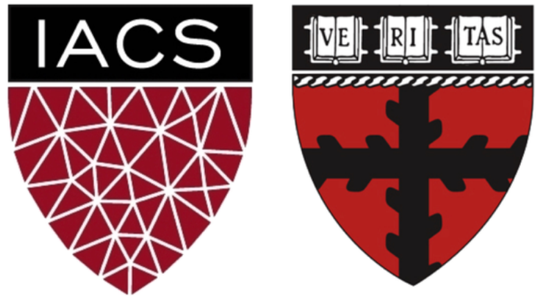
Lecture #14: Hamiltonian Monte Carlo

AM 207: Advanced Scientific Computing

Stochastic Methods for Data Analysis, Inference and Optimization

Fall, 2020


IACS
VE RI TAS

Outline

1. Hierachical Generalized Linear Models
2. Sampling as Optimization
3. Hamiltonian Monte Carlo

Hierarchical Generalized Linear Models

Generalized Linear Models

We can add covariates to all the other common statistical models of data $Y^{(n)} \sim p(Y^{(n)}|\theta^{(n)})$!

The general scheme is:

1. Set $\mathbb{E}[Y^{(n)}] = \theta^{(n)} = g(\mathbf{w}^\top \mathbf{X}^{(n)})$, where g is typically a non-linear function called the ***link function***

   Note: in literature, g^{-1} is called the link function.

2. Set the variance according to how the variance of $p(Y^{(n)}|\theta^{(n)})$ depends on $\theta^{(n)}$; $\mathrm{Var}[Y^{(n)}] = \phi f(\theta^{(n)})$, where ϕ is a constant.

These models are called ***generalized linear models***. Logistic regression is so called because g^{-1} is the ***logistic function***.

Kidney Cancer Dataset: Pooling vs Fully Independent Estimation

Rather than modeling the total incidents of kidney cancer, we collect information for each individual in each county. We can explain how the cancer rate depends on demogaphics.

$$\theta^{(n)} = \text{sigm}\left(\mathbf{w}^\top \mathbf{X}^{(n)}\right)$$

$$Y^{(n)} \sim Ber(\theta^{(n)})$$

1. $Y^{(n)}$ is the cancer status of n-th individual
2. $\mathbf{X}^{(n)}$ include income, level of education, access to preventative care, cost of living etc
3. $\theta^{(n)}$ is the underlying cancer rate for the n-th individual.

Pooled Estimation: In the above model, we pooled the data across all counties. We implicitly assume that the cancer rate $\theta^{(n)}$ depends on demographic factors $\mathbf{X}$ in the **same way for every individual**.

$$\text{Prob}[Y^n = 1|\mathbf{X}^{(n)}] = \text{sigm}\,(-1 * \text{income} + 2 * \text{BMI} + \ldots)$$

Independent Estimation: Alternatively, we can estimate a $\mathbf{w}^{(c)}$ for each county c, fully independently:

$$\theta^{(n,c)} = \text{sigm}\left((\mathbf{w}^{(c)})^\top \mathbf{X}^{(n,c)}\right)$$

$$Y^{(n,c)} \sim Ber(\theta^{(n,c)})$$

We implicitly assume that the way $\theta^{(n,c)}$ depends on demographic factors $\mathbf{X}$ **varies for every county independently**.

$$\text{Prob}[Y^{(n,1)} = 1|\mathbf{X}^{(n,1)}] = \text{sigm}\,(0.1 * \text{income} - 0.2 * \text{BMI} + \ldots)$$

$$\text{Prob}[Y^{(n,100)} = 1|\mathbf{X}^{(n,100)}] = \text{sigm}\,(-1 * \text{income} + 2 * \text{BMI} + \ldots)$$

Kidney Cancer Dataset : Hierachical vs Non-Hiearchical Models

Bayesian Model: Instead of assuming that the cancer rates in differen counties are fully independent. We can assume that the rates are **similar but differ from each other stochastically**.

$$
\begin{aligned}
\mathbf{w}^{(c)} &\sim p(\mathbf{w}|\phi) \\
\theta^{(n,c)} &= \text{sigm}\left((\mathbf{w}^{(c)})^\top \mathbf{X}^{(n,c)}\right) \\
Y^{(n,c)} &\sim Ber(\theta^{(n,c)})
\end{aligned}
$$

where ϕ is the set of hyperparameters of this model (i.e. you have to specify it). The prior $p(\mathbf{w}|\phi)$ expresses our belief that $\theta^{(n,c)}$'s are similar.

Hierachical Model: Rather than choosing ϕ arbitrarily, we can **infer ϕ from the data** by putting a prior on it.

$$
\begin{aligned}
\phi &\sim p(\phi|a) \\
\mathbf{w}^{(c)} &\sim p(\mathbf{w}|\phi) \\
\theta^{(n,c)} &= \text{sigm}\left((\mathbf{w}^{(c)})^\top \mathbf{X}^{(n,c)}\right) \\
Y^{(n,c)} &\sim Ber(\theta^{(n,c)})
\end{aligned}
$$

where a is the set of hyperparameters of this model (i.e. you have to specify it).

