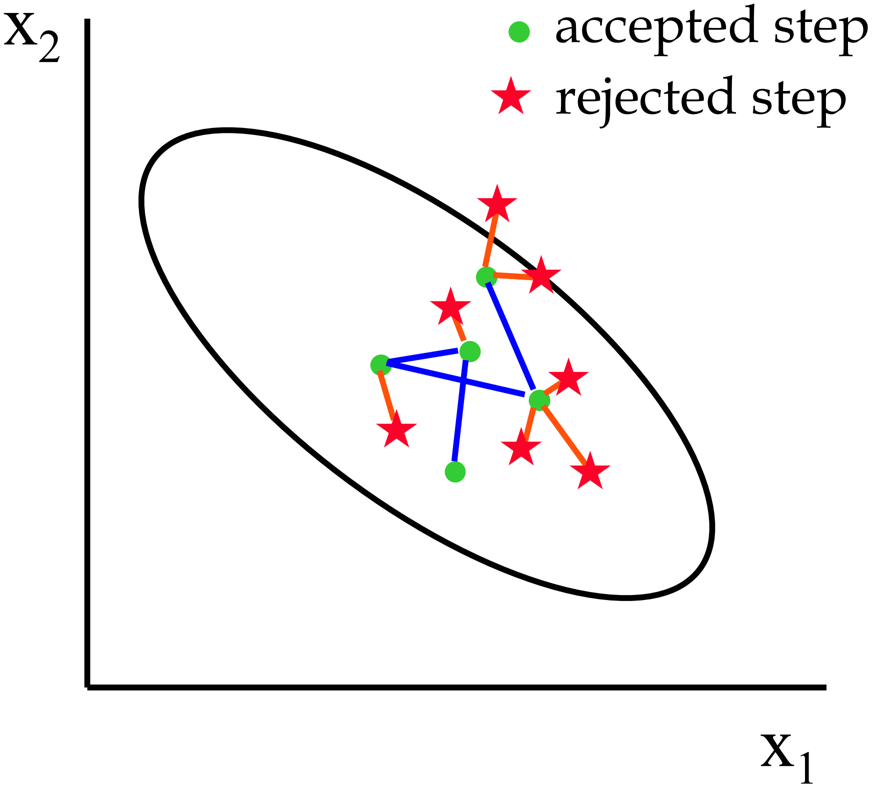
  - typical use is to calculate mean $\langle \mathbf{x}\rangle$ and variance $\langle (\mathbf{x} - \langle \mathbf{x}\rangle)^2\rangle$
- Useful for evaluating integrals, such as the partition function for properly normalizing the pdf
- Dynamic display of sequences provides visualization of uncertainties in model and range of model variations
- Automatic marginalization; when considering any subset of parameters of an MCMC sequence, the remaining parameters are marginalized over (integrated out)

Markov Chain Monte Carlo

Generates sequence of random samples from an arbitrary probability density function

- Metropolis algorithm:
  - draw trial step from symmetric pdf, i.e., $t(\Delta \mathbf{x}) = t(-\Delta \mathbf{x})$
  - accept or reject trial step
  - simple and generally applicable
  - relies only on calculation of target pdf for any **x**

Probability(x_1, x_2) = q(**x**)



Metropolis algorithm

- Target pdf is $q(\mathbf{x})$
- Select initial parameter vector $\mathbf{x}_0$
- Iterate as follows: at iteration number k

  (1) create new trial position $\mathbf{x}^* = \mathbf{x}_k + \Delta\mathbf{x}$,
  where $\Delta\mathbf{x}$ is randomly chosen from $t(\Delta\mathbf{x})$

  (2) calculate ratio $r = q(\mathbf{x}^*)/q(\mathbf{x}_k)$

  (3) accept trial position, i.e. set $\mathbf{x}_{k+1} = \mathbf{x}^*$
  if $r \geq 1$ or with probability r, if $r < 1$
  otherwise stay put, $\mathbf{x}_{k+1} = \mathbf{x}_k$

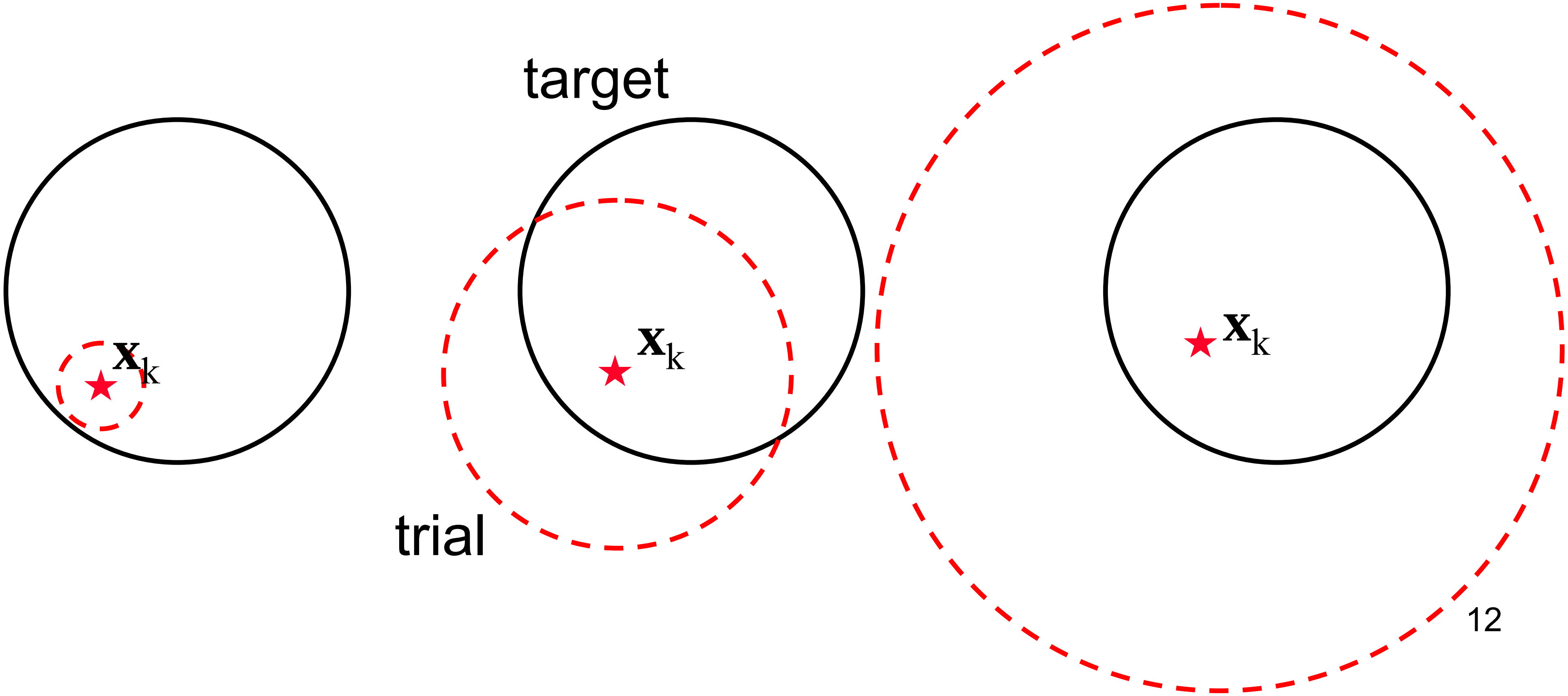
- Requires only computation of $cq(\mathbf{x})$, where c is a constant
- Trail distribution must be symmetric: $t(\Delta\, \mathbf{x}) = t(-\Delta\, \mathbf{x})$
- Maintains detailed balance: $p(\mathbf{x}_k \rightarrow \mathbf{x}_{k+1}) = p(\mathbf{x}_{k+1} \rightarrow \mathbf{x}_k)$
- "Markov chain" since $\mathbf{x}_{k+1}$ depends probabilistically only on $\mathbf{x}_k$

Choice of trial distribution

- Algorithm places loose requirements on trial distribution *t*()
  - stationary; independent of position
- Often used functions include
  - *n*-D Gaussian, isotropic and uncorrelated
  - *n*-D Cauchy, isotropic and uncorrelated
- Choose width to “optimize” MCMC efficiency
  - rule of thumb: aim for acceptance fraction of about 25%

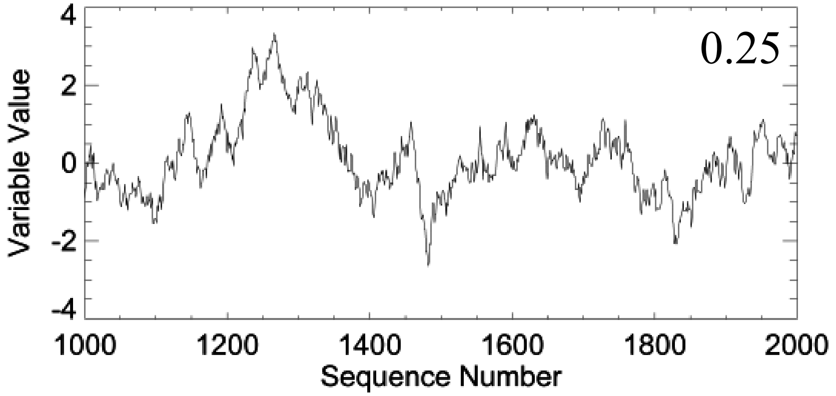
Choice of trial distribution – experiments

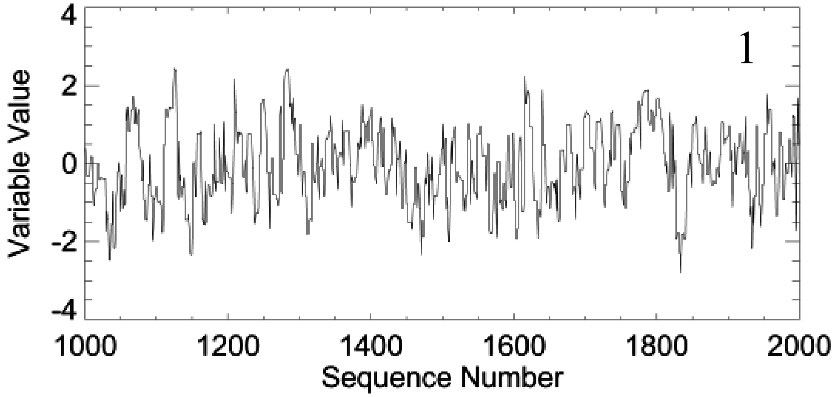
- Target distribution $q(\mathbf{x})$ is n dimensional Gaussian
  - uncorrelated, univariate (isotropic with unit variance)
  - most generic case
- Trial distribution $t(\Delta\mathbf{x})$ is n dimensional Gaussian
  - uncorrelated, equivariate; various widths

