Sampling

Muchang Bahng

Spring 2024

Contents

Monte carlo algorithms is a general term for computational techniques that use random numbers, which can be used both in classical and Bayesian statistics. This is extremely important when working with distributions that are cannot be simply stated using elementary densities (Gaussian, Beta, etc.). The entire goal of Bayesian inference is to maintain a full posterior probability distribution over a set of random variables. However, maintaining and using this distribution requires computing integrals which, for most non-trivial models, is intractable.

The basic idea of MCMC is that we want to construct a Markov chain which will travel between different possible states (e.g. the hypotheses/parameter values in a Bayesian analysis), where the amount of time spent in any particular state is proportional to the posterior probability of the state. That is, the stationary distribution of the chain is the posterior distribution. As a result, the computer explores the set of possible parameter values, spending a lot of time in the regions with high posterior probability, and only rarely visiting regions of low posterior probability.

1 Metropolis Hastings

Say that with initial distribution $p(\theta)$, we have calculated the posterior as

$$p(\theta \mid x) \propto p(\theta)\, p(x \mid \theta) \tag{1}$$

It is often the case that the set of possible values of θ is very large, so it is computationally inefficient to compute the normalizing factor

$$p(x) = \sum_\theta p(\theta)\, p(x \mid \theta) \tag{2}$$

Therefore, we only have this function $f(\theta) = p(\theta)\, p(x \mid \theta)$ that is directly proportional to $p(\theta \mid x)$. That is, we don't know the normalizing constant c such that

$$p(\theta \mid x) = \frac{f(\theta)}{c} \tag{3}$$

Using this information, we wish to construct and run an algorithm that converges onto the true posterior distribution $p(\theta \mid x)$ at a sufficiently fast rate.

We begin by constructing a discrete-time irreducible Markov chain with state space $\mathcal{S} = \{1, 2, \ldots, N\}$ representing the set of possible parameter values (the labels for the elements of $\mathcal{S}$ does not matter, since we can construct whatever bijection we want from the actual states to a subset of $\mathbb{N}$). Like a normal Markov chain, we will choose the next state to go to at each step, *but now, we will then choose to accept this proposal to go to that step with an additional probability.* That is, we will construct two matrices:

- An $|\mathcal{S}| \times |\mathcal{S}|$ **proposal transition matrix** Q_{prop}, with

$$p(\text{propose } i \mapsto j) = (Q_{prop})_{ij} = q_{prop}(i, j) \tag{4}$$

  being the probability of getting a *proposal* to transition from state i to state j. This matrix is constructed by the user and is completely well-defined and known; this choice may also affect the convergence rate. Note that with this formulation, the rows will sum up to 1 and Q^T is a stochastic matrix. We can also construct Q_{prop} to be symmetric, that is $q_{prop}(i, j) = q_{prop}(j, i)$, for easier calculations.

- An $|\mathcal{S}| \times |\mathcal{S}|$ **acceptance probability matrix** A, with

$$\begin{aligned}
(\text{accept proposal } i \mapsto j \mid \text{propose } i \mapsto j) &= (A)_{ij} = \alpha(i, j) \\
&= \min\left(1, \frac{p(\theta = j \mid x)\; q_{prop}(j, i)}{p(\theta = i \mid x)\; q_{prop}(i, j)}\right) \\
&= \min\left(1, \frac{f(\theta = j)\; q_{prop}(j, i)}{f(\theta = i)\; q_{prop}(i, j)}\right) \\
&= \min\left(1, \frac{f(\theta = j)}{f(\theta = i)}\right) \qquad (\text{if } Q_{prop} \text{ symmetric})
\end{aligned}$$

Then, we element-wise multiply the two matrices, except the diagonals, to get the **true transition matrix** Q defined

$$(Q)_{ij} = q(i, j) = \begin{cases} q_{prop}(i, j) \cdot \alpha(i, j) = q_{prop}(i, j) \cdot \min\left(1, \frac{f(\theta = j)\; q(j,i)}{f(\theta = i)\; q(i,j)}\right) & \text{if } i \neq j \\ 1 - \sum_{j \neq i} q(i, j) & \text{if } i = j \end{cases} \tag{5}$$

where $q(i, j)$ represents the **true transition probability** of going from state i to state j. Note that we have element-wise multiplied every non-diagonal element, and we have defined $(Q)_i i$ such that the sum of each row is 1 (so that this becomes a viable transition matrix). Note also that this element-wise multiplication makes sense because

$$\begin{aligned} p(\theta_{k+1} = j \,|\, \theta_k = i) &= p(\text{accept proposal } i \mapsto j, \text{ propose } i \mapsto j) \\ &= p(\text{accept proposal } i \mapsto j \,|\, \text{propose } i \mapsto j)\; p(\text{propose } i \mapsto j) \\ &= \alpha(i,j) \cdot q_{prop}(i,j) \end{aligned}$$

This is the Markov chain we wish to get, where "one" step is really a two-step process of proposing and accepting/rejecting. We wish to get the steady state distribution $\pi(\theta)$ of this chain, which can be found in two well-known ways:

- Calculate the left-eigenvector of Q with eigenvalue 1.
- Randomly initialize θ_0 and run the chain for a sufficiently long time to record where it lands at each step
$$\theta_0 = i_0, \theta_1 = i_1, \theta_2 = i_2, \theta_3 = i_3, \ldots, \theta_n = i_n \tag{6}$$
which can be used to approximate $\pi(\theta)$ by defining
$$\pi(\theta = i) = \frac{\text{proportion of states in state } i \text{ in the n-step process}}{n} \tag{7}$$

Finally, we claim that this steady state distribution $\pi(\theta)$ is precisely the posterior we are looking for: $p(\theta \,|\, x)$.

1.1 Algorithm

Given that we have computed a scalar multiple of a high dimensional posterior $\pi = \frac{f}{c}$ defined in $\mathbb{R}^n$ for $n >> 1$, we would like to either optimize f or sample from f to find its true normalizing factor c. There are some overlaps in the methods used to achieve these goals. Let us denote our (parameter) state as $\theta \in \mathbb{R}^n$, with a discrete time step denoted by t and step size h.

Markov Chain Monte Carlo algorithms are extremely simple and computationally efficient, since they only require to compute $f(\theta)$, without any gradient information. They generate a sequence of correlated samples which on the long run converge to a sequence of independent samples. The degree of correlation of nearby samples is called the *autocorrelation* of the MCMC sampler. We first generate a proposal step according to some kernel and then decide whether to accept or reject that proposal. Usually, we have a series of "burn-in" steps that allow the chain to first converge to a local maximum, which we can then throw away. The simplest version of this is with an isotropic Gaussian kernel.

Note that the step size is very important here: If h is too small, then this chain would behave like a random walk. If h it too big, then this chain would mainly stay at one state. Ideally, the acceptance probability should be between 0.2 and 0.7.

This isotropic MH is not robust, since it would not work well if some parameters of θ are correlated and the estimated covariance of f at some local maximum is more "diagonal." Therefore, some adaptive mechanism is needed, which we can implement by estimating the covariance matrix of the proposal kernel using the empirical covariance of the proposal steps. To reduce memory allocation, we should use a recursive algorithm to compute the mean and covariance, rather than having to store all the θ_t's. To maintain stability, we may start adapting after a certain number of steps B and compute covariance estimates every U steps.

On top of this even, we can precondition the initial Σ_0 to be some other estimate of the posterior and weight it accordingly so that our proposal covariance is some "balance" of this computed estimate and the empirical estimate, using a damping parameter α.

$$\Sigma_t = \alpha \Sigma_0 + (1-\alpha)\Sigma^{\text{emp}}, \qquad 0 \leq \alpha \leq 1 \tag{8}$$

The lower the α, the more the precomputed estimate is "washed away" by the empirical covariance. We can also treat the α as a variable function $\alpha(t)$ and adapt its value as the chain runs. For example, if we

Algorithm 1 Random Walk Metropolis Hastings w/ Isotropic Gaussian Kernel

```
Require: Initial $\boldsymbol{\theta}_0$, Stepsize $h$, Burn-in steps $\mathcal{B}$
  for $t = 0$ to $T$ do
    $\epsilon_t \sim \mathcal{N}(0, I)$
    $P_{t+1} \leftarrow \theta_t + \epsilon_t$
    if $f(P_{t+1}) \geq f(\theta_t)$ then
      $\theta_{t+1} \leftarrow P_{t+1}$
    else
      $\delta \sim \text{Uniform}[0, 1]$
      if $\delta < f(P_{t+1})/f(\theta_t)$ then
        $\theta_{t+1} \leftarrow P_{t+1}$
      else
        $\theta_{t+1} \leftarrow \theta_t$
      end if
    end if
  end for
  Delete first $\mathcal{B}$ states of $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_T]$
```

Algorithm 2 Adaptive Random Walk Metropolis

```
Require: Initial $\boldsymbol{\theta}_0$, Stepsize $h$, Burn-in steps $\mathcal{B}$, Adaptation burn-in $B$, Adaptation frequency $U$
  $\mu_0^{\text{emp}} \leftarrow 0$
  $\Sigma_0 \leftarrow I$
  $\Sigma_0^{\text{emp}} \leftarrow I$
  for $t = 0$ to $T$ do
    $\epsilon_t \sim \mathcal{N}(0, \Sigma_t)$
    $P_{t+1} \leftarrow \theta_t + \epsilon_t$
    if $f(P_{t+1}) \geq f(\theta_t)$ then
      $\theta_{t+1} \leftarrow P_{t+1}$
    else
      $\delta \sim \text{Uniform}[0, 1]$
      if $\delta < f(P_{t+1})/f(\theta_t)$ then
        $\theta_{t+1} \leftarrow P_{t+1}$
      else
        $\theta_{t+1} \leftarrow \theta_t$
      end if
    end if
    $\Sigma_{t+1}^{\text{emp}} \leftarrow \frac{1}{t+1}\big[(\theta^{t+1} - \mu_t)(\theta^{t+1} - \mu_t)^T - \Sigma_t^{\text{emp}}\big]$
    $\mu_{t+1}^{\text{emp}} \leftarrow \mu_t + \frac{1}{t+1}[\theta_{t+1} - \mu_t]$
    if $t > B$ and $t$ is divisible by $U$ then
      $\Sigma_{t+1} \leftarrow \Sigma_{t+1}^{\text{emp}}$
    end if
  end for
  Delete first $\mathcal{B}$ states of $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_T]$
```

would like the precomputed covariance to have more weight in the beginning (for stability), but eventually completely overpowered by the empirical covariance, we can choose it such that $\alpha(0) = 1$ and $\alpha \to 0$ as $t \to +\infty$, with the specific behavior customized to the problem.

Algorithm 3 Adaptively Preconditioned Random Walk Metropolis

```
Require: Initial $\boldsymbol{\theta}_0$, Stepsize $h$, Burn-in steps $\mathcal{B}$, Adaptation burn-in $B$, Adaptation frequency $U$, Damping
    function $\alpha$, Precomputed covariance estimate $\Sigma^{\text{pre}}$
  $\mu_0^{\text{emp}} \leftarrow 0$
  $\Sigma_0^{\text{emp}} \leftarrow I$
  $\Sigma_0 \leftarrow I$
  for $t = 0$ to $T$ do
      $\epsilon_t \sim \mathcal{N}(0, \Sigma_t)$
      $P_{t+1} \leftarrow \theta_t + \epsilon_t$
      if $f(P_{t+1}) \geq f(\theta_t)$ then
          $\theta_{t+1} \leftarrow P_{t+1}$
      else
          $\delta \sim \text{Uniform}[0, 1]$
          if $\delta < f(P_{t+1})/f(\theta_t)$ then
              $\theta_{t+1} \leftarrow P_{t+1}$
          else
              $\theta_{t+1} \leftarrow \theta_t$
          end if
      end if
      $\Sigma_{t+1}^{\text{emp}} \leftarrow \frac{1}{t+1}\big[(\theta^{t+1} - \mu_t)(\theta^{t+1} - \mu_t)^T - \Sigma_t^{\text{emp}}\big]$
      $\mu_{t+1}^{\text{emp}} \leftarrow \mu_t + \frac{1}{t+1}[\theta_{t+1} - \mu_t]$
      if $t > B$ and $t$ is divisible by $U$ then
          $\Sigma_{t+1} \leftarrow \alpha(t) \cdot \Sigma^{\text{pre}} + (1 - \alpha(t)) \cdot \Sigma_{t+1}^{\text{emp}}$
      end if
  end for
  Delete first $\mathcal{B}$ states of $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_T]$
```

1.2 Detailed Balance: Justification of the Metropolis Algorithm

But why does $\pi(\theta) = p(\theta \,|\, x)$? Given a Markov chain θ_0 with transition matrix Q, the chain is said to satisfy **detailed balance** with respect to a distribution $\pi(\theta)$ if

$$\pi(\theta = i)q(i, j) = \pi(\theta = j)q(j, i) \tag{9}$$

for all $i, j \in \mathcal{S}$. In fact, we claim that θ_i does satisfy detailed balance with respect to $p(\theta \,|\, x)$. That is, it satisfies

$$p(\theta = i \,|\, x)q(i, j) = p(\theta = j \,|\, x)q(j, i) \tag{10}$$

This case is trivial for when $i = j$, so assume $i \neq j$. A transition from i to a different j can only be achieved with an accepted proposed step, which happens with probability

$$\begin{aligned}
q(i, j) &= q_{prop}(i, j) \cdot \alpha(i, j) \\
&= q_{prop}(i, j) \cdot \min\left(1, \frac{p(\theta = j \,|\, x)\; q_{prop}(j, i)}{p(\theta = i \,|\, x)\; q_{prop}(i, j)}\right) \\
&= \frac{q_{prop}(i, j)}{p(\theta = i \,|\, x)} \min\big(p(\theta = i \,|\, x), \frac{p(\theta = j \,|\, x)\; q_{prop}(j, i)}{q_{prop}(i, j)}\big) \\
&= \frac{1}{p(\theta = i \,|\, x)} \min\big(p(\theta = i \,|\, x)\; q_{prop}(i, j), p(\theta = j \,|\, x)\; q_{prop}(j, i)\big)
\end{aligned}$$

Applying the same method from transitioning from j to i gives the same equation, but with the i and j's switched.

$$q(j,i) = \frac{1}{p(\theta = j \mid x)} \min\big(p(\theta = j \mid x)\, q_{prop}(j,i), p(\theta = i \mid x)\, q_{prop}(i,j)\big) \tag{11}$$

But switching the i and j leaves the terms inside the minimum invariant. Therefore, we can see that

$$p(\theta = i \mid x)\, q(i,j) = \min\big(p(\theta = j \mid x)\, q_{prop}(j,i), p(\theta = i \mid x)\, q_{prop}(i,j)\big) = p(\theta = j \mid x)\, q(j,i) \tag{12}$$

proving detailed balance. Now, we can sum the left hand side of the detailed balance equation over i to get

$$\sum_i p(\theta = i \mid x) q(i,j) = \sum_i p(\theta = j \mid x) q(j,i) = p(\theta = j \mid x) \sum_i q(j,i) = p(\theta = j \mid x) \tag{13}$$

which in matrix form, says

$$p(\theta \mid x) Q = p(\theta \mid x) \tag{14}$$

where $p(\theta \mid x) = \big(p(\theta = 1 \mid x) \ldots p(\theta = N \mid x)\big)$ and $Q_{ij} = q(i,j)$. This implies that $p(\theta \mid x)$ is a stationary distribution, and therefore, computing the stationary distribution is equivalent to computing $p(\theta \mid x)$.