A Hierarchical Generalized Linear Model for Kidney Cancer

If we wanted to build an explanatory model for the regional differences in how cancer rate depends on demographic factors, we might want to consider a ***hierarchical generalized linear model*** as below:

$$
\begin{aligned}
\sigma^2_{\mathbf{w}} &\sim exp(1.5) \\
\mu &\sim \mathcal{N}(0, \mathbf{I}) \\
\mathbf{w}^{(c)} &\sim \mathcal{N}(\mu, \sigma^2_{\mathbf{w}}\mathbf{I}) \\
\theta^{(n,c)} &= \text{sigm}\left((\mathbf{w}^{(c)})^\top \mathbf{X}^{(n,c)}\right) \\
Y^{(n,c)} &\sim Ber(\theta^{(n,c)})
\end{aligned}
$$

The key object of interest for this model is the posterior

$$p(\mu, \sigma^2_{\mathbf{w}}, \{\mathbf{w}^{(c)}\}_c, \{\theta^{(n,c)}\}_{n,c} \mid \{Y_{n,c}\}_{n,c}).$$

Since there is no closed for this posterior, we must sample from it.

Question: Should we expect that sampling from the posterior of these hierarchical models to be easy? What happened in HW#5 with your sampler?

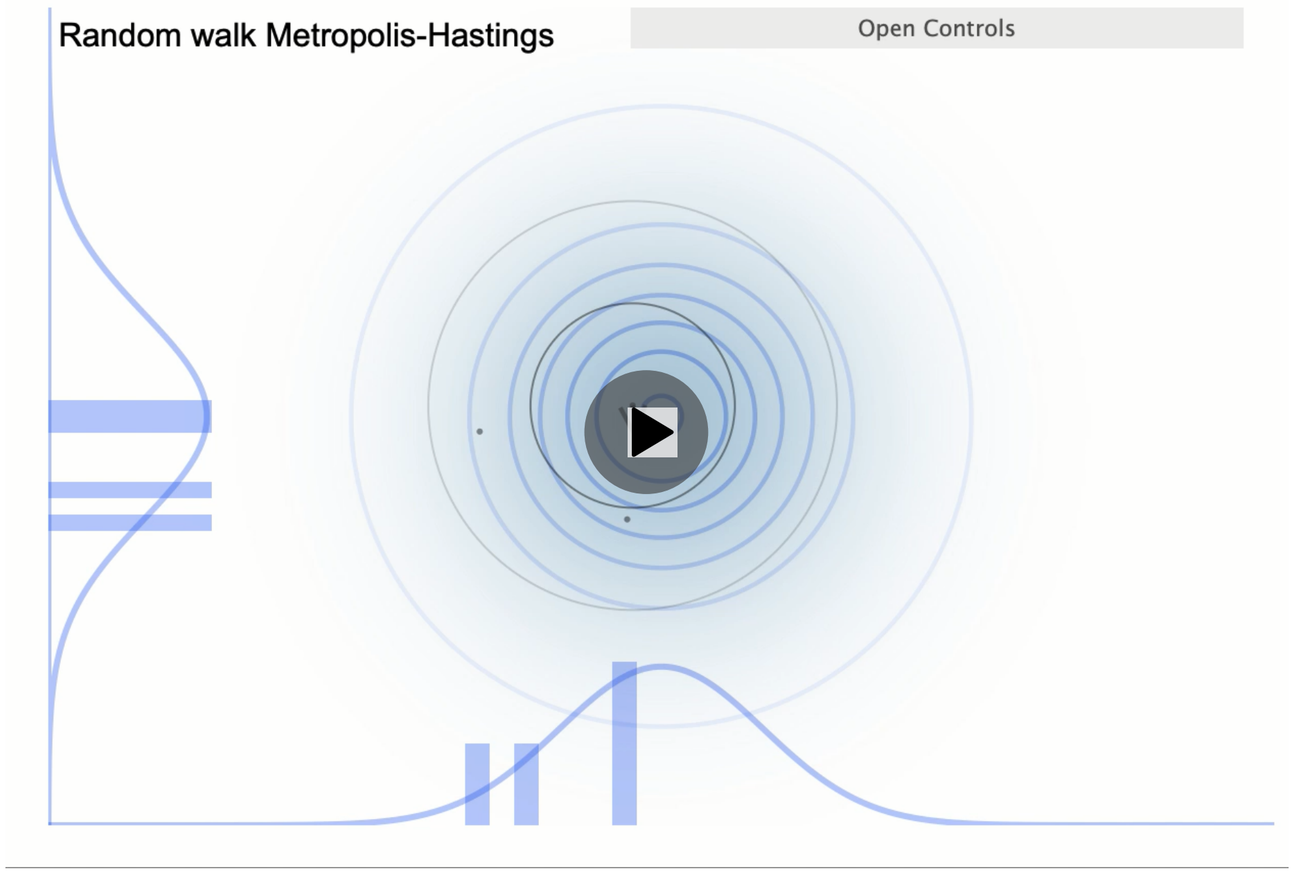
Sampling as Optimization

When is Metropolis Hastings Efficient?

Come up with a rule for what kinds of proposals get accepted. How many proposals will generally satisfy this rule?