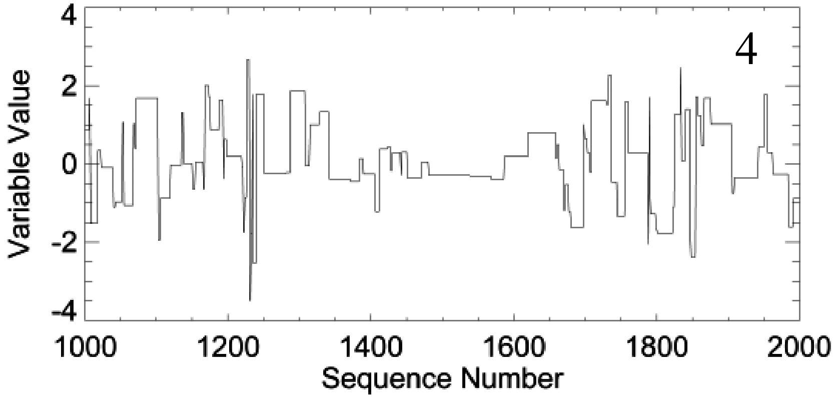


MCMC sequences for 2D Gaussian

- Results of running Metropolis with ratios of width of trial pdf to target pdf of 0.25, 1, and 4
- When trial pdf is much smaller than target pdf, movement across target pdf is slow
- When trial width same as target, samples seem to better sample target pdf
- When trial width much larger than target, trials stay put for long periods, but jumps are large

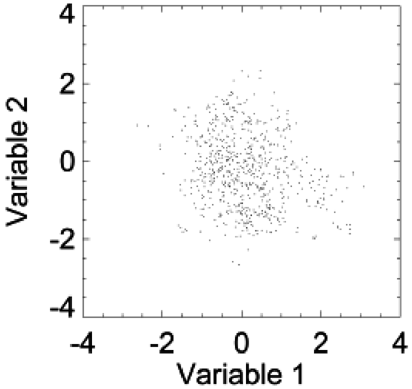




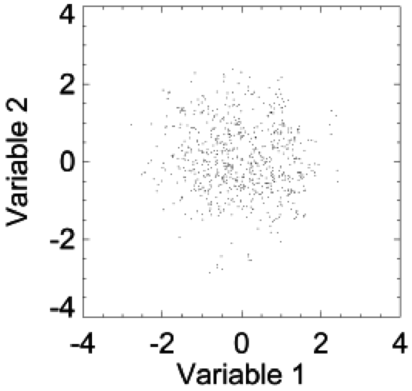


MCMC sequences for 2D Gaussian

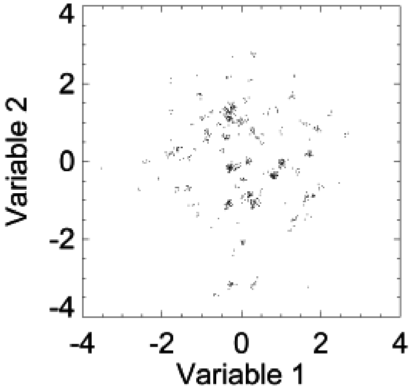
- Results of running Metropolis with ratios of width of trial pdf to target pdf of 0.25, 1, and 4
- Display accumulated 2D distribution for 1000 trials
- Viewed this way, it is difficult to see difference between top two images
- When trial pdf much larger than target, fewer splats, but further apart



0.25



1



4

MCMC - autocorrelation and efficiency

- In MCMC sequence, subsequent parameter values are usually correlated
- Degree of correlation quantified by autocorrelation function:

$$\rho(l) = \frac{1}{N}\sum_{i=1}^{N} y(i)y(i-l)$$

  - where *y(x)* is the sequence and *l* is lag

- For Markov chain, expect exponential

$$\rho(l) = \exp[-\left|\frac{l}{\lambda}\right|]$$

- Sampling efficiency is

$$\eta = [1+2\sum_{l=1}^{\infty}\rho(l)]^{-1} = \frac{1}{1+2\lambda}$$

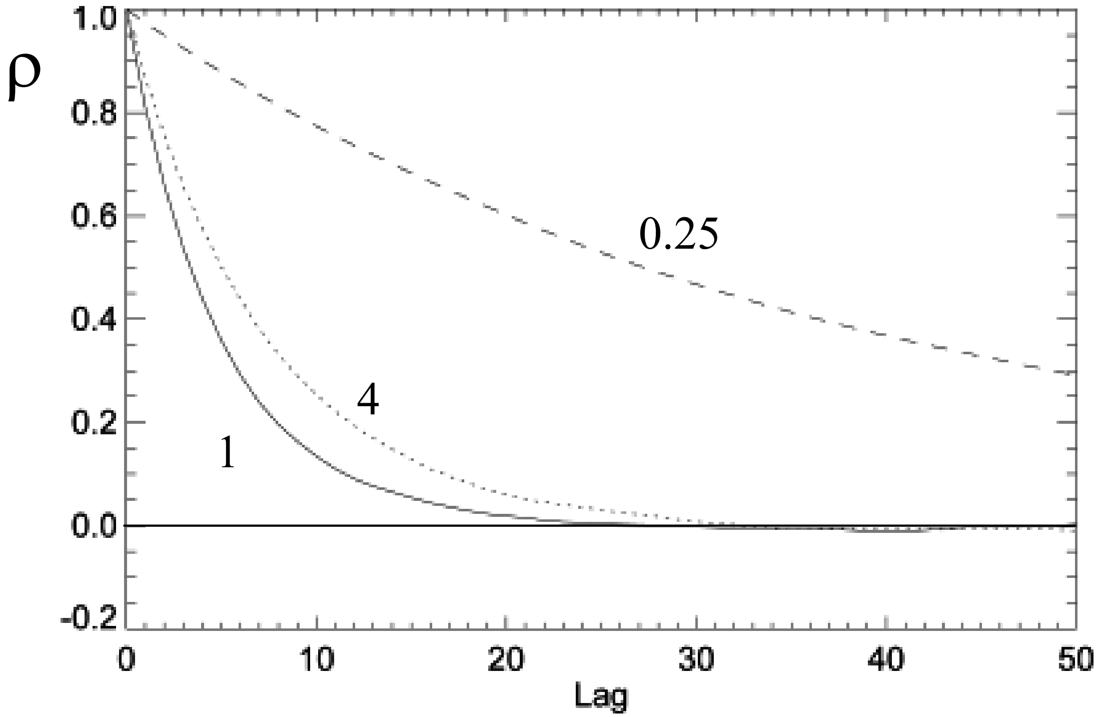
- In other words, η^{-1} iterates required to achieve one statistically independent sample

Autocorrelation for 2D Gaussian

- Plot confirms that the autocorrelation drops slowly when the trial width is much smaller than the target width; MCMC efficiency is poor
- Sampling efficiency is

$$\eta = \frac{1}{1 + 2\lambda}$$

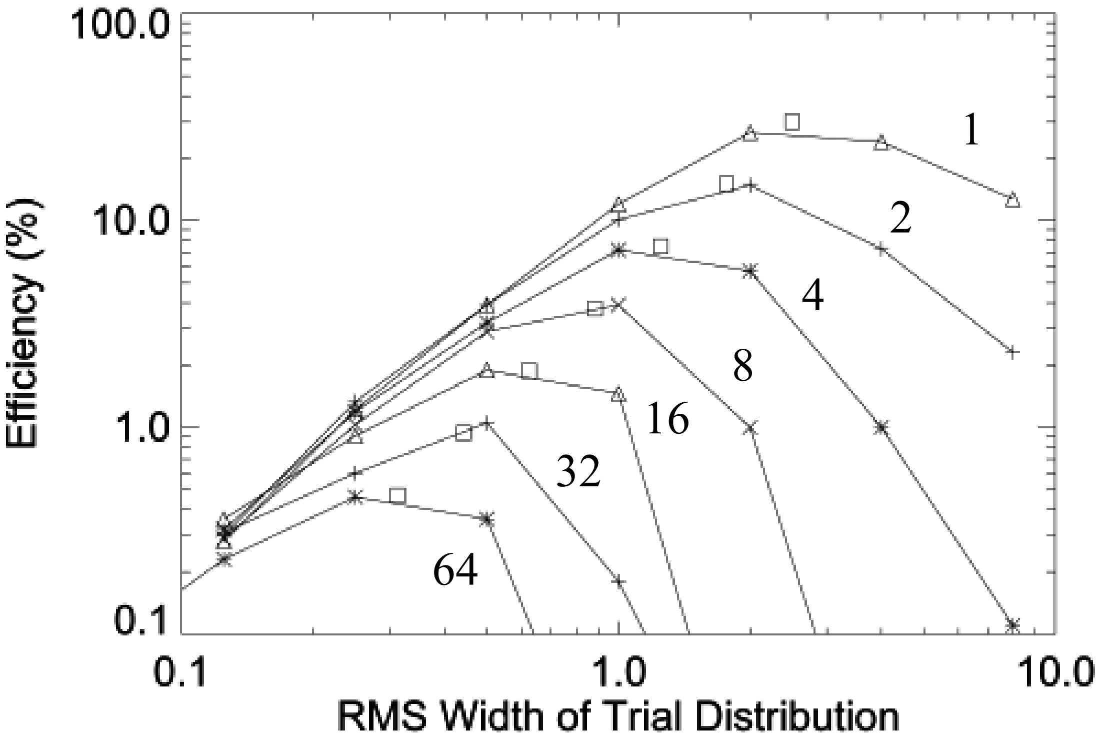
- Best efficiency occurs when trial about same size as target (for 2D)



Normalized autocovariance for various widths of trial pdf relative to target: 0.25, 1, and 4

Efficiency as function of width of trial pdf

- for univariate, uncorrelated Gaussians, with 1 to 64 dimensions
- efficiency as function of width of trial distributions
- boxes are predictions of optimal efficiency from diffusion theory [A. Gelman, et al., 1996]
- efficiency drops reciprocally with number of dimensions

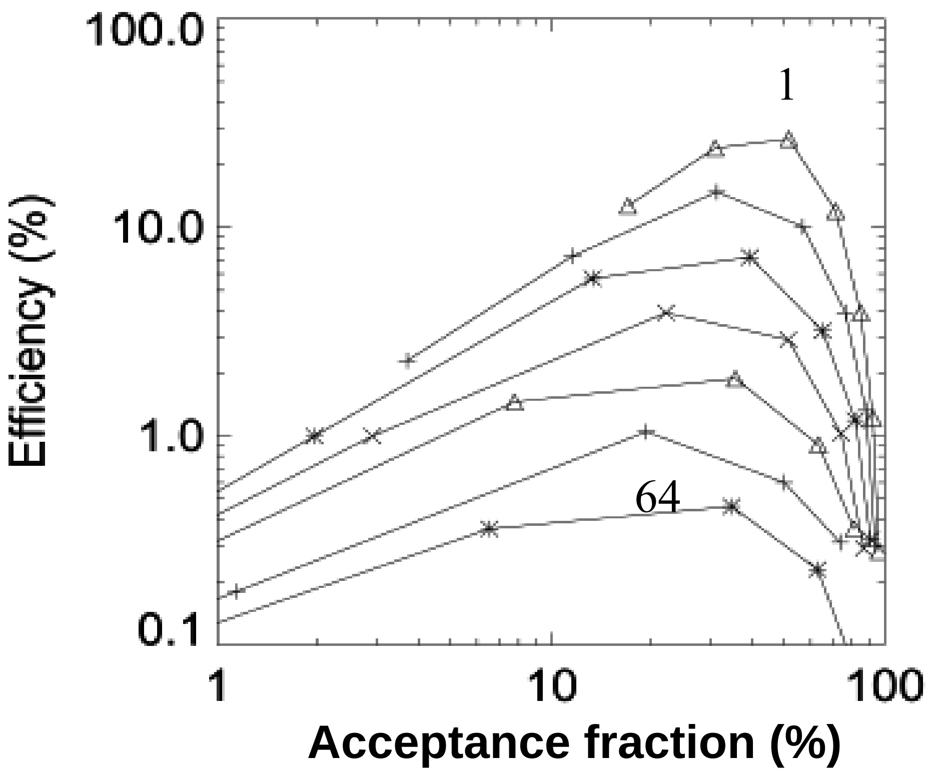


Efficiency as function of acceptance fraction

- For univariate Gaussians, with 1 to 64 dimensions
- Efficiency as function of acceptance fraction
- Best efficiency is achieved when about 25% of trials are accepted for moderate number of dimensions
- Optimal statistical efficiency:

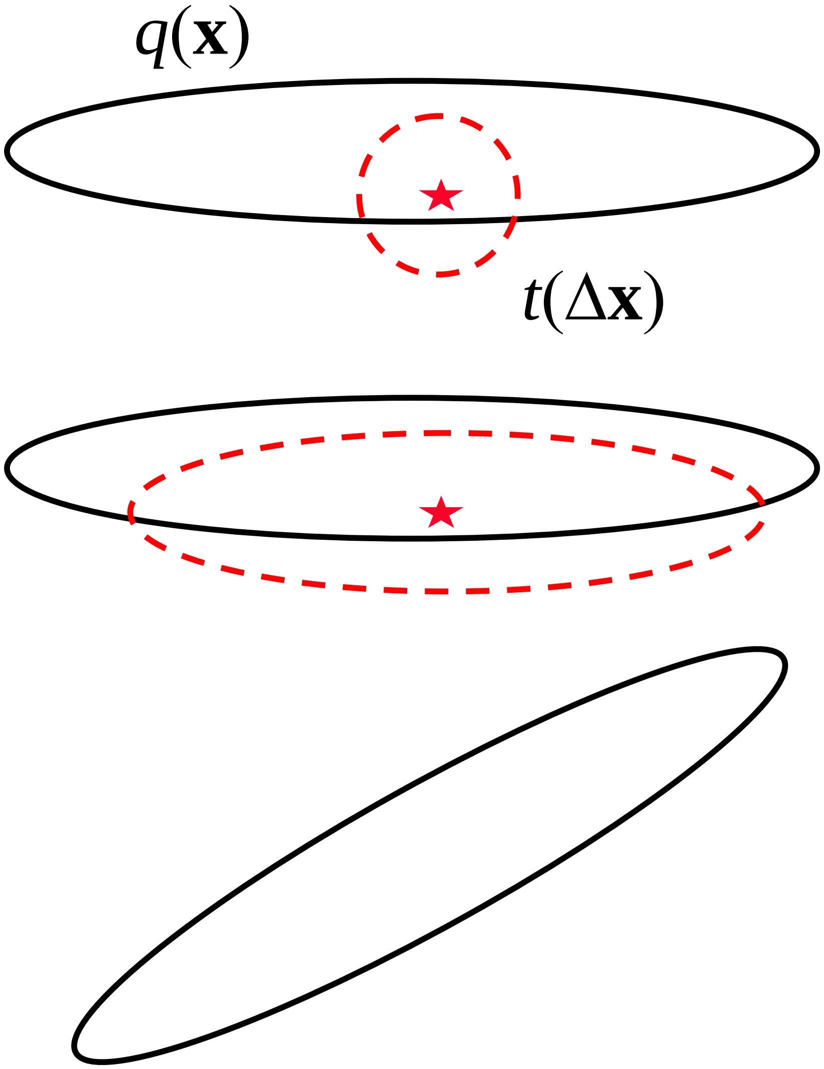
  $\eta \sim 0.3/n$

  - for uncorrelated, equivariate Gaussian
  - generally decreases correlation and variable variance
  - consistent with diffusion theory derivation [A. Gelman, et al., 1996]



Further considerations

- When target distribution $q(\mathbf{x})$ not isotropic
  - difficult to accommodate with isotropic $t(\Delta\mathbf{x})$
  - each parameter can have different efficiency
  - desirable to vary width of different $t(\mathbf{x})$ to approximately match $q(\mathbf{x})$
  - recovers efficiency of univariate case
- When $q(\mathbf{x})$ has correlations
  - $t(\mathbf{x})$ should match shape of $q(\mathbf{x})$

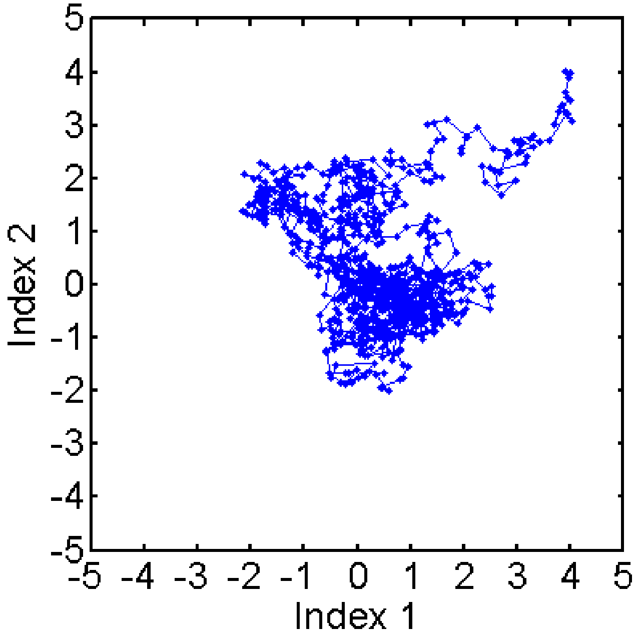


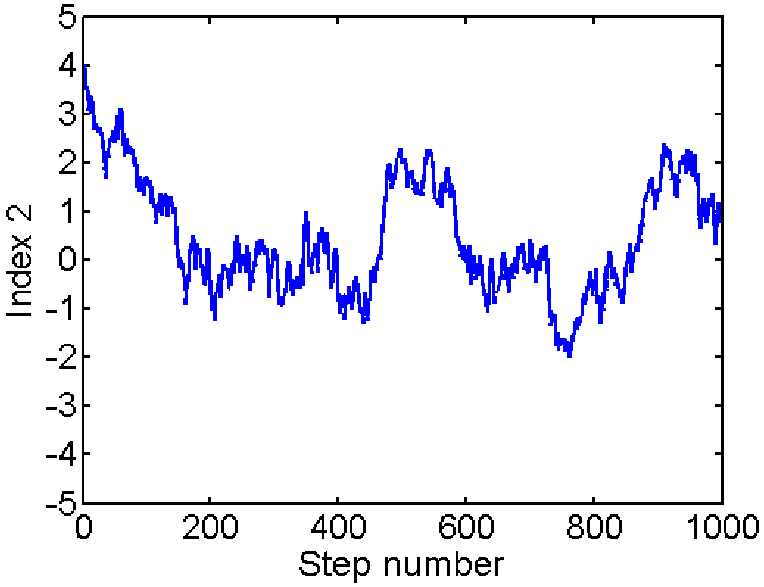
MCMC - Issues

- Identification of convergence to target pdf
  - is sequence in thermodynamic equilibrium with target pdf?
  - validity of estimated properties of parameters (covariance)
- Burn in
  - at beginning of sequence, may need to run MCMC for awhile to achieve convergence to target pdf
- Use of multiple sequences
  - different starting values can help confirm convergence
  - natural choice when using computers with multiple CPUs
- Accuracy of estimated properties of parameters
  - related to efficiency, described above
- Optimization of efficiency of MCMC

MCMC – convergence and burn in

- Example: sequence obtained for 2 D unit-variance Gaussian pdf
  - Metropolis algorithm
  - starting point is (4, 4)
  - trial pdf is Gaussian, σ = 0.2
  - 1000 steps
  - avg acceptance = 0.87
- Observe:
  - large number of steps required before sequence has converged to core region (burn in)
  - hard to tell whether sequence has converged, either from 2D plot or by looking at individual coordinate (convergence)





Annealing

- Introduction of fictitious temperature
  - define functional $\varphi(\mathbf{x})$ as minus-logarithm of target probability

    $$\varphi(\mathbf{x}) = - \log(q(\mathbf{x}))$$

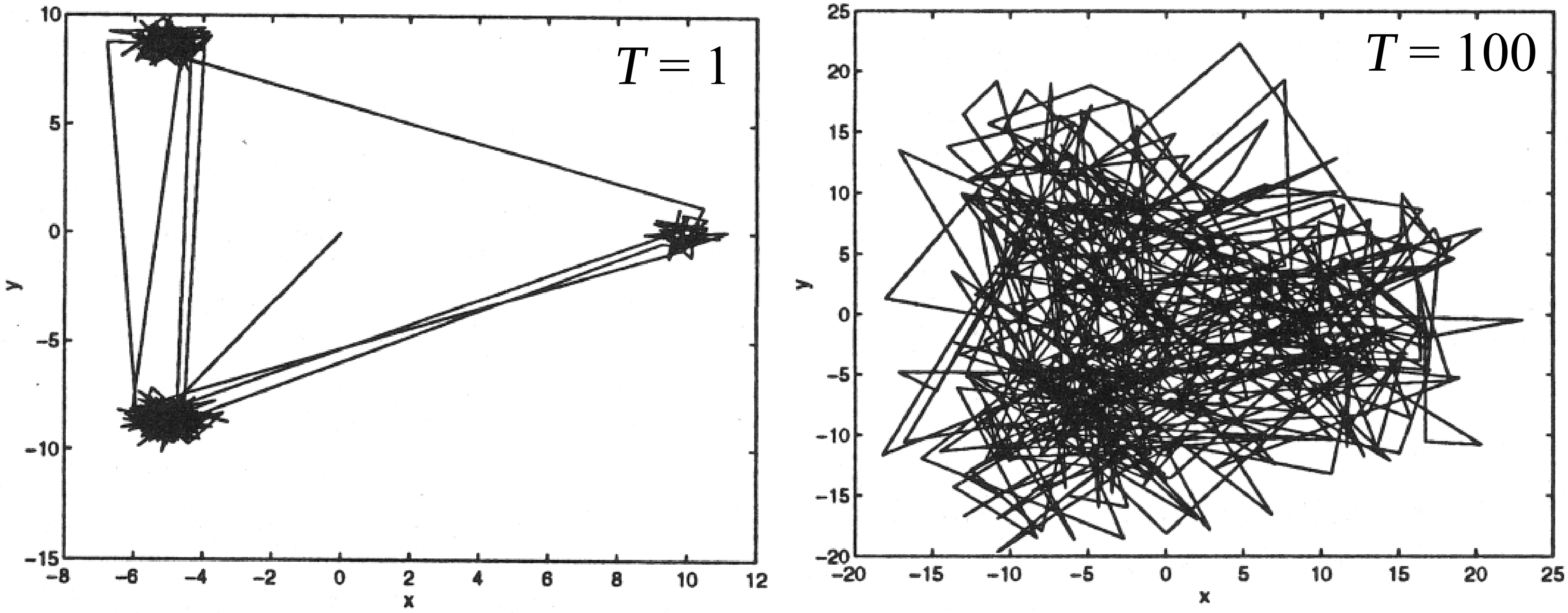
  - scale φ by an inverse "temperature" to form new pdf

    $$q'(\mathbf{x}, T) = \exp[- \varphi(\mathbf{x})/ T]$$

  - $q'(\mathbf{x}, T)$ is flatter than $q(\mathbf{x})$ for $T > 1$ (called annealing)
- Uses of annealing (also called tempering)
  - allows MCMC to move between multiple peaks in $q(\mathbf{x})$
  - simulated-annealing optimization algorithm (takes $\lim T \rightarrow 0$)

Annealing helps handle multiple peaks

- Scale minus-log-prob: $q'(\mathbf{x}, T) = \exp[-\varphi(\mathbf{x})/T]$, T = temperature
- Example: target distribution is three narrow, well separated peaks
- For original distribution ($T = 1$), an MCMC run of 10000 steps rarely moves between peaks
- At temperature $T = 100$ (right), MCMC moves easily between peaks and through surrounding regions



from M-D Wu and W. J. Fitzgerald, *Maximum Entropy and Bayesian Methods* (1996)

Other MCMC algorithms

- Gibbs
  - vary only one component of **x** at a time
  - draw new value of x_j from conditional $q(x_j | x_1\, x_2 \ldots x_{j-1}\, x_{j+1} \ldots)$
- Metropolis-Hastings
  - allows use of nonsymmetric trial functions, $t(\Delta\mathbf{x}; \mathbf{x}_k)$
  - uses acceptance criterion $r = [t(\Delta\mathbf{x}; \mathbf{x}_k)\, q(\mathbf{x}^*)] / [t(-\Delta\mathbf{x}; \mathbf{x}^*)\, q(\mathbf{x}_k)]$
- Langevin technique
  - variation of Metropolis-Hastings approach
  - uses gradient* of minus-log-prob to shift trial function towards regions of higher probability
- Hamiltonian hybrid algorithm
  - based on particle dynamics; requires gradient* of minus-log-prob
  - provides potentially higher efficiency for large number of variables
- Many others

* adjoint differentiation affords efficient gradient calculation

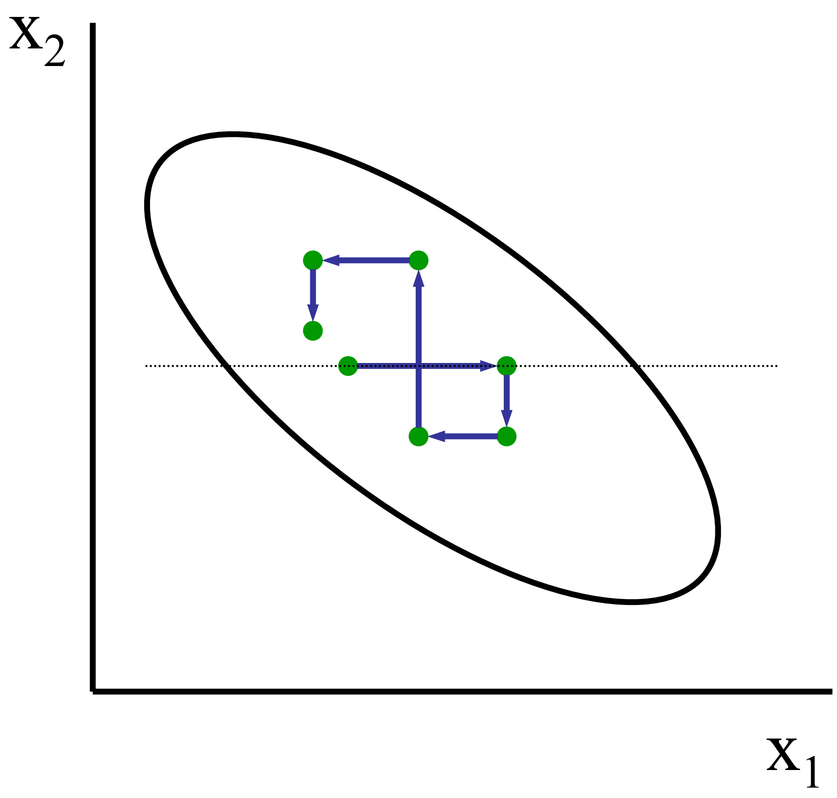
Gibbs algorithm

- Vary only one component of **x** at a time
- Draw new value of x_j from conditional pdf

  $q(x_j| x_1 x_2 \ldots x_{j-1} x_{j+1} \ldots)$

  - algorithm typically used only when draws from *q* are relatively easy to do
- Cycle through all components

Probability(x_1, x_2)



Hamiltonian hybrid algorithm

- Hamiltonian hybrid algorithm
  - called hybrid because it alternates Gibbs & Metropolis steps
  - associate with each parameter x_i a momentum p_i
  - define a Hamiltonian

    $H = \varphi(\mathbf{x}) + \Sigma\, p_i^2/(2\, m_i)$; where $\varphi = -\log\,(q\,(\mathbf{x}\,))$
  - new pdf:

    $$q'(\mathbf{x}, \mathbf{p}) = \exp(-\, H(\mathbf{x}, \mathbf{p})) = q(\mathbf{x})\, \exp(-\Sigma\, p_i^2/(2\, m_i))$$
  - can easily move long distances in ($\mathbf{x}$, $\mathbf{p}$) space at constant H using Hamiltonian dynamics, so Metropolis step is very efficient
  - uses gradient* of φ (minus-log-prob)
  - Gibbs step in constant $\mathbf{p}$ is easy
  - efficiency may be better than Metropolis for large dimensions

* adjoint differentiation affords efficient gradient calculation

Hamiltonian algorithm

- Gibbs step: randomly sample momentum distribution
- Follow trajectory of constant *H* using leapfrog algorithm:

$$p_i(t+\frac{\tau}{2}) = p_i(t) - \frac{\tau}{2}\left.\frac{\partial\varphi}{\partial x_i}\right|_{\mathbf{x}(t)}$$

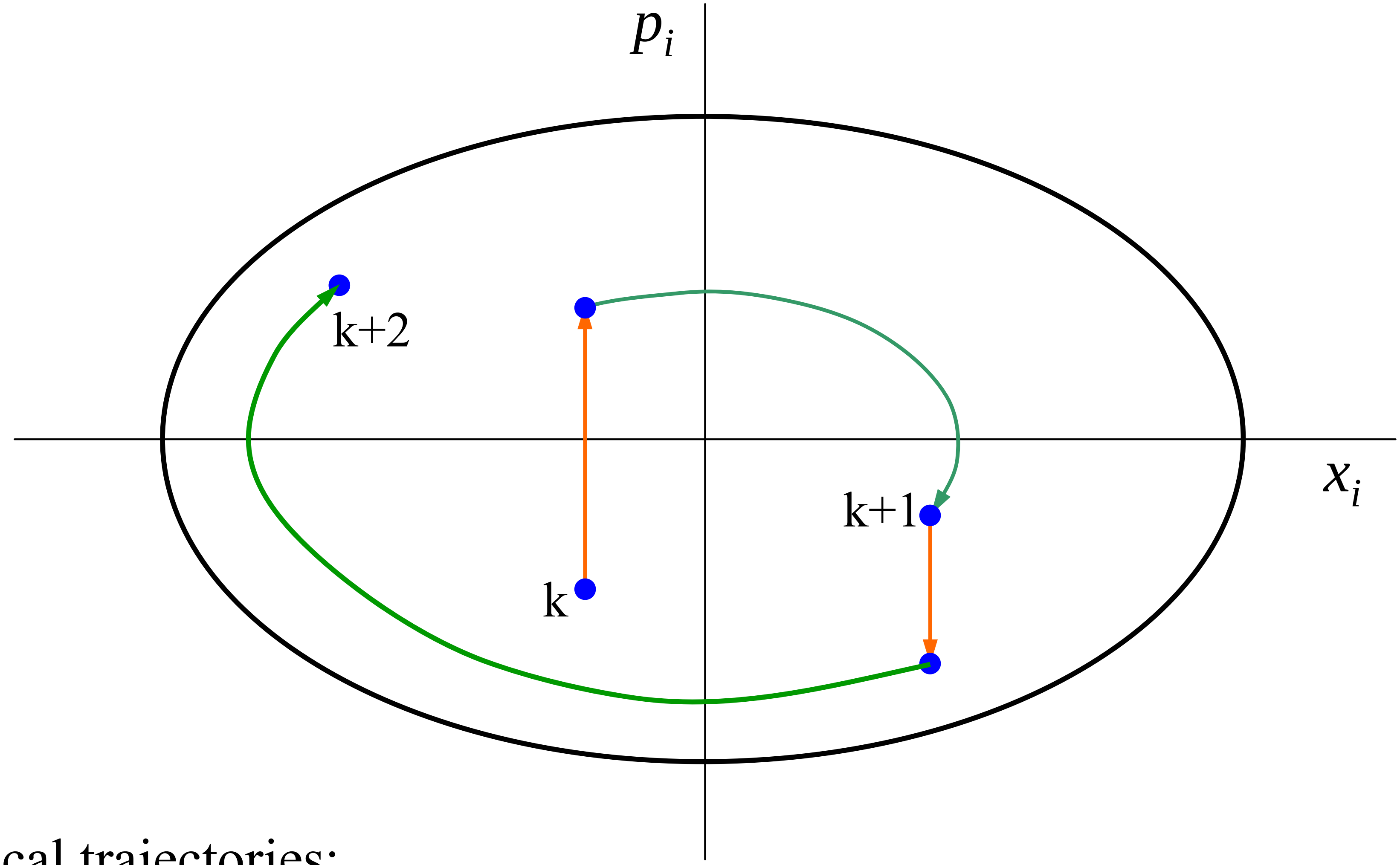
$$x_i(t+\tau) = x_i(t+\tau) + \frac{\tau}{m_i}\,p_i(t+\frac{\tau}{2})$$

$$p_i(t+\tau) = p_i(t+\frac{\tau}{2}) - \frac{\tau}{2}\left.\frac{\partial\varphi}{\partial x_i}\right|_{\mathbf{x}(t+\tau)}$$

where τ is leapfrog time step.

- Repeat leapfrog a predetermined number of times
- Metropolis step: accept or reject on basis of *H* at beginning and end of H trajectory

Hamiltonian hybrid algorithm



Typical trajectories:

red path - Gibbs sample from momentum distribution

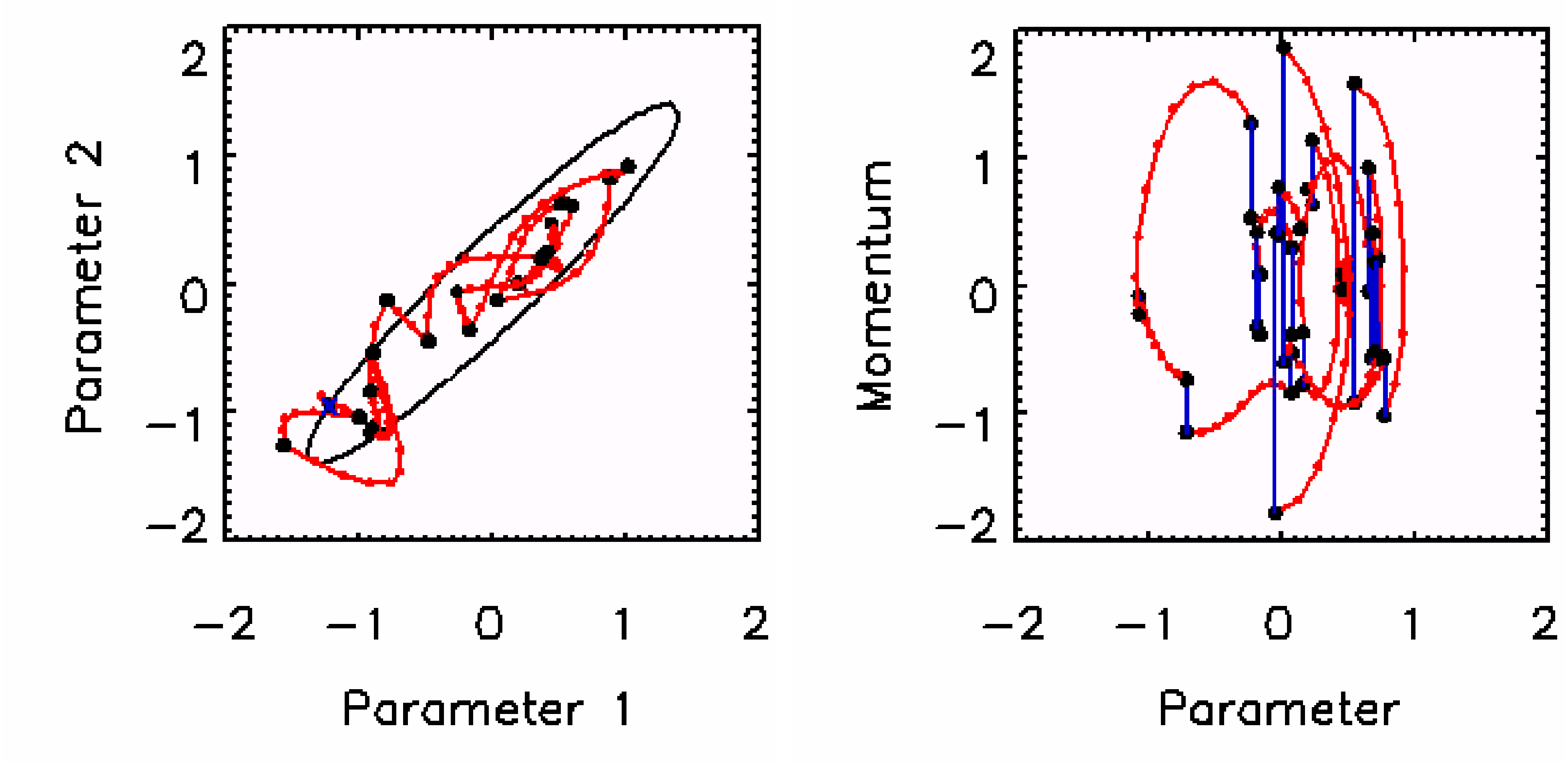
green path - trajectory with constant *H*, follow by Metropolis

Hamiltonian algorithm

- Gibbs step - easy because draws are from uncorrelated Gaussian
- H trajectories followed by several leapfrog steps permit long jumps in (**x**, **p**) space, with little change in *H*
  - specify total time = *T* ; number of leapfrog steps = T/τ
  - randomize *T* to avoid coherent oscillations
  - reverse momenta at end of H trajectory to guarantee that it is symmetric process (condition for Metropolis step)
- Metropolis step - no rejections if *H* is unchanged

- Adjoint differentiation efficiently provides gradient

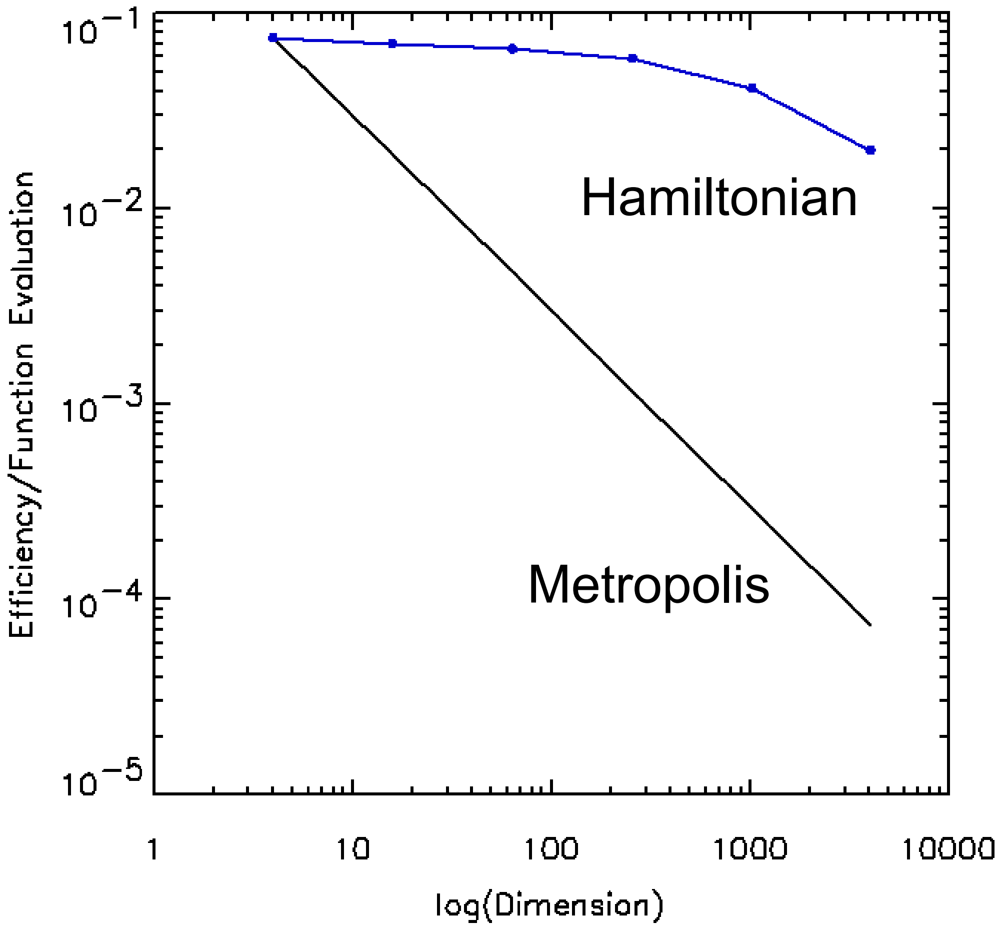
2D correlated Gaussian distribution



- 2D Gaussian pdf with high correlation (r =0.95)
- Length of H trajectories randomized

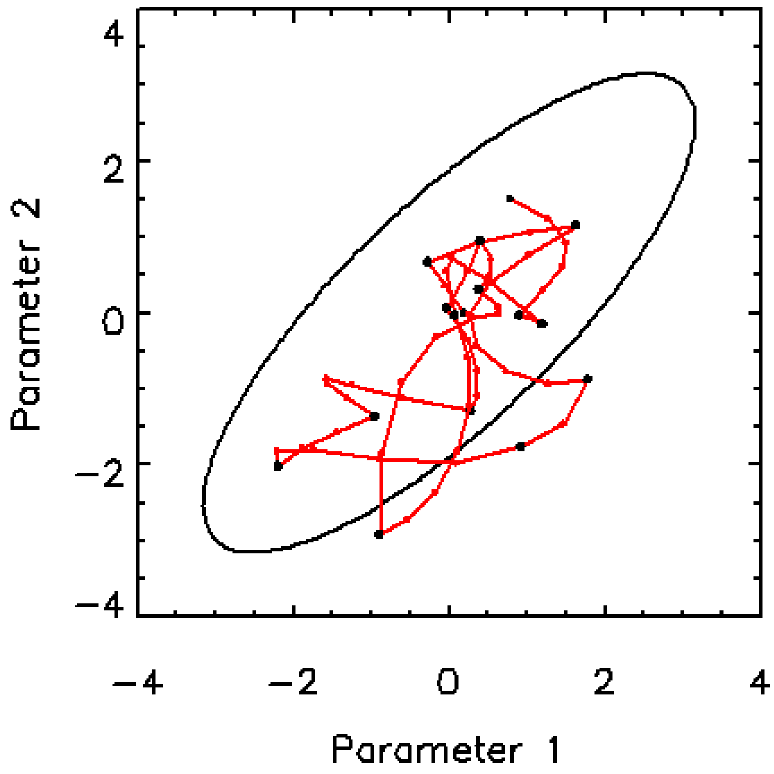
n-D isotropic Gaussian distributions

- Assume that gradient of φ are calculated as quickly as φ itself (e.g., using adjoint differentiation)
- MCMC efficiency versus number dimensions
    - Hamiltonian method: drops little
    - Metropolis method: goes as 0.3/*n*



- Hamiltonian method much more efficient at high dimensions

16D correlated Gaussian distribution



- 16D Gaussian pdf related to smoothness prior based on integral of L2 norm of second derivative
- Efficiency/(function evaluation) =

  2.2% (Hamiltonian algorithm)

  0.11% or 1.6% (Metropolis; without and with covariance adaptation)

Conclusions – Hamiltonian MCMC

- MCMC provides good tool for exploring the Bayesian posterior and hence for drawing inferences about models and parameters
- Hamiltonian method
  - based on Hamiltonian dynamics
  - efficiency for isotropic Gaussians is about 7% per function evaluation, independent of number of dimensions
  - caveat – must be able to calculate gradient of minus-log-posterior in time comparable to the posterior itself (e.g., through adjoint differentiation)
  - much better efficiency than Metropolis for large dimensions
  - more robust to correlations among parameters than Metropolis

Conclusions – MCMC

- MCMC provides good tool for exploring the posterior and hence for drawing inferences about models and parameters
- For valid results, care must be taken to
  - verify convergence of the sequence
  - exclude early part of sequence, before convergence reached
  - be wary of multiple peaks that need to be sampled
- For good efficiency with Metropolis alg., care must be taken to
  - adjust the size and shape of the trial distribution; rule of thumb is to aim for 25% trial acceptance for $5 < n < 100$

- A lot of MCMC research is going on
- Software libraries for MCMC are available for most computer languages, or as stand-alone applications, e.g., OpenBUGS (formerly WinBUGS)

Rossi analysis – example of MCMC

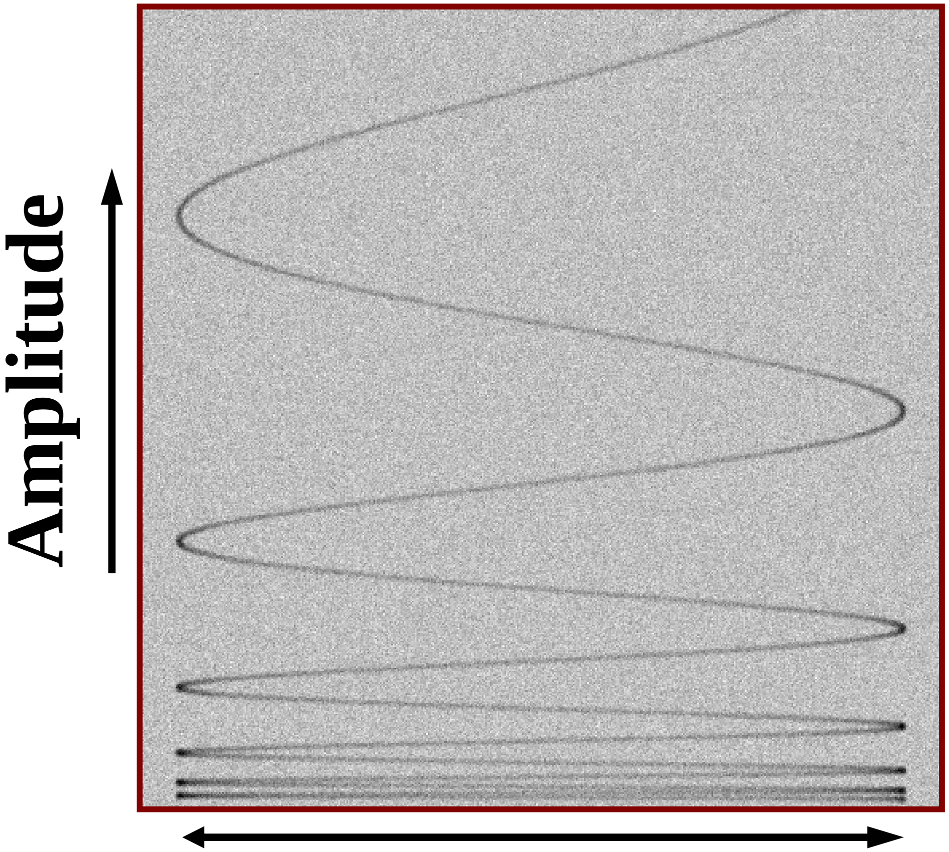
- Goal: measure flux as function of time, Φ(*t*), to obtain alpha, a measure of criticality, versus time

$$\alpha(t) = \frac{1}{\Phi}\frac{d\Phi}{dt} = \frac{d(\ln\Phi)}{dt}$$

- Experimental issues
  - measurements made using Rossi technique
  - signal displayed on oscilloscope, photographed, read
  - recorded signal is band limited
- Analysis complicated by intricate error model for measurements

The Rossi technique

- Rossi technique - photograph oscilloscope screen
  - horizontal sweep is driven sinusoidally in time
  - signal amplitude vertical
- Records rapidly increasing signal while keeping trace in middle of CRT, which minimizes oscilloscope nonlinearities



$$x = x_R \cos(2\pi f_R t + \phi_0)$$

Bayesian analysis of an experiment

- The pdf describing uncertainties in model parameter vector **a**, called **posterior**:
  - $p(\mathbf{a}|\mathbf{d}) \sim p(\mathbf{d}|\mathbf{d}^*)\, p(\mathbf{a})$ (Bayes law)

    where **d** is vector of measurements, and

    d*(**a**) is measurement vector predicted by model
  - $p(\mathbf{d}|\mathbf{d}^*)$ is likelihood, probability of measurements **d** given the values **d*** predicted by simulation of experiment
  - $p(\mathbf{a})$ is prior; summarizes previous knowledge of **a**
  - "best" parameters estimated by
    - maximizing posterior (called MAP solution)
    - mean of posterior
  - uncertainties in **a** are fully characterized by $p(\mathbf{a}|\mathbf{d})$

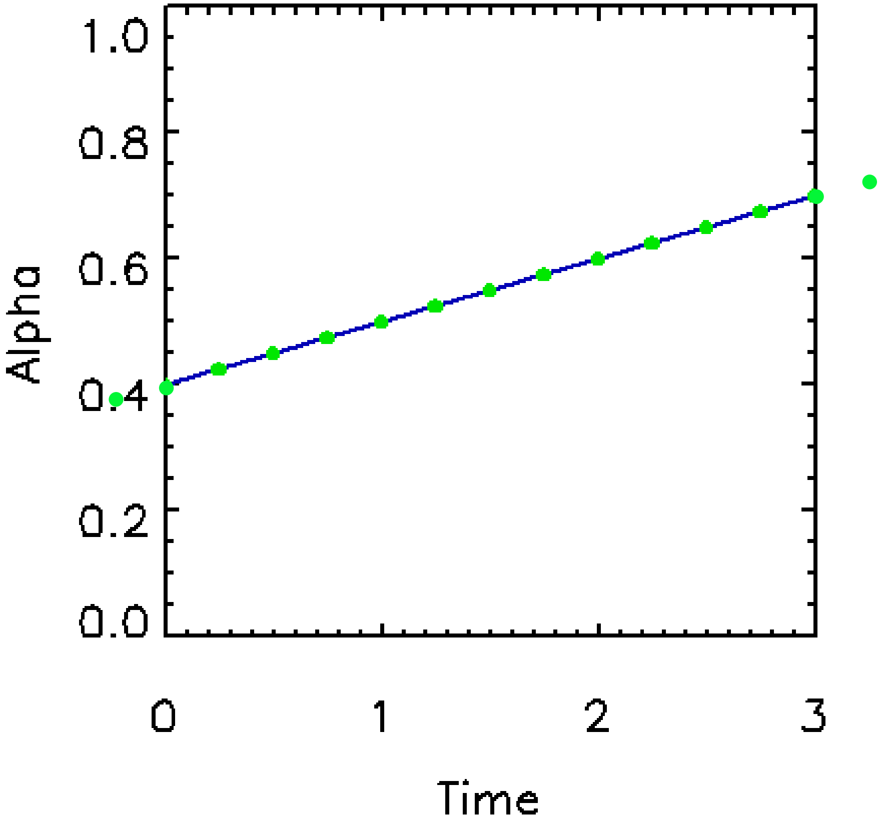
Cubic spline expansion of alpha curve

- Expand $\alpha(t)$ in terms of basis functions:

$$\alpha(t) = \sum_k a_k \, \phi\left[\frac{t - t_k}{\Delta t}\right]$$

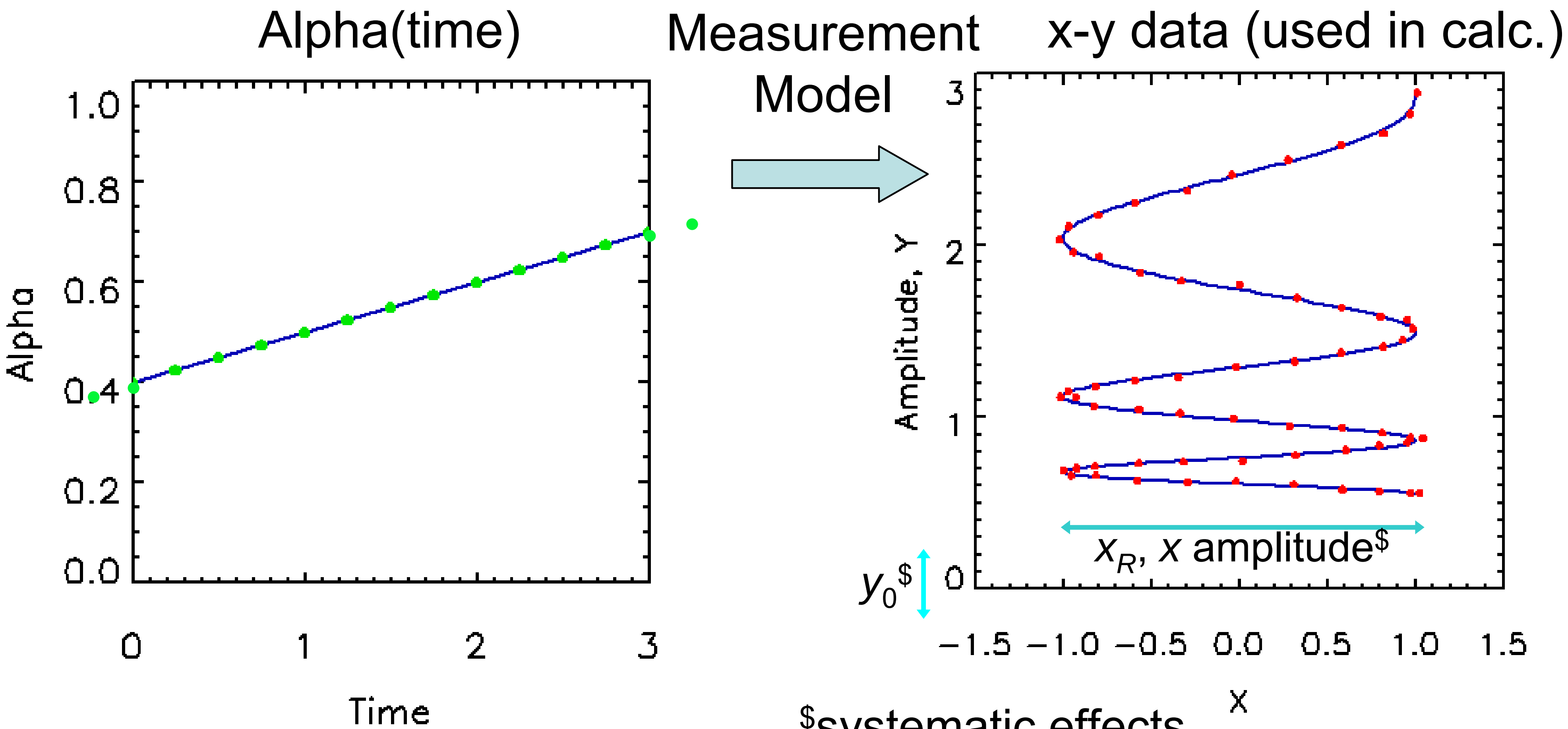
  where
  - a_k is the expansion coefficient,
  - ϕ is a spline basis function,
  - t_k is the position of the kth knot
  - Δt is the knot spacing
- Use 15 evenly-space knots
  - spacing chosen on basis of limited bandwidth of signal y
  - two are outside data interval to handle end conditions
- Parameters a_k are to be determined

Alpha(time)



Modeling the Rossi data

- $\alpha(t)$ represented as cubic spline
- measurement model predicts data
- can include systematic effects of measurement system



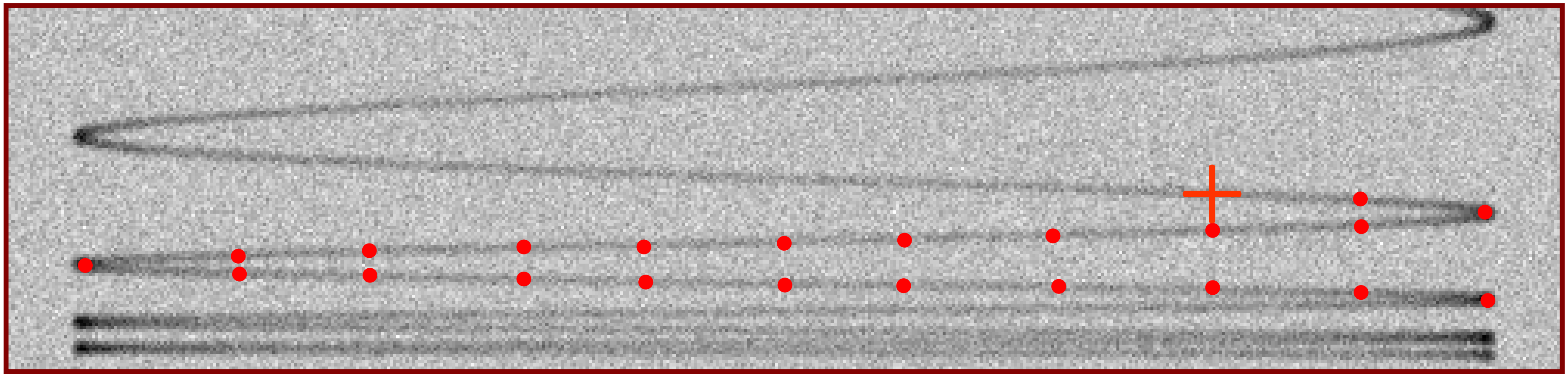
Alpha(time)

Measurement Model

x-y data (used in calc.)

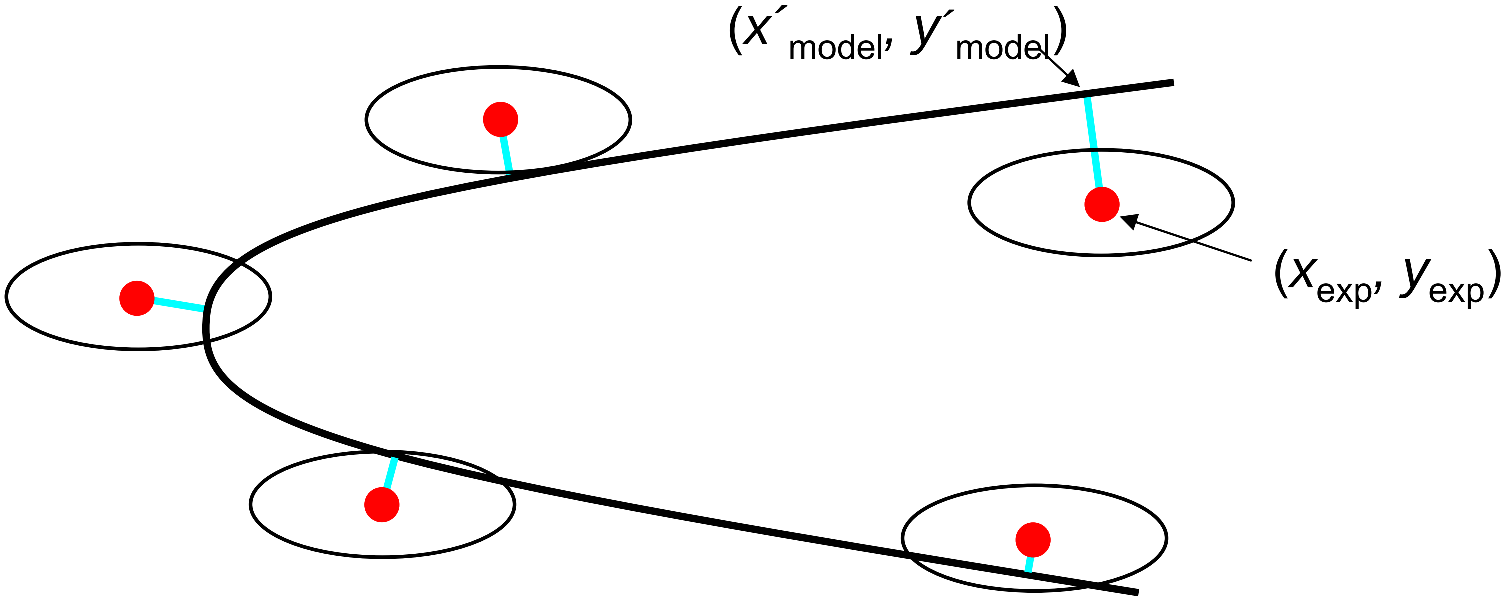
$systematic effects

Reading a Rossi trace



- Technician reads points by centering cross hairs of a reticule on trace; computer records positions, $\{x_i, y_i\}$
- Points are read with intent to:
  - place point at peaks
  - achieve otherwise arbitrary placement along curve with even spacing along trace

Likelihood model - uncertainties in Rossi data



- minus-log-likelihood, p(**d**|**a**), for measured point (x_{exp} , y_{exp}):

$$\Delta\frac{\chi^2}{2} = \frac{(x_{\text{exp}} - x'_{\text{model}})^2}{2\sigma_x^2} + \frac{(y_{\text{exp}} - y'_{\text{model}})^2}{2\sigma_y^2}$$

where $(x'_{\text{model}}, y'_{\text{model}})$ is the model point closest to $(x_{\text{exp}}, y_{\text{exp}})$

Smoothness constraint

- Cubic splines tend to oscillate in some applications
- Smoothness of $\alpha(t)$ can be controlled by minimizing

$$S(\alpha) = T^3 \int \left| \frac{d^2\alpha}{dt^2} \right|^2 dt$$

where T is the time interval; T^3 factor removes T dependence

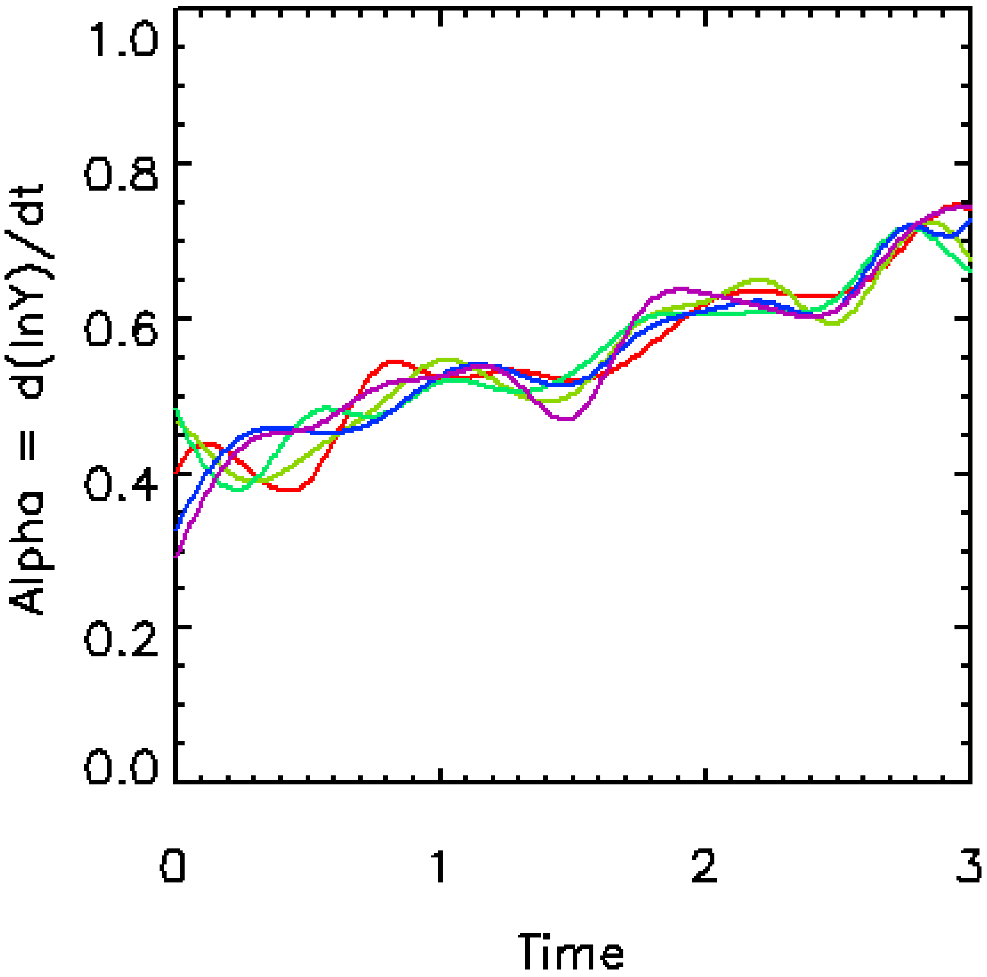
- Smoothness can be incorporated in Bayesian context by setting prior on spline coefficients to

$$-\log p(\mathbf{a}) = \lambda\, S(\alpha(\mathbf{a}))$$

- Hyperparameter λ can be determined in Bayesian approach by maximizing $p(\lambda|\mathbf{d})$

MCMC - alpha uncertainty

- MCMC samples from posterior
  - plot shows several $\alpha(t)$ curves consistent with data
  - uncertainties in model visualized as variability among curves
- Smoothness parameter, $\lambda = 0.4$



MCMC – estimation of λ

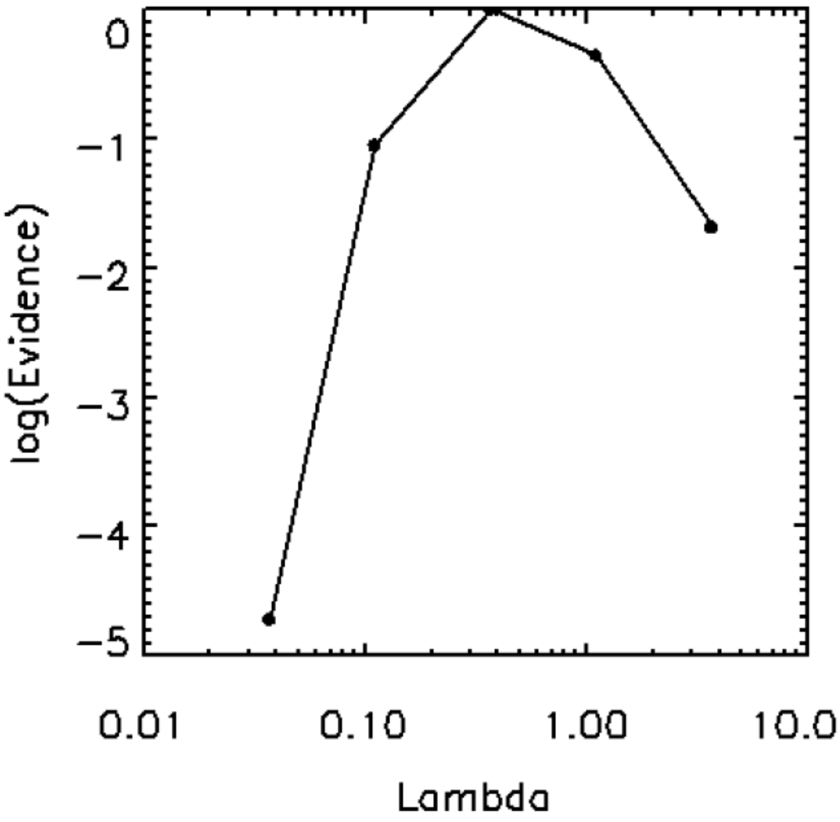
- Strength of smoothness prior given by λ
- Determine λ using Bayes law

$$p(\lambda\,|\,\mathbf{d}) = \int p(\mathbf{a},\lambda\,|\,\mathbf{d})\,d\mathbf{a}$$

$$\propto \int p(\mathbf{d}\,|\,\mathbf{a},\lambda)\;p(\mathbf{a},\lambda)\,d\mathbf{a}$$

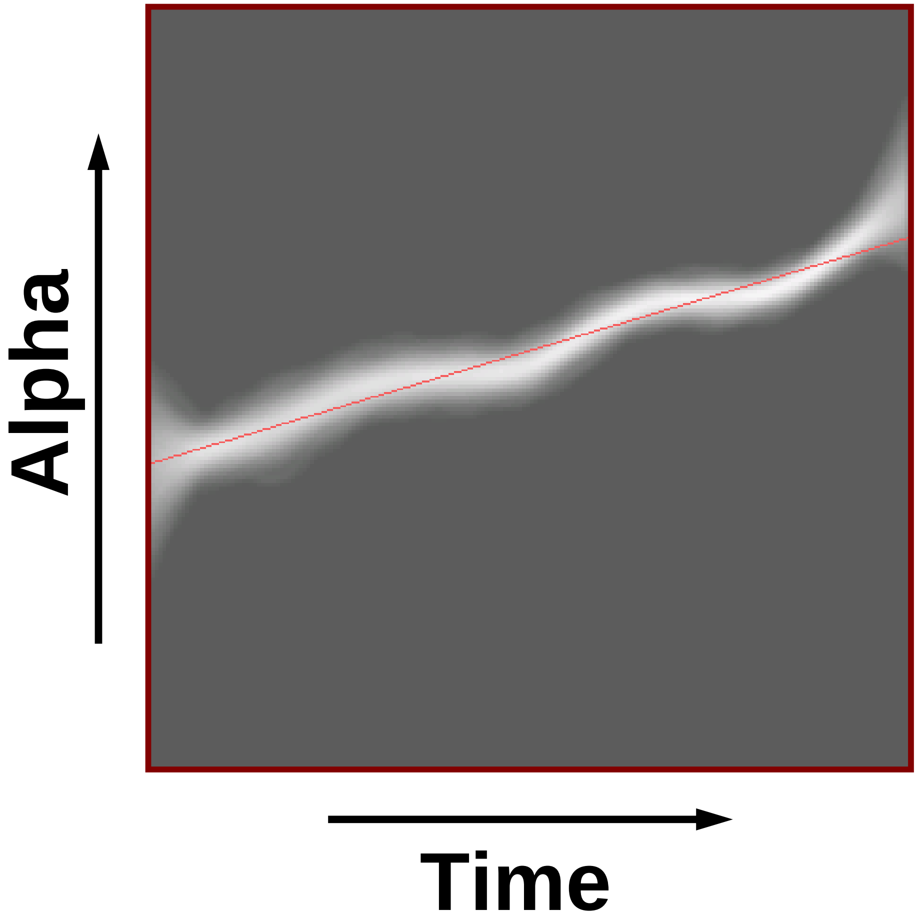
$$= p(\lambda)\int p(\mathbf{d}\,|\,\mathbf{a},\lambda)\;p(\mathbf{a})\,d\mathbf{a}$$

- Last integral, called **evidence**, is estimated as value of integrand at its peak times its volume
- Volume given by determinant of covariance matrix of **a**, estimated using MCMC sequence
- At maximum λ = 0.4



MCMC - Alpha

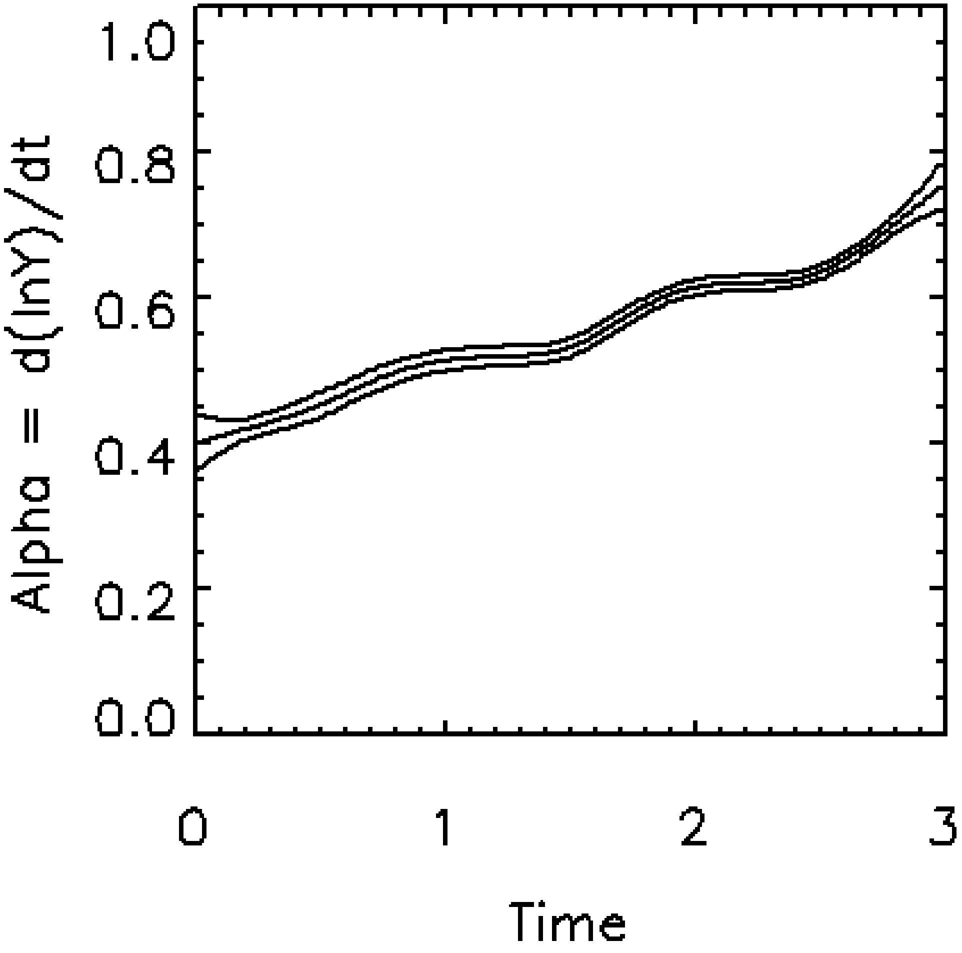
- For MCMC sequence with 10^5 samples, image shows accumulated MCMC curves in alpha domain
- Effectively shows PDF for uncertainty distribution in alpha, estimated from data
- However, does **not** show correlations between uncertainties at two different times, as do individual MCMC samples



$\lambda = 0.4$ (best value)

MCMC - Alpha

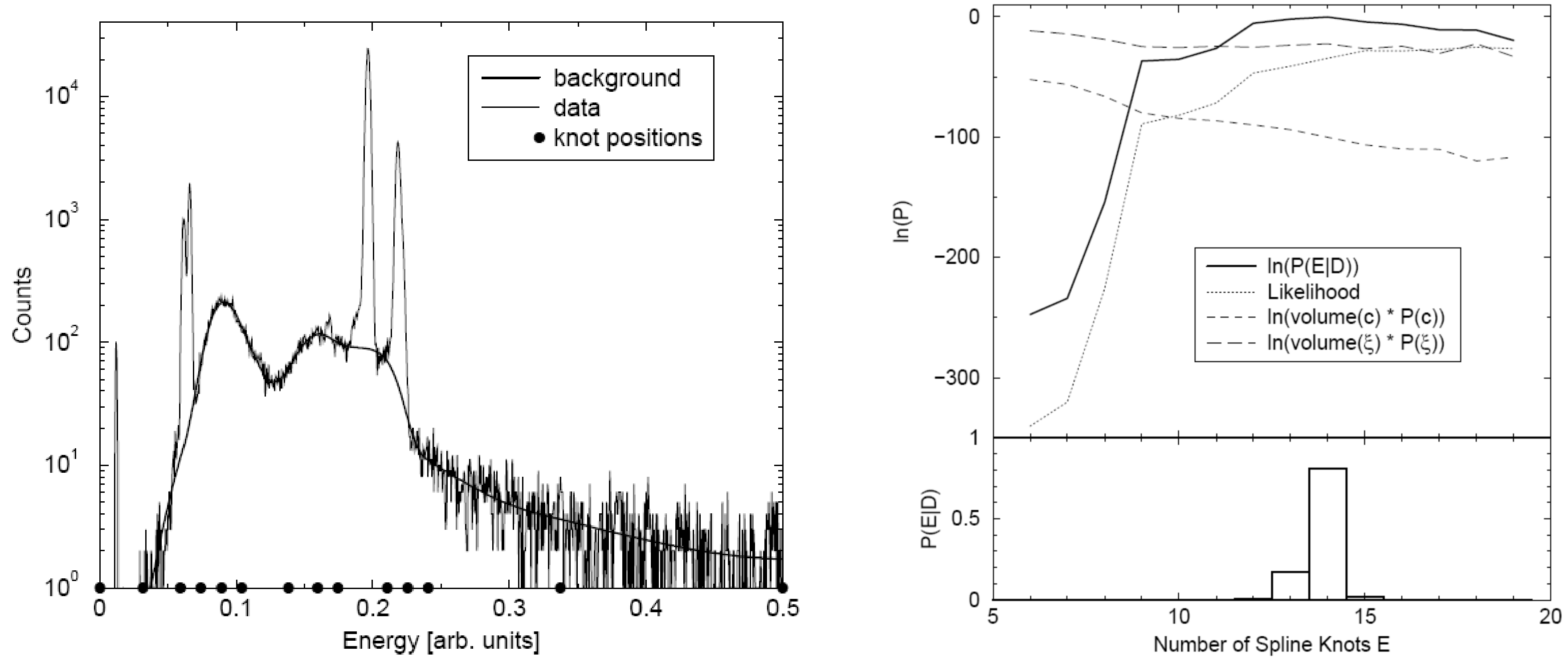
- Interpreting accumulated alpha curve as a PDF, one can estimate $\alpha(t)$ in terms of
  - posterior mean
  - posterior max. (MAP estimate)
- Or characterize uncertainties
  - standard deviations
  - covariance matrix (correlations)
  - credible intervals (envelope)
- Plot on right shows
  - posterior mean
  - posterior mean +/- standard dev. (one standard dev. envelope)



$\lambda = 0.4$ (best value)

Background estimation in spectral data

- Problem: estimate background for PIXE spectrum
- Approach is based on assuming background is smooth and treating resonances as outlying data
- Fully Bayesian calculation using MCMC to estimate spline parameters, their knot positions, and number of knots



from Fischer et al., Phys. Rev. E **61**, 1152 (2000)

Summary

In this tutorial:

- MCMC provides random draws from calculational pdf
- Metropolis algorithm
  - choosing the trial function
  - diagnositics
- Hamiltonian (hybrid) algorithm
  - potentially more efficient than Metropolis, provided ∇φ can be calculated as quickly as φ
- Examples:
  - analysis of Rossi traces; complex likelihood function
    - possibility of elaborating on model to include systematic effects
  - background estimation using splines and treating signal as outliers