The intuition behind detailed balance is quite easy to understand, too. Suppose we start a chain in the stationary distribution, so that the respective probabilities $\theta_0 \sim \pi(\theta)$ of starting at position are "smeared" across all states i. Then, the quantity $\pi(\theta = i) q(i,j)$ represents the "amount" of probability that flows down edge $i \to j$ in one time step. If detailed balance holds, then the amount of probability flowing from $i \to j$ equals the amount that flows from $j \to i$ (which is $\pi(\theta = j) q(j,i)$). Therefore, there is no *net* flux of probability along the edge $i \leftrightarrow j$ during one time step (remember this holds only for when the chain is in the stationary distribution).

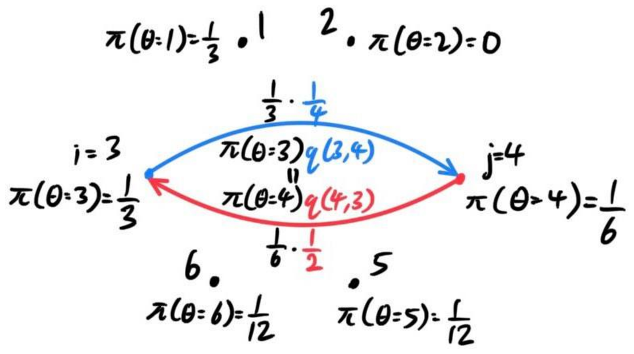

$\pi(\theta=1)=\frac{1}{3}$
1
2
$\pi(\theta=2)=0$
$\frac{1}{3}\cdot\frac{1}{4}$
$i=3$
$\pi(\theta=3)q(3,4)$
$=$
$\pi(\theta=4)q(4,3)$
$j=4$
$\pi(\theta=3)=\frac{1}{3}$
$\pi(\theta=4)=\frac{1}{6}$
$\frac{1}{6}\cdot\frac{1}{2}$
6
5
$\pi(\theta=6)=\frac{1}{12}$
$\pi(\theta=5)=\frac{1}{12}$

Figure 1: Visualization of detailed balance in Markov chain

1.3 Metropolis-Hastings: Example

Suppose we want a Markov chain of state space $\mathcal{S} = \{1, 2\}$ with the steady state distribution

$$\pi = \begin{pmatrix} \frac{3}{4} & \frac{1}{4} \end{pmatrix} \iff \pi(\theta = 1) = \frac{3}{4},\ \pi(\theta = 2) = \frac{1}{4} \tag{15}$$

To implement the Metropolis-Hastings algorithm, we calculate the proposal matrix and acceptance matrix

$$Q_{prop} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \text{ and } A = \begin{pmatrix} 1 & \frac{1}{3} \\ 1 & 1 \end{pmatrix} \tag{16}$$

which is calculated since $\alpha(1,2) = \min\left(1, \frac{1/4}{3/4}\right) = 1/3$ and $\alpha(2,1) = \min\left(1, \frac{3/4}{1/4}\right) = 1$. We multiply the nondiagonal entries together and fill in the diagonals to get

$$\begin{pmatrix} & \frac{1}{2} \cdot \frac{1}{3} \\ \frac{1}{2} \cdot 1 & \end{pmatrix} \implies Q = \begin{pmatrix} \frac{5}{6} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \tag{17}$$

Which can be visualized as an object jumping between two nodes with the following transitions.

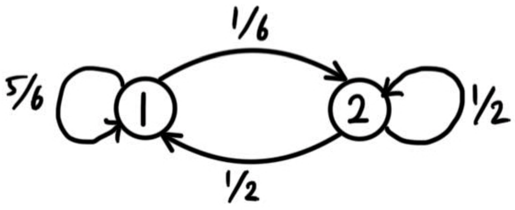


Figure 2: Two-state Markov chain with transition probabilities

2 Gibbs Sampling

Gibbs Sampling is a special case of the Metropolis-Hastings in which the newly proposed state is accepted with probability one. With observed data x, say that we have calculated the D-dimensional posterior

$$p(\theta \mid x) \propto f(\theta) = p(\theta)\, p(x \mid \theta) \tag{18}$$

where the parameter $\theta = (\theta^1, \ldots, \theta^D)$ is an element of the D-dimensional state space $\mathcal{S} = \{1, \ldots, n\}^D$ (actually, each θ^i does not need to be derived from the same $\{1, \ldots, n\}$ and we can generalize this algorithm to account for this). Remember that:

- It is hard to calculate $p(\theta \mid x) = p(\theta^1, \ldots, \theta^D \mid x)$ because calculating the constant c that normalizes $f(\theta)$ is hard (since D may be large). This makes it difficult to sample from the posterior.
- It is easy to calculate $f(\theta) = f(\theta^1, \ldots, \theta^D)$. We just don't know how to scale the individual values appropriately and so this function is useless in of itself, even though it is directly proportional to $p(\theta \mid x)$.

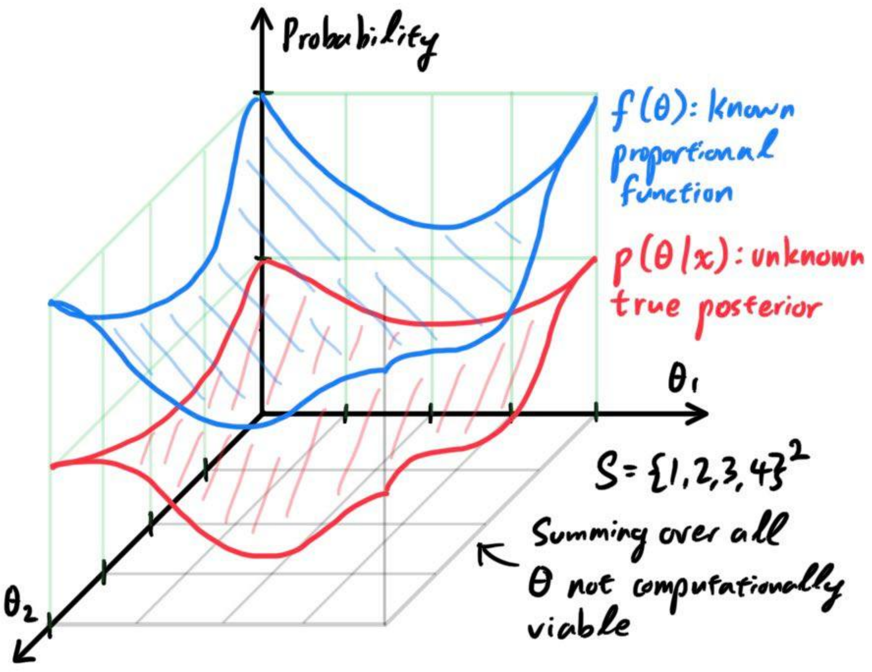

Probability
f(θ): known proportional function
p(θ|x): unknown true posterior
θ1
θ2
S={1,2,3,4}^2
Summing over all θ not computationally viable

Figure 3: Comparison of unknown posterior vs known proportional function

With the D-dimensional state space $\mathcal{S}$, we construct the true transition matrix. Say that the ith state of the chain is located at node θ_i with given coordinates

$$\theta_i = (\theta_i^1, \theta_i^2, \ldots, \theta_i^D) \tag{19}$$

The step to transition from this given θ_i to the next θ_{i+1} consists of two parts:

1. Pick a component index $j = d \in \{1, 2, \ldots, D\}$ uniformly at random. Many algorithms also pick $d = 1$ for the first step, $d = 2$ for the second, and so on.

$$p(\text{Index } d \text{ chosen}) = \frac{1}{D} \tag{20}$$

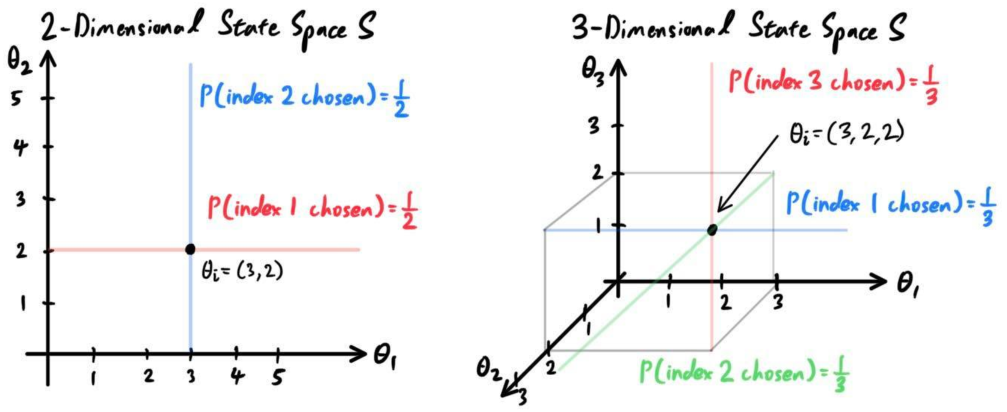


Figure 4: Choosing a component index in Gibbs sampling

2. With this well-defined d, we would like to update the Markov chain from state θ_i to θ_{i+1} by updating only the dth component of θ_i, and keeping every component fixed. When θ_i^d is updated, the new θ_{i+1}^d must take some value of $k^* \in \{1, \ldots, n\}$. As expected, it chooses which value k^* to update to according to the marginal distribution of $p(\theta \,|\, x)$ given $\theta_i^1, \ldots, \theta_i^{d-1}, \theta_i^{d+1}, \ldots, \theta_i^D$.

$$\begin{aligned} p(\theta_i^d \mapsto \theta_{i+1}^d = k^* \,|\, \text{Index } d \text{ chosen}) &= p(\theta_{i+1}^d = k^* \,|\, \theta_i^1, \ldots, \theta_i^{d-1}, \theta_i^{d+1}, \ldots, \theta_i^D, x) \\ &= \frac{p(\theta_i^1, \ldots, \theta_i^{d-1}, k^*, \theta_i^{d+1}, \ldots, \theta_i^D \,|\, x)}{\sum_{k=1}^n p(\theta_i^1, \ldots, \theta_i^{d-1}, k, \theta_i^{d+1}, \ldots, \theta_i^D \,|\, x)} \\ &= \frac{f(\theta_i^1, \ldots, \theta_i^{d-1}, k^*, \theta_i^{d+1}, \ldots, \theta_i^D)}{\sum_{k=1}^n f(\theta_i^1, \ldots, \theta_i^{d-1}, k, \theta_i^{d+1}, \ldots, \theta_i^D)} \end{aligned}$$

where the last step is justified by the proportionality of f and p. It turns out that the probability of where θ_{i+1}^d will land on does not actually depend on where θ_i^d is currently.

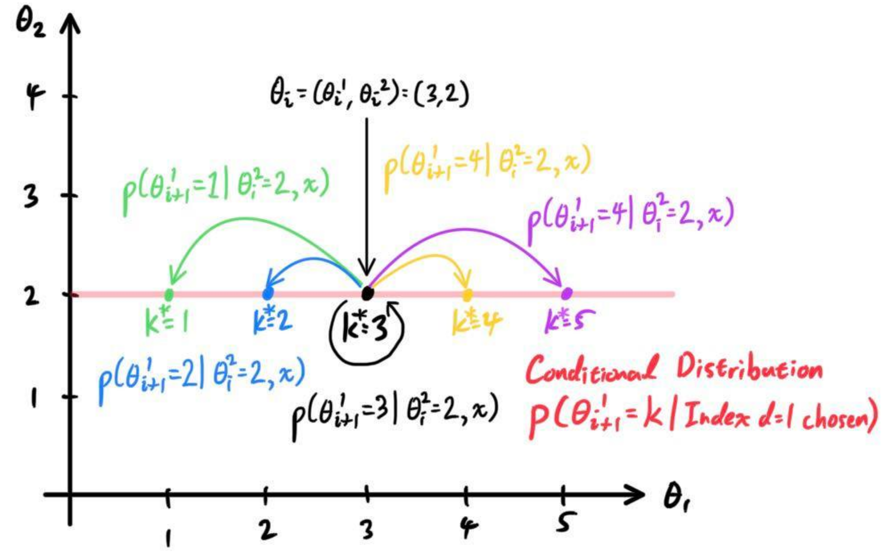
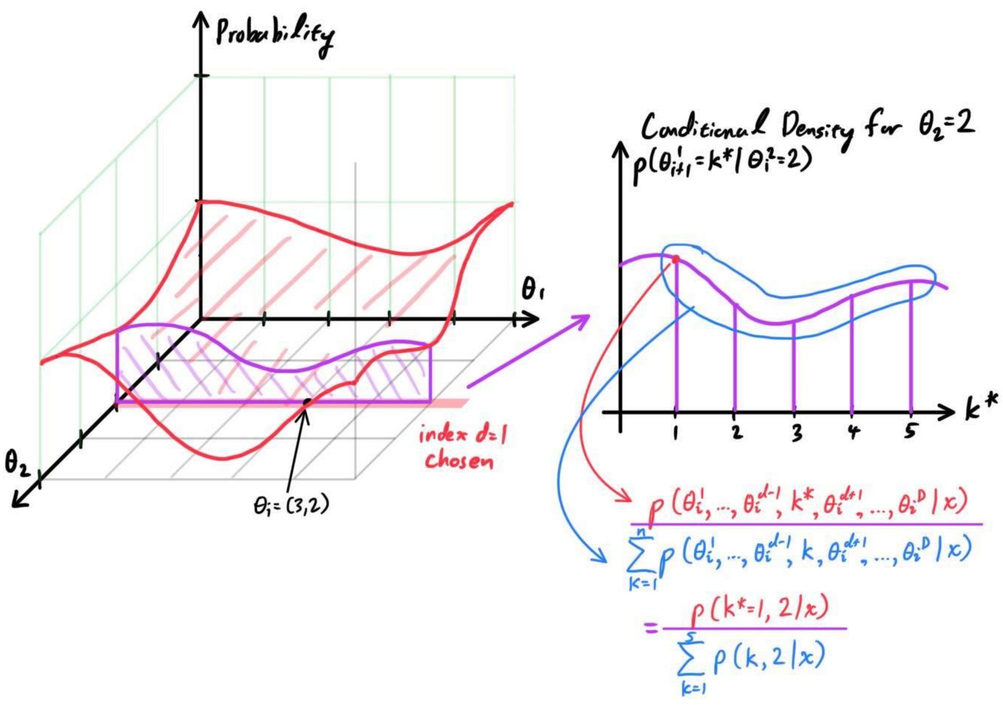

θi = (θi¹, θi²) = (3,2)
p(θ¹i+1 = 1 | θ²i = 2, x)
p(θ¹i+1 = 4 | θ²i = 2, x)
p(θ¹i+1 = 4 | θ²i = 2, x)
k* = 1
k* = 2
k* = 3
k* = 4
k* = 5
p(θ¹i+1 = 2 | θ²i = 2, x)
p(θ¹i+1 = 3 | θ²i = 2, x)
Conditional Distribution
p(θ¹i+1 = k | Index d=1 chosen)

Figure 5: Example for $D = 2, n = 5$ showing possible states (within red line) that θ_{i+1} can transition to

Do not be daunted by the notation. Just remember that $p(\theta_{i+1}^d = k^* \,|\, \theta_i^1, \ldots, \theta_i^{d-1}, \theta_i^{d+1}, \ldots, \theta_i^D, x)$ is just the conditional probability of $p(\theta \,|\, x)$ given that every $\theta_i^j, j \neq d$ are constant. This is easily visualized by taking the 1-dimensional cross section of the density $p(\theta \,|\, x)$ defined on $\mathcal{S}$.



Probability
θ_1
θ_2
index d=1 chosen
$\theta_i = (3,2)$
Conditional Density for $\theta_2 = 2$
$p(\theta_{i+1}^1 = k^* \mid \theta_i^2 = 2)$
k^*
1
2
3
4
5
$\dfrac{p(\theta_i^1, \ldots, \theta_i^{d-1}, k^*, \theta_i^{d+1}, \ldots, \theta_i^D \mid x)}{\sum_{k=1}^n p(\theta_i^1, \ldots, \theta_i^{d-1}, k, \theta_i^{d+1}, \ldots, \theta_i^D \mid x)}$
$= \dfrac{p(k^*=1, 2 \mid x)}{\sum_{k=1}^5 p(k, 2 \mid x)}$

Figure 6: Cross-sectional view of density in Gibbs sampling

Therefore, we can construct a Markov chain with the following transition probabilities. Given two states $\theta_r, \theta_s \in \mathcal{S}$, if θ_s differs in θ_r in at most one component, call it the dth component (i.e. $\theta_r^j = \theta_s^j$ for all $j \neq d$), then the probability of transition from θ_r to θ_s is

$$\begin{aligned} p(\theta_r, \theta_s) &= p(\theta_r^d \mapsto \theta_s^d \mid \text{Index } d \text{ chosen})\, p(\text{Index } d \text{ chosen}) \\ &= \frac{f(\theta_r^1, \ldots, \theta_r^{d-1}, \theta_s^d, \theta_r^{d+1}, \ldots, \theta_r^D)}{\sum_{k=1}^n f(\theta_r^1, \ldots, \theta_r^{d-1}, k, \theta_r^{d+1}, \ldots, \theta_r^D)} \cdot \frac{1}{D} \end{aligned}$$

Therefore, given that the chain is in state $\theta_i = (3,2)$ in state space $\mathcal{S} = \{1,2,3,4,5\}^2$, it may be able to get to the point in red or blue, depending on which index d was chosen. But it is impossible to go to any of the yellow states, so the transition probabilities are all 0.

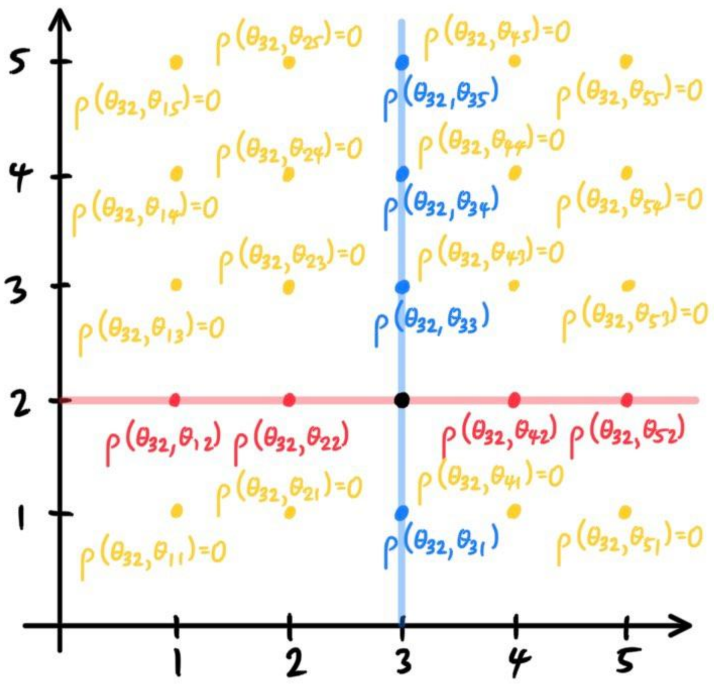

$p(\theta_{32},\theta_{15})=0$
$p(\theta_{32},\theta_{25})=0$
$p(\theta_{32},\theta_{35})$
$p(\theta_{32},\theta_{45})=0$
$p(\theta_{32},\theta_{55})=0$
$p(\theta_{32},\theta_{14})=0$
$p(\theta_{32},\theta_{24})=0$
$p(\theta_{32},\theta_{34})$
$p(\theta_{32},\theta_{44})=0$
$p(\theta_{32},\theta_{54})=0$
$p(\theta_{32},\theta_{13})=0$
$p(\theta_{32},\theta_{23})=0$
$p(\theta_{32},\theta_{33})$
$p(\theta_{32},\theta_{43})=0$
$p(\theta_{32},\theta_{53})=0$
$p(\theta_{32},\theta_{12})$
$p(\theta_{32},\theta_{22})$
$p(\theta_{32},\theta_{42})$
$p(\theta_{32},\theta_{52})$
$p(\theta_{32},\theta_{11})=0$
$p(\theta_{32},\theta_{21})=0$
$p(\theta_{32},\theta_{31})$
$p(\theta_{32},\theta_{41})=0$
$p(\theta_{32},\theta_{51})=0$
1
2
3
4
5

Figure 7: Possible transitions from state $(3, 2)$ in Gibbs sampling

With this, we can calculate the stationary distribution by either:

- Calculating the left-eigenvector of the transition matrix defined $p(\theta_r, \theta_s)$ with eigenvalue 1.
- Randomly initialize $\theta_0 = (\theta_0^1, \ldots, \theta_0^D)$ and run the chain for sufficiently long time to find out the proportion of steps in which a Markov chain lands on each $\theta \in \mathcal{S}$.

Now, it is easy to see why Gibbs sampling is a special case of Metropolis-Hastings. The Gibbs transition algorithm that we just mentioned is clearly a Markov chain, and within the context of Metropolis, we can interpret it as the proposal transition matrix with acceptance probability 1. By the same justification for Metropolis, we can prove that the stationary distribution of Gibbs sampling is $p(\theta \mid x)$.

3 Phase Spaces and Phase Flows

An ordinary differential equation describes the evolutional trajectory of some system as a function of time. Numerical methods are used to approximate this trajectory within a certain polynomial error bound. Given the (possibly vectored-valued) differential equation

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}) \tag{21}$$

with initial value $\mathbf{z}(0) = \boldsymbol{\zeta}$, the exact solution would be a function $\mathbf{z} : \mathbb{R} \longrightarrow \mathcal{M}$ from time to some phase manifold (s.t. $\mathbf{z}(0) = \zeta$). We can write this solution as a flow map Φ

$$\Phi_t(\boldsymbol{\zeta}) = \mathbf{z}(t, \zeta) \tag{22}$$

Numerically solving this differential equation relies on the idea of a discretization with a finite stepsize h, and an iterative procedure $\hat{\Phi}$ that computes, starting from $\mathbf{z}_0 = \boldsymbol{\zeta}$, a sequence $\mathbf{z}_1, \mathbf{z}_2, \ldots$, where $\mathbf{z}_n \approx \mathbf{z}(nh)$. The simplest scheme is simply Euler's method which advances the solution from timestep to timestep by the formula

$$\mathbf{z}_{n+1} = \mathbf{z}_n + h\mathbf{f}(\mathbf{z}_n) \tag{23}$$

which is based on the observation that $\mathbf{z}(t+h) \approx \mathbf{z}(t) + h\,\dot{\boldsymbol{z}}(t)$, i.e. the beginning of a Taylor series expansion in powers of h, and the further observation that the solution satisfies the differential equation, hence $\dot{\boldsymbol{z}}(t)$ may be replaced by $\mathbf{f}(\mathbf{z}(t))$. The discretized, approximate flow map will be denoted $\hat{\Phi} : \mathbf{z}_n \mapsto \mathbf{z}_{n+1}$.

Let us talk about the convergence of these schemes. The **order of accuracy** is the exponent in the power law by which the local error in the method is related to the stepsize. For example, when we say that a scheme is of third order, we mean that the local error (on a fixed finite-time interval) can be bounded by Kh^3, where h is a sufficiently small timestep and K is a number which depends on the length of the time interval and the features of the problem, but which is independent of h. It is known that the Euler method is a first order method. That is, let the length of the time interval be τ, with the step size h and number of steps ν. Then, $\nu h = \tau$, and defining the error at step n to be $e_n := ||\mathbf{z}_n - \mathbf{z}(t_n)||$, where $t_n = nh$, the maximum error of the Euler method at step ν satisfies

$$\bar{e} := \max_{0<n\leq\nu} e_n \leq C(\tau)h \tag{24}$$

Using the order notation, we have $\bar{e} = \mathcal{O}(h)$. Summing this over an interval gives a 0th order global error. For discretizations, local errors of order n correspond to global errors of order $n+1$. In simulations, it is apparent that the errors are larger at the end of the interval than at earlier times. We know that molecular dynamics trajectories need to be very long compared to the time-step used in order for them to be useful, so how the error grows in long simulations is quite important.

3.1 Higher Order Methods

One approach to higher accuracy is to to decrease the step size, but a more efficient way is to use a higher order method, which must satisfy a global error estimate (for finite time intervals) of the form

$$\bar{e} \approx C(\tau)\, h^r \tag{25}$$

For example, the Taylor series expansion of the solution may be written

$$\mathbf{z}(t+h) = \mathbf{z}(t) + h\dot{\boldsymbol{z}}(t) + \frac{1}{2}h^2\ddot{\boldsymbol{z}}(t) + \ldots \tag{26}$$

and while truncating after the first term leads to Euler's method, truncating after the second order term leads to

$$\mathbf{z}_{n+1} = \mathbf{z}_n + h\dot{\boldsymbol{z}}_n + \frac{1}{2}h^2\ddot{\boldsymbol{z}}_n \tag{27}$$

In this formula, the second derivative is obtained by differentiating the differential equation itself:

$$\ddot{\boldsymbol{z}}(t) = \frac{d}{dt}\dot{\boldsymbol{z}}(t) = \frac{d}{dt}\mathbf{f}'\big(\mathbf{z}(t)\big) = \mathbf{f}'\big(\mathbf{z}(t)\big)\,\mathbf{f}\big(\mathbf{z}(t)\big) \tag{28}$$

So we can write the Taylor series method as below, which describes the flow map approximation:

$$\hat{\Phi}(\mathbf{z}_n) = \mathbf{z}_{n+1} = \mathbf{z}_n + h\,\mathbf{f}(\mathbf{z}_n) + \frac{1}{2}h^2\mathbf{f}'(\mathbf{z}_n)\,\mathbf{f}(\mathbf{z}_n) \tag{29}$$

3.2 Convergence and the Order of Accuracy

A typical integrator computes successive steps (of stepsize h) from the formulas

$$\mathbf{z}_{n+1} = \hat{\Phi}_h(\mathbf{z}_n), \qquad \mathbf{z}_0 = \boldsymbol{\zeta} \tag{30}$$

Assume that $\hat{\Phi}_h$ is a smooth map for all $h > 0$. The exact solution $\mathbf{z}$ satisfies

$$\mathbf{z}(t_{n+1}) = \Phi_h(\mathbf{z}(t_n)) \tag{31}$$

since Φ_h simply takes point $\mathbf{z}(t_n)$ and flows it for time period h. Therefore to each $h > 0$ we can associate a finite set of space points $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_\nu$, which represents the numerical solutions at $t_0 = 0, t_1 = h, t_2 = 2h, \ldots, t_\nu = \nu h = \tau$. Taking the difference of the numerical and exact solutions, we have

$$\mathbf{z}_{n+1} - \mathbf{z}(t_{n+1}) = \hat{\Phi}(\mathbf{z}_n) - \Phi_h\big(\mathbf{z}(t_n)\big) \tag{2.1}$$

We first assume that $\hat{\Phi}_h$ is an $\mathcal{O}(h^{p+1})$ approximation of Φ in the sense that there is a constant $K \geq 0$ and a constant $\Delta > 0$ such that, for $t \in [0, \tau]$, we have

$$||\Phi_t\big(\mathbf{z}(t)\big) - \hat{\Phi}_h\big(\mathbf{z}(t)\big)|| \leq K h^{p+1}, \qquad h < \Delta \tag{32}$$

This assumption is usually verified by expanding the numerical and exact solutions in powers of h, using Taylor series expansions. To tackle the question of growth of error, we make an additional assumption on $\hat{\Phi}_h$, namely that is satisfies a **Lipshitz condition** of the form

$$||\hat{\Phi}_h(\mathbf{u}) - \hat{\Phi}_h(\mathbf{w})|| \leq (1 + hL)||\mathbf{u} - \mathbf{w}|| \tag{33}$$

for all $\mathbf{u}, \mathbf{w} \in D$, where D is some subdomain of $\mathbb{R}^{6N}$ that contains the exact solution for $[0, \tau]$, and $h \leq \Delta$. This stability assumption, in words, says that the method does not increase the separation between two nearby trajectories by more than a factor of the form $1 + hL$. Therefore, from (2.1), we can write

$$\mathbf{z}_{n+1} - \mathbf{z}(t_{n+1}) = \hat{\Phi}(\mathbf{z}_n) - \hat{\Phi}_h\big(\mathbf{z}(t_n)\big) + \hat{\Phi}_h\big(\mathbf{z}(t_n)\big) - \Phi_h\big(\mathbf{z}(t_n)\big) \tag{34}$$

and take norms with the triangle inequality to get the following, where $\varepsilon_n = ||\mathbf{z}_n - \mathbf{z}(t_n)||$.

$$\varepsilon_{n+1} \leq (1 + Lh)\varepsilon_n + \bar{K}h^{p+1} \tag{35}$$

and from this, we can calculate the bound

$$\varepsilon_n \leq \frac{\bar{K}}{L}e^{Lnh}h^p \tag{36}$$

4 Hamiltonian Dynamics Inspired Samplers and Integrators

Let us have a system of N point particles in $\mathbb{R}^3$, with the state of each particle fully characterized by its position and momentum vectors. Let us denote the masses of the particles as m_i, which will be commonly represented as the $3N \times 3N$ matrix

$$\mathbf{M} = \operatorname{diag}(m_1, \ldots, m_N) \otimes I_3 = \begin{pmatrix} m_1 &&&&&&&& \\ & m_1 &&&&&&& \\ && m_1 &&&&&& \\ &&& m_2 &&&&& \\ &&&& \ddots &&&& \\ &&&&& m_{N-1} &&& \\ &&&&&& m_N && \\ &&&&&&& m_N & \\ &&&&&&&& m_N \end{pmatrix}, \tag{37}$$

the position vector of all particles as $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_N) \in \Omega_{\mathbf{q}} \subset \mathbb{R}^{3N}$, and the momentum vector of all particles as $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_N) \in \Omega_{\mathbf{p}} \subset \mathbb{R}^{3N}$. The configuration space is therefore $\Omega_{\mathbf{q}} \times \Omega_{\mathbf{p}} = \Omega \subset \mathbb{R}^{3N} \times \mathbb{R}^{3N}$. The collective kinetic energy of the system is

$$E(\mathbf{p}) = \frac{1}{2} \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p} \tag{38}$$

and hence the total energy/Hamiltonian of the particle system is

$$H(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}) + E(\mathbf{p}) \tag{39}$$

Note that the potential energy depends only on the position vector $\mathbf{q}$, while the kinetic energy depends on the momentum $\mathbf{p}$. The equations of motion for Hamiltonian flow states that the derivative of the position is the momentum, and the derivative of the momentum is the force, which is the gradient of the potential. Therefore, finding the time evolution of a system of particles boils down to solving the coupled equations below:

$$\begin{aligned} \dot{\boldsymbol{q}} &= \mathbf{M}^{-1} \mathbf{p} \\ \dot{\boldsymbol{p}} &= \mathbf{F}(q) = -\nabla_{\mathbf{q}} U(\mathbf{q}) \end{aligned}$$

The gradient of $H : \Omega_{\mathbf{q}} \times \Omega_{\mathbf{p}} \longrightarrow \mathbb{R}$ can be represented as

$$\nabla H(\mathbf{q}, \mathbf{p}) = \begin{pmatrix} \nabla_{\mathbf{q}} H \\ \nabla_{\mathbf{p}} H \end{pmatrix} = \begin{pmatrix} \frac{\partial H}{\partial \mathbf{q}} \\ \frac{\partial H}{\partial \mathbf{p}} \end{pmatrix} = \begin{pmatrix} \partial H / \partial q_1 \\ \vdots \\ \partial H / \partial q_{3N} \\ \partial H / \partial p_1 \\ \vdots \\ \partial H / \partial p_{3N} \end{pmatrix} \tag{40}$$

But since $H(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}) + E(\mathbf{p})$ is separable and since

$$\begin{aligned} \nabla_{\mathbf{p}} E_{\text{kin}}(p) &= \nabla_{\mathbf{p}} \frac{1}{2} \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p} \\ &= \nabla_{\mathbf{p}} \frac{1}{2} (m_1^{-1} p_{11}^2 + m_1^{-1} p_{12}^2 + m_1^{-1} p_{13}^2 + m_2^{-1} p_{21}^2 + \ldots + m_N^{-1} p_{N3}^2) \\ &= \big(m_1^{-1} p_{11},\ m_1^{-1} p_{12},\ m_1^{-1} p_{13},\ m_2^{-1} p_{21}, \ldots, m_N^{-1} p_{N3} \big)^T \\ &= \mathbf{M}^{-1} \mathbf{p} \end{aligned}$$

we have

$$\nabla H(\mathbf{q}, \mathbf{p}) = \begin{pmatrix} \nabla_{\mathbf{q}} U(\mathbf{q}) \\ \nabla_{\mathbf{p}} E_{\text{kin}}(\mathbf{p}) \end{pmatrix} = \begin{pmatrix} \nabla_{\mathbf{q}} U(\mathbf{q}) \\ \mathbf{M}^{-1}\mathbf{p} \end{pmatrix} \tag{41}$$

and therefore, the equations of motions can be rewritten as

$$\begin{cases} \dot{\boldsymbol{q}} & = \mathbf{M}^{-1}\mathbf{p} \\ \dot{\boldsymbol{p}} & = -\nabla_{\mathbf{q}} U(\mathbf{q}) \end{cases} \implies \begin{pmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{I}_{3N} \\ -\mathbf{I}_{3N} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \nabla_{\mathbf{q}} U(\mathbf{q}) \\ \mathbf{M}^{-1}\mathbf{p} \end{pmatrix} = \mathbf{J}\nabla H(\mathbf{q}, \mathbf{p}) \tag{42}$$

Given an initial point $(\mathbf{q}(0), \mathbf{p}(0)) \in \Omega_{\mathbf{q}} \times \Omega_{\mathbf{p}}$, the Hamiltonian flow map satisfies

$$\Phi_t\big(\mathbf{q}(0), \mathbf{p}(0)\big) = \big(\mathbf{q}(t), \mathbf{p}(t)\big) \tag{43}$$

4.1 Properties of Hamiltonian Flow Maps

Hamiltonian flow maps $\Phi_t : \Omega \longrightarrow \Omega$ have important properties.

1. The collection of flow maps form an algebraic group under the composition operator

$$\Phi_t \circ \Phi_s = \Phi_{t+s} \tag{44}$$

with the identity element $\Phi_0 = \text{Id}$ (the path map that doesn't go anywhere), and well-defined inverse

$$\Phi_t^{-1} = \Phi_{-t} \tag{45}$$

2. Symmetry holds in the sense that

$$S \circ \Phi_t \circ S = \Phi_{-t} \tag{46}$$

where the function $S : (\mathbf{q}, \mathbf{p}) \mapsto (\mathbf{q}, -\mathbf{p})$ flips the momentum.

3. Total energy is conserved under Φ_t.

$$H\big(\mathbf{q}(t), \mathbf{p}(t)\big) = H\big(\mathbf{q}(0), \mathbf{p}(0)\big) \tag{47}$$

4. In the absence of an external force, the total momentum is conserved under Φ_t.

4.1.1 The Symplectic Property

The final property is less obvious. A fundamental property of solutions of Hamiltonian differential equations is that the collection $(\Phi_t)_{t\in\mathbb{R}}$ of associated flow maps has a symplectic group structure, which means that the symplectic 2-form is preserved under the action of each group element.

1. A **1-form** α defined on $\mathbb{R}^{6N}$ is a family of linear mappings such that for every $\mathbf{x} \in \mathbb{R}^{6N}$, $\alpha(\mathbf{x})$ is a linear map from $\mathbb{R}^{6N}$ to $\mathbb{R}$. That is, given a linear map $\mathbf{a} : \mathbb{R}^{6N} \longrightarrow \mathbb{R}^{6N}$, we may define a one-form associated to this vector field $\mathbf{a} \mapsto \alpha$ by

$$\alpha(\mathbf{x})(\boldsymbol{\xi}) = \mathbf{a}(\mathbf{x})^T \boldsymbol{\xi} \tag{48}$$

2. The **differential** of a function $g : \mathbb{R}^{6N} \longrightarrow \mathbb{R}$, denoted $\mathrm{d}g$, is a family of linear mappings from vectors $\boldsymbol{\xi} \in \mathbb{R}^{6N}$ into the reals defined by

$$\mathrm{d}g(\mathbf{q}, \mathbf{p})(\boldsymbol{\xi}) = \nabla g(\mathbf{q}, \mathbf{p})^T \boldsymbol{\xi} \tag{49}$$

Therefore, we can see that the differential is an example of a 1-form.

3. The wedge product of 1-forms α, β is a **2-form**, which can be viewed as a quadratic form, i.e. a scalar-valued function of two vectors which is linear in each argument. It is written $\alpha \wedge \beta$ and is defined, for vectors $\boldsymbol{\xi}, \boldsymbol{\eta} \in \mathbb{R}^{6N}$ by

$$(\alpha \wedge \beta)(\boldsymbol{\xi}, \boldsymbol{\eta}) \coloneqq \alpha(\boldsymbol{\xi})\beta(\boldsymbol{\eta}) - \alpha(\boldsymbol{\eta})\beta(\boldsymbol{\xi}) \tag{50}$$

Now, let $q_i, p_j : \mathbb{R}^{6N} \longrightarrow \mathbb{R}$ be the component functions mapping $(\mathbf{q}, \mathbf{p}) \mapsto q_i, p_j$, respectively, where $1 \leq i, j \leq 3N$. Then, $\mathrm{d}q_i, \mathrm{d}p_i$ are examples of differential 1-forms. The wedge product of the coordinate differentials $\mathrm{d}q_i, \mathrm{d}p_i$ can be written

$$(\mathrm{d}q_i \wedge \mathrm{d}p_i)(\boldsymbol{\xi}, \boldsymbol{\eta}) = \xi_i \eta_{i+3N} - \xi_{i+3N}\eta_i = \boldsymbol{\xi}^T \mathbf{J}^{(i)} \boldsymbol{\eta} \tag{51}$$

where $\mathbf{J}$ is the matrix which has zeros everywhere except for $(\mathbf{J}^{(i)})_{i,3N+i} = 1, (\mathbf{J}^{(i)})_{i+3N,i} = -1$. Summing these terms results in the symplectic 2-form, denoted ψ_S:

$$\psi_S \coloneqq \sum_{i=1}^{3N} \mathrm{d}q_i \wedge \mathrm{d}p_i(\boldsymbol{\xi}, \boldsymbol{\eta}) = \boldsymbol{\xi}^T \bigg(\sum_{i=1}^{3N} \mathbf{J}^{(i)} \bigg) \boldsymbol{\eta} = \boldsymbol{\xi}^T \mathbf{J} \boldsymbol{\eta} \tag{52}$$

That is, since $\Phi_t : \mathbb{R}^{6N} \longrightarrow \mathbb{R}^{6N}$, then its Jacobian $\nabla \Phi_t$ is a $6N \times 6N$ matrix, and the condition above implies that

$$\nabla \Phi_t^T \mathbf{J} \nabla \Phi_t = \mathbf{J} \text{ for all } t \in \mathbb{R} \tag{53}$$

Denoting the Jacobian $\nabla \Phi_t$ as Φ_t', we can take the determinant of both sides to find that

$$\det \big(\nabla \Phi_t^T \mathbf{J} \nabla \Phi_t \big) = \det(\mathbf{J}) \implies \det(\nabla \Phi_t^T) \det(\mathbf{J}) \det(\nabla \Phi_t) = \det(\mathbf{J}) \tag{54}$$

and so $\det {\Phi_t'}^2 = 1$. In the case of a flow map, we know that for $t \to 0$, the flow map Φ_t would essentially reduce to the identity map Id, and so

$$\lim_{t \to 0} \Phi_t' = 1 \implies \det \Phi_t' = +1 \tag{55}$$

due to the determinant being a continuous function of t. Therefore a consequence of this is that Φ_t is volume preserving.

4.2 Common Symplectic Integrators

We wish to solve for Φ_t numerically by constructing an approximation with acceptable error.

$$(\hat{\mathbf{q}}_{n+1}, \hat{\mathbf{p}}_{n+1}) = \hat{\Phi}_h(\hat{\mathbf{q}}_n, \hat{\mathbf{p}}_n) \tag{56}$$

with $(\hat{\mathbf{q}_0}, \hat{\mathbf{p}_0}) = \big(\mathbf{q}(0), \mathbf{p}(0)\big)$. There is the obvious error stemming from the choice of a large h, but what is more important is that the geometric structure of the manifold $\big(\mathbf{q}(t), \mathbf{p}(t)\big)_{t>0}$ corresponding to the trajectory of the exact solution is replicated by the discrete approximation $\big(\hat{\mathbf{q}}_n, \hat{\mathbf{p}}_n\big)_{n \in \mathbb{N}}$. The best way to do this is to construct such a structure preserving integration scheme by designing the integration map $\hat{\Phi}_h$ in such a way that the symplectic 2-form is preserved. That is, construct a **symplectic integration scheme** $\hat{\Phi}_h$ such that

$$(\hat{\Phi}_h')^T \mathbf{J} \, \hat{\Phi}_h' = \mathbf{J} \tag{57}$$

4.2.1 Euler and Symplectic Euler

The standard Euler integration scheme has many shortfalls, such as error growth and stability issues. Its algorithmic form reads

$$\begin{aligned} \mathbf{q}_{k+1} &= \mathbf{q}_k + h \, \mathbf{M}^{-1} \mathbf{p}_k \\ \mathbf{p}_{k+1} &= \mathbf{p}_k - h \, \nabla U(\mathbf{q}_k) \end{aligned}$$

Therefore, modified version of this scheme, called the **symplectic Euler integration scheme**, is used, which reads

$$\begin{aligned} \mathbf{p}_{k+1} &= \mathbf{p}_k - h \, \nabla U(\mathbf{q}_k) \\ \mathbf{q}_{k+1} &= \mathbf{q}_k + h \mathbf{M}^{-1} \mathbf{p}_{k+1} \end{aligned}$$

which has the slight modification that to advance the timestep, we use the first equation to compute $\mathbf{p}_{k+1}$ and then insert this in the second.

4.2.2 Verlet

One of the most commonly used symplectic numerical integrators is the **Stormer-Verlet method**, which in algorithmic form reads

$$\begin{aligned}\mathbf{q}_{k+1/2} &= \mathbf{q}_k + \frac{h}{2}\mathbf{M}^{-1}\mathbf{p}_k \\ \mathbf{p}_{k+1} &= \mathbf{p}_k - h\,\nabla U(\mathbf{q}_{k+1/2}) \\ \mathbf{q}_{k+1} &= \mathbf{q}_{k+1/2} + \frac{h}{2}\mathbf{M}^{-1}\mathbf{q}_{k+1}\end{aligned}$$

We can see that this algorithm updates $\mathbf{q}_k \mapsto \mathbf{q}_{k+1/2}$ with the given $\mathbf{p}_k$ over half-time step, and then updates the force field vector with the new position vector $-\nabla U(\mathbf{q}_{k+1/2})$. This new force is used to update the momentum $\mathbf{p}_k \mapsto \mathbf{p}_{k+1}$. Finally, the position is updated with the new momentum: $\mathbf{q}_{k+1/2} \mapsto \mathbf{q}_{k+1}$. A closely related, alternative form is the **Velocity-Verlet method**, which updates the momentum first, then position, and finally momentum.

$$\begin{aligned}\mathbf{p}_{k+1/2} &= \mathbf{p}_k - \frac{h}{2}\nabla U(\mathbf{q}_k) \\ \mathbf{q}_{k+1} &= \mathbf{q}_k + h\,\mathbf{M}^{-1}\mathbf{p}_{k+1/2} \\ \mathbf{p}_{k+1} &= \mathbf{p}_{k+1/2} - \frac{h}{2}\nabla U(\mathbf{q}_{k+1})\end{aligned}$$

The Velocity Verlet method is 2nd order (globally). While the algorithm may not look like a second order, we can see that with simple substitution, we have a second order evaluation of $\mathbf{q}_{k+1}$ (up to h^2 term) followed by an evaluation of $\mathbf{p}_{k+1}$.

$$\begin{aligned}\mathbf{q}_{k+1} &= \mathbf{q}_k + h\mathbf{M}^{-1}\mathbf{p}_{k+1/2} \\ &= \mathbf{q}_k + h\mathbf{M}^{-1}\Big(\mathbf{p}_k - \frac{h}{2}\nabla U(\mathbf{q}_k)\Big) \\ &= \mathbf{q}_k + h\mathbf{M}^{-1}\mathbf{p}_k - \frac{1}{2}h^2\nabla U(\mathbf{q}_k) \\ \mathbf{p}_{k+1} &= \mathbf{p}_{k+1/2} - \frac{h}{2}\nabla U(\mathbf{q}_{k+1}) \\ &= \Big(\mathbf{p}_k - \frac{h}{2}\nabla U(\mathbf{q}_k)\Big) - \frac{h}{2}\nabla U(\mathbf{q}_{k+1}) \\ &= \mathbf{p}_k - \frac{h}{2}\big[\nabla U(\mathbf{q}_k) + \nabla U(\mathbf{q}_{k+1})\big]\end{aligned}$$

Setting $\mathbf{M} = I$, $\mathbf{F} = -\nabla_{\mathbf{q}} U$ and expanding the $\mathbf{q}_{k+1}$ in the inner term, we get

$$\begin{aligned}\mathbf{q}_{k+1} &= \mathbf{q}_k + h\mathbf{p}_k + \frac{1}{2}h^2\mathbf{F}(\mathbf{q}_k) \\ \mathbf{p}_{k+1} &= \mathbf{p}_k + \frac{h}{2}\Big[\mathbf{F}(\mathbf{q}_k) + \mathbf{F}\Big(\mathbf{q}_k + h\mathbf{p}_k + \frac{1}{2}h^2\mathbf{F}(\mathbf{q}_k)\Big)\Big]\end{aligned}$$

The first equation is already a polynomial, i.e. it is in the form of a series expansion in powers of h where the coefficients are functions of the starting point $(\mathbf{q}_k, \mathbf{p}_k)$. The second equation may be written as a series expansion in powers of h as well.

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \frac{h}{2}\mathbf{F}(\mathbf{q}_k) + \frac{h}{2}\Big[\mathbf{F}(\mathbf{q}_k) + h\mathbf{F}'(\mathbf{q}_k)\big(\mathbf{p}_k + \frac{h}{2}\mathbf{F}(\mathbf{q}_k)\big) + \frac{h^2}{2}\mathbf{F}''(\mathbf{q}_k)\big(\mathbf{p}_k + \frac{h}{2}\mathbf{F}(\mathbf{q}_k)\big)^2 + \ldots\Big]$$

which we will neglect terms involving 4th and higher powers of h. Combining terms of like powers of h, we have

$$\mathbf{p}_{k+1} = \mathbf{p}_k + h\mathbf{F}(\mathbf{q}_k) + \frac{h^2}{2}\mathbf{p}_k\mathbf{F}'(\mathbf{q}_k) + \frac{h^3}{4}\big[\mathbf{F}'(\mathbf{q}_k)\mathbf{F}(\mathbf{q}_k) + \mathbf{p}_k^2\mathbf{F}''(\mathbf{q}_k)\big] + \mathcal{O}(h^4) \tag{58}$$

We compare this against the Taylor expansion of the exact solution $\mathbf{z}(t) \coloneqq \big(\mathbf{q}(t), \mathbf{p}(t)\big)$. We evaluate

$$\begin{aligned}\mathbf{q}(t+h) &= \mathbf{q}(t) + h\mathbf{q}'(t) + \frac{h^2}{2}\mathbf{q}''(t) + \frac{h^3}{6}\mathbf{q}'''(t) + \mathcal{O}(h^4) \\ &= \mathbf{q}(t) + h\mathbf{p}(t) + \frac{h^2}{2}\mathbf{F}(\mathbf{q}(t)) + \frac{h^3}{6}\mathbf{F}'(\mathbf{q}(t))\,\mathbf{p}(t) + \mathcal{O}(h^4)\end{aligned}$$

where the calculations followed from the fact that $\mathbf{q}' = \mathbf{p}$, which means that $\mathbf{q}'' = \mathbf{p}' = \mathbf{F}(\mathbf{q})$, which means that $\mathbf{q}''' = \frac{d}{dt}\mathbf{F}(\mathbf{q}) = \mathbf{F}'(\mathbf{q})\,\mathbf{q}' = \mathbf{F}'(\mathbf{q})\,\mathbf{p}$. Then, we have

$$\begin{aligned}\mathbf{p}(t+h) &= \mathbf{p}(t) + h\mathbf{p}'(t) + \frac{h^2}{2}\mathbf{p}''(t) + \frac{h^3}{6}\mathbf{p}'''(t) + \mathcal{O}(h^4) \\ &= \mathbf{p}(t) + h\mathbf{F}(\mathbf{q}(t)) + \frac{h^2}{2}\mathbf{p}(t)\,\mathbf{F}'(\mathbf{q}(t)) + \frac{h^3}{6}\big[\mathbf{p}(t)^2\,\mathbf{F}''(\mathbf{q}(t)) + \mathbf{F}'(\mathbf{q}(t))\,\mathbf{F}(\mathbf{q}(t))\big] + \mathcal{O}(h^4)\end{aligned}$$

which follows from the fact that $\mathbf{p}''' = (\mathbf{p}\,\mathbf{F}')' = \mathbf{p}^2\,\mathbf{F}'' + \mathbf{F}'\mathbf{F}$.
Now, let's compare them. Let us have initial point $\mathbf{z}_k = \mathbf{z}(t) = \big(\mathbf{q}(t), \mathbf{p}(t)\big) = \big(\mathbf{q}_k, \mathbf{p}_k\big)$ at time t. The actual flow and the integrator takes in $\mathbf{z}(t)$ and $\mathbf{z}_k$, respectively, but they are the same initial point, so we will label them with $\mathbf{z} = (\mathbf{q}, \mathbf{p})$. We can use the flow map $\Phi_h\big(\mathbf{z}(t)\big)$ to evaluate the exact position $\mathbf{z}(t+h)$ after time h. That is, $\Phi_h\big(\mathbf{z}(t)\big) \coloneqq \mathbf{z}\big(h, \mathbf{z}(t)\big) = \mathbf{z}(t+h)$. The Taylor expansion of this flow map is

$$\Phi_h\big(\mathbf{z}(t)\big) = \mathbf{z}(t+h) = \begin{cases}\mathbf{q}(t+h) = \mathbf{q} + h\mathbf{p} + \frac{h^2}{2}\mathbf{F} + \frac{h^3}{6}\mathbf{F}'\,\mathbf{p} + \mathcal{O}(h^4) \\ \mathbf{p}(t+h) = \mathbf{p} + h\mathbf{F} + \frac{h^2}{2}\mathbf{p}\,\mathbf{F}' + \frac{h^3}{6}\big[\mathbf{p}^2\,\mathbf{F}'' + \mathbf{F}'\,\mathbf{F}\big] + \mathcal{O}(h^4)\end{cases} \tag{59}$$

The numerical integrator would calculate something slightly different. That is, given the initial point $\mathbf{z}(t) = \mathbf{z}_k$, $\hat{\Phi}_h\big(\mathbf{z}(t)\big) = \mathbf{z}_{k+1}$ is the numerical approximation after time h. The Taylor expansion of this integrator is

$$\hat{\Phi}_h\big(\mathbf{z}(t)\big) = \mathbf{z}_{k+1} = \begin{cases}\mathbf{q}_{k+1} = \mathbf{q} + h\mathbf{p} + \frac{h^2}{2}\mathbf{F} \\ \mathbf{p}_{k+1} = \mathbf{p} + h\mathbf{F} + \frac{h^2}{2}\mathbf{p}\mathbf{F}' + \frac{h^3}{4}\big[\mathbf{F}'\mathbf{F} + \mathbf{p}^2\mathbf{F}''\big] + \mathcal{O}(h^4)\end{cases} \tag{60}$$

We should get $\mathbf{z}_{k+1} \approx \mathbf{z}(t+h)$, by looking at the differences, we find that these differ in the third (and higher) order terms.

$$\begin{aligned}\mathbf{q}_{k+1} - \mathbf{q}(t+h) &= -\frac{h^3}{6}\mathbf{F}'\mathbf{p} + \mathcal{O}(h^4) \\ \mathbf{p}_{k+1} - \mathbf{p}(t+h) &= \frac{h^3}{12}\big[\mathbf{p}^2\mathbf{F}'' + \mathbf{F}'\mathbf{F}\big] + \mathcal{O}(h^4)\end{aligned}$$

We can, in summary, state that the *local* error is third order

$$||\hat{\Phi}_h(\mathbf{z}) - \Phi_h(\mathbf{z})|| = \kappa(\mathbf{z})h^3 + \mathcal{O}(h^4) \tag{61}$$

where $\kappa(\mathbf{z}) \coloneqq \kappa(\mathbf{q}, \mathbf{p})$ is a function of the position and momentum. We may then define the maximum local error as

$$\bar{K} \coloneqq \max_{t \in [0,\tau]} \kappa\big(\mathbf{z}(t)\big) \tag{62}$$

and summing this all up leads to the total error being bounded by a constant multiple of h^2, achieving consistency of order 2.

4.2.3 Yoshida 4th-Order

The Yoshida Fourth Order Scheme is overall three iterations of velocity Verlet (making it also a symplectic integrator), using stepsizes $\tau_0 h, \tau_1 h, \tau_0 h$, respectively. We write this with subindices α, β to indicate the intermediate stages, and abuse our notation to be similar to those in computer science.

$$\begin{aligned}
\mathbf{p}_\alpha &= \mathbf{p} - (\tau_0\, h/2)\nabla U(\mathbf{q}) \\
\mathbf{q}_\alpha &= \mathbf{q} + (\tau_0\, h)\mathbf{M}^{-1}\mathbf{p}_\alpha \\
\mathbf{p}_\alpha &= \mathbf{p}_\alpha - (\tau_0\, h/2)\nabla U(\mathbf{q}_\alpha) \\
\mathbf{p}_\beta &= \mathbf{p}_\alpha - (\tau_1\, h/2)\nabla U(\mathbf{q}_\alpha) \\
\mathbf{q}_\beta &= \mathbf{q}_\alpha + (\tau_1\, h)\mathbf{M}^{-1}\mathbf{p}_\beta \\
\mathbf{p}_\beta &= \mathbf{p}_\beta - (\tau_1\, h/2)\nabla U(\mathbf{q}_\beta) \\
\mathbf{p} &= \mathbf{p}_\beta - (\tau_0\, h/2)\nabla U(\mathbf{q}_\beta) \\
\mathbf{q} &= \mathbf{q}_\beta + (\tau_0\, h)\mathbf{M}^{-1}\mathbf{p} \\
\mathbf{p} &= \mathbf{p} - (\tau_0\, h/2)\nabla U(\mathbf{q})
\end{aligned}$$

The equations can be written in a simplified form, combining several of the steps. This scheme requires three new evaluations of the force ∇U per iteration, making it significantly more expensive than the vanilla second-order Verlet method. However, this method is of 4th order.

4.3 Adjoint Method

For the true flow map $\Phi_t : \Omega \longrightarrow \Omega$, we know that due to time-reversibility, the inverse map is the same as the same map with a backward timestep:

$$\Phi_t^{-1} = \Phi_{-t} \tag{63}$$

For a discretized integrator $\hat{\Phi}$, this may not always be the case (even though symplectic forms might be preserved). Therefore, we can define the **adjoint** of the numerical scheme to be

$$(\hat{\Phi}_h)^\dagger \coloneqq \hat{\Phi}_{-h}^{-1} \tag{64}$$

Clearly, the adjoint of the true flow map is the same as the original, i.e. Φ_t is self-adjoint. Furthermore, the adjoint of the adjoint of any flow map is the original flow map.

4.3.1 Adjoint of Euler's Method

Consider Euler's method $\hat{\Phi}_h$ in fully general form, with $\mathbf{z} = (\mathbf{q}, \mathbf{p})^T$. Then, we have

$$\begin{aligned}
\hat{\Phi}_h &: \mathbf{z}_k \mapsto \mathbf{z}_{k+1} \text{ such that } \mathbf{z}_{k+1} = \mathbf{z}_k + h\,\mathbf{f}(\mathbf{z}_k) \\
\hat{\Phi}_h^{-1} &: \mathbf{z}_k \mapsto \mathbf{z}_{k+1} \text{ such that } \mathbf{z}_k = \mathbf{z}_{k+1} + h\,\mathbf{f}(\mathbf{z}_{k+1}) \\
\hat{\Phi}_h^\dagger = \hat{\Phi}_{-h}^{-1} &: \mathbf{z}_k \mapsto \mathbf{z}_{k+1} \text{ such that } \mathbf{z}_k = \mathbf{z}_{k+1} - h\,\mathbf{f}(\mathbf{z}_{k+1}) \iff \mathbf{z}_{k+1} = \mathbf{z}_k + h\,\mathbf{f}(\mathbf{z}_{k+1})
\end{aligned}$$

and so and clearly, $\hat{\Phi}_{-h}^{-1}$ defines $\mathbf{z}_{k+1}$ implicitly.

4.3.2 Adjoint of Symplectic Euler's Method

To construct the adjoint method of the symplectic Euler scheme, we see that that $\hat{\Phi}_h^{-1}$ maps $(\mathbf{q}_k, \mathbf{p}_k) \mapsto (\mathbf{q}_{k+1}, \mathbf{p}_{k+1})$ such that

$$\begin{aligned}
\mathbf{p}_k &= \mathbf{p}_{k+1} - h\,\nabla U(\mathbf{q}_{k+1}) \\
\mathbf{q}_k &= \mathbf{q}_{k+1} + h\,\mathbf{M}^{-1}\mathbf{p}_k
\end{aligned}$$

and therefore $\hat{\Phi}_{-h}^{-1}$ maps $(\mathbf{q}_k, \mathbf{p}_k) \mapsto (\mathbf{q}_{k+1}, \mathbf{p}_{k+1})$ such that

$$\begin{aligned}
\mathbf{q}_{k+1} &= \mathbf{q}_k + h\,\mathbf{M}^{-1}\mathbf{p}_k \\
\mathbf{p}_{k+1} &= \mathbf{p}_k - h\,\nabla U(\mathbf{q}_{k+1})
\end{aligned}$$

We find that the adjoint of the symplectic Euler scheme is explicitly defined.

4.4 Building Symplectic Integrators: Splitting Methods

Let $H(\mathbf{q},\mathbf{p}) = H_1(\mathbf{q},\mathbf{p}) + H_2(\mathbf{q},\mathbf{p})$ have flow map Φ_t, and let Φ_t^1, Φ_t^2 be the flow maps for the systems with Hamiltonians H_1, H_2 respectively. We propose that the map

$$\Psi_t \coloneqq \Phi_t^1 \circ \Phi_t^2 \tag{65}$$

is an approximation of Φ_t. Notice that the order of composition can be arbitrary due to commutativity of addition. For Ψ_t to be a first order approximation of Φ_t, we need at least

$$||\Psi_t(\mathbf{z}) - \Phi_t(\mathbf{z})|| \leq C(\mathbf{z})t^2 \tag{66}$$

That is, the local error must be 2nd order. Let $\mathbf{z}_0 = (\mathbf{q}_0, \mathbf{p}_0) \in \Omega$ be some arbitrary initial point, and let us flow it across time t to the new point $\Phi_t(\mathbf{z}_0)$. We do a first-order Taylor expansion the flow with respect to the time t,

$$\begin{aligned}\Phi_t(\mathbf{z}_0) &= \mathbf{z}_0 + t\big[\Phi_t(\mathbf{z}_0)\big]' + \mathcal{O}(t^2)\\ &= \mathbf{z}_0 + t\mathbf{J}\nabla_{\mathbf{z}} H(\mathbf{z}_0) + \mathcal{O}(t^2)\\ &= \mathbf{z}_0 + t(\mathbf{J}\nabla_{\mathbf{z}} H_1 + \mathbf{J}\nabla_{\mathbf{z}} H_2)(\mathbf{z}_0) + \mathcal{O}(t^2)\end{aligned} \tag{3.1}$$

On the other hand, we have

$$\begin{aligned}\Phi_t^1(\mathbf{z}_0) &= \mathbf{z}_0 + t\mathbf{J}\nabla_{\mathbf{z}} H_1(\mathbf{z}_0) + \mathcal{O}(t^2)\\ \Phi_t^2(\mathbf{z}_0) &= \mathbf{z}_0 + t\mathbf{J}\nabla_{\mathbf{z}} H_2(\mathbf{z}_0) + \mathcal{O}(t^2)\end{aligned}$$

and composing them gives

$$\begin{aligned}\Phi_t^1 \circ \Phi_t^2 &= \mathbf{z} + t\mathbf{J}\nabla H_2(\mathbf{z}) + t\mathbf{J}\nabla H_1\big(\mathbf{z} + t\mathbf{J}\nabla H_2(\mathbf{z})\big) + \mathcal{O}(t^2)\\ &= \mathbf{z} + t\mathbf{J}\nabla H_2(\mathbf{z}) + t\mathbf{J}\nabla H_1(\mathbf{z}) + \underbrace{t^2(\mathbf{J}\nabla H_1)\circ(\mathbf{J}\nabla H_2)(\mathbf{z})\big)}_{\mathcal{O}(t^2)} + \mathcal{O}(t^2)\\ &= \mathbf{z} + t(\mathbf{J}\nabla H_2 + \mathbf{J}\nabla H_1)(\mathbf{z}) + \mathcal{O}(t^2)\end{aligned}$$

which agrees with the terms of (3.1) up to second order, and therefore the local error is indeed second order.

4.4.1 Symplectic Euler Constructed from Splitting Schemes

Let $H_1(\mathbf{q},\mathbf{p}) = \mathbf{p}^T\mathbf{M}^{-1}\mathbf{p}/2$ and $H_2(\mathbf{q},\mathbf{p}) = U(\mathbf{q})$, then the splitting method for $H = H_1 + H_2$ can be obtained by determining the flow maps for each of the two parts. For H_1 and H_2 we have the differential equations

$$\begin{cases}\dot{\boldsymbol{q}} = \mathbf{M}^{-1}\mathbf{p}\\ \dot{\boldsymbol{p}} = -\nabla_{\mathbf{q}} U(\mathbf{q})\end{cases} \implies H_1\begin{cases}\dot{\boldsymbol{q}} = \mathbf{M}^{-1}\mathbf{p}\\ \dot{\boldsymbol{p}} = \mathbf{0}\end{cases} \text{ and } H_2\begin{cases}\dot{\boldsymbol{q}} = \mathbf{0}\\ \dot{\boldsymbol{p}} = -\nabla_{\mathbf{q}} U(\mathbf{q})\end{cases} \tag{67}$$

The fact that $\dot{\boldsymbol{p}} = 0$ for H_1 tells us that the momentum is constant and therefore the trajectory $\mathbf{q}$ is linear, and hence the discrete flow map $\hat{\Phi}_h^1$ is

$$\hat{\Phi}_h^1\begin{pmatrix}\mathbf{q}_k\\ \mathbf{p}_k\end{pmatrix} = \begin{pmatrix}\mathbf{q}_{k+1}\\ \mathbf{p}_{k+1}\end{pmatrix} = \begin{pmatrix}\mathbf{q}_k + h\mathbf{M}^{-1}\mathbf{p}_k\\ \mathbf{p}_k\end{pmatrix} \tag{68}$$

The flow map $\hat{\Phi}_h^2$ is

$$\hat{\Phi}_h^2\begin{pmatrix}\mathbf{q}_k\\ \mathbf{p}_k\end{pmatrix} = \begin{pmatrix}\mathbf{q}_{k+1}\\ \mathbf{p}_{k+1}\end{pmatrix} = \begin{pmatrix}\mathbf{q}_k\\ \mathbf{p}_k - h\nabla U(\mathbf{q}_k)\end{pmatrix} \tag{69}$$

The composition of these maps is

$$\hat{\Phi}_h^1 \circ \hat{\Phi}_h^2 = \begin{cases}\mathbf{q}_{k+1} = \mathbf{q}_k + h\mathbf{M}^{-1}\mathbf{p}_{k+1}\\ \mathbf{p}_{k+1} = \mathbf{p}_k - h\nabla U(\mathbf{q}_k)\end{cases} \tag{70}$$

which is precisely the symplectic Euler method. Composing the same two maps in the opposite order gives the adjoint symplectic Euler method.

$$\big(\hat{\Phi}_h^1 \circ \hat{\Phi}_h^2\big)^\dagger = \hat{\Phi}_h^2 \circ \hat{\Phi}_h^1 = \begin{cases}\mathbf{q}_{k+1} = \mathbf{q}_k + h\mathbf{M}^{-1}\mathbf{p}_k\\ \mathbf{p}_{k+1} = \mathbf{p}_k - h\nabla U(\mathbf{q}_{k+1})\end{cases} \tag{71}$$

4.4.2 Symplectic Verlet Method from Splitting Schemes

For the symplectic Euler method $\hat{\Phi}_h$ and its adjoint method $\hat{\Phi}_h^\dagger$, consider the composition

$$\mathcal{K}_h \coloneqq \hat{\Phi}_{h/2}^\dagger \circ \hat{\Phi}_{h/2} \tag{72}$$

Computing this, we have $\hat{\Phi}_{h/2}(\mathbf{q}_k, \mathbf{p}_k) = (\mathbf{q}_{k+1/2}, \mathbf{p}_{k+1/2})$, and $\hat{\Phi}_{h/2}^\dagger(\mathbf{q}_{k+1/2}, \mathbf{p}_{k+1/2}) = (\mathbf{q}_{k+1}, \mathbf{p}_{k+1})$ defined

$$\hat{\Phi}_{h/2}\begin{pmatrix}\mathbf{q}_k \\ \mathbf{p}_k\end{pmatrix} = \begin{cases}\mathbf{q}_{k+1/2} = \mathbf{q}_k + \frac{h}{2}\mathbf{M}^{-1}\mathbf{p}_{k+1/2} \\ \mathbf{p}_{k+1/2} = \mathbf{p}_k - \frac{h}{2}\nabla U(\mathbf{q}_k)\end{cases}$$
$$\hat{\Phi}_{h/2}^\dagger\begin{pmatrix}\mathbf{q}_{k+1/2} \\ \mathbf{p}_{k+1/2}\end{pmatrix} = \begin{cases}\mathbf{q}_{k+1} = \mathbf{q}_{k+1/2} + \frac{h}{2}\mathbf{M}^{-1}\mathbf{p}_{k+1/2} \\ \mathbf{p}_{k+1} = \mathbf{p}_{k+1/2} - \frac{h}{2}\nabla U(\mathbf{q}_{k+1})\end{cases}$$

This composition simplifies to

$$\mathcal{K}_h \coloneqq \hat{\Phi}_{h/2}^\dagger \circ \hat{\Phi}_{h/2} = \begin{cases}\mathbf{p}_{k+1/2} & = \mathbf{p}_k - \frac{h}{2}\nabla U(\mathbf{q}_k) \\ \mathbf{q}_{k+1} & = \mathbf{q}_k + h\mathbf{M}^{-1}\mathbf{p}_{k+1/2} \\ \mathbf{p}_{k+1} & = \mathbf{q}_{k+1/2} - \frac{h}{2}\nabla U(\mathbf{q}_{k+1})\end{cases} \tag{73}$$

which is precisely the velocity Verlet method in Hamiltonian form. Since we have obtained this method as the composition of two symplectic maps, and the symplectic maps form a group, we know that this method will also be symplectic. Similarly, we can construct the adjoint map of $\mathcal{K}_h$ by simply taking the composition in the other direction, which we see to be the same (i.e. $\mathcal{K}_h$ is symmetric/self-adjoint).

$$\mathcal{K}_h^\dagger \coloneqq \big(\hat{\Phi}_{h/2}^\dagger \circ \hat{\Phi}_{h/2}\big)^\dagger = \hat{\Phi}_{h/2}^\dagger \circ \hat{\Phi}_{h/2} = \mathcal{K}_h \tag{74}$$

4.4.3 General Composition Methods

In general, if we have any two symplectic numerical methods, say $\hat{\Phi}_h^1$ and $\hat{\Phi}_h^2$, then the composition

$$\hat{\Phi}_h \coloneqq \hat{\Phi}_h^1 \circ \hat{\Phi}_h^2 \tag{75}$$

is another symplectic numerical method. The order of this new method is typically the minimum of the orders of the two methods involved, but it can be higher, as the example of the Verlet method (constructed by composing the Euler and its adjoint, both of order 1).

4.5 Modified, Shadow Hamiltonians

We already know that our symplectic discretized schemes successfully conserves the 2-form, but these schemes do not actually conserve the Hamiltonian. Let us take the 1-dimensional harmonic oscillator with frequency ω, which has the Hamiltonian $H(q,p) = p^2/2 + \omega^2 q^2/2$. Let us discretize it with the adjoint symplectic Euler method (regarding the mass $m = 1$):

$$q_{k+1} = q_k + hp_k$$
$$p_{k+1} = q_{k+1} - h\omega^2 q_{k+1}$$

Then, we can do simple algebra to see that $H(q_k, p_k) \neq H(q_{k+1}, p_{k+1})$.

$$\begin{aligned} H(q_{k+1}, p_{k+1}) &= \frac{1}{2}(p - h\omega^2 q_{k+1})^2 + \frac{1}{2}\omega^2(q + hp)^2 \\ &= \frac{1}{2}(p_k^2 - 2p_k h\omega^2 q_{k+1} + h^2\omega^4 q_k^2)^2 + \frac{1}{2}\omega^2(q_k^2 + 2hq_kp_k + h^2p_k^2)^2 \\ &= \ldots \neq H(q_k, p_k) \end{aligned}$$

However, if we modify the Hamiltonian from $H(q,p) = p^2/2 + \omega^2 q^2/2$ to

$$\tilde{H}(q,p) = \frac{1}{2}\big(p^2 + h\omega^2 pq + \omega^2 q^2\big) \tag{76}$$

Then it turns out that $\tilde{H}(q_k, p_k) = \tilde{H}(q_{k+1}, p_{k+1})$, and so this new modified Hamiltonian is conserved. This is significant, since now we have some other invariant property of the numerical method. The phase space of this 1-dimensional oscillator is simply $\Omega_q \times \Omega_p \subset \mathbb{R} \times \mathbb{R}$. If we were to visualize the level sets of the Hamiltonian, then we can imagine the level sets of the modified Hamiltonian to be a "perturbed" version of the original ones. The fact that there even exists a modified Hamiltonian invariant is another special property of symplectic integrators. If we used Euler's method to solve the harmonic oscillator, we would find that energy grows without bound.

4.5.1 Lie Derivatives and Poisson Brackets

Let us have $\mathbf{z} \in \mathbb{R}^m$ in our phase space. Furthermore, let us assume that at any point $\boldsymbol{\zeta} \in \mathbb{R}^m$ there is a unique solution $\mathbf{z}(t, \boldsymbol{\zeta})$ such that $\dot{\mathbf{z}}(t) = \mathbf{f}\big(\mathbf{z}(t)\big),\ \mathbf{z}(0) = \boldsymbol{\zeta}$ is globally defined for all t. This also means that $\mathbf{f} : \mathbb{R}^m \longrightarrow \mathbb{R}^m$ is a vector field defining the phase flow on $\mathbb{R}^m$. On this phase space let us define a scalar field $\phi : \mathbb{R}^m \longrightarrow \mathbb{R}$. Letting $\hat{\phi} = \phi \circ \mathbf{z} : \mathbb{R} \longrightarrow \mathbb{R}$ be the function outputting the value of ϕ across the path ϕ, we can take its derivative using chain rule

$$\begin{aligned}
\frac{d}{dt}\hat{\phi}\bigg|_{t=0} &= \begin{pmatrix} \frac{\partial \phi}{\partial z_1}(\zeta) \\ \frac{\partial \phi}{\partial z_2}(\zeta) \\ \cdots \\ \frac{\partial \phi}{\partial z_m}(\zeta) \end{pmatrix} \begin{pmatrix} \frac{dz_1}{dt}(0) & \frac{dz_2}{dt}(0) & \ldots & \frac{dz_m}{dt}(0) \end{pmatrix} \\
&= \nabla \phi(\boldsymbol{\zeta}) \cdot \dot{\mathbf{z}}(0) \\
&= \mathbf{f}(\boldsymbol{\zeta}) \cdot \nabla \phi(\boldsymbol{\zeta})
\end{aligned}$$

That is, the derivative of $\hat{\phi}$ at $t = 0$ is the dot product of the derivative vector at $\boldsymbol{\zeta}$ and the gradient of the scalar potential at $\boldsymbol{\zeta}$. From this, we can define the **Lie derivative $\mathcal{L}_{\mathbf{f}}$** as

$$\mathcal{L}_{\mathbf{f}}\phi := \mathbf{f} \cdot \nabla \phi \tag{77}$$

This is similar to the directional derivative of ϕ in direction $\mathbf{f}$. Therefore, the equation for the evolution of ϕ can be written as follows. Note also that the origin of time is irrelevant since we can always just shift the time frame by a constant, so we can really focus on the point on the path in $\mathbb{R}^m$ rather than the associated time. Therefore, at a certain time t with associated point $\mathbf{z}(t)$, this Lie derivative at $\mathbf{z}(t)$ in direction $\mathbf{f}$ under scalar field ϕ is

$$\frac{d}{dt}\phi\big(\mathbf{z}(t)\big) = (\mathcal{L}_{\mathbf{f}}\phi)\big(\mathbf{z}(t)\big) \tag{78}$$

Similarly, the second derivative at $\mathbf{z}(t)$ in direction $\mathbf{f}$ under ϕ will be denoted

$$\frac{d^2}{dt^2}\phi\big(\mathbf{z}(t)\big) = (\mathcal{L}_f^2\phi)\big(\mathbf{z}(t)\big) \tag{79}$$

and so on for higher derivatives. The Taylor series expansion of $\phi(\mathbf{z}(t))$, centered at 0, along a solution of the differential equation can therefore be written as

$$\begin{aligned}
\phi\big(\mathbf{z}(t)\big) &= \phi\big(\mathbf{z}(0)\big) + t\left(\frac{d}{dt}\phi\big(\mathbf{z}(t)\big)\bigg|_{t=0}\right) + \frac{t^2}{2}\left(\frac{d^2}{dt^2}\phi\big(\mathbf{z}(t)\big)\bigg|_{t=0}\right) + \ldots \\
&= \phi\big(\mathbf{z}(0)\big) + t(\mathcal{L}_{\mathbf{f}}\phi)\big(\mathbf{z}(0)\big) + \frac{t^2}{2}(\mathcal{L}_f^2\phi)\big(\mathbf{z}(0)\big) + \ldots \\
&= \big(e^{t\mathcal{L}_{\mathbf{f}}}\phi\big)\big(\mathbf{z}(0)\big)
\end{aligned}$$

In summary, given the initial value problem $\dot{\mathbf{z}}(t) = \mathbf{f}\big(\mathbf{z}(t)\big),\ \mathbf{z}(0) = \boldsymbol{\zeta}$ and any scalar field ϕ defined on $\mathbb{R}^m$, we have $\phi(\mathbf{z}(t)) = (e^{t\mathcal{L}_{\mathbf{f}}}\phi)(\mathbf{z}(0))$. By setting ϕ to simply be the component functions z_i of $\mathbf{z}$, this fully defines

$\mathbf{z}(t)$ given $\mathbf{z}(0) = \boldsymbol{\zeta}$. We also don't need to fix the initial point, since the flow map Φ_t is a collection of all the flows for every single possible initial point. Concisely, by abuse of notation, the flow map Φ_t can be represented by

$$\Phi_t = \exp(t\mathcal{L}_{\mathbf{f}}) \tag{80}$$

More strictly speaking, what is really meant by the equality above is that the individual components satisfy

$$z_i(t, \boldsymbol{\zeta}) = \big[\Phi_t(\boldsymbol{\zeta})\big]_i = \big(\exp(t\mathcal{L}_{\mathbf{f}})z_i\big)(\boldsymbol{\zeta}) \tag{81}$$

Ignoring the initial term in the Taylor series allows us write

$$e^{t\mathcal{L}_{\mathbf{f}}}\phi = \phi + t\mathcal{L}_{\mathbf{f}}\phi + \frac{t^2}{2}\mathcal{L}_{\mathbf{f}}^2\phi + \dots \tag{82}$$

which may or may not be bounded with respect to functions ϕ. Though significant, we will ignore this problem for now. We now introduce the **Poisson bracket**, which is defined for two smooth scalar-valued functions g_1 and g_2 of phase variables $(\mathbf{q}, \mathbf{p}) \in \mathbb{R}^m$ by

$$\{g_1, g_2\} \coloneqq \nabla g_1^T \mathbf{J} \nabla g_2 = \sum_{i=1}^m \left(\frac{\partial g_1}{\partial q_i}\frac{\partial g_2}{\partial p_i} - \frac{\partial g_2}{\partial q_i}\frac{\partial g_1}{\partial p_i}\right) \tag{83}$$

where $\mathbf{J}$ is the skew symmetric symplectic structure matrix.

$$\mathbf{J} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{pmatrix} \tag{84}$$

As the name suggests, the Poisson bracket has the bracket structure: it is bilinear, skew-symmetric, and satisfied the Jacobi identity defined

$$\{g_1, \{g_2, g_3\}\} + \{g_3, \{g_1, g_2\}\} + \{g_2, \{g_3, g_1\}\} = 0 \tag{85}$$

We can in fact write differential equations in terms of Poisson brackets. For example, the simple component DEQ $\dot{q}_i = p_i$ (of vector DEQ $\dot{\mathbf{q}} = \mathbf{p}$) can be written in terms of brackets as the first line, with the derivation shown in the following lines.

$$\begin{aligned}
\dot{q}_i &= \{q_i, H\} \\
&= \nabla q_i^T \mathbf{J} \nabla H \\
&= \begin{pmatrix} \mathbf{e}_i & 0 \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \nabla_{\mathbf{q}} H \\ \nabla_{\mathbf{p}} H \end{pmatrix} \\
&= \nabla_{p_i} H = \frac{\partial}{\partial p_i} H = \frac{\partial}{\partial p_i} \frac{1}{2}||\mathbf{p}||^2 + U(\mathbf{q}) = p_i
\end{aligned}$$

More generally, if $F(\mathbf{q}, \mathbf{p})$ is any smooth, scalar-valued function of the phase variables, we may write

$$\dot{F} = \frac{d}{dt} F\big(\mathbf{q}(t), \mathbf{p}(t)\big) = \{F, H\} \tag{86}$$

We can see this because

$$\begin{aligned}
\{F, H\} &= \nabla F^T \mathbf{J} \nabla H \\
&= \begin{pmatrix} \nabla_{\mathbf{q}} F & \nabla_{\mathbf{p}} F \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \nabla_{\mathbf{q}} H \\ \nabla_{\mathbf{p}} H \end{pmatrix} \\
&= \nabla_{\mathbf{q}} F \cdot \nabla_{\mathbf{p}} H - \nabla_{\mathbf{q}} H \cdot \nabla_{\mathbf{p}} F \\
&= \nabla_{\mathbf{q}} F \cdot \nabla_{\mathbf{p}} \left(\frac{1}{2}||\mathbf{p}||^2 + U(\mathbf{q})\right) - \nabla_{\mathbf{q}} \left(\frac{1}{2}||\mathbf{p}||^2 + U(\mathbf{q})\right) \cdot \nabla_{\mathbf{p}} F \\
&= \nabla_{\mathbf{q}} F \cdot \mathbf{p} - \nabla_{\mathbf{q}} F \cdot \nabla_{\mathbf{q}} U(\mathbf{q}) \\
&= \frac{\partial F}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} - \frac{\partial F}{\partial \mathbf{p}} \frac{d\mathbf{p}}{dt} \\
&= \frac{d}{dt} F\big(\mathbf{q}(t), \mathbf{p}(t)\big)
\end{aligned}$$

Therefore, we have the following relation between the Lie derivative and the Poisson bracket.

$$\mathcal{L}_{\mathbf{J}\nabla H} F = \mathbf{J}\nabla H \cdot \nabla F = \nabla F^T \mathbf{J} \nabla H = \{F, H\} \tag{87}$$

What this says is that given the coupled Hamiltonian equations

$$\begin{cases} \dot{\boldsymbol{q}} & = \mathbf{M}^{-1}\mathbf{p} \\ \dot{\boldsymbol{p}} & = -\nabla_{\mathbf{q}} U(\mathbf{q}) \end{cases} \implies \begin{pmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{p}} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{I}_{3N} \\ -\mathbf{I}_{3N} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \nabla_{\mathbf{q}} U(\mathbf{q}) \\ \mathbf{M}^{-1}\mathbf{p} \end{pmatrix} = \mathbf{J}\nabla H(\mathbf{q}, \mathbf{p}) \tag{88}$$

the Lie derivative $\mathcal{L}_{\mathbf{J}\nabla H} F$ under scalar field F can be represented in terms of the Lie bracket $\{F, H\}$, and therefore $\exp(t\mathcal{L}_{\mathbf{J}\nabla H})$ can be regarded as the Hamiltonian flow of the system. Following the convention, we will simplify the expression $\mathcal{L}_{\mathbf{J}\nabla H}$ to simply $\mathcal{L}_H$. Note that $\mathcal{L}_H$ takes in a smooth scalar field F and outputs another scalar field that tells the directional derivative in direction H.

$$\mathcal{L}_H : C^\infty(\mathbb{R}^m) \longrightarrow C^\infty(\mathbb{R}^m) \tag{89}$$

4.5.2 Backward Error Analysis for Hamiltonian Splitting Methods

Note that the relation $\mathcal{L}_H \phi = \{\phi, H\}$ is linear in H (due to bilinearity). Let us have a system with Hamiltonian $H = H_1 + H_2$. Then, $\mathcal{L}_H = \mathcal{L}_{H_1 + H_2} = \mathcal{L}_{H_1} + \mathcal{L}_{H_2}$, and thus the flow map of the system is

$$\Phi_t = e^{t(\mathcal{L}_{H_1} + \mathcal{L}_{H_2})} \tag{90}$$

The splitting method based on a composition of flows on H_1 and H_2 is $e^{t\mathcal{L}_{H_1}} e^{t\mathcal{L}_{H_2}}$. It is well known that given noncommuting operators A, B, e^{A+B} does not necessarily equal $e^A e^B$. Expanding and subtracting gives us the difference to be (where $[A, B] = AB - BA$ is the commutator):

$$e^{h\mathcal{L}_{H_1}} e^{h\mathcal{L}_{H_2}} - e^{h\mathcal{L}_H} = \frac{h^2}{2}[\mathcal{L}_{H_1}, \mathcal{L}_{H_2}] + \mathcal{O}(h^3) \tag{91}$$

We can see that since $\mathcal{L}_H f = \{f, H\}$, $\mathcal{L}_{H_1}\mathcal{L}_{H_2} f = \{\{f, H_2\}, H_1\}$ and thus the commutator reduces to

$$\begin{aligned} [\mathcal{L}_{H_1}, \mathcal{L}_{H_2}] f &= \{\{f, H_2\}, H_1\} - \{\{f, H_1\}, H_2\} \\ &= \{\{f, H_2\}, H_1\} - \{\{H_1, f\}, H_2\} && \text{(skew symmetry)} \\ &= -\{\{H_2, H_1\}, f\} && \text{(Jacobi identity)} \\ &= \{f, \{H_2, H_1\}\} && \text{(skew symmetry)} \\ &= \mathcal{L}_{\{H_1, H_2\}} f \end{aligned}$$

This means that it is possible to relate the commutator of Lie derivatives of Hamiltonian fields H_1, H_2 to the Lie derivative of the Poisson bracket of the corresponding Hamiltonians. Ignoring the $\mathcal{O}(h^3)$ term, we can interpret the error $[\mathcal{L}_{H_1}, \mathcal{L}_{H_2}] = \mathcal{L}_{\{H_1, H_2\}}$ as itself being derived from another Hamiltonian. Let us expand $e^{h\mathcal{L}_{H_1}} e^{h\mathcal{L}_{H_2}} = e^{h\mathcal{L}_{\tilde{H}}}$ using the Baker-Campbell-Hausdorff formula:

$$e^{h\mathcal{L}_{H_1}} e^{h\mathcal{L}_{H_2}} = \exp\bigg(h\big(\mathcal{L}_{H_1} + \mathcal{L}_{H_2}\big) + \frac{h^2}{2}[\mathcal{L}_{H_1}, \mathcal{L}_{H_1}] + \frac{h^3}{12}\big([\mathcal{L}_{H_1}, [\mathcal{L}_{H_1}, \mathcal{L}_{H_2}]] - [\mathcal{L}_{H_2}, [\mathcal{L}_{H_1}, \mathcal{L}_{H_2}]]\big) + \ldots \bigg) \tag{92}$$

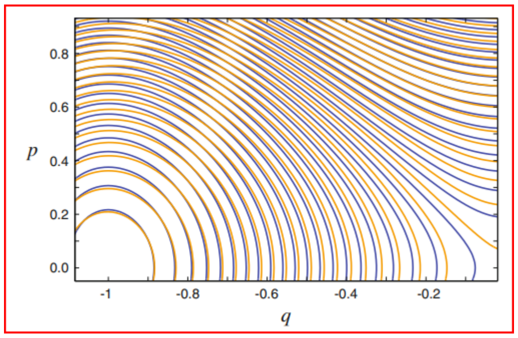
Then, $h\mathcal{L}_{\tilde{H}}$ would be the term in the exponent. Dividing by h and substituting $\mathcal{L}_H = \mathcal{L}_{H_1} + \mathcal{L}_{H_2}$ gives

$$\begin{aligned} \mathcal{L}_{\tilde{H}} &= \mathcal{L}_H + \frac{h}{2}[\mathcal{L}_{H_1}, \mathcal{L}_{H_2}] + \frac{h^2}{12}\big([\mathcal{L}_{H_1}, [\mathcal{L}_{H_1}, \mathcal{L}_{H_2}]] - [\mathcal{L}_{H_2}, [\mathcal{L}_{H_1}, \mathcal{L}_{H_2}]]\big) + \ldots \\ &= \mathcal{L}_H + \frac{h}{2}\mathcal{L}_{\{H_1, H_2\}} + \frac{h^2}{12}\big(\mathcal{L}_{\{H_1, \{H_1, H_2\}\}} - \mathcal{L}_{\{H_2, \{H_1, H_2\}\}}\big) + \ldots \\ &= \mathcal{L}_{H + \frac{h}{2}\{H_1, H_2\} + \frac{h^2}{12}(\{H_1, \{H_1, H_2\}\} - \{H_2, \{H_1, H_2\}\}) + \ldots} \end{aligned}$$

which implies that the Hamiltonian $\tilde{H}$ of the splitting approximation deviates from the true Hamiltonian H through the BCH formula.

$$\tilde{H} = H + \underbrace{\frac{h}{2}\{H_1, H_2\} + \frac{h^2}{12}(\{H_1, \{H_1, H_2\}\} - \{H_2, \{H_1, H_2\}\}) + \dots}_{\text{error term}} \tag{93}$$

This series $\tilde{H}$ is referred to as the **shadow Hamiltonian** corresponding to the splitting method. The numerical method may be viewed as being equivalent to the exact solution of a nearby Hamiltonian system, rather than the true one. We can visualize the isocontour lines for a double-well model along with its modified Verlet Hamiltonian below.



Note that we still haven't addressed the convergence of this series, but we simply assume that the error term is bounded (which may not always be justified). Furthermore if H_1 and H_2 commute, i.e. $\{H_1, H_2\} = 0$, then there is no error in splitting. There are few special splitting cases where this would happen. An alternative approach to numerically solving the SDE is to find a scheme with Hamiltonian that has *its* shadow Hamiltonian to be our target one. That is, we use a perturbed version of the original SDE and discretize it, which should lead to a higher order scheme.

4.5.3 Symplectic Euler

Recall that splitting our Hamiltonian using

$$H_1 = \frac{1}{2}\mathbf{p}^T\mathbf{M}^{-1}\mathbf{p}, \quad H_2 = U(\mathbf{q}) \tag{94}$$

gives us the symplectic Euler method. The BCH expansion gives us the following perturbed Hamiltonian, which we can see has a leading error term of power 1, making it a first-order scheme.

$$\begin{aligned}
\tilde{H}_h &= H + \frac{h}{2}\{H_1, H_2\} + \frac{h^2}{12}\big(\{H_1, \{H_1, H_2\}\} - \{H_2, \{H_1, H_2\}\}\big) + \dots \\
&= H + \frac{H}{2}\nabla H_1^T \mathbf{J} \nabla H_2 + \dots \\
&= H + \frac{h}{2}\left[\begin{pmatrix} \mathbf{0} & \mathbf{p}^T\mathbf{M}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \nabla_{\mathbf{q}} U(\mathbf{q}) \\ \mathbf{0} \end{pmatrix} \right] + \dots \\
&= H - \frac{h}{2}\mathbf{p}^T\mathbf{M}^{-1}\nabla U(\mathbf{q}) + \frac{h^2}{12}\big[\mathbf{p}^T\mathbf{M}^{-1}U''\mathbf{M}^{-1}\mathbf{p} + \nabla U(\mathbf{q})^T\mathbf{M}^{-1}\nabla U(\mathbf{q})\big] \\
&- \frac{h^3}{12}\nabla U(\mathbf{q})^T\mathbf{M}^{-1}U''(\mathbf{q})\mathbf{M}^{-1}\mathbf{p} + \mathcal{O}(h^4)
\end{aligned}$$

4.5.4 Velocity Verlet

Given a Hamiltonian symmetrically split into three parts

$$H(\mathbf{q}, \mathbf{p}) = H_1 + H_2 + H_3 = \frac{1}{2}U(\mathbf{q}) + \frac{1}{2}T(\mathbf{p}) + \frac{1}{2}U(\mathbf{q}) \tag{95}$$

calculating the estimate $\exp\left(h\mathcal{L}_H\right) \approx \exp(\frac{h}{2}\mathcal{L}_{H_1})\exp(\frac{h}{2}\mathcal{L}_{H_2})\exp(\frac{h}{2}\mathcal{L}_{H_3})$ using the BCH lemma gives the following. Notice that the symmetricity of the splitting scheme allows us to cancel out the odd powered terms.

$$\tilde{H}_h = T + U + \frac{h^2}{12}\Big(\{T,\{T,U\}\} - \frac{1}{2}\{U,\{U,T\}\}\Big) + \dots \tag{96}$$

The shadow Hamiltonian of the Velocity Verlet scheme applied to a single degree of freedom system of the form $H(q,p) = U(q) + \frac{1}{2}p^2$ then gives

$$\begin{aligned}\tilde{H}(q,p) = H(q,p) + \frac{h^2}{24}\big(2pU''(q)p - (U'(q))^2\big) + h^4\Big(\frac{1}{720}p^4U''''(q) - \frac{1}{120}p^2U'(q)U'''(q) \\ - \frac{1}{240}(U'(q))^2U''(q) - \frac{1}{60}p^2(U''(q))^2 + U'(q)U'''(q)\Big) + \mathcal{O}(h^6)\end{aligned}$$

Higher order symplectic integrators give higher order error terms (e.g. Yoshida-4, Imada-4).

4.6 Hamiltonian Monte Carlo (HMC)

Hamiltonian Monte Carlo is one type of MCMC Metropolis-Hastings algorithms, with a Hamiltonian dynamics evolution simulated using a time-reversible, symplectic integrator (usually Velocity-Verlet). We first initialize our chain $\mathbf{X}_0 = (\mathbf{q}^0, \mathbf{p}^0)$ and compute the Hamiltonian $H(\mathbf{q},\mathbf{p}) = U(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T\mathbf{M}^{-1}\mathbf{p}$. Given that we have $\mathbf{X}_k = (\mathbf{q}_k, \mathbf{p}_k)$ at the end of the kth step, we then repeat the following steps:

1. Fix $\mathbf{q}$ but pick $\mathbf{p}_{k+1} \sim \mathcal{N}(\mathbf{p}_k, \mathbf{\Sigma})$.
2. We run Velocity Verlet (or some other symplectic scheme) for some fixed number of steps L of stepsize h, which models Hamiltonian flow to some new position $(\mathbf{q}_k', \mathbf{p}_k')$. This is our transition proposal. Note that for every step in Velocity Verlet, we must compute the gradient of the potential. In order to simulate Hamiltonian flow, this gradient must be exactly computed; our batch approximation will lead to discretized steps that is not deterministic anymore and do not fulfill our symplectic properties and energy preservation.
3. We accept this proposal with probability

$$\alpha = \min\left(1, \frac{\exp\big[-H(\mathbf{q}_k', \mathbf{p}_k')\big]}{\exp\big[-H(\mathbf{q}_k, \mathbf{p}_k)\big]}\right)$$

   and assign $\mathbf{X}_{k+1} = (\mathbf{q}_{k+1}, \mathbf{p}_{k+1}) = (\mathbf{q}_k', \mathbf{p}_k')$ upon acceptance and $\mathbf{X}_{k+1} = \mathbf{X}_k$ if not. Note that in this step, we require the exact evaluations of our Hamiltonian.

Hamiltonian Monte Carlo is very useful if we could efficiently calculate the true log-posterior, but otherwise, the batch approximation will not model a Hamiltonian flow (and thus will not preserve the symplectic, time-reversibility, etc. properties), rendering HMC useless.

HMC is able to draw samples in high dimensions with greater efficiency than classical MCMC. Its key advantage is its ability to draw samples that are large distances apart by evolving them via Hamiltonian dynamics. The acceptance rate depends on the error accumulated along the sample trajectory (i.e. the error of the shadow Hamiltonian), and remains large even in high dimensions. However, a large step size (leading to greater error of shