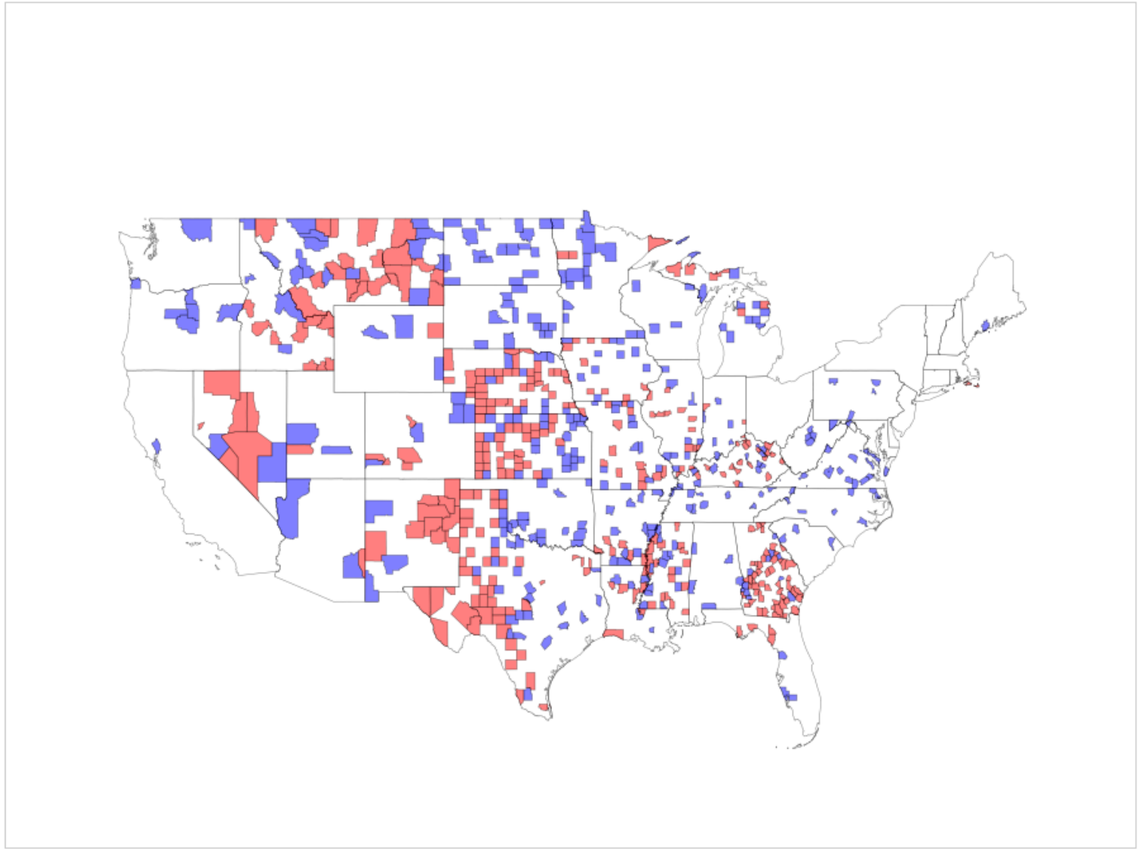
Lecture 17

Data Augmentation to The theory of HMC via Slice

Last week

- the normal model to regression
- identifiability and regression
- posterior and posterior predictive for regression, vith theano shareds
- gaussian processes as picking functions from a covariance matrix compatible prior
- its something where any finite set of points sampled from it is a multivariate-normal
- $GP(post) \propto G(likelihood) x GP(prior)$
- with data marginal not caring about the size of the other block
- but posterior-predictive conditional caring about the data block size

Homework



Levels of Bayes

Method	Definition
Maximum Likelihood	$\hat{\theta} = argmax_\theta p(D\|\theta)$
MAP estimation	$\hat{\theta} = argmax_\theta p(D\|\theta)p(\theta\|\eta)$
ML-2 (Empirical Bayes)	$\hat{\eta} = argmax_\eta \int d\theta\, p(D\|\theta)p(\theta\|\eta) = argmax_\eta p(D\|\eta)$
MAP-2	$\hat{\eta} = argmax_\eta \int d\theta\, p(D\|\theta)p(\theta\|\eta)p(\eta) = argmax_\eta p(D\|\eta)p(\eta)$
Full Bayes	$p(\theta, \eta\|D) \propto p(D\|\theta)p(\theta\|\eta)p(\eta)$

This week

- Back to Gibbs
- Data Augmentation
- Slice Sampling as augmentation
- HMC as augmentation
- Back to hierarchical model with HMC and NUTS

Back to Gibbs

- Back to the Gibbs Sampling structure
- we sample from the conditionals of the true pdf
- gives us a non-local proposal
- we have a DAG, with observations at the bottom of a tree, next layer intermediate parameters, upper layers hyper-parameters
- sample conditionals from parents up the tree.

The idea of Gibbs

$$f(x) = \int f(x,y)dy = \int f(x|y)f(y)dy = \int dy f(x|y) \int dx' f(y|x')f(x')$$

Thus: $f(x) = \int h(x,x')f(x')dx'$ integral fixed point equation

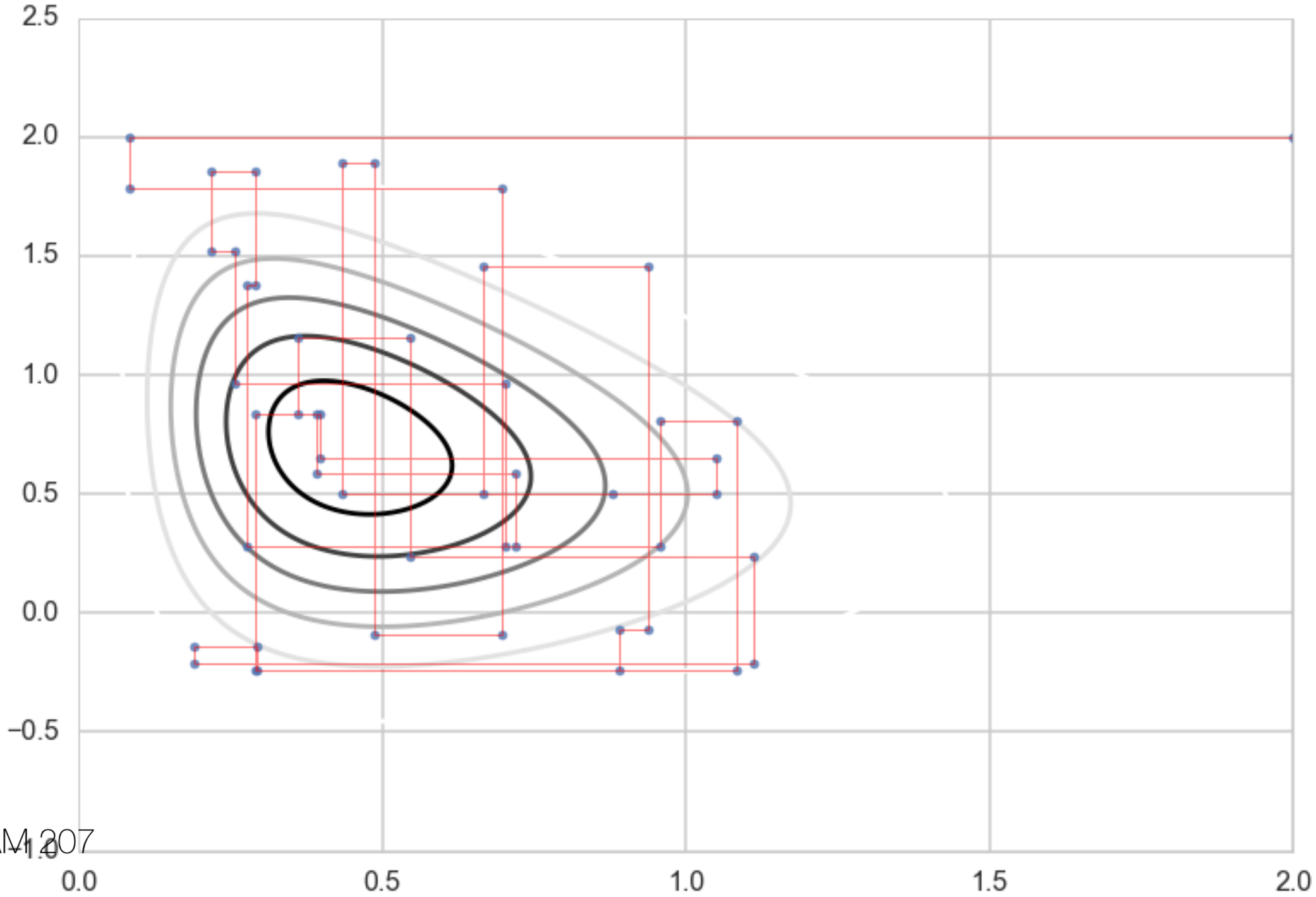
where $h(x,x') = \int dy f(x|y)f(y|x')$.

Iterative scheme in which the "transition kernel" $h(x, x')$ is used to create a proposal for metropolis-hastings moves:

$$f(x_t) = \int h(x_t, x_{t-1}) f(x_{t-1}) dx_{t-1}$$, a Stationary distribution.

$$h(x, x') = \int dy f(x|y) f(y|x')$$.: Sample alternately to get transitions.

Can sample x marginal and $x|y$ so can sample the joint x, y.


2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
0.0
0.5
1.0
1.5
2.0

Data Augmentation

want to sample a $p(x)$

The difference from Gibbs Sampling: the other variable, say y, is to be treated as latent.

The game is to construct a joint $p(x, y)$ such that we can sample from $p(x|y)$ and $p(y|x)$, and then find the marginal

$$p(x) = \int dy\, p(x, y).$$

Where we are going: Latent Variables

- critical to our subsequent understanding
- dont think of bayes/frequentist, think of observed/Latent
- anything unobserved is latent (this is the posterior predictive point of view)
- standard bayesian viewpoint: nuisance parameters are latent
- latent factors in matrix factorization, mixtures, recommendations...

Simplest form of a DA algo:

1. Draw $Y \sim p_{Y|X}(.|x)$ and call the observed value y

2. Draw $X_{n+1} \sim p_{X|Y}(.|y)$

3. Histo the X

Usual "Fake News" Example

Sample from $p(x) = e^{-x^2/2}/\sqrt{2\pi}$.

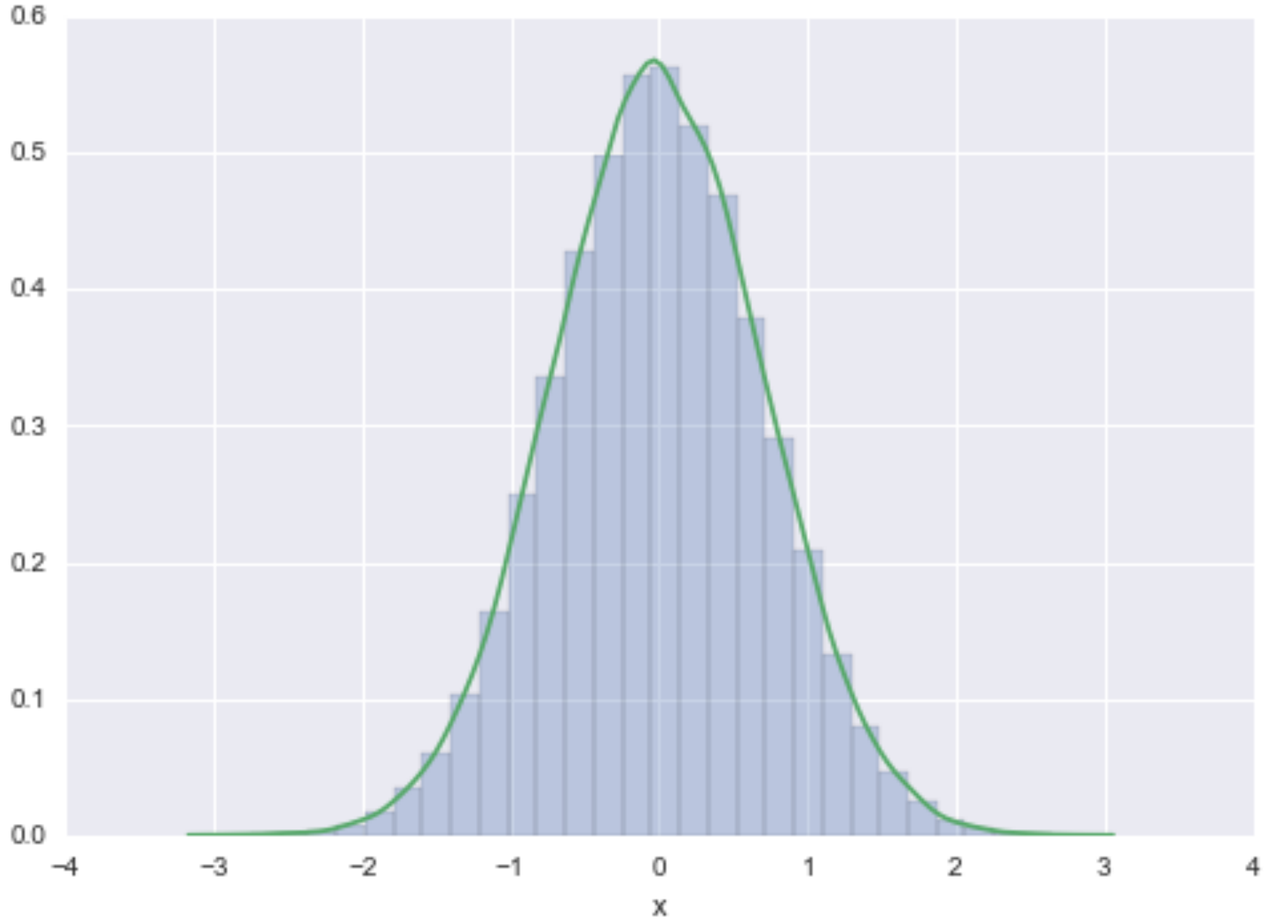
Take $p(x, y) = 1/(\sqrt{2\pi}) \exp\left\{-(x^2 - \sqrt{2}xy + y^2)\right\}$

$$Y|X = x \sim N(x/\sqrt{2}, 1/2) \text{ and } \mathrm{X}|\mathrm{Y} = \mathrm{y} \sim \mathrm{N}(\mathrm{y}/\sqrt{2}, 1/2)$$

The x-marginal is $\propto e^{-x^2/2} \int e^{-(y - x/\sqrt{2})^2} dy$

Example (contd)

```
N=100000
x=np.zeros(N)
x[0] = np.random.rand() # INITIALIZE
for i in np.arange(1,N):
    Y=sp.stats.norm.rvs(x[i-1]/np.sqrt(2), 0.5)
    x[i]=sp.stats.norm.rvs(Y/np.sqrt(2), 0.5)
```



Transition kernel

$h(x', x) = h(x'|x) = \int_Y p(x'|y)\, p(y|x)\, dy$ has stationarity by construction from Gibbs.

Its a probability: $\int h(x'|x)dx' = \int_X \int_Y p(x'|y)\, p(y|x)\, dy\, dx'$

$= \int_Y p(y|x) \left[\int_X p(x'|y)dx' \right] dy = \int_Y p(y|x)dy = 1$

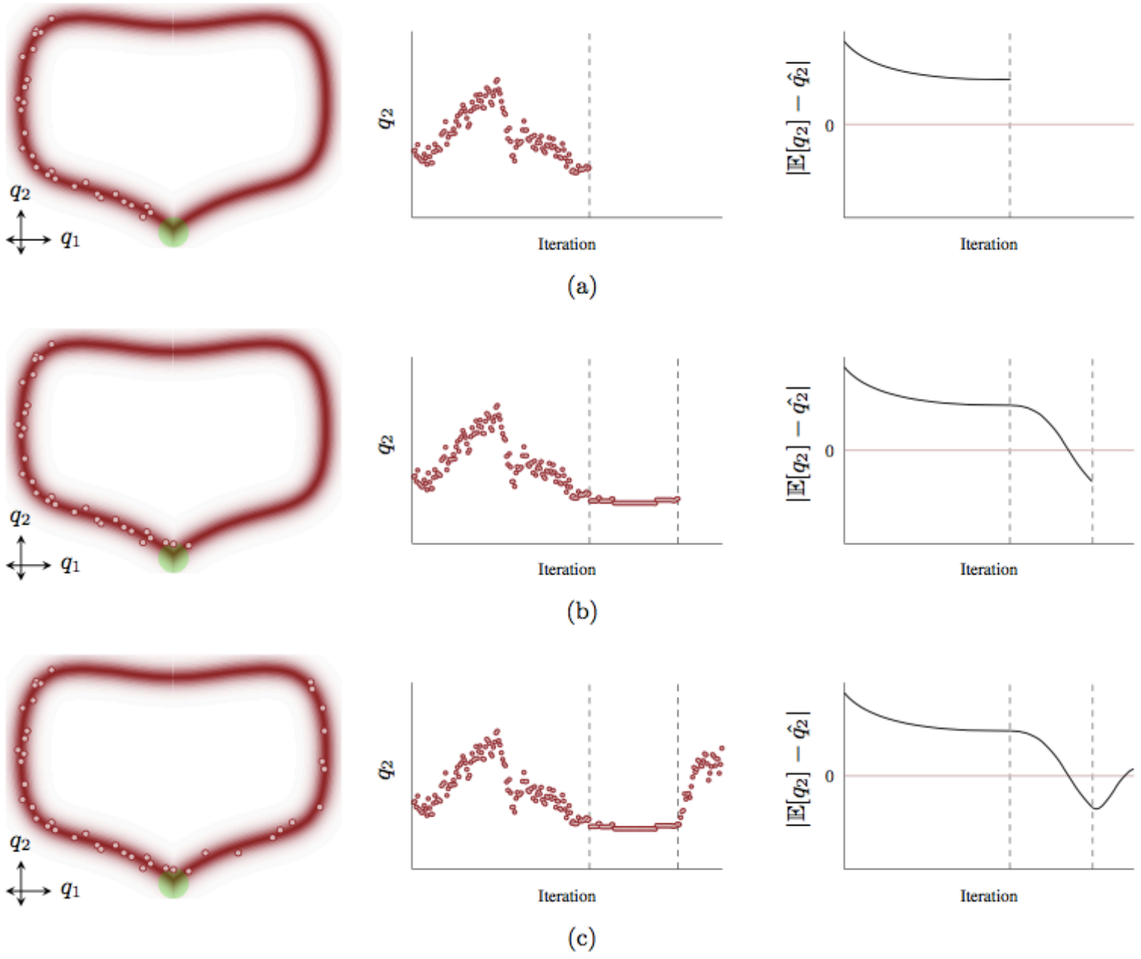
$h(x'|x)\,p(x)$ is symmetric in (x, x'):

$$h(x'|x)\,p(x) = p(x)\int_Y p(x'|y)\,p(y|x)\,dy = \int_Y \frac{p(x',y)\,p(x,y)}{p(y)}\,dy.$$

The rhs is symmetric in (x, x') and so is $h(x'|x)p(x)$.

The Markov chain generated with transition probability $h(x'|x)$ is **REVERSIBLE** and thus supports detailed balance.

Problems with MCMC

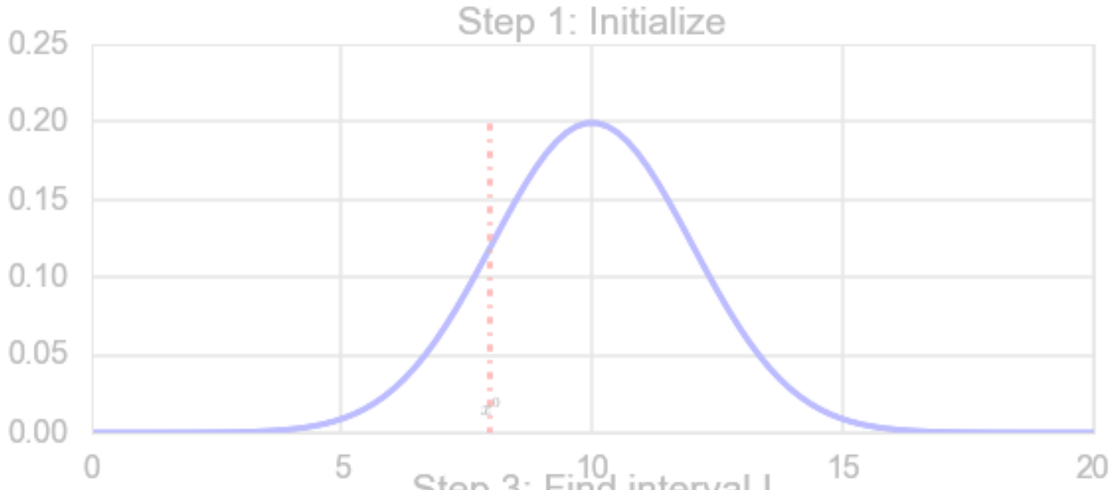


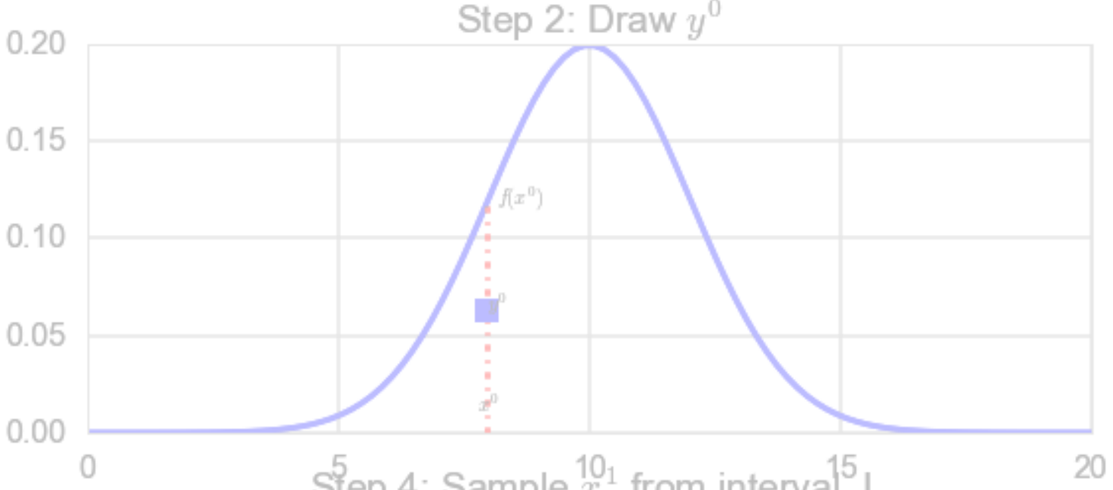
- overshoot and oscillate at pinches
- need to specify step step sizes
- large steps go outside typical set and are not accepted
- small steps accepted but go nowhere
- large correrlations

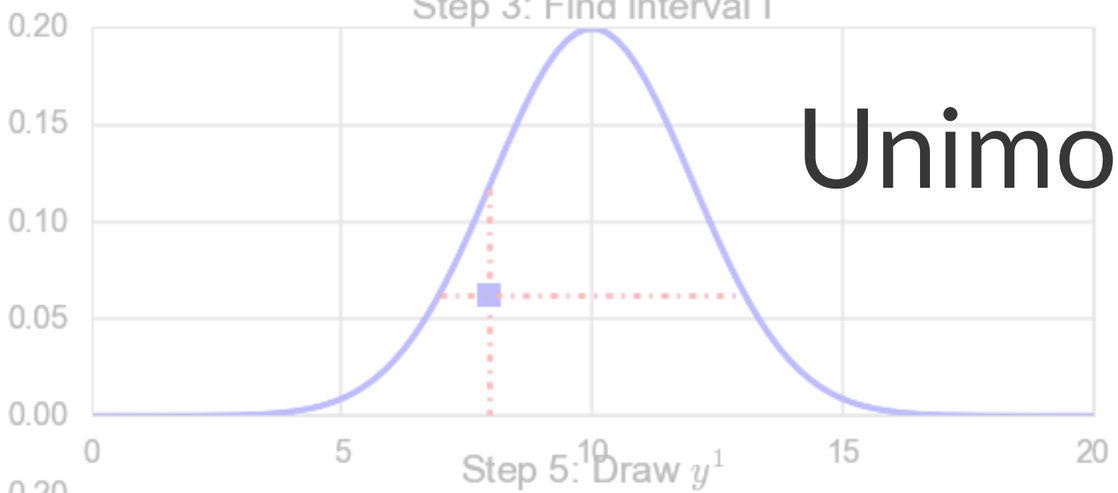
SLICE

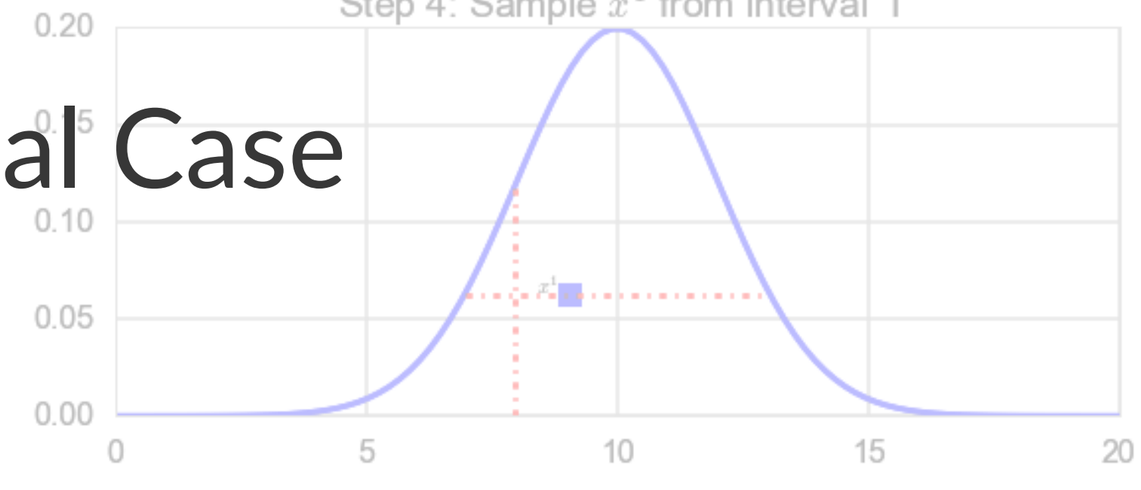
to the rescue

- Pick an initial point x_0 from our posterior
- Draw y_0 from U(0, f(x_0))
- Repeat for N samples
  - Select the interval (e.g. stepping out, etc)
  - Sample x_i from that interval (e.g. shrinkage)
  - Draw y_i from U(0, f(x_i))

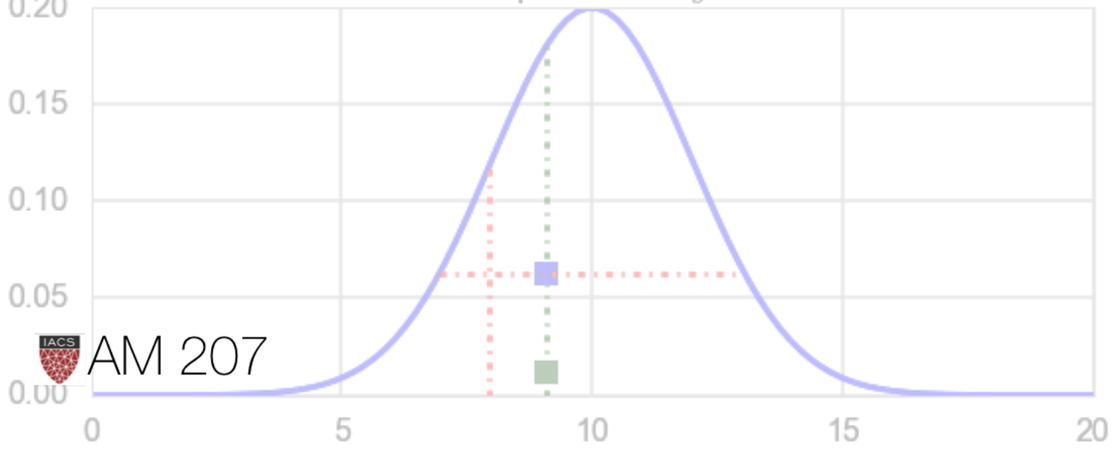




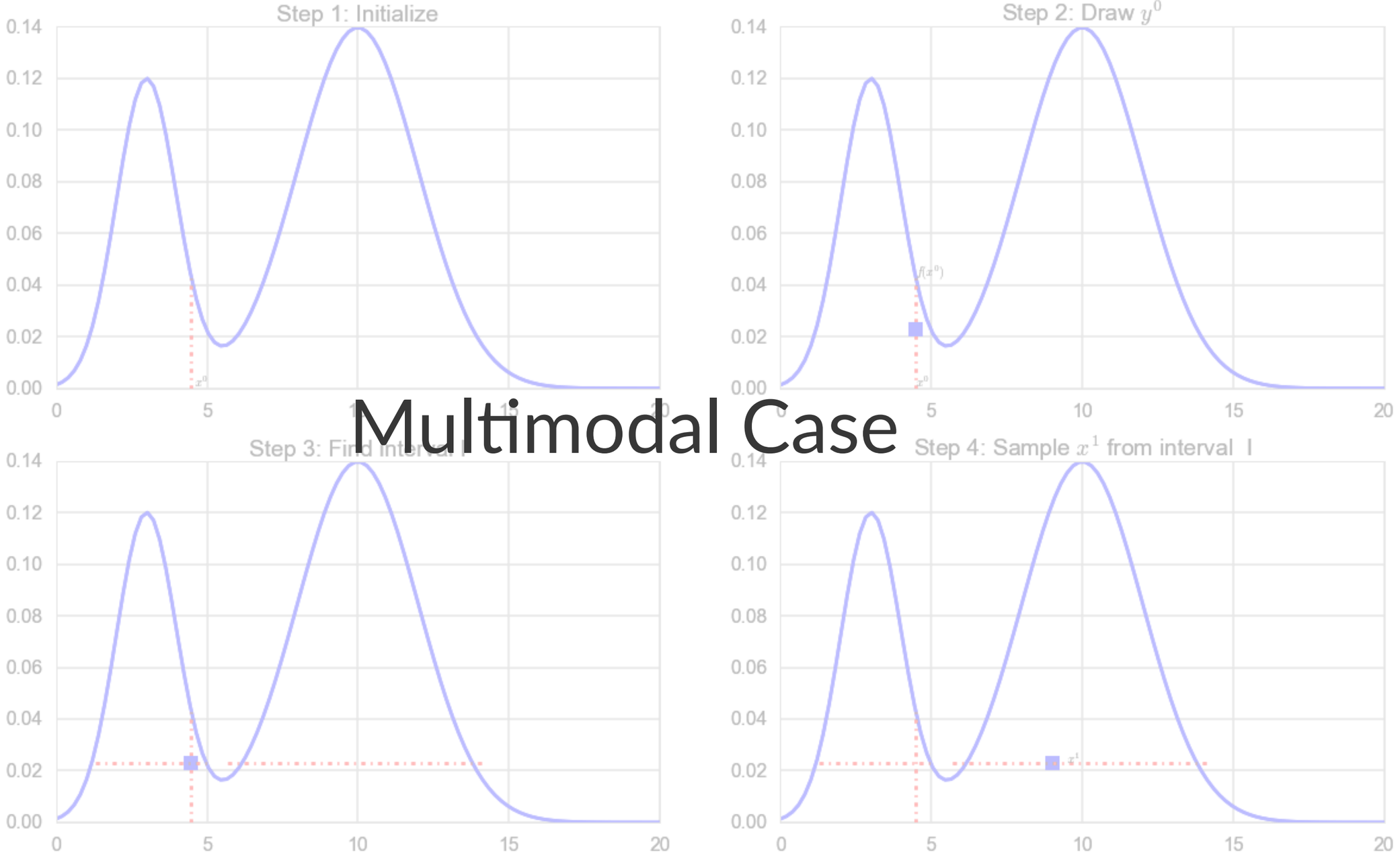




Unimodal Case



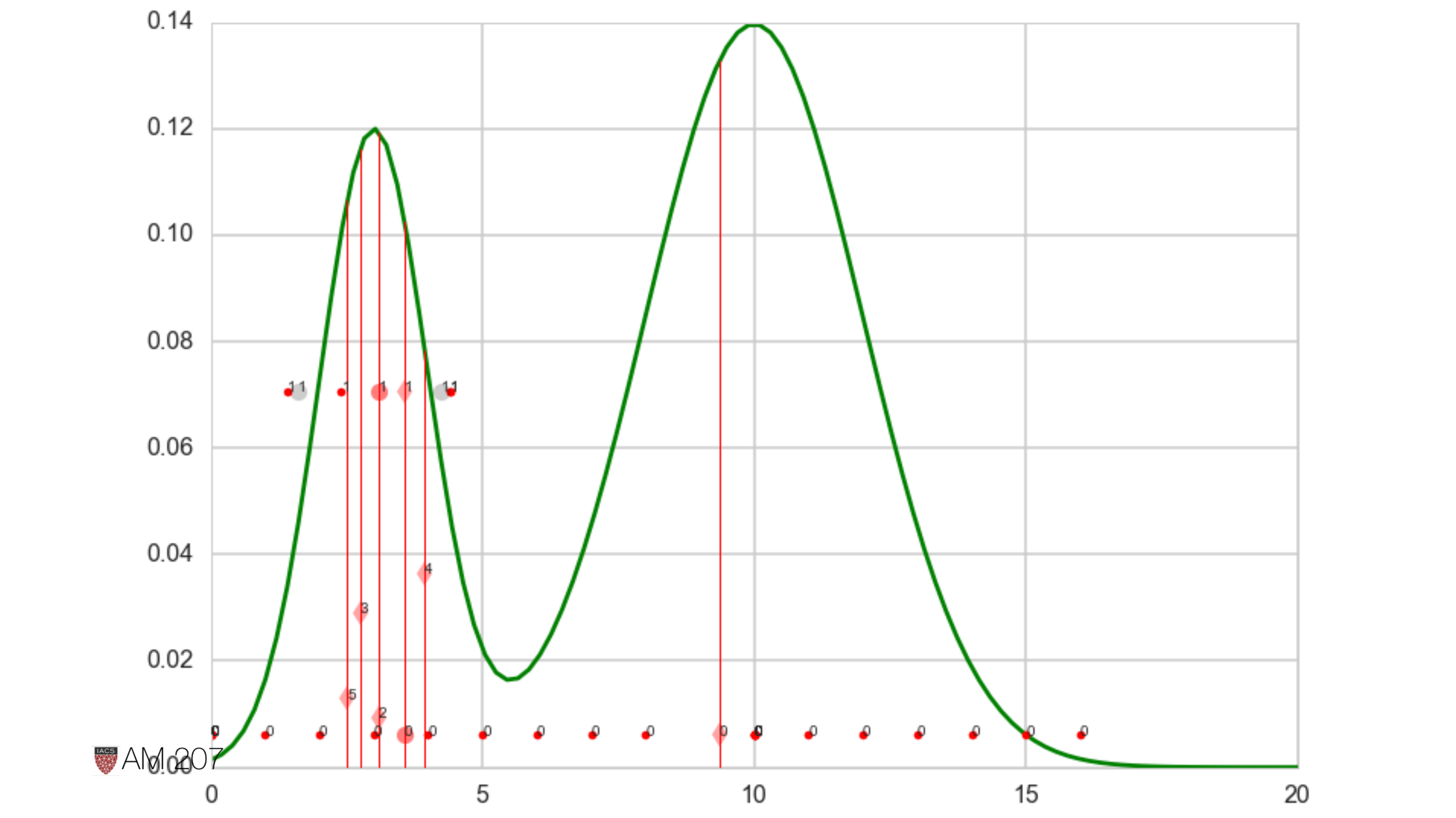
Multimodal Case



Stepping Out

- set w width and draw u ~ Unif(0,1)
- set L = $x^{(0)} - wu, R = L + w$ (so $x^{(0)}$ lies in [L, R])
- while y < f(L) (here's where we extend left interval) L = L - w
- while y < f(R) (here's where we extend the right interval) R = R + w

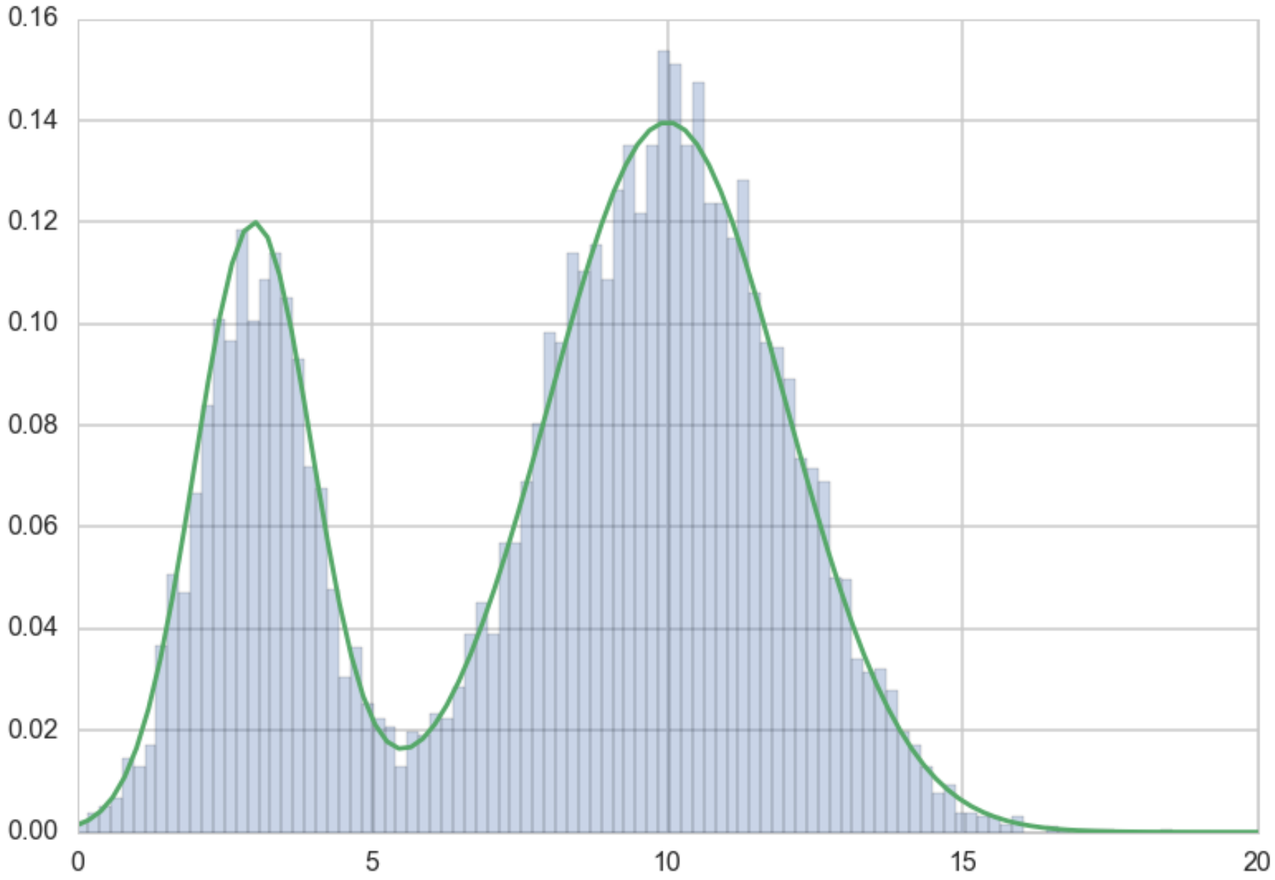
The final interval will be larger than S.


0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
5
10
15
20

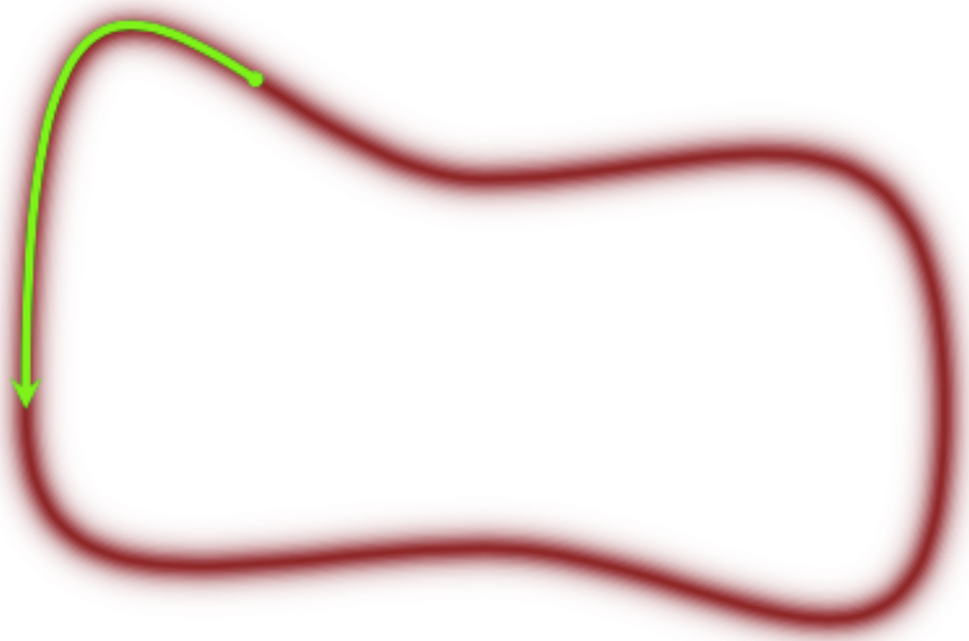
Shrinkage

- start with interval $I = (L, R)$
- current sample is $x^{(k)}$ and $y^{(k)}$
- repeat until loop exits
  - sample x^* uniformly from (L, R)
  - if $y^{(k)} < f(x^*)$

```
w=1.0
L=0; R=0;
x_prev = np.random.uniform(low=0, high=17)
iters=10000
trace=[]
kmax=1
for k in range(iters):
    y_samp = np.random.uniform(low=0, high=fun(x_prev))
    # widen left
    U = np.random.rand()
    L=x_prev-U*w
    R=x_prev+w*(1.0-U)
    while fun(L)>y_samp:
        L = L-w
    while fun(R)>y_samp:
        R = R+w
    #now propose new x on L,R

    while 1:
        x_prop= np.random.uniform(low=L, high=R)
        if k <= kmax:
            print("L,R, xprop", L, R, x_prop)
        if y_samp < fun(x_prop):
            x_prev = x_prop
            trace.append(x_prop)
            break
        elif x_prop > x_prev:
                R = x_prop
        elif x_prop < x_prev:
                L = x_prop
```



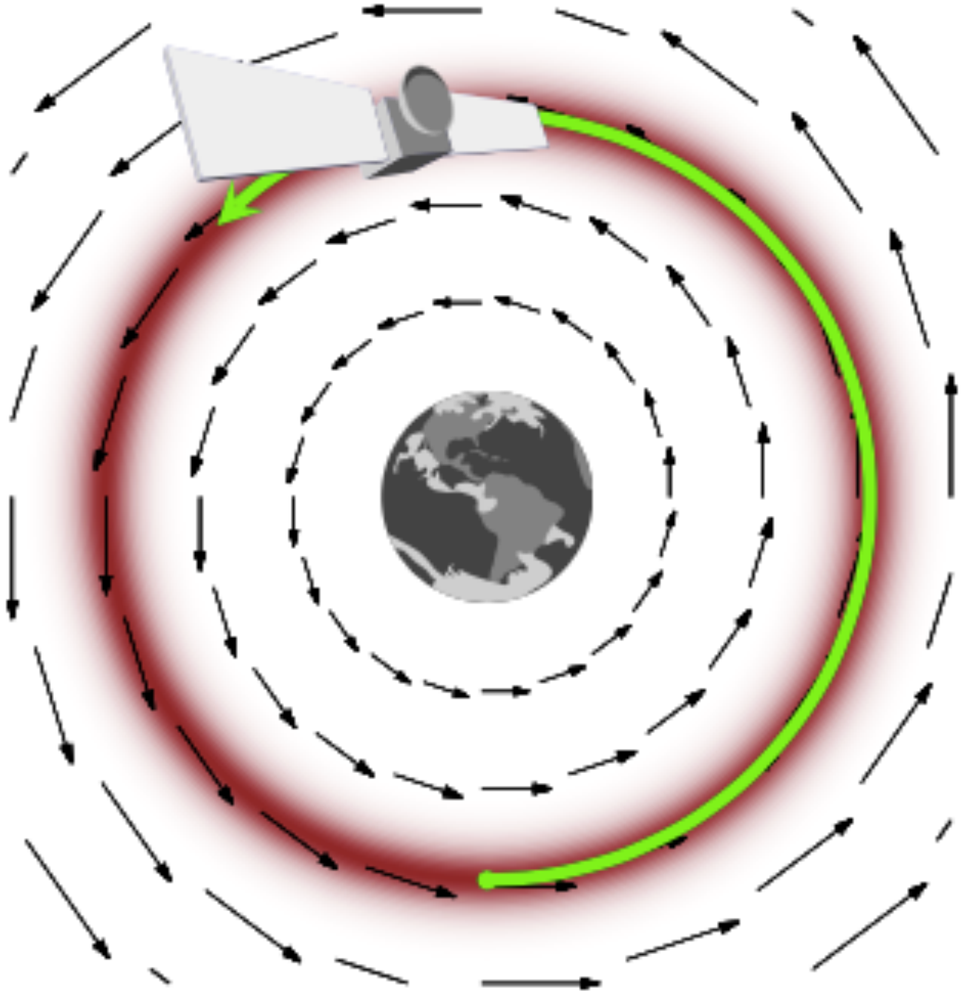
Hamiltonian Monte Carlo



Need a Coherent Glide

- want to cover on $p(q)$ better than a drunkard
- move smoothly on $p(q)$
- instead we will augment with a "momentum" variable p
- try to move smoothly on $p(p, q)$
- and then marginalize:

$$\int dp p(p, q) = \int dp p(p|q)p(q) = p(q)$$



Balance between gravity and momentum in a rocket provides it

Now, like in annealing, let
$p(p,q) = e^{-Energy}$

Carry out an augmentation with an additional momentum with the energy
Hamiltonian

$$H(p,q) = \frac{p^2}{2m} + V(q)$$

$$p(p,q) = e^{-H(p,q)} = e^{-K(p,q)}e^{-V(q)} = p(p|q)p(q)$$

and thus: $H(p,q) = -log(p(p,q)) = -logp(p|q) - logp(q)$

The choice of a kinetic energy term then is the choice of a conditional probability distribution over the "augmented" momentum which ensures that

$$\int dp p(p,q) = \int dp p(p|q)p(q) = p(q)\int p(p|q)dp = p(q).$$

Canonical distribution

Distribution of a physical system in connection with a heat bath.

Its temperature if thus fixed.

$p(p, q)$ is our canonical distribution

$$p(p, q) = e^{-H(p,q)} = e^{-K(p,q)} e^{-V(q)} = p(p|q)p(q)$$

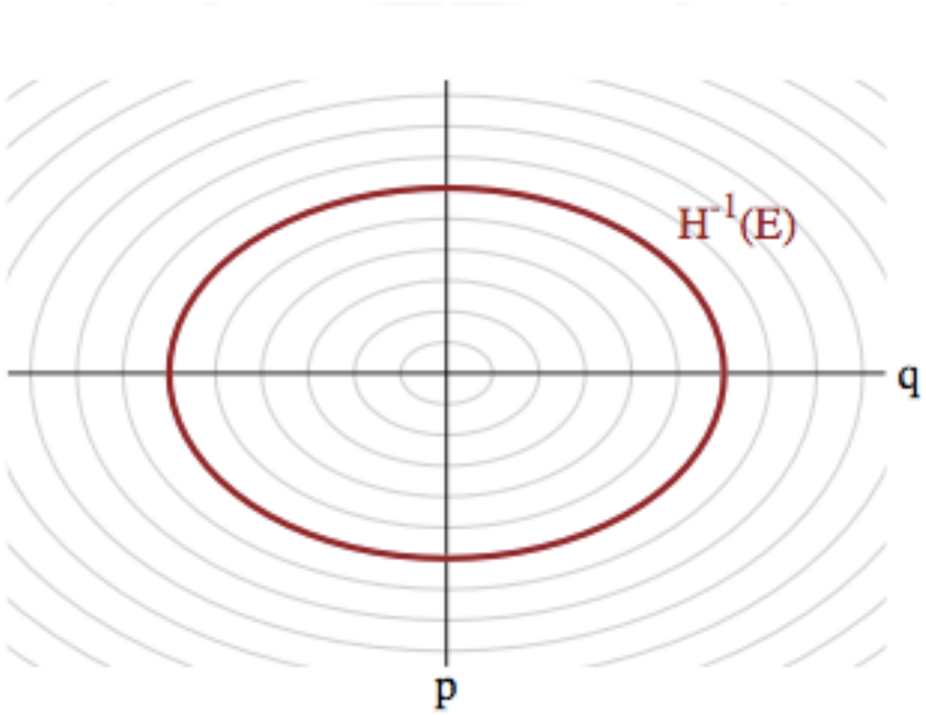
Phase Space level sets

Typical Set decomposes into level sets of constant probability(energy)

The energy **Hamiltonian**

$$H(p,q) = \frac{p^2}{2m} + V(q) = E_i\,,$$

with E_i constants (constant energies) for each level-set foliate and where the **potential energy** $V(q) = -log(p(q))$ replaces the energy term we had earlier in simulated annealing.



We are looking at level sets of the

Joint distribution

Why do it this way?

Because Hamiltonian flow conserves energy, leading naturally to using level sets and the

Microcanonical distribution

Microcanonical distribution: states for given energy.

Time implicit H: flows **constant energy, vol preserving, reversible**.

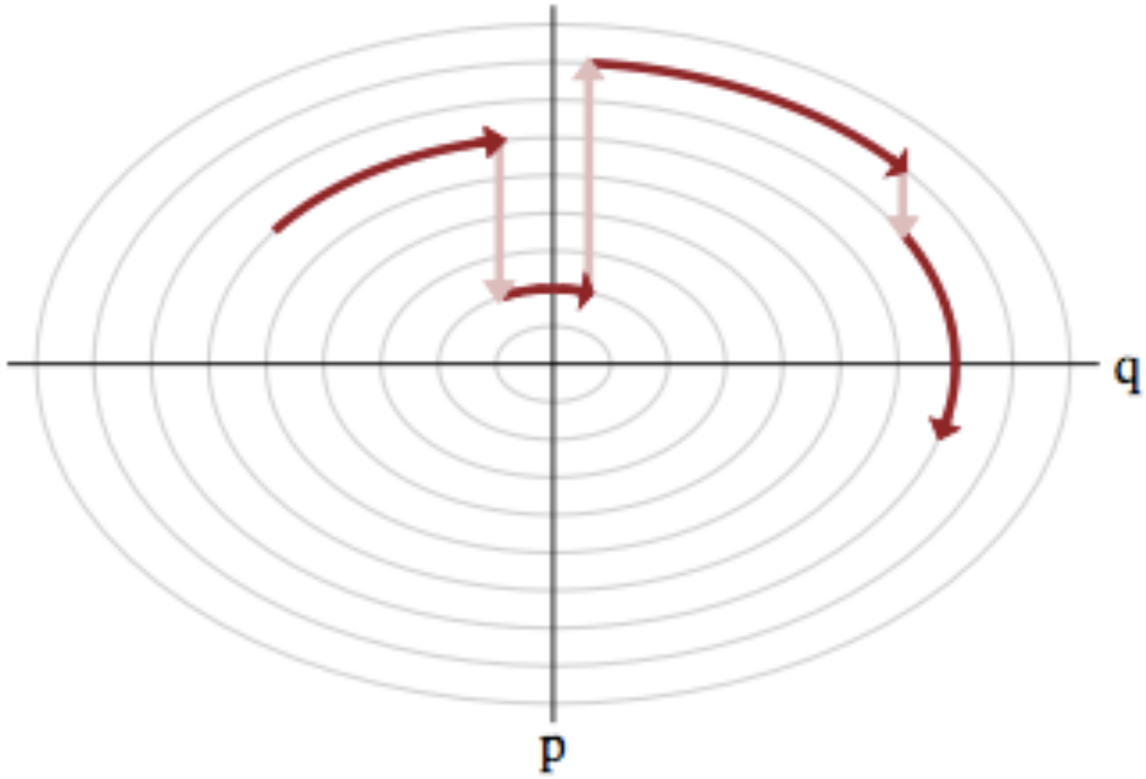
The canonical distribution can be written as a product of this microcanonical distribution and a **marginal energy distribution**:

$$p(q,p) = p(\theta_E|E)p(E)$$

where θ_E indexes the position on the level set.

Also need to sample **Marginal Energy Distrib**: probability of level set in the typical set.

Momentum resampling (thruster fire) moves us between level sets



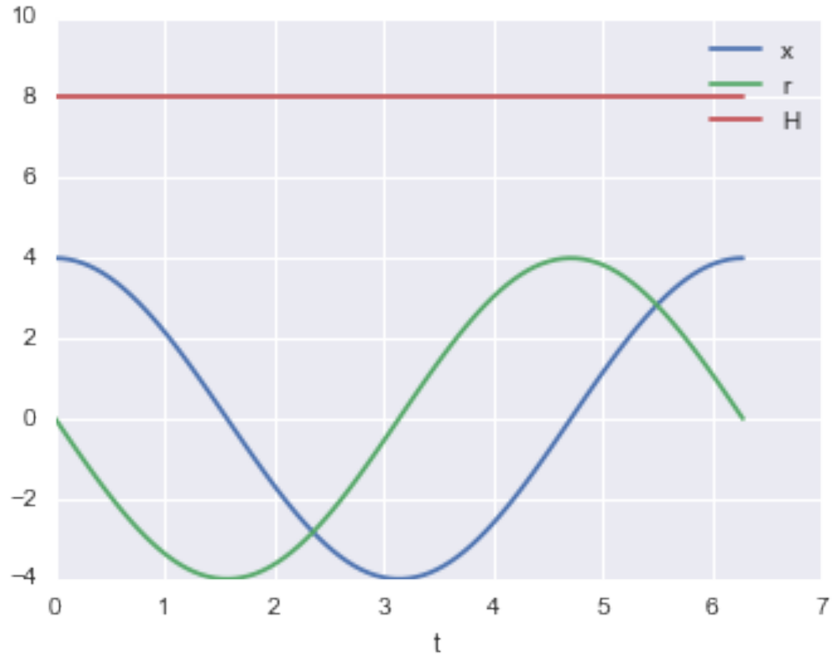
Traverse a level set: Hamiltonian Mechanics

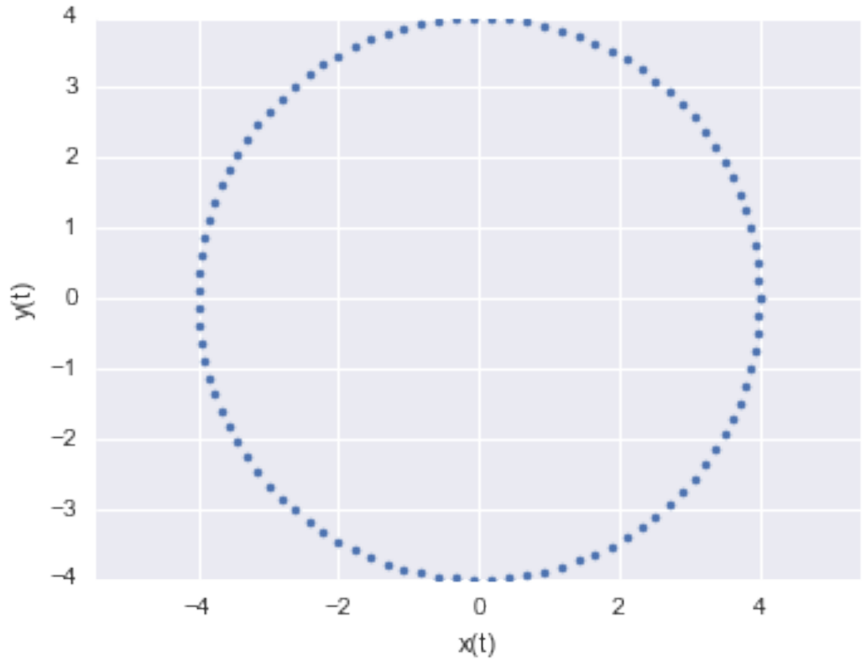
Physics equations of motion in the **Hamiltonian Formalism** set up the "glide" (over a level set).

$$\frac{dp}{dt} = -\frac{\partial H}{\partial q}$$
$$\frac{dq}{dt} = \frac{\partial H}{\partial p}$$

$H = p^2/2m + V(q)$, $\frac{dp}{dt} = -\frac{\partial H}{\partial q} = -\frac{\partial V}{\partial q} = Force$: Newton's law.

Oscillator: an EXACT solution!






```
q_t = lambda t: 4. * np.cos(t)
p_t = lambda t: -4. * np.sin(t)
```

Explicitly time-independent Hamiltonian is conserved

If the Hamiltonian H doesn't have a functional dependence on time we see that

$$\frac{dH}{dt} = \sum_i \left[\frac{\partial H}{\partial q_i} \frac{dq_i}{dt} + \frac{\partial H}{\partial p_i} \frac{dp_i}{dt} \right] + \frac{\partial H}{\partial t}$$

$$\frac{dH}{dt} = \sum_i \left[\frac{\partial H}{\partial q_i} \frac{\partial H}{\partial p_i} + (\frac{\partial H}{\partial p_i})(-\frac{\partial H}{\partial q_i}) \right] + \frac{\partial H}{\partial t}$$

if

$$\frac{\partial H}{\partial t} = 0, \frac{dH}{dt} = 0$$

Then

$$H(t + \Delta t) = H(t) \, \forall t.$$

This time independence is crucial to reversibility: cannot figure which direction equations are being run

Reversibility

T_s from $(q, p) \to (q', p')$ to a "later" time $t' = t + s$. Mapping is 1-1, inverse T_{-s}. This can be obtained by simply negating time:

$$\frac{dp}{d(-t)} = -\frac{\partial H}{\partial q}$$
$$\frac{dq}{d(-t)} = \frac{\partial H}{\partial p}$$

Superman Transform

If we then transform $p \rightarrow -p$, we have the old equations back:

$$\frac{d(-p)}{d(-t)} = -\frac{\partial H}{\partial q}$$
$$\frac{dq}{d(-t)} = \frac{\partial H}{\partial(-p)}$$

To reverse T_s, flip the momentum, run Hamiltonian equations backwords in time until you get back to the original position and momentum in phase space at time t, then flip the momentum again so it is pointing in the right direction.

Volume in phase space is conserved

T_s for small change $s = \delta$ can be written as:

$$T_\delta = \begin{pmatrix} q \\ p \end{pmatrix} + \delta \begin{pmatrix} \frac{dq}{dt} \\ \frac{dp}{dt} \end{pmatrix} + O(\delta^2)$$

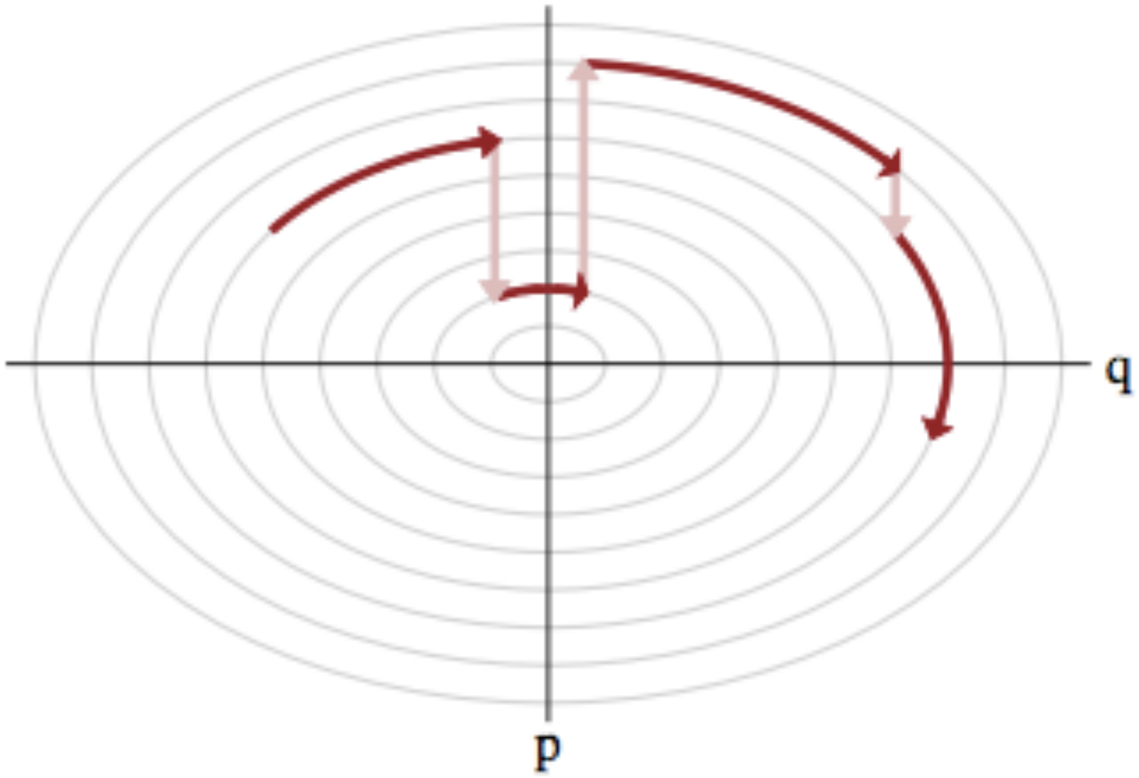
Jacobian:

$$\begin{bmatrix} 1 + \delta \frac{\partial^2 H}{\partial q \partial p} & \delta \frac{\partial^2 H}{\partial p^2} \\ \delta \frac{\partial^2 H}{\partial q^2} & 1 - \delta \frac{\partial^2 H}{\partial p \partial q} \end{bmatrix}$$ and thus the determinant is $1 + O(\delta^2)$.

Thus as our system evolves, any contraction or expansion in position space must be compensated by a respective expansion or compression in momentum space.

As a result of this, the momenta we augment our distribution with must be **dual** to our pdf's parameters, transforming in the opposite way so that phase space volumes are invariant.

Between level sets: Momentum resampling



Draw p from a distribution that is determined by the distribution of momentum, i.e. $p \sim N(0, \sqrt{M})$ for example, and attempt to explore the level sets.

Firing the thruster moves us between level sets!

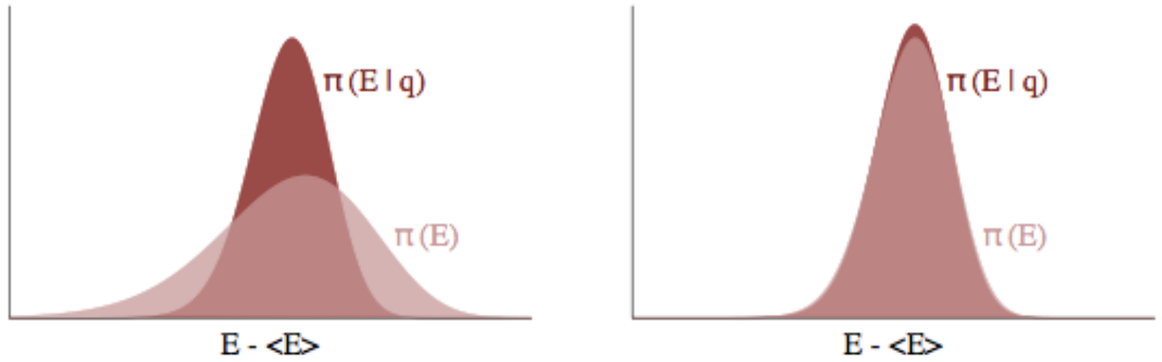
That is, we sample the marginal energy distribution.

Resampling Efficiency

Let $p(E|q)$ as the transition distribution of energies induced by a momentum resampling using $p(p|q) = -log\, K(p, q)$ at a given position q.

If $p(E|q)$ narrow compared to the marginal energy distribution $p(E)$: random walk amongst level sets proceeds slowly.

If $p(E|q)$ matches $p(E)$: independent samples generated from the marginal energy distribution very efficiently.



Tuning: Choice of Kinetic energy

- The ideal kinetic energy interacts with target to make microcanonical exploration easy and uniform and marginal exploration well matched by the transition distribution.
- In practice we often use $K(p) = p' M^{-1} p$
- Set inverse mass matrix to the covariance of the target distribution: maximally decorrelate the target. Do in warmup phase. Warmup replaces burnin.

See this for Gaussian:

$$H = \frac{1}{2} p^T M^{-1} p + \frac{1}{2} q^T \Sigma^{-1} q$$

On transformation $p' = p\sqrt{M^{-1}}$, then $q' = q\sqrt{M}$

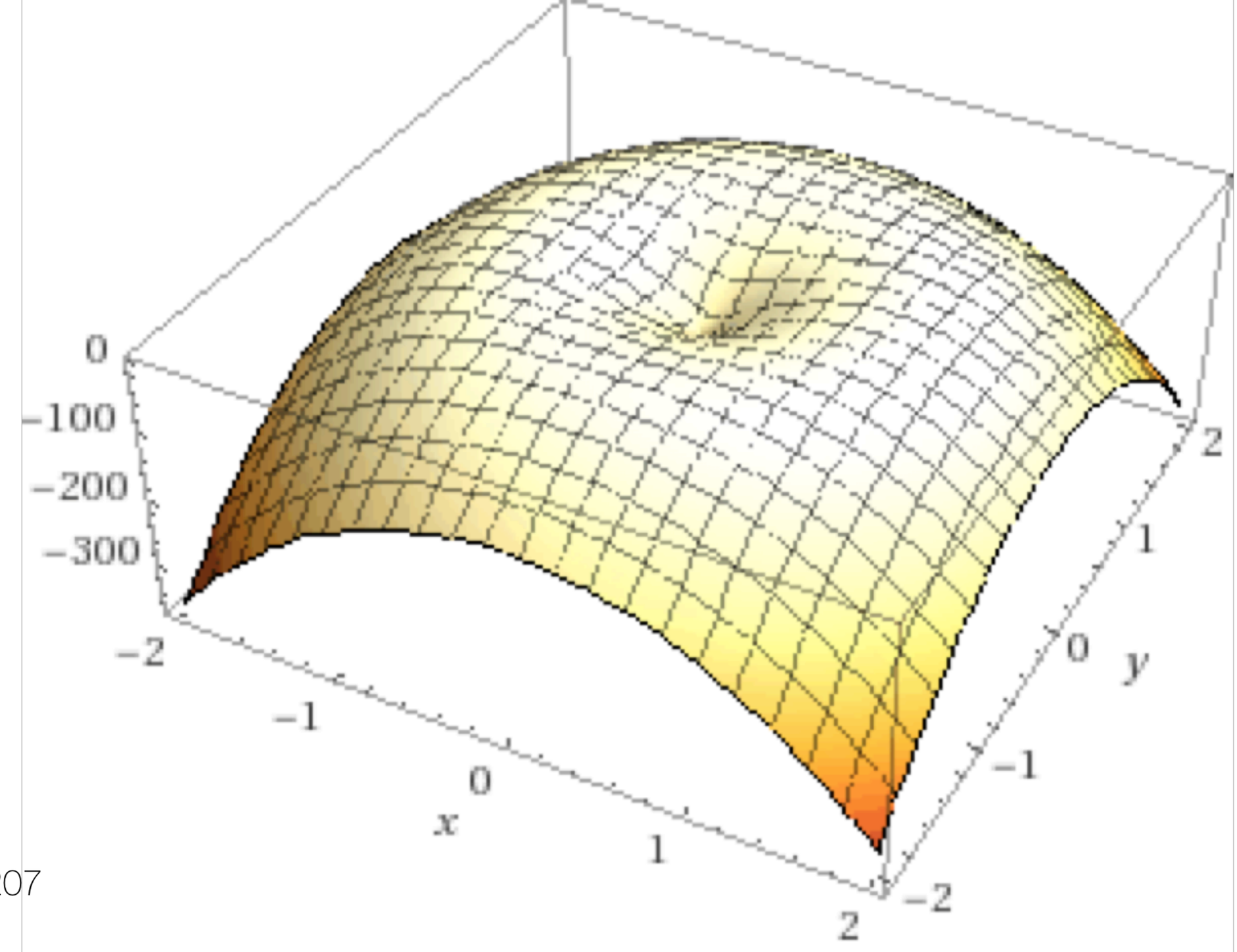
$$H = \frac{1}{2}\left(p'^T p' + q'^T q'\right) \text{ if } M^{-1} = \Sigma$$

Thus de-correlate target.

Generalize to arbitrary distributions.

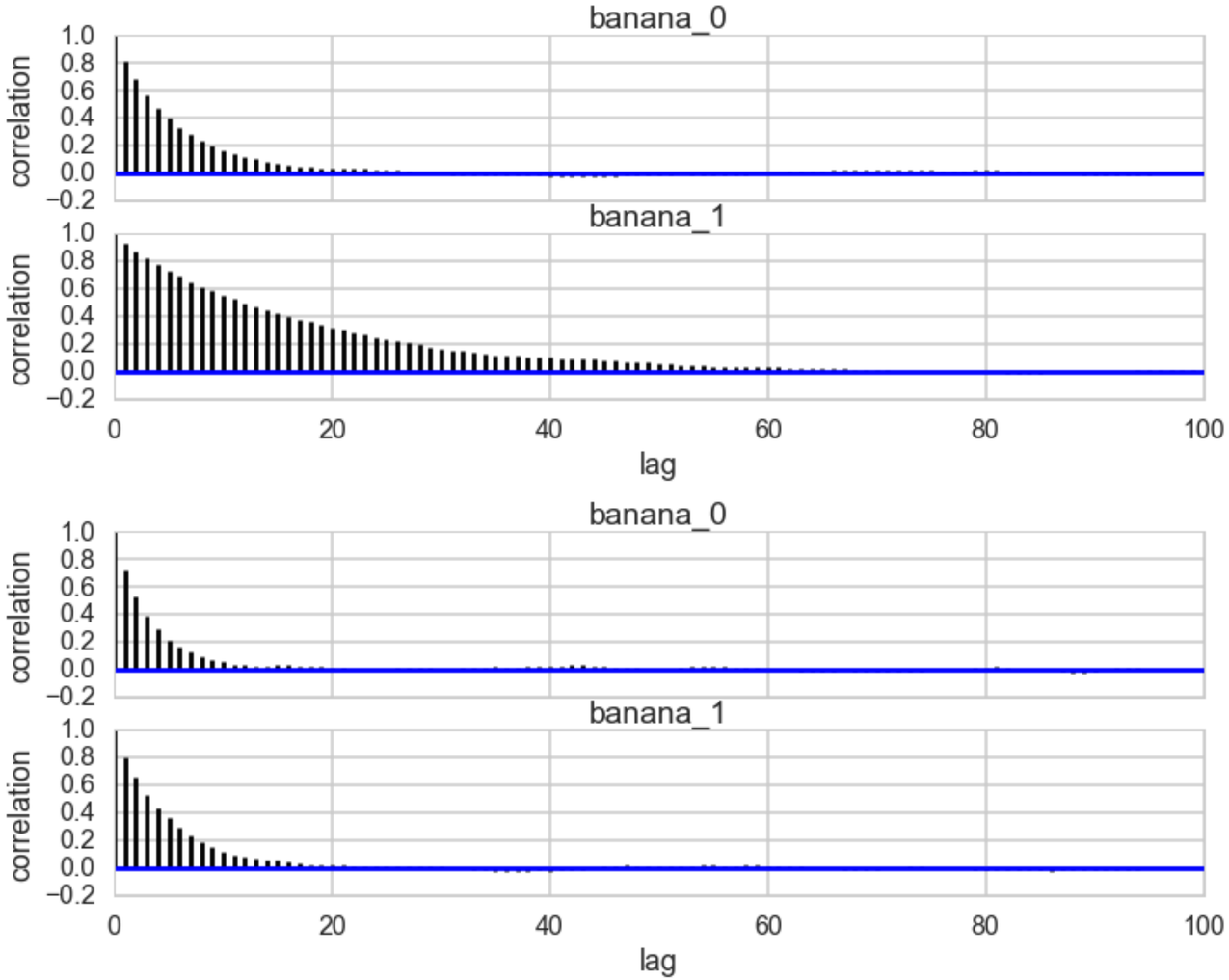
Tuning: integration time

- whats the best integration time?
- should we glide for a long time? then we wont get too may samples
- if our integration was exact we could glide for arbitrary short times
- but integration is not exact and will infact take us off the level set
- thus too many samples/too short time will get us back to MH


0
−100
−200
−300
−2
−1
0
1
2
x
2
1
0
−1
−2
y

HMC/NUTS in pymc3

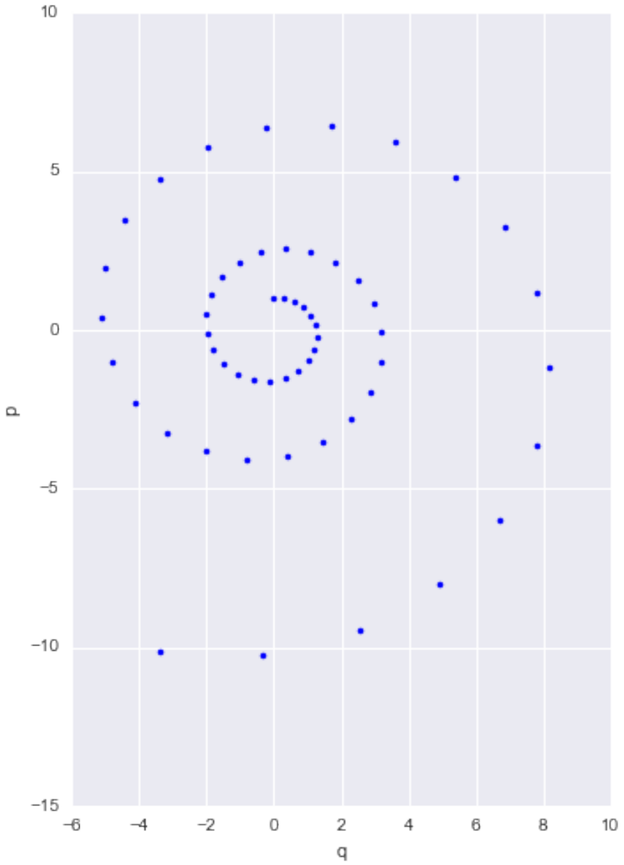
```
def clike2(value):
    x = value[0]
    y = value[1]
    val = -100 * (T.sqrt(y**2+x**2)-1)**2 + (x-1)**3 - y -5
    return (val)

with pm.Model() as model:
    banana = pm.DensityDist("custom", clike2, shape=2, testval=[1,1])

with model:
    start = pm.find_MAP()
    stepper=pm.Metropolis()
    trace=pm.sample(100000, step=stepper, start=start)
pm.autocorrplot(trace[20000::5])

with model:
    stepper_nuts=pm.NUTS()
    trace_nuts=pm.sample(100000, step=stepper_nuts)
pm.autocorrplot(trace_nuts[:16000])
```

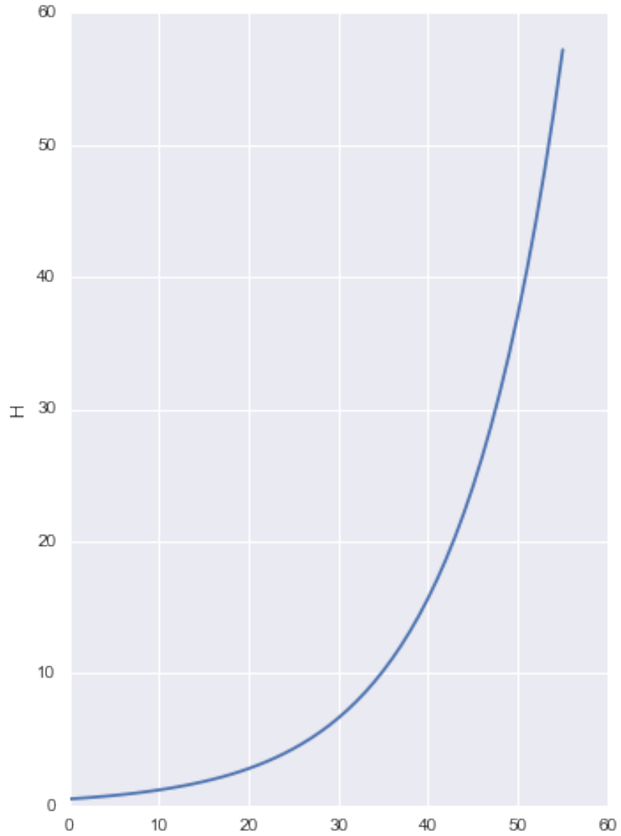


Problems

- discretization to solve differential equaions and the need for symplecticity
- lack of reversibility even with symplecticity (we are marginally off the level set)

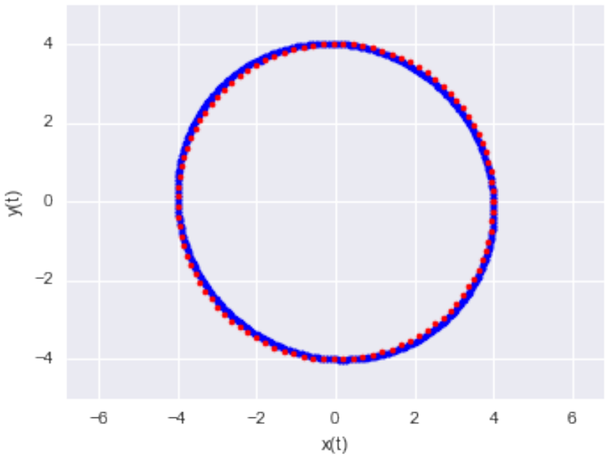
Practical implementation: Discretization and our problems

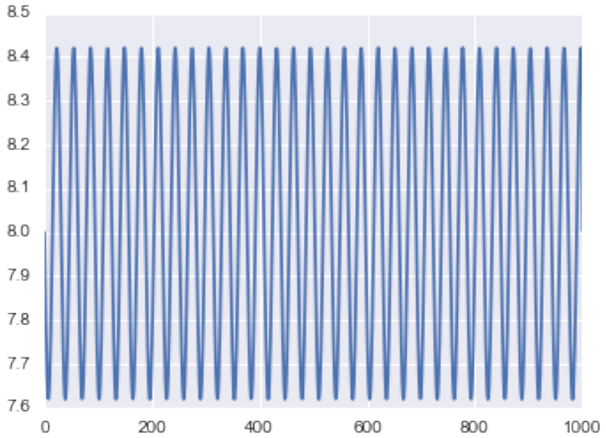




- $p_i(t+\epsilon) = p_i(t) - \epsilon \frac{\partial U}{\partial q_i}|_{q(t)}$
- $q_i(t+\epsilon) = q_i(t) + \epsilon \frac{p_i(t)}{m_i}$
- off-diagonal terms of size ϵ makes volume not preserved
- leads to drift over time

Sympletic Leapfrog





- Only *shear* transforms allowed, will preserve volume.
- $p_i(t+\frac{\epsilon}{2}) = p_i(t) - \frac{\epsilon}{2}\frac{\partial V}{\partial q_i}|_{q(t)}$
- $q_i(t+\epsilon) = q_i(t) + \epsilon\frac{p_i(t+\frac{\epsilon}{2})}{m_i}$
- $p_i(t+\epsilon) = p_i(t+\frac{\epsilon}{2}) - \frac{\epsilon}{2}\frac{\partial V}{\partial q_i}|_{q(t+\epsilon)}$
- still error exists, oscillatory, so reversibility not achieved