In [22]:
```
HTML("""<video height="440" controls><source src="fig/mh_gaussian.mov" type="vid
eo/mp4"></video>""")
```

Out[22]:

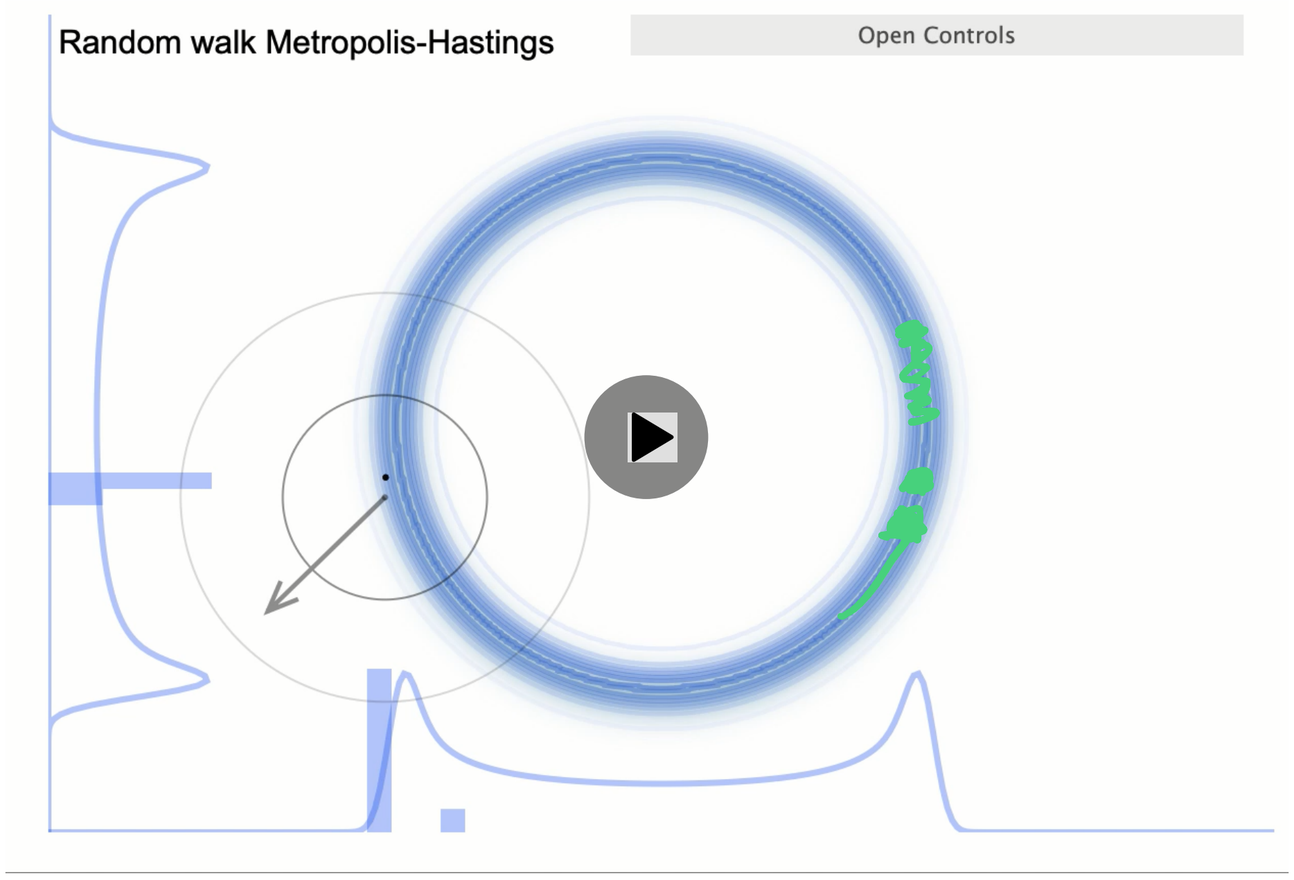


When is Metropolis Hastings Inefficient?

Come up with a rule for what kinds of proposals get accepted. How many proposals will generally satisfy this rule?

In [23]:
```
HTML("""<video height="440" controls><source src="fig/mh_donut.mov" type="video/
mp4"></video>""")
```

Out[23]:



The Connection Between Energy and Density Functions

In ***simulated annealing*** we turn the objective function $U(q)$ into a pdf $\pi(q)$, but with higher mass where U has lower values. This is the ***Gibbs distribution***:

$$\pi(q) = \frac{1}{Z}\exp\left\{-\frac{U(q)}{T}\right\}, \quad T \text{ is a constant}$$

where Z is the normalizing constant of the exponential part. The Gibbs distribution has a physical interpretation: $\pi(q)$ is the likelihood of being in state q given the energy $U(q)$ of state q and the temperature T. For now, we set $T = 1$.

Using the same relationship, we can turn a probability distribution into an energy function

$$U(q) = -\log \pi(q).$$

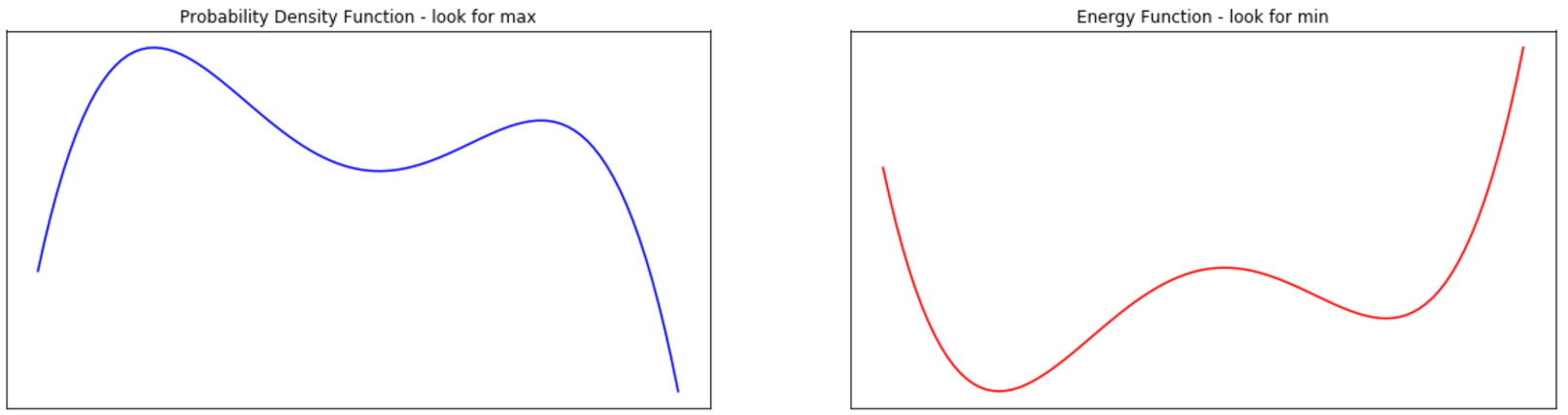
Motto: negative gradient of the energy function points to the direct of steepest decrease in U and the **steepest increase** in π.

The Connection Between Optimization and Sampling

If we wanted to sample from high density regions of π, why don't we turn π into an energy function $U(q) = -\log \pi(q)$ and use gradient descent to find the valleys of U (and hence modes of π)?

Why isn't gradient descent on U a valid way to sample from π?

In [14]:
```
fig, ax = plt.subplots(1, 2, figsize=(20, 5))
ax = plot_energy_gibbs(f, x, ax)
plt.show()
```

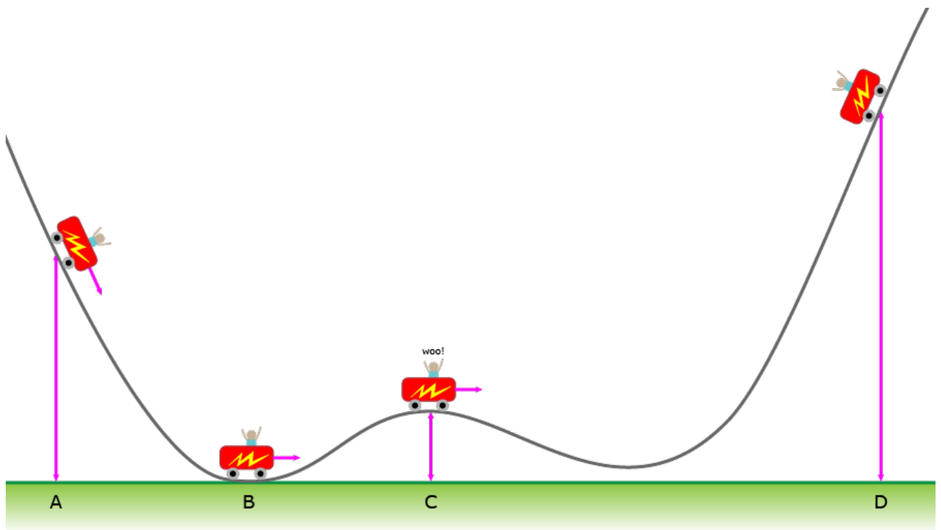


Gradient Descent with Random Momentum: Simulating Mechanics of Moving Particles

To randomly sample from π we need to perform gradient-baesed descent on U with some randomness to make sure we move around. But we need to make sure this randomness is controlled (i.e. doesn't overwhelm the information in the gradient).

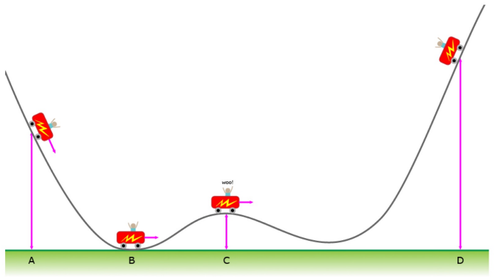
What if we performed gradient descent with an initial random momentum? That is, we simulated the way a particle with mass would move on the graph of U if we started its motion with a push (an intitial direction and magnitude of force):

I. at first, the particle would tend to travel in the direction of the initialization
II. eventually, gravity will pull it towards the minima (it will perform gradient descent)

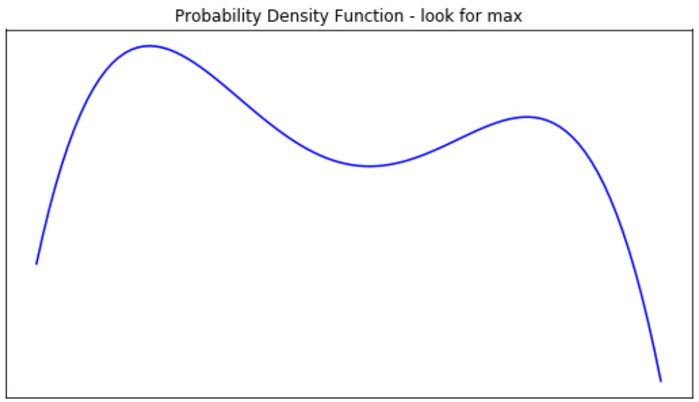


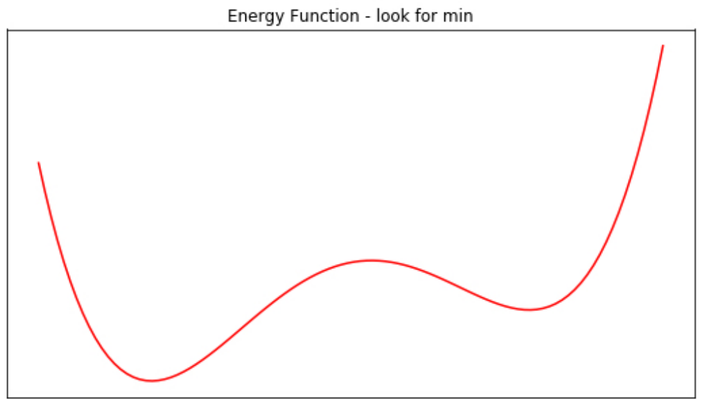
Sampling by Physical Simulation

We can now sketch a sampler of π based on the simulations physical explorations of the graph of it's energy function U:



1. start at a random q, pick a mass M
2. repeat:
    A. start at the current sample q_{current}
    B. pick a random momentum vector (random direction and magnitude of force)
    C. simulate, for a period of T seconds, the movement of a particle with mass M moving on the graph of U with this momentum
    D. the q-coordinate(s) of where the particle lands at the T-th second is the new sample q_{new}
    E. set $q_{\text{current}} \leftarrow q_{\text{new}}$





Hamiltonian Monte Carlo

Hamiltonian Motion as Differential Equations

What does it mean to simulate the movement of a particle on the graph of U? It means to exchange potential energy (height) with kinetic energy (related to speed) in a way that obeys the ***conservation of total energy***.

A physical properties of system with position $q(t)$ and momentum $p(t)$ is described by a function $H(q, p)$ called the ***Hamiltonian***. We will always choose a Hamiltonian that decomposes as the sum of potential and kinetic energy as follows:

$$\underbrace{H(q, p)}_{\text{total}} = \underbrace{U(q(t))}_{\text{potential}} + \underbrace{K(p(t))}_{\text{kinetic}}$$

How the system (q, p) changes over time is described by the partial derivatives of H:

$$\frac{dq}{dt} = \underbrace{\frac{\partial H}{\partial p}}_{\text{function of } t}, \qquad \frac{dp}{dt} = \underbrace{-\frac{\partial H}{\partial q}}_{\text{function of } t}$$

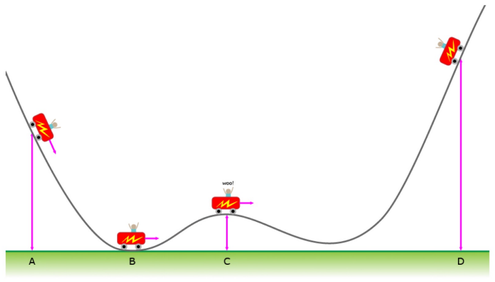
This forms a system of ***differential equations*** -- equations describing the derivatives/gradients of a function. We call this a ***Hamiltonian system***.

Say the system is at (q_0, p_0) at time $t = 0$, the system's state at time $t = T$ is given by **integrating** the differential equations from $t = 0$ to $t = T$ and solving for constants based on $q(0) = q_0, p(0) = p_0$. The map $\phi(T)$ transforming the state (q_0, p_0) to the state at (q_T, p_T) is called the ***flow*** of the Hamiltonian system. The principle of conservation of total energy says that $H(q, p)$ is constant along every point q_t, p_t on every flow.

Hamiltonian Monte Carlo: with Exact Integration

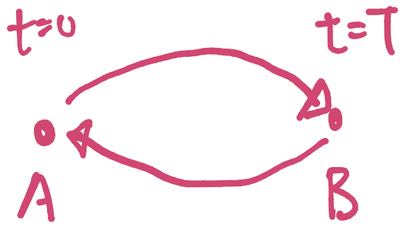
Let $\pi(q)$ be our target distribution with $q \in \mathbb{R}^D$. We turn π into an energy function $U(q)$ by $U(q) = -\log(\pi(q))$. We choose a kinetic energy function $K(p)$. We fix $t = T$.

1. start with a random $q^{(0)} \in \mathbb{R}^D$
2. repeat:
    A. (**kick-off**) sample a random momentum from the Gibbs distribution of $K(p)$, i.e. $p^{(current)} \sim \frac{1}{Z}\exp\{-K(p)\}$.
    B. (**simulate movment**) simulate Hamiltonian motion for a time interval of length T, i.e. start at $(q^{(current)}, p^{(current)})$ and find $(q^{(\text{time } T)}, p^{(\text{time } T)})$. You need to integrate $\frac{dq}{dt} = \frac{\partial H}{\partial p}, \frac{dp}{dt} = -\frac{\partial H}{\partial q}$!
    C. accept $q^{(current)}$ as a new sample: $q^{(current)} \leftarrow q^{(\text{time } T)}$.



Questions: is this a correct way to sample from π? But isn't the integration in Step B hard?

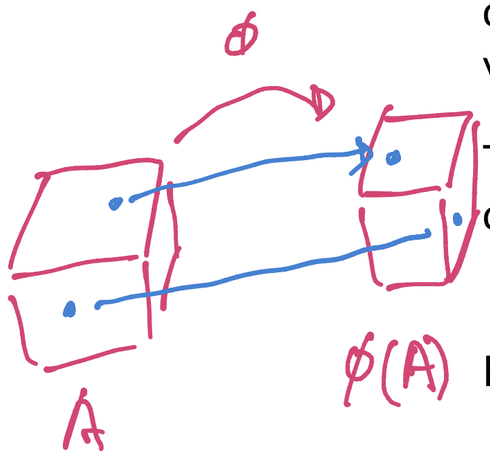
Correctness of HMC with Exact Integration



Two important properties of Hamiltonian dynamics:

I. (**Reversibility**) The flow $\phi(T)$ of a Hamiltonian system is one-to-one and onto for every interval length T, i.e. given $(q^{(\text{time } T)}, p^{(\text{time } T)})$ we can compute the point of origin $(q^{(\text{time } 0)}, p^{(\text{time } 0)})$.

II. (**Volume Preserving**) It's easy to verify that the vector field of a Hamiltonian system is divergence free, which will imply that the flow $\phi(T)$ is volume preserving -- if $A \subset \mathbb{R}^D$ has volume V then the image of A under $\phi(T)$ has volume V.



The above two properties imply detailed balance for HMC with respect to the joint distribution (see readings):

$$\pi(q, p) = \frac{1}{Z}\exp\{-H(q, p)\}.$$

But we can rewrite the joint distribution $\pi(q, p)$:

$$\begin{aligned}\pi(q, p) = \frac{1}{Z}\exp\{-U(q) - K(p)\} &= \exp\{-U(q)\}\frac{1}{Z}\exp\{-K(p)\} \\ &= \exp\{\log(\pi(q))\}\frac{1}{Z}\exp\{-K(p)\} \\ &= \pi(q)\frac{1}{Z}\exp\{-K(p)\}\end{aligned}$$

That is, q and p are independent. Hence, sampling from $\pi(q, p)$ and ignoring p gives us samples from $\pi(q)$.

Approximating Flows: Sympletic Integrators

Simulating Hamiltonian dynamics for a time period of length T, using the flow $\phi(T)$, requires that we integrate the equations in the Hamiltonian system with respect to t,

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

This is generally intractable to do analytically! In HMC, we approximate the dynamics by:

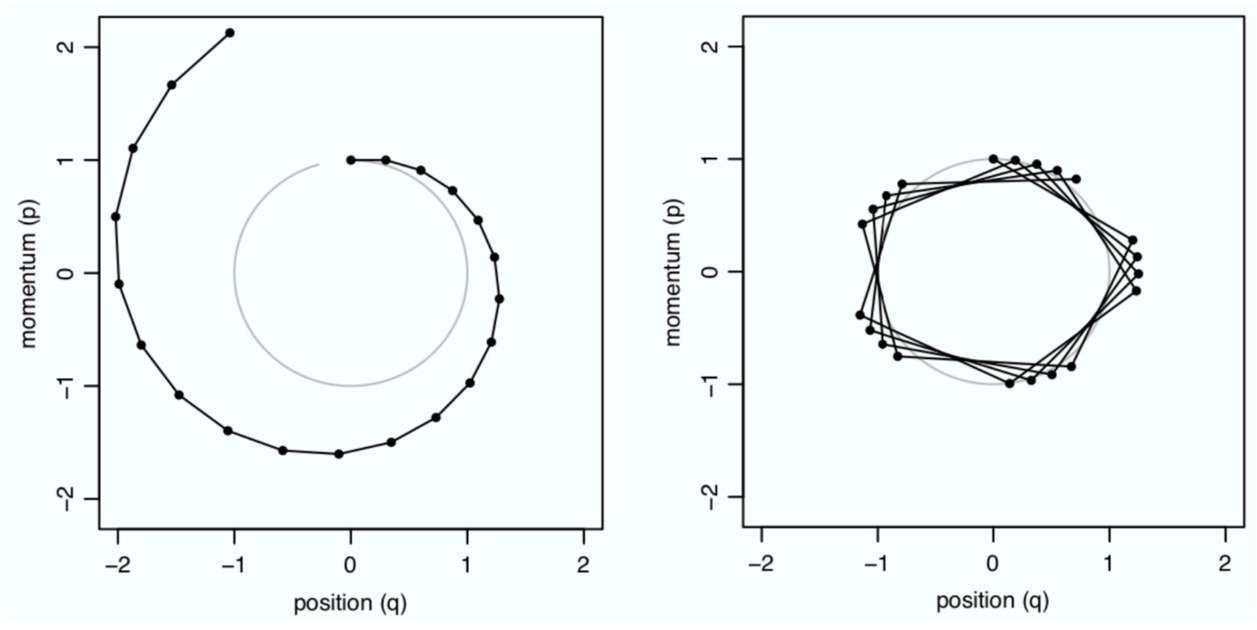
1. discretizing the time period $[0, T]$ into L chunks each with length ϵ.
2. **approximating** the flow for a time period of length ϵ.

A common method of approximating the flow is called the ***leap-frog integrator***. This integrator has the special property of being **reversible** and **volume-preserving**.

This means that replacing exact Hamiltonian dynmaics in HMC with the leap-frog integration will not obviously ruin detailed balance!

Approximating Flows: Adjusting for Error

But the approximation is not exact! The longer the time period ϵ the more error in the approximation. The more steps in your simulation (i.e. the longer T is) the more these errors will accumulate. To ensure the correctness of the sampling, **we need to add a Metropolis-Hastings step** so that we can potentially reject the sample from the approximation.



Hamiltonian Monte Carlo: with the Leap Frog Integrator

Let $\pi(q)$ be our target distribution with $q \in \mathbb{R}^D$. We turn π into an energy function $U(q)$ by $U(q) = -\log(\pi(q))$. We choose a kinetic energy function $K(p)$.

0. start with a random $q^{(0)} \in \mathbb{R}^D$

1. repeat:

A. (**kick-off**) sample a random momentum from the Gibbs distribution of $K(p)$, i.e. $p^{(current)} \sim \frac{1}{Z}\exp(-K(p))$.

B. (**simulate movement**) simulate Hamiltonian motion for L steps each with time interval ϵ, using the leap-frog integrator.

a. Repeat for T - 1 times, for $p^{(\text{step } 0)} = p^{(current)}, q^{(\text{step } 0)} = q^{(current)}$:

i. (half-step update for momentum) $p^{(\text{step } t+1/2)} \leftarrow p^{(\text{step } t)} - \epsilon/2 \frac{\partial U}{\partial q}(q^{(\text{step } t)})$

ii.* (full-step update for position) $q^{(\text{step } t+1)} \leftarrow q^{(\text{step } t)} + \epsilon \frac{\partial K}{\partial p}(p^{(\text{step } t)})$

iii. (half-step update for momentum) $p^{(\text{step } t+1)} \leftarrow p^{(\text{step } t+1/2)} - \epsilon/2 \frac{\partial U}{\partial q}(q^{(\text{step } t+1)})$

b. (**reverse momentum**) $p^{(\text{step } T)} \leftarrow -p^{(\text{step } T)}$

C.(**correction for simulation error**) implement Metropolis-Hasting accept mechanism:

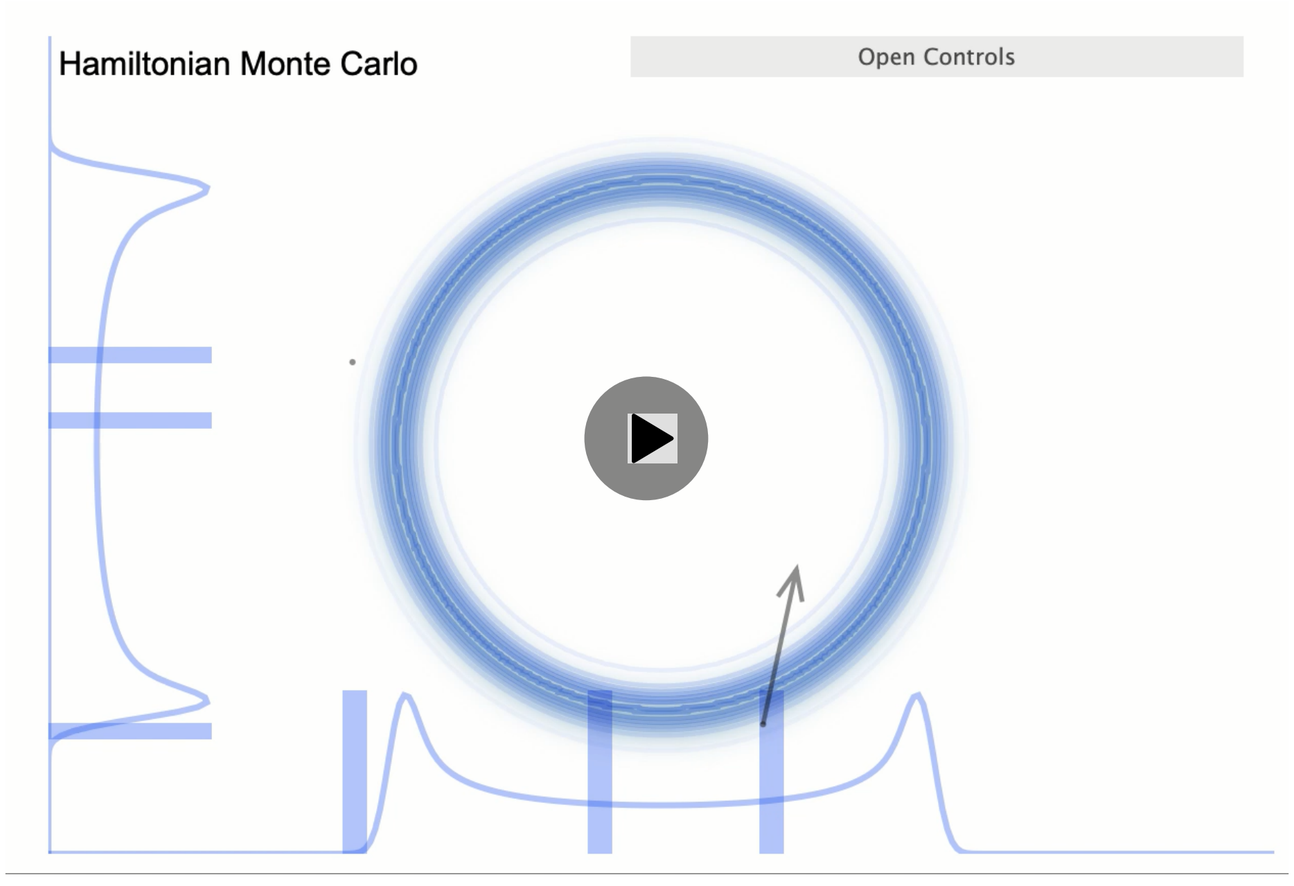
a. compute $\alpha = \min\left(1, \exp\left\{H(q^{(current)}, p^{(current)}) - H(q^{(\text{step } T)}, p^{(\text{step } T)})\right\}\right)$

b. sample $U \sim U(0, 1)$, if $U \leq \alpha$ then accept, else keep old sample

HMC with Leap Frog Integrator in Action

In [57]:
```
HTML("""<video height="440" controls><source src="fig/hmc_donut.mov" type="video
/mp4"></video>""")
```

Out[57]:



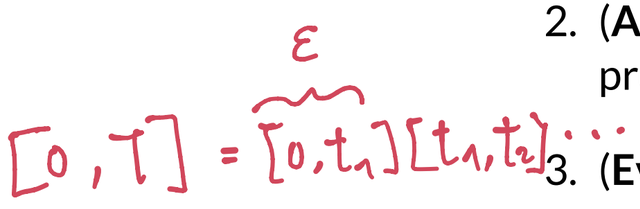
Hamiltonian Monte Carlo: Summary

Given a target distribution π, we

1. turn $\pi(q)$ into a potential energy function $U(q) = -\log(\pi(q))$. We choose a kinetic energy function $K(p)$.
2. generate samples $\{q_s\}$ by approximately simulating Hamiltonian dynamics for a particle (i.e. realistic movement on the graph of $U(q)$) with random intialization.
3. for each sample we need to accept based on a Metropolis-Hastings mechanism because the simulation is not exact.

Design choices:

1. (**Advanced**) kinetic energy function $K(p)$ -- needs to be one such that you can easily sample from its Gibbs distribuion $\frac{1}{Z}\exp(-K(p))$.
2. (**Advanced**) integrator to approximate Hamiltonian flow -- this needs to be volume-preserving and reversible.
3. (**Everyone**) number and size of discrete time steps for discrete integrators.

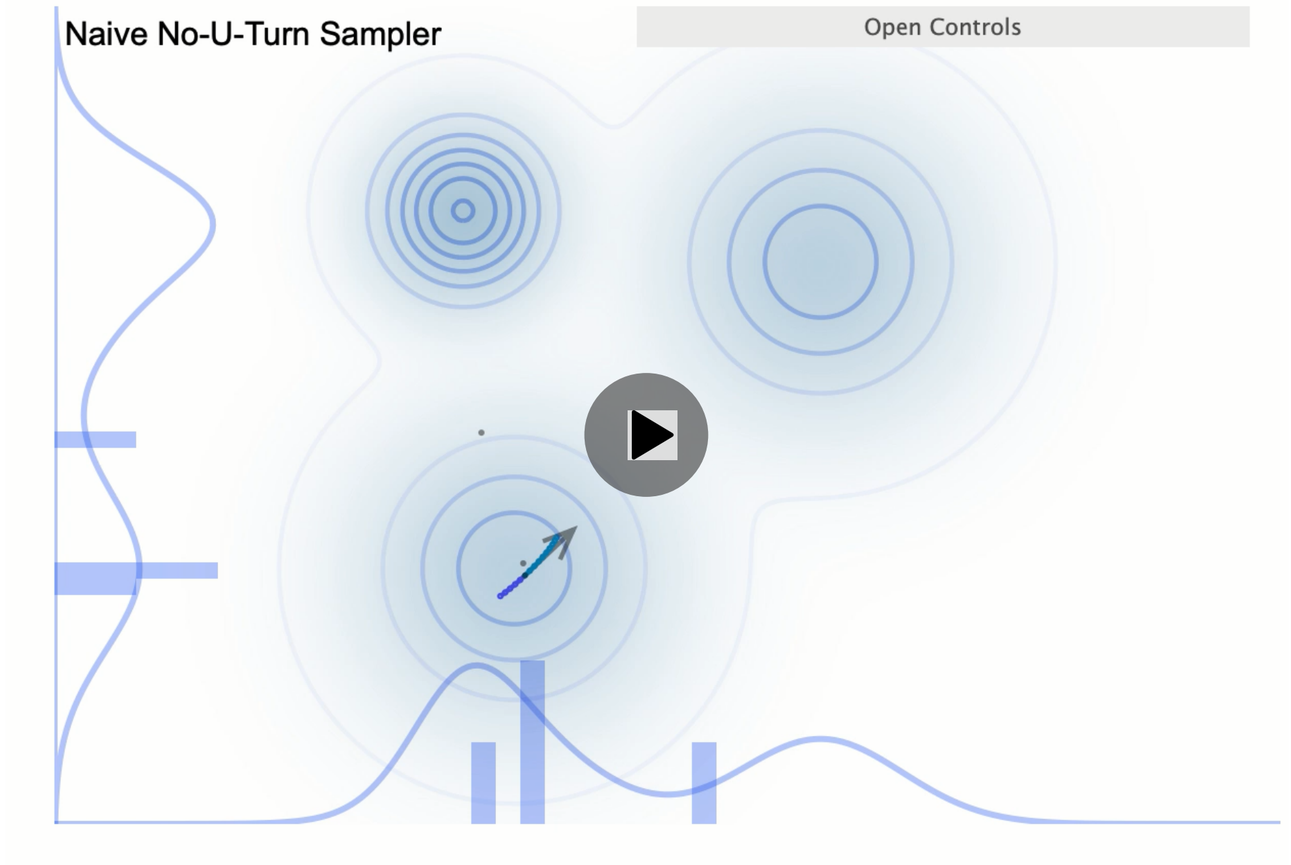


What happens when the step sizes are large and we take many steps? What happens when the step sizes are tiny and we take few?

HMC for Multimodal Distributions

In [55]:
```
HTML("""<video height="440" controls><source src="fig/hmc_multimodal.mov" type="
video/mp4"></video>""")
```

Out[55]:



In [ ]: