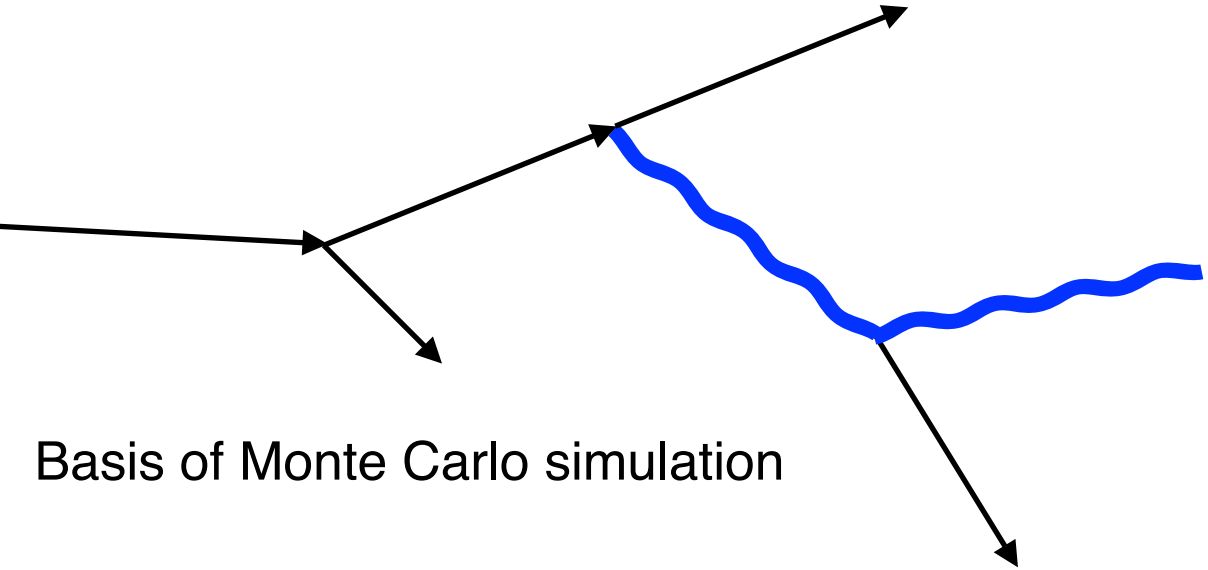
Lecture 10:

- Markov
- Metropolis, Hastings
- Gibbs, Gauss and Hamilton!

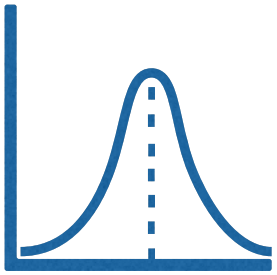
Markov Chain Monte Carlo (MCMC)

A method to numerically integrate over composite functions by probabilistically sampling the function space with a succession of linked iterations

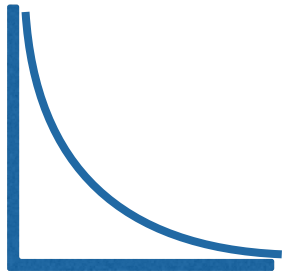


Basis of Monte Carlo simulation

Markov Chain: Each new step only depends on the previous one



Deposited Energy



Penetration Depth

Can also use such an approach to map out the parameter space of a function in the vicinity of its minimum/maximum.

Provides a robust approach for complex, multi-dimensional parameter space with lots of local minima and maxima.

Computationally intensive, but chains can be run in parallel.

Convergence of chains requires:

- Irreducibility: From any initial state, there is non-zero probability of reaching any other state. This prevents the chain getting stuck in local minima.
- Aperiodicity: The chain must not be periodic. This means the chain never gets stuck in a loop between the same states.
- Recurrence: All subsequent steps sample from the same stationary distribution once it has been reached. This means once a stationary state has been achieved, adding more steps gives a more accurate approximation to the target distribution.

Such chains are 'ergodic'

Say you're at some position, q (a vector of fit parameters), in the function of interest, such as the likelihood. Assume there is some proposed probability, $P(q' \mid q)$, for jumping to another point, q'.

$$P(q'|q) = \underbrace{g(q'|q)}_{\text{proposal}}\ \underbrace{A(q'|q)}_{\text{acceptance}}$$

Also apply "Principle of Detailed Balance" to ensure the chain direction is reversible so that we will will reach an equilibrium "stationary" state:

$$\rho(q)\ P(q \to q') = \rho(q')\ P(q' \to q)$$

or

Bayesian Posterior probability

$$P(q|D)\ P(q'|q) = P(q'|D)\ P(q|q')$$

$$\alpha(q'|q) \equiv \frac{A(q'|q)}{A(q|q')} = \frac{P(q'|D)\ g(q|q')}{P(q|D)\ g(q'|q)}$$

$$= \frac{[P(D|q')\ P(q')]\ g(q|q')}{[\underbrace{P(D|q)}_{\text{Likelihood}}\ \underbrace{P(q)}_{\text{Prior}}]\ g(q'|q)}$$

Metropolis-Hastings Algorithm

Probability to accept the proposed jump is given by:

Hastings bit

$$P_{accept} = min\left[1, \left(\frac{P(q'|D)}{P(q|D)}\frac{g(q|q')}{g(q'|q)}\right)\right]$$

i.e. always accept if the new point is better, but potentially accept if the new point is worse based on the balance of relative probabilities (so that you explore the parameter space around the best point).

So throw a random number between 0 and 1, and move to the new point if the number is less than this probability.

Then generate a new proposed position to jump to, and go again…

The frequency of visiting a particular point in the parameter space will be proportional to the overall posterior probability of that point as a solution

Generating Proposals: Gibbs Sampling

$$q = (q_1, q_2, q_3 \ldots)$$

$$q_1' \rightarrow P(q_1' \,|\, q_1)$$

$$q_2' \rightarrow P(q_2' \,|\, q_2, q_1')$$

$$q_3' \rightarrow P(q_3' \,|\, q_3, q_1', q_2')$$

$$q_n' \rightarrow P(q_n' \,|\, q_n, q_1', q_2', \ldots q_{n-1}')$$

For independent, Gaussian probabilities, this is simply:

Need to tune step sizes, guided by any parameter constraints: if too large, acceptance will be low; if too small, convergence will be slow

$$q_1' \rightarrow \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(q_1' - q_1)^2}{2\sigma_1^2}\right)$$

$$q_2' \rightarrow \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(q_2' - q_2)^2}{2\sigma_2^2}\right)$$

etc.

Generating Proposals: Hamiltonian Sampling

Analogy with system of particles at some temperature T: particles correspond to the model parameters being fit, temperature allows their values to 'jiggle about' and explore the phase space.

Boltzmann distribution:

units of kT

$$P(E) = e^{-E}$$

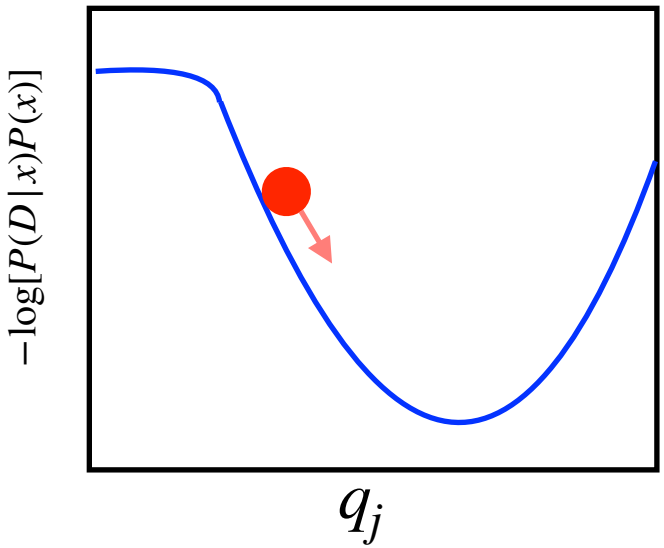
$$= e^{-(U(q)+KE)}$$

$$= e^{-U(q)} e^{-\sum \frac{p_i^2}{2m_i}}$$

Average probability to be where we are without jiggling:

$P(q \mid D) \propto P(D \mid q)P(q)$

Probability for where to move next (proposal)



$$U(q) \propto -\log[P(D \mid q)P(q)]$$

$$P(E) = P(D \mid q)P(q)e^{-\sum \frac{p_i^2}{2m_i}}$$

Sample momenta from Gaussian, tuning values of $\sigma_i = \sqrt{m_i}$ …as before

However, the subsequent evolution is then defined by Hamiltonian dynamics:

For a conservative system

$$H = E = U(q) + KE = -\log[P(D \mid q)P(q)] + \sum \frac{p_i^2}{2m_i}$$

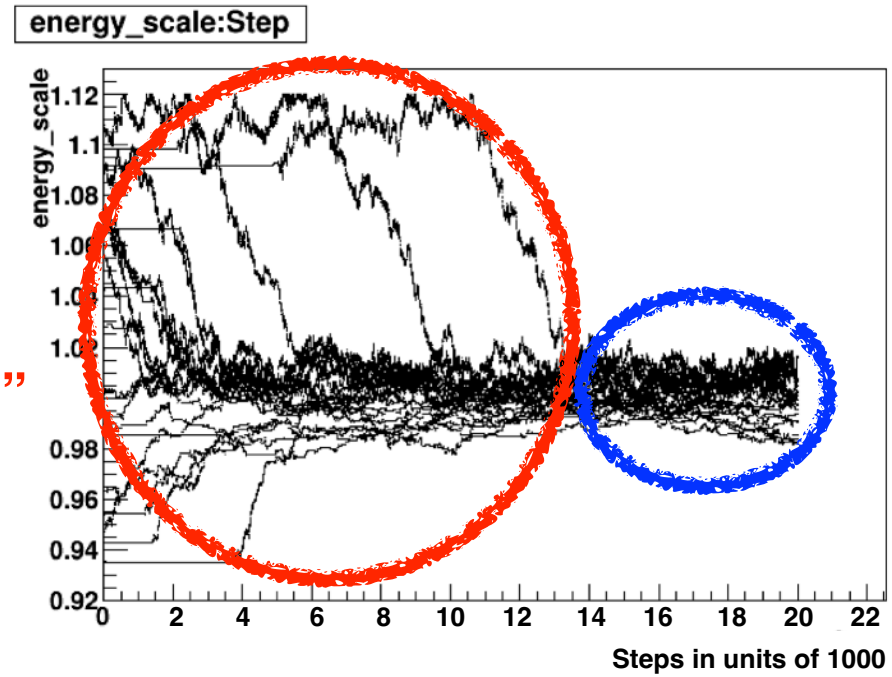
$$\frac{dq}{dt} = \frac{\partial H}{\partial p} = \frac{p}{m} \qquad q \to q + \frac{p}{m}\Delta t$$

$$\frac{dp}{dt} = -\frac{\partial H}{\partial q} = -\frac{\partial U(q)}{\partial q} \qquad p \to p - \frac{\partial U(q)}{\partial q}\Delta t$$

Tune step sizes

See “Leap Frog” algorithm for how to handle this better (basically uses interleaved average of p to compute next q)

Hamiltonian-guided path is much more efficient for reaching the stationary phase and dealing with discontinuities, but is more computationally intensive for each step. Need to tune algorithm for the specific problem at hand.

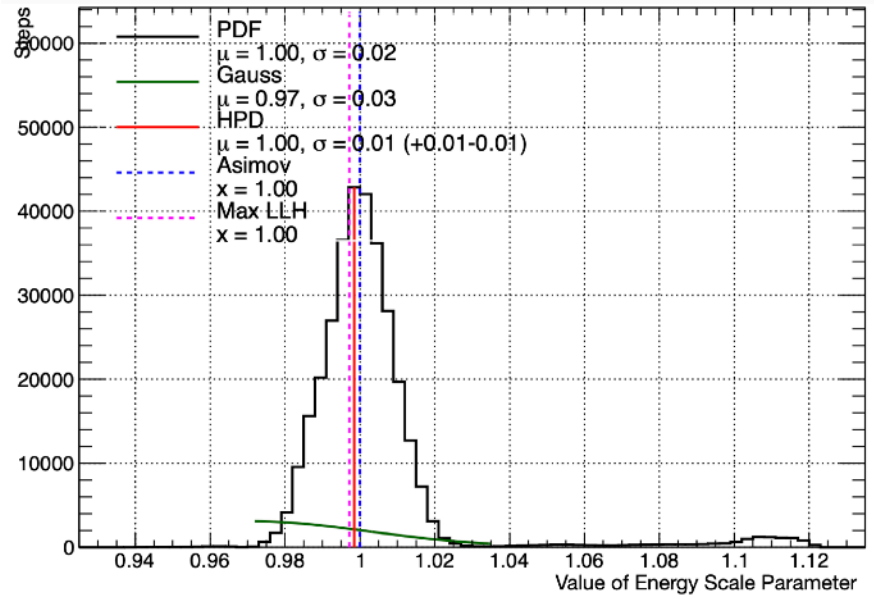


"Burn-In"

"Stationary State"

MCMC applied to simulated SNO+ data to determine signal content, normalisations to various backgrounds and systematic uncertainties

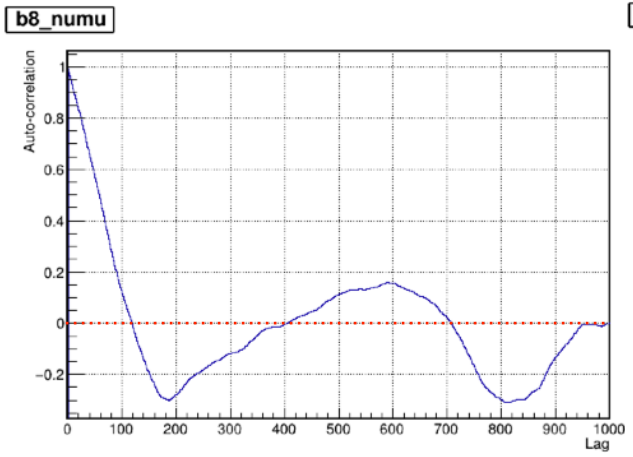
(thanks to Will Parker)



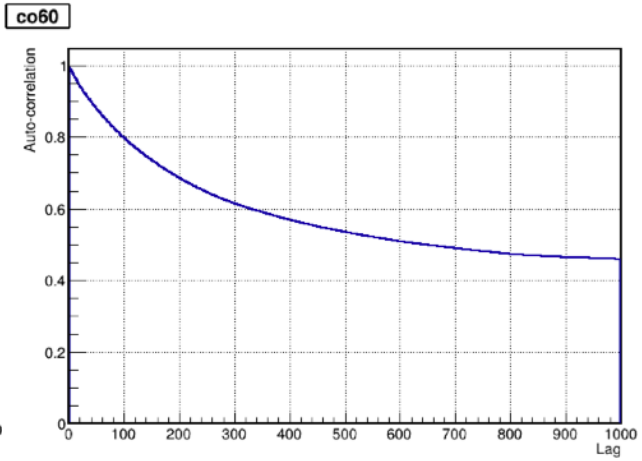
Autocorrelation as Test of Convergence

$$a = \frac{\Sigma_{i=1}^{N-k}(X_i - \bar{X})(X_{i+k} - \bar{X})}{\Sigma_{i=1}^{N}(X_i - \bar{X})^2}$$

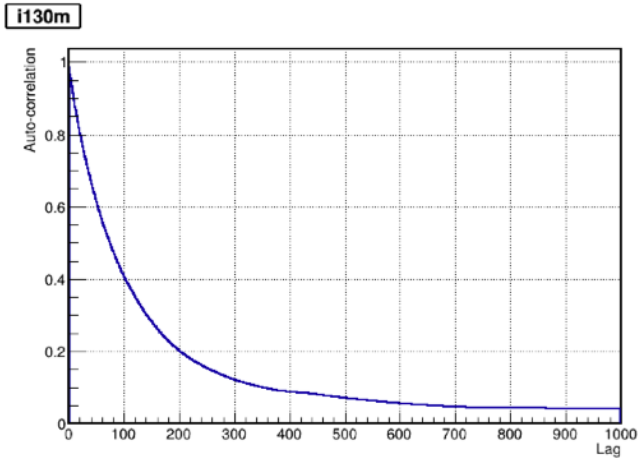
where k = “lag”



Hasn’t converged yet



Isn’t converging quickly enough
(should probably change the step size)



Converging nicely!

Can also use the MCMC information around the stationary phase to produce posterior density maps of the relevant model parameter space (if using parameters with uniform priors, this is also the likelihood map)

(Thanks to Daniel Cookman)