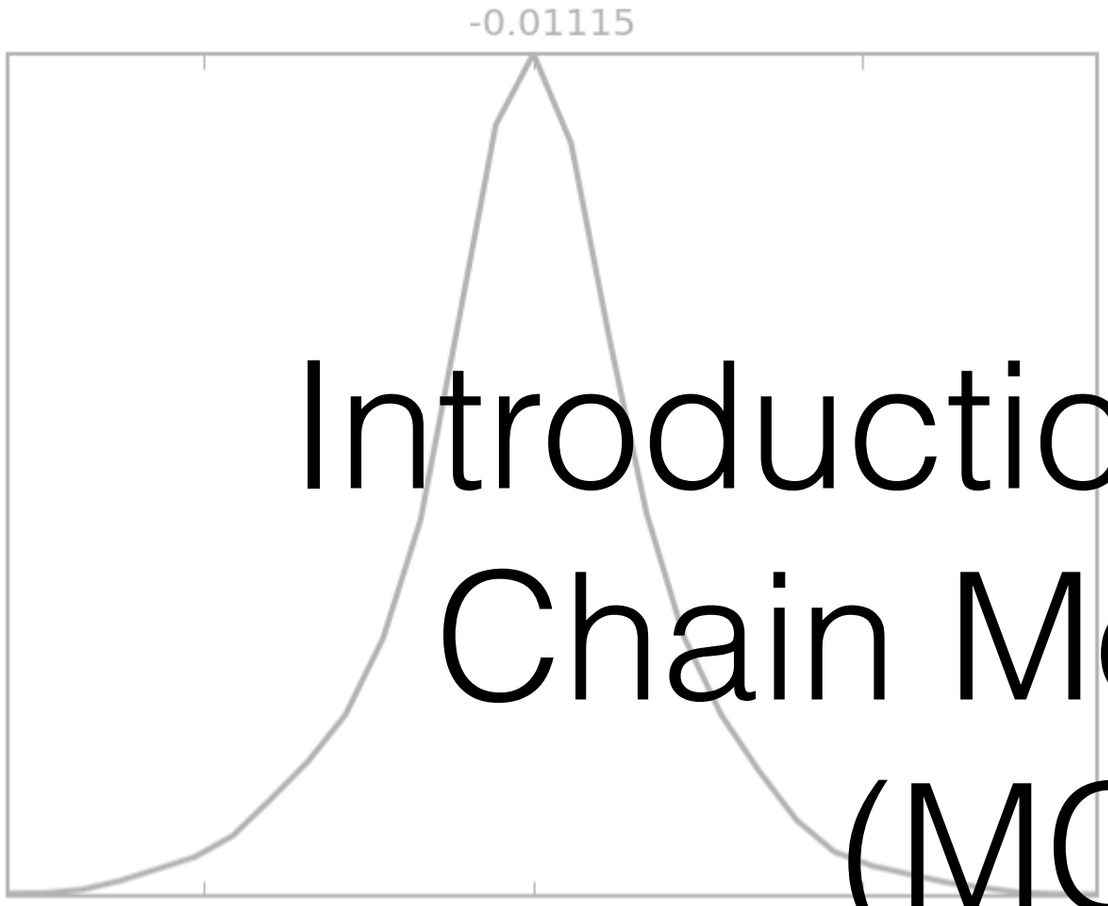
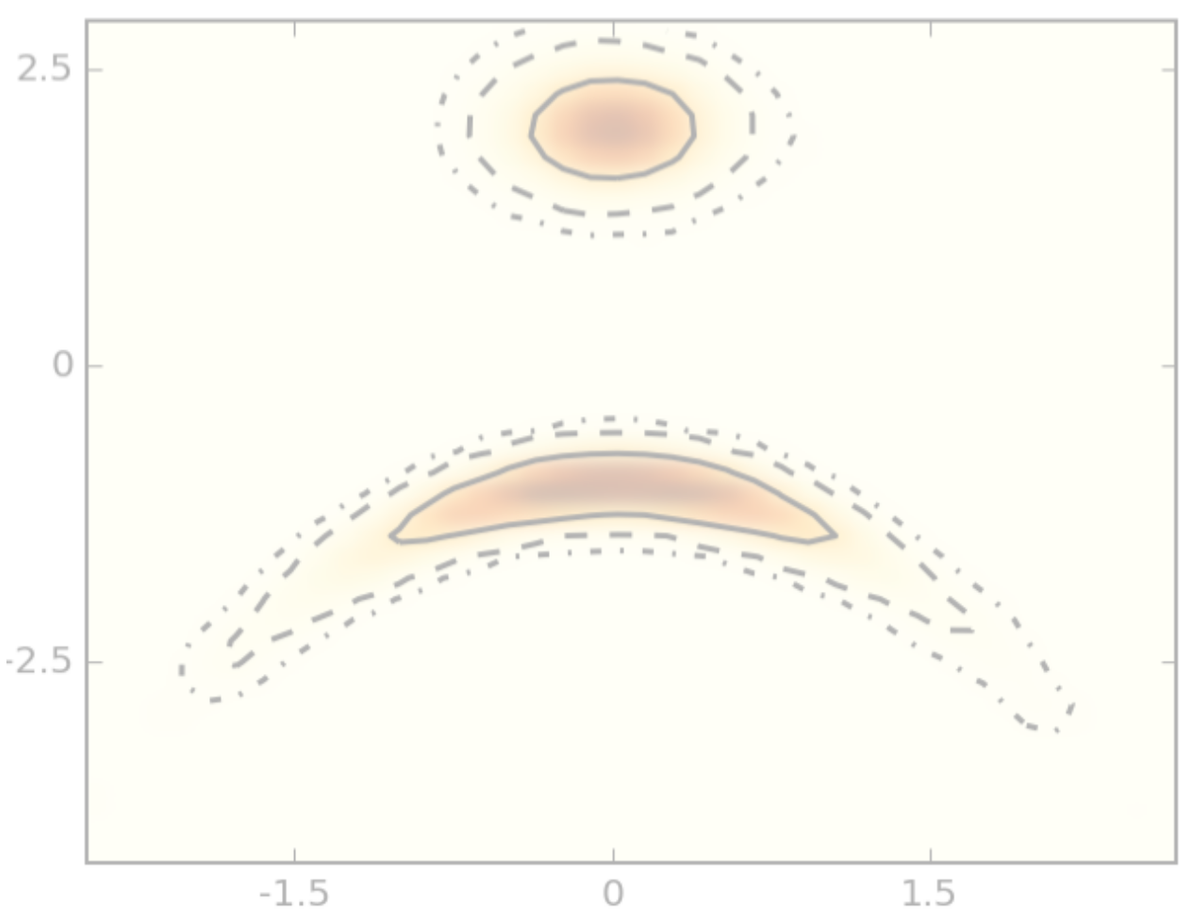
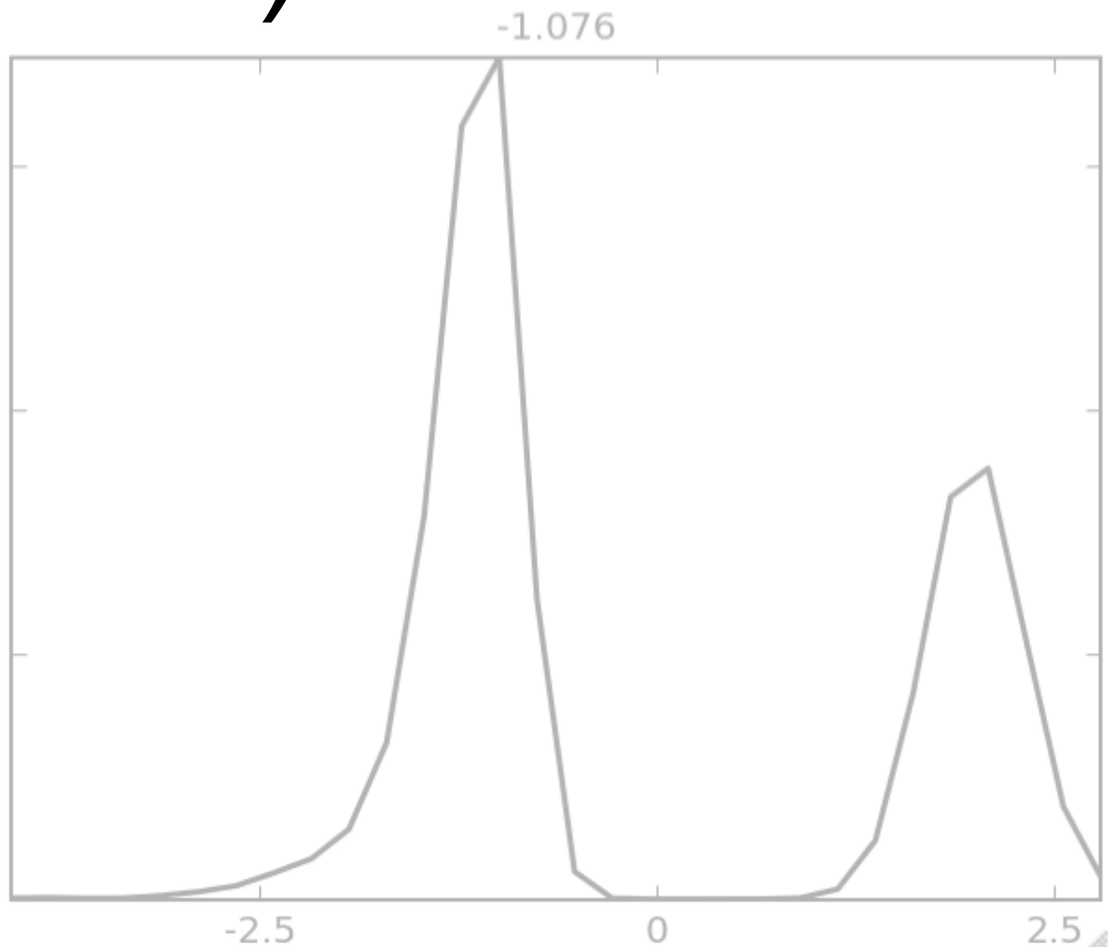


Introduction to Markov Chain Monte Carlo (MCMC)





Outline

- Motivation for MCMC
- MCMC algorithm
- Parallel Tempering (PT)
- Reverse Jump MCMC (RJMCMC)
- Tips and Tricks

Motivation

- MCMCs are used to map out large and complicated parameter spaces

- In modern PTA data analysis we need to explore the likelihood function. We either seek to maximize or to map the entire likelihood surface.

- Noise + GW models now require at least 10s of unknown parameters to fully describe the data. This number can be in the 1000s for full PTA.

- Impossible to grid parameter space and other methods such as nested sampling are inefficient in large parameter spaces (this is slightly better now but we are going to focus on MCMC)

MCMC Algorithm (1)

- MCMCs stochastically export parameter space in such a way that the histogram of their samples approaches the target distribution.

- Markovian: Evolution of chain on depends on current state and a fixed transition matrix.

- Chain will converge to target distribution if transition matrix is:

  - *Irreducible*: For any state of the chain, there is a non-zero probability to transition to any other position in the space.

  - *Aperiodic: Chain should not get trapped in cycles*

MCMC Algorithm (2)

- These conditions are satisfied if detailed balance is satisfied

$$\pi(x_i)T(x_{i+1}|x_i) = \pi(x_{i+1})T(x_i|x_{i+1})$$

- Don't know transition matrix a-priori so use proposal $q(x_{i+1}|x_i)$ which does not likely satisfy detailed balance.

- To satisfy detailed balance we add an acceptance rate

$$\pi(x_i)q(x_{i+1}|x_i)\kappa(x_{i+1}|x_i) = \pi(x_{i+1})q(x_i|x_{i+1})$$

- Solving for the transition probability

$$\kappa(x_{i+1}|x_i) = \min\left(1, \frac{\pi(x_{i+1})q(x_i|x_{i+1})}{\pi(x_i)q(x_{i+1}|x_i)}\right) = \min(1, H)$$

MCMC Algorithm (3)

1. Initialize x_0

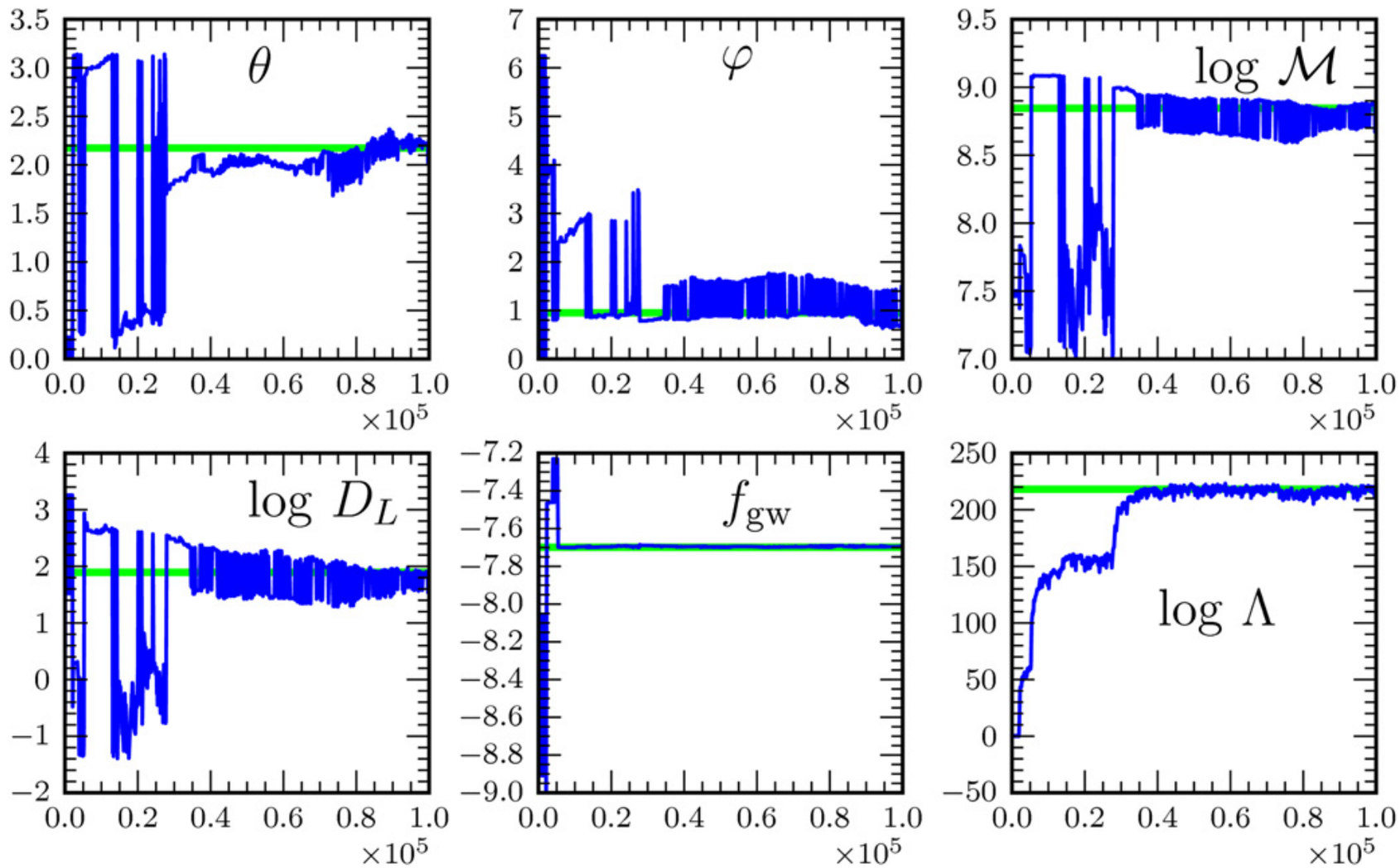
2. for i = 0 to N-1

   - Sample $u \sim U(0,1)$
   - Sample $x_* \sim q(x_*|x_i)$
   - if $u < \min\left(1, \frac{\pi(x_{i+1})q(x_i|x_{i+1})}{\pi(x_i)q(x_{i+1}|x_i)}\right)$
     - $x_{i+1} = x_*$
   - else

     $x_{i+1} = x_i$

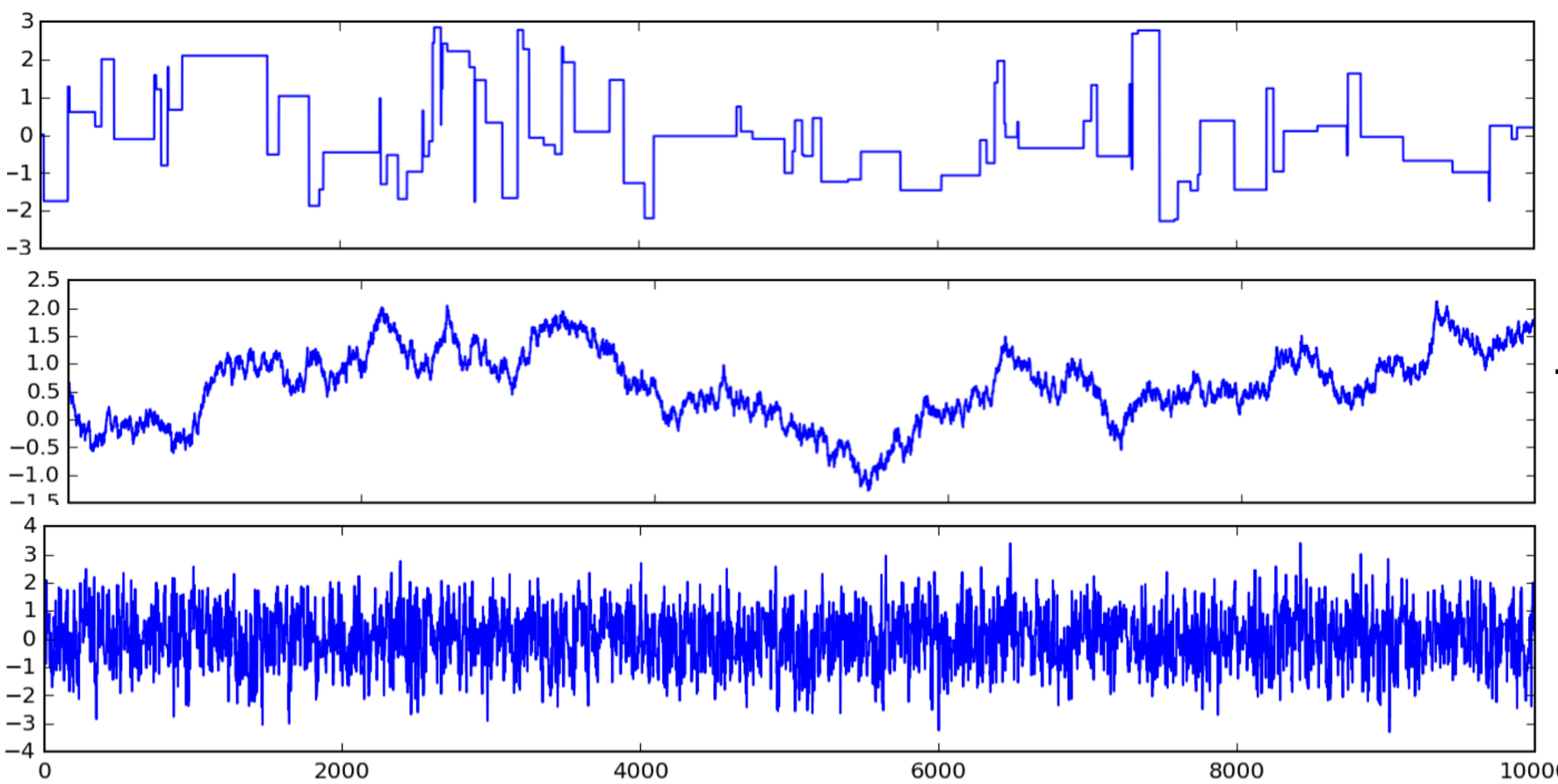
Burn-in

- All of the properties and derivations of the previous slides are only true once the chain is at equilibrium (i.e. it is exploring the high probability region of the posterior distribution)



Jump Proposals

- To effectively explore the parameter space we must have jump proposals that quickly move around the parameter space.
- Gaussian distribution is typical $q(x_{i+1}|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_{i+1}-x_i)^2}{2\sigma^2}}$



Too big

Too small

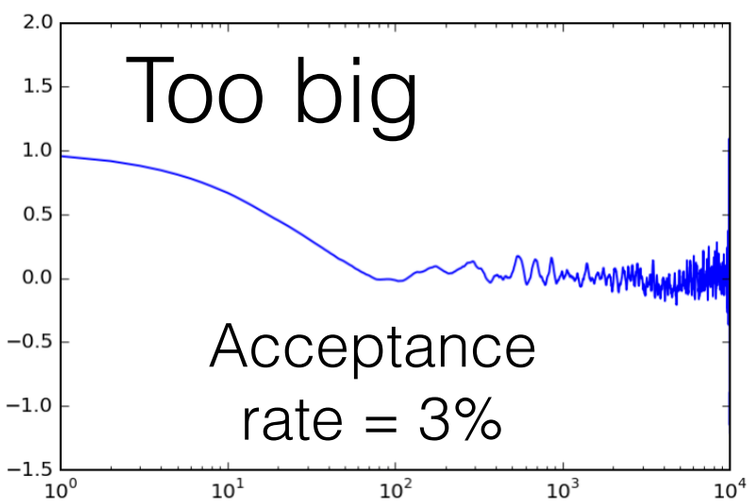
Just right

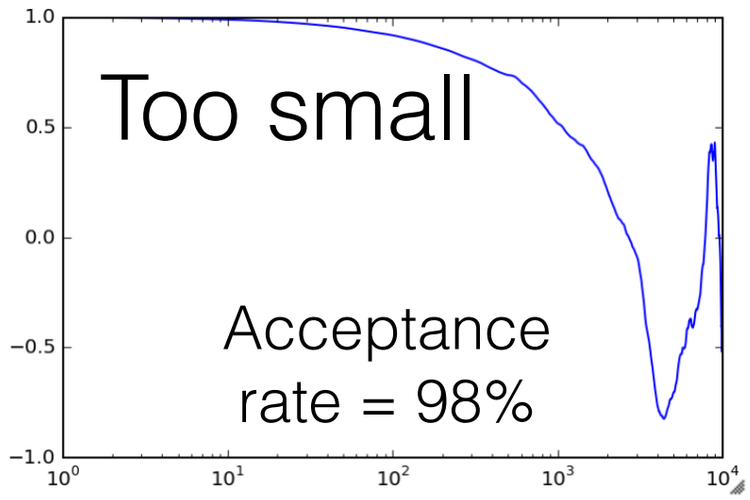
Some Diagnostics

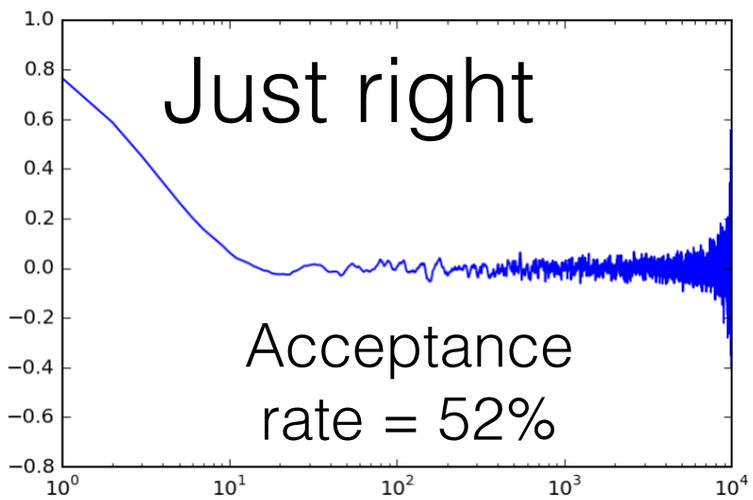
- One indicator of how well a chain is mixing (i.e., how well it is exploring the parameter space) is the autocorrelation function and autocorrelation length

$$ACF_t(x) = \frac{\frac{1}{N-1}\sum_{i=1}^{N-t}(x_i - \bar{x})(x_{i+t} - \bar{x})}{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

- Acceptance rate ~55% for 1-d, 23% for >5-d



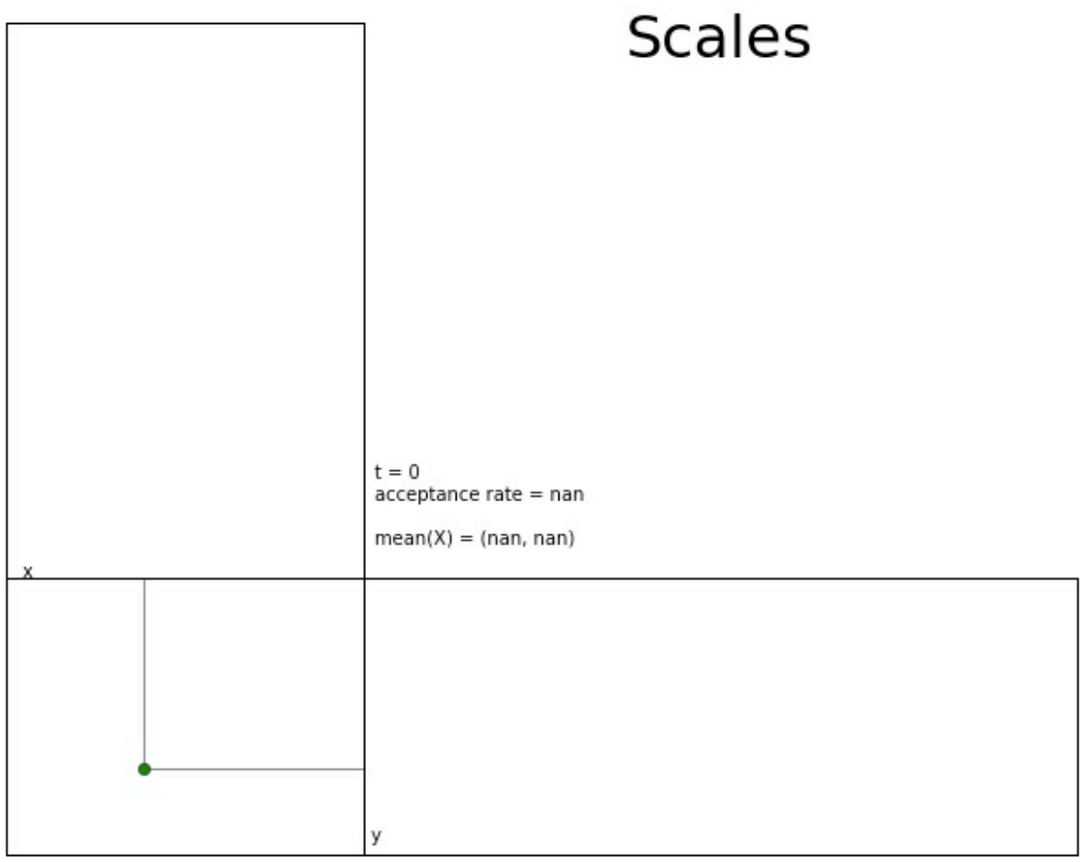




Different jump sizes

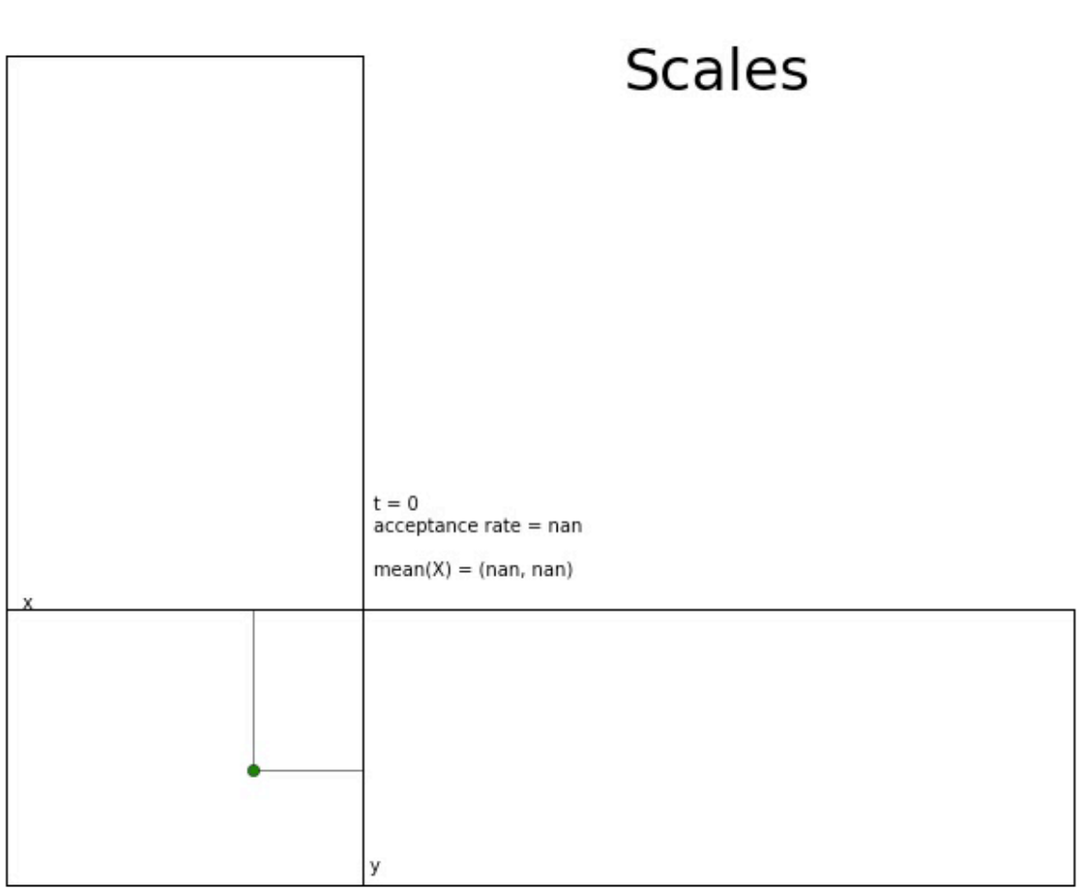
- Ok so we just play with the step size until we get a low ACL and good acceptance right?

Equal step size



credit: Rutger van Haasteren & Justin Ellis

Different Step Size



credit: Rutger van Haasteren & Justin Ellis

Correlated Parameters: Fisher Matrix

- Many times our parameters are correlated and un-correlated proposals do poor job mixing

- For strong signals and analytically tractable problems we can use the Fisher covariance matrix in our gaussian jumps

$$\Gamma_{ij} = -\left\langle \frac{\partial^2}{\partial\theta_i \partial\theta_j} \pi(\vec{\theta}) \right\rangle$$

- Not computationally feasible for many of our problems because usually these derivatives involve large matrix multiplications. However we do use them occasionally, especially for timing model parameters.

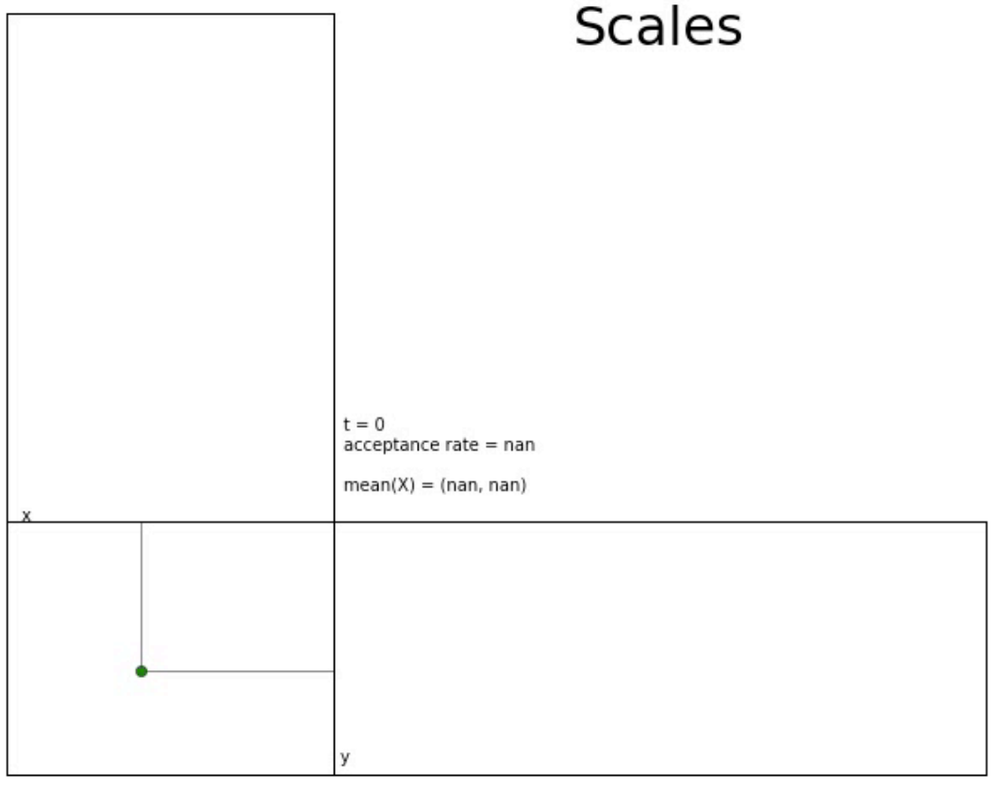
Correlated Parameters: Differential Evolution (DE)

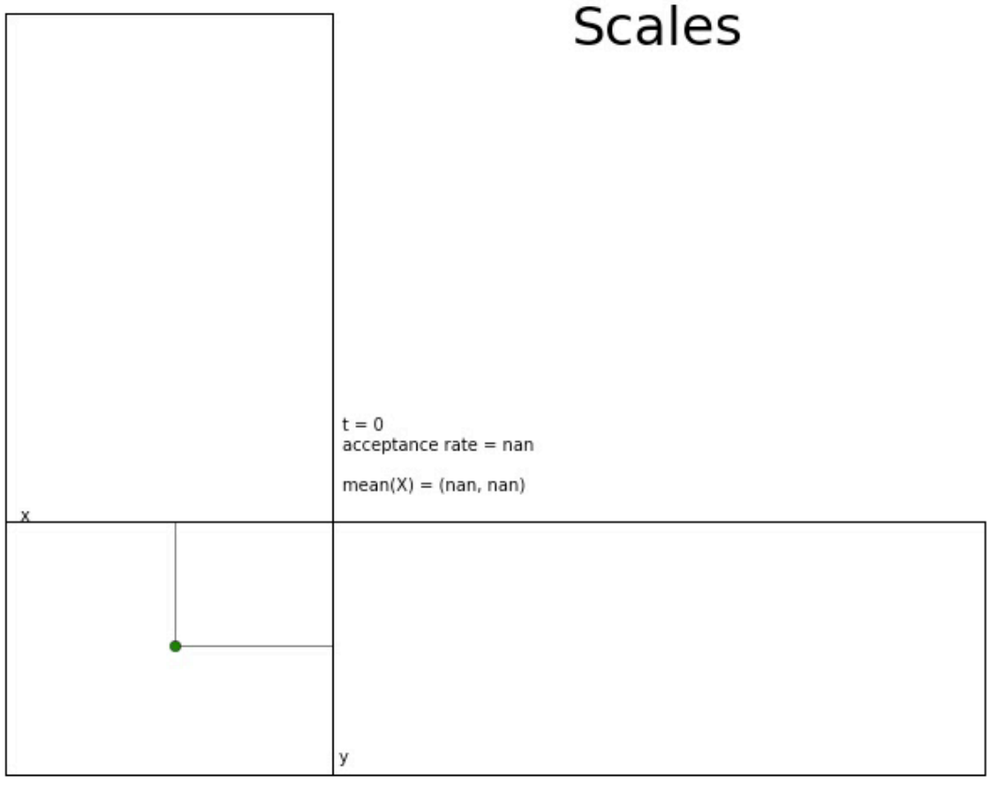
- So the Fisher matrix may not be tractable, a simple yet effective jump is known as Differential Evolution.

- This jump makes use of the past history of the chain to propose jumps

$$\vec{x}_* = \vec{x}_i + \gamma(\vec{x}_k - \vec{x}_j)$$

k and j are randomly chosen iterations of the chain history

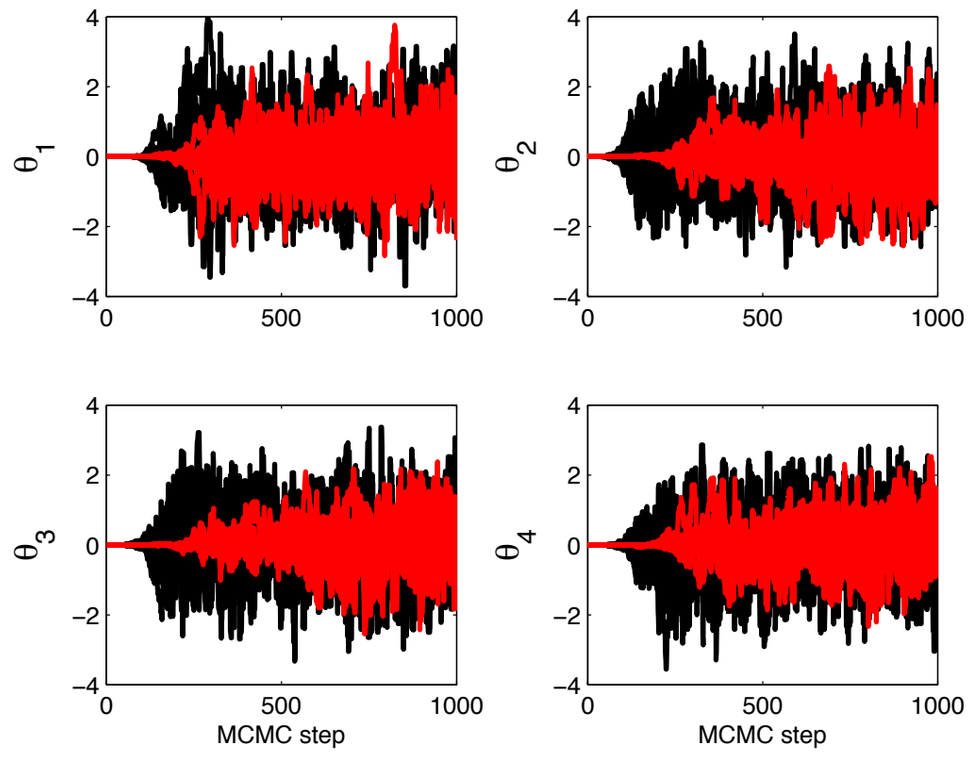




Correlated Parameters: Adaptive Metropolis (AM)

- Adaptive metropolis proposals are a multi-variate gaussian proposal with a covariance matrix that is built from previous samples in the chain.

- If the target distribution is Gaussian then this jump is optimal (i.e. the proposal approaches the target distribution).



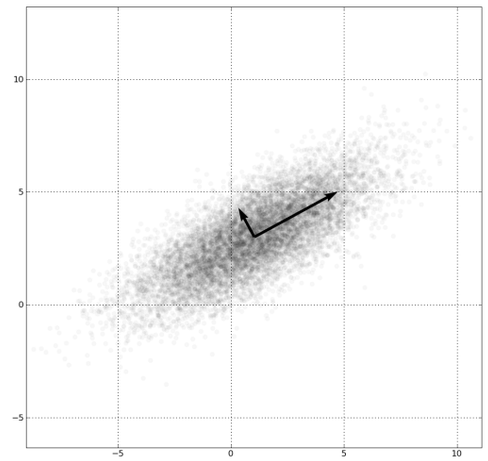
Correlated Parameters: Single Component Adaptive Metropolis (SCAM)

- Both Differential Evolution and AM work great in low dimensions but in large dimensions they perform poorly because they propose a jump in every parameter and not just a few.

- SCAM uses the same covariance matrix as AM but only jumps in one correlated parameter at a time.

- Decompose Covariance into eigenvectors, $\hat{e}_{(j)}$ and eigenvalues λ_j

$$\vec{x}_* = \vec{x}_i + \frac{\delta}{\sqrt{\lambda_i}}\hat{e}_{(i)}$$



Other options

- The standard Metropolis Hastings algorithm with the jumps described so far does a good job on most PTA applications.

- There are other MCMC samplers that are useful but currently not used much in PTAs

1. Gibbs Sampling: Draw directly from *conditional* probability distribution. Caveat: must be able to analytically or easily numerically draw from conditional pdf

2. Hamiltonian Monte-Carlo: Uses information about the gradient of the pdf and evolves the chains through a modified version of Hamiltons equations. Caveat: much more complicated and extremely sensitive to tuning parameters

3. Ensemble Samplers: uses a cloud of different MCMC chains and proposes jumps based on current position of all chains. Caveats: seems inefficient in large parameters spaces but needs further looking in to.

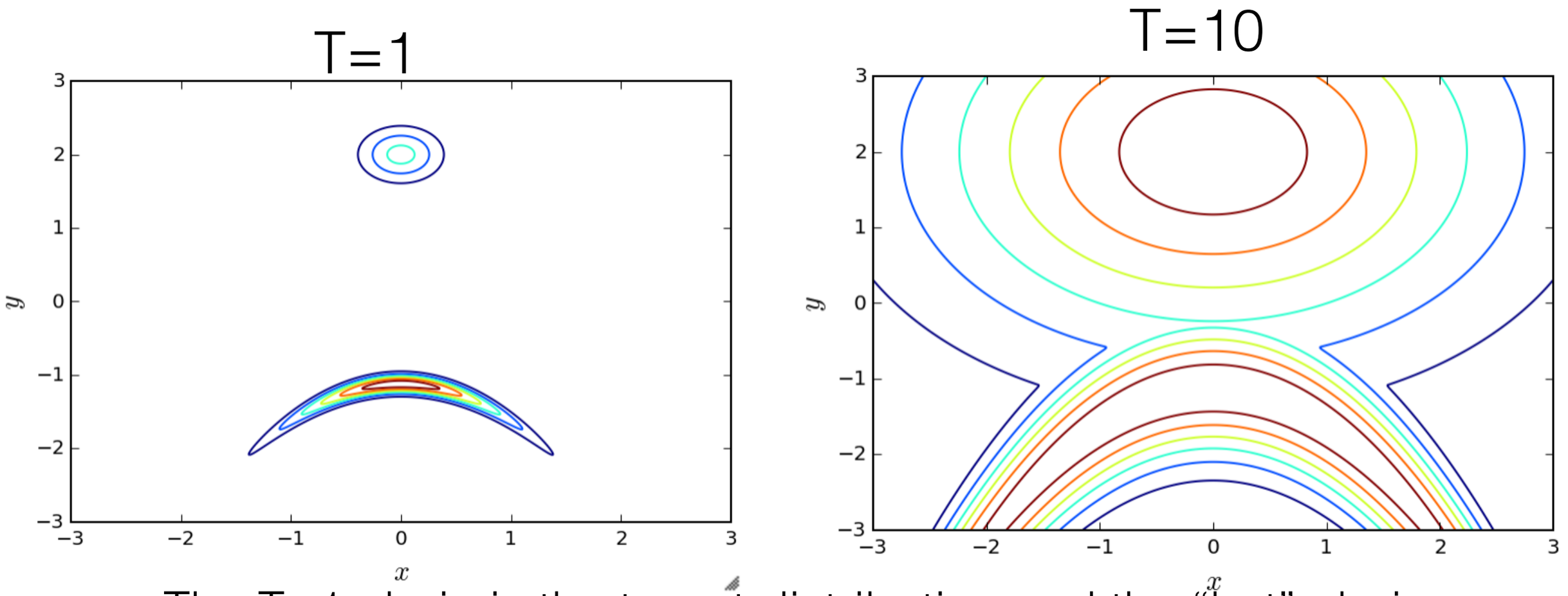
Parallel Tempering (1)

- Standard MCMC works well in moderate dimensions and unimodal distributions.

- In large dimensions with many degeneracies or multiple modes it very hard for traditional MCMC to locate the high probability regions and to efficiently move between modes.

- With parallel tempering several chains are run in *parallel* with each chain at a different *temperature:* $L(x,T) = L(x)^{1/T}$

Parallel Tempering (2)



- The T=1 chain is the target distribution and the “hot” chains explore a more flattened out likelihood.

- Throughout the MCMC run each chain will propose a “swap” with another. This way the “hot” chains can tell the “cold” chains where to go.

PTMCMC Algorithm

1. for i = 0 to Ntemp - 1
    - $T_i = \left(1 + \sqrt{2/N_{dim}}\right)^i$
    - Generate different x_0 for each temperature chain

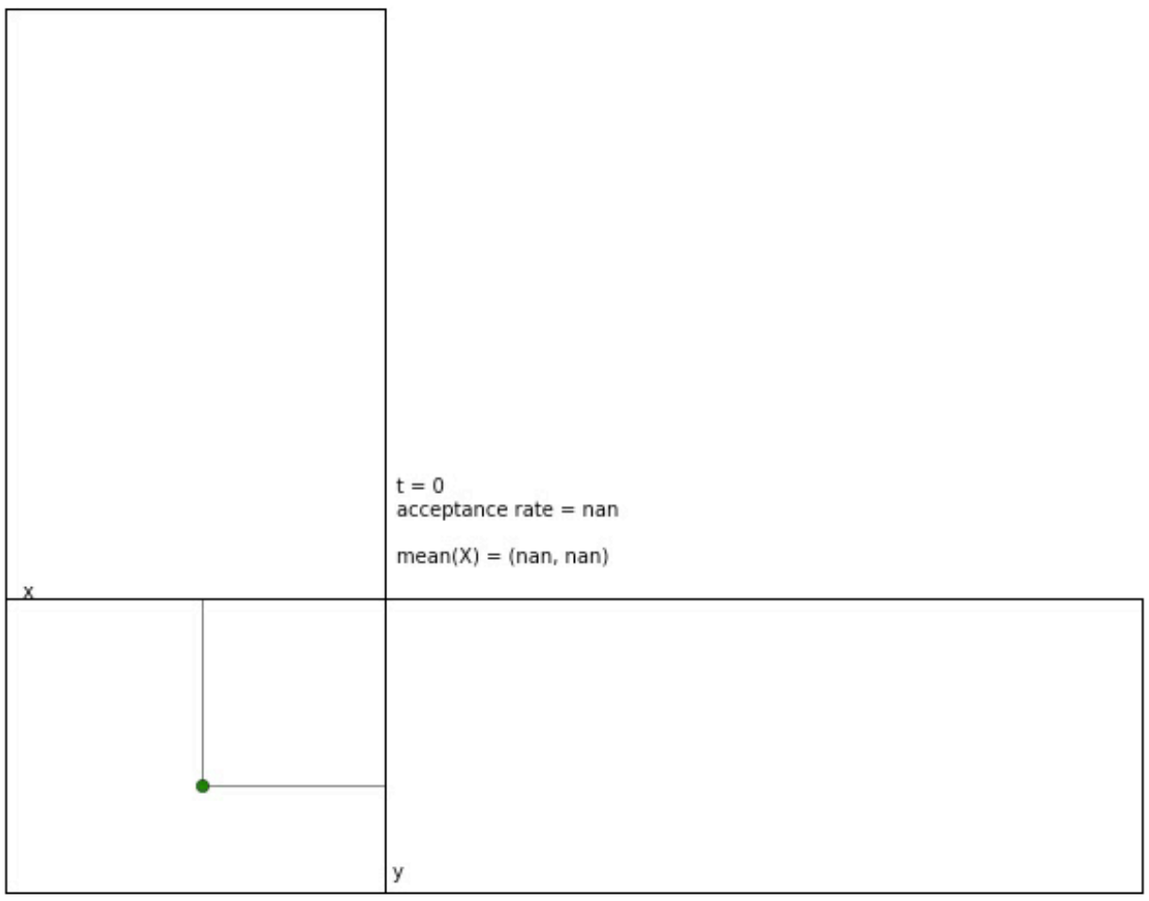
2. for i in range 0 to N -1

    1. for j = 0 to Ntemp - 1
        - Carry out standard MCMC step for chain j

    2. if i mod 10 = 0
        - for j = Ntemp-1 to 0
            - compute hastings ratio $H = \left(\frac{L(x_{j-1})}{L(x_j)}\right)^{(1/T_j - 1/T_{j-1})}$
            - if $u \sim U(0,1) < H$
                - $j \Leftrightarrow j-1$

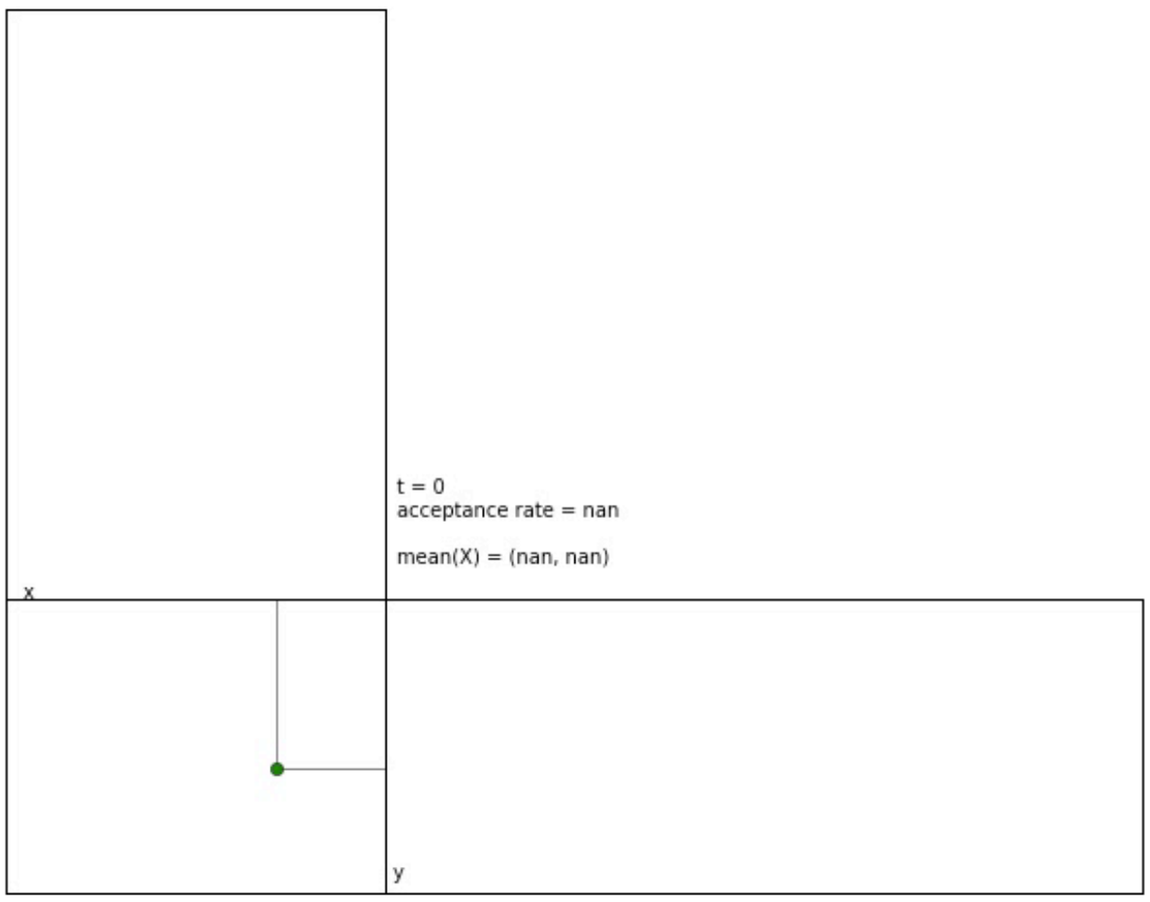
PTMCMC Example

T=1



credit: Rutger van Haasteren & Justin Ellis

T=1024

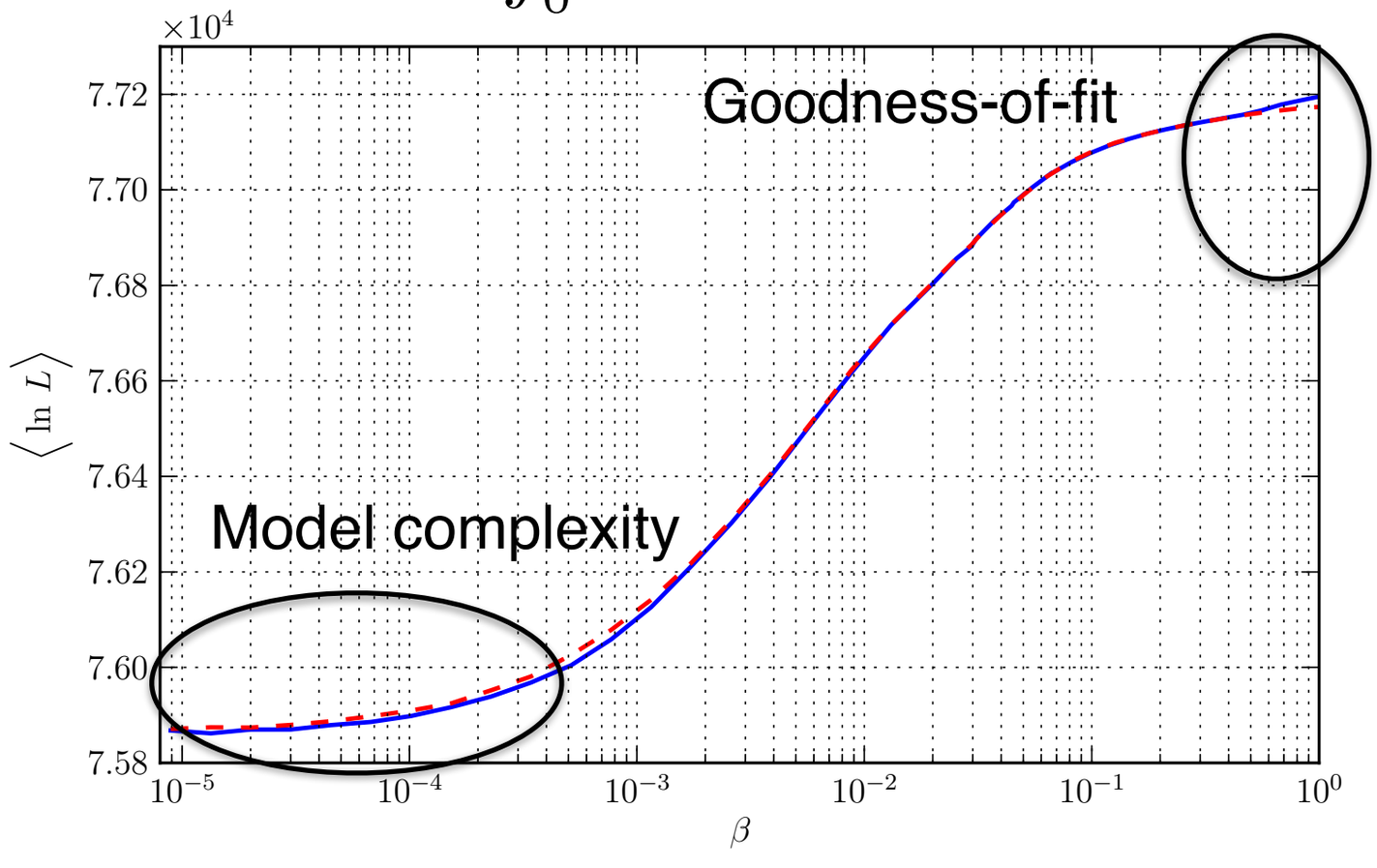


credit: Rutger van Haasteren & Justin Ellis

Thermodynamic Integration for Bayesian Evidence

- Bayesian Evidence is notoriously hard to calculate. One method is thermodynamic integration. This can almost come for free if you are already running PTMCMC.

- The evidence is $\log \mathcal{Z} = \int_0^1 d\beta \langle L(x) \rangle_\beta$



Tips & Tricks

- Zero-log-likelihood test. Set log-likelihood=0 and run MCMC. Should return prior. This is a good test of detailed balance.

- Jump cycle: Should use a variety of jumps to improve mixing. Can be done with randomly chosen step sizes (i.e some small and some big) or with a distribution of different jump proposals (i.e. DE for 50%, gaussian for 30% and AM for 20%)

- Parameter Blocks: Choose blocks of parameters to jump in while holding others fixed. Greatly improves mixing in large dimensions

- Uniform draws from prior. GREATLY improves mixing in weak parameters

- Look at acceptance rates for different jumps to gauge effectiveness of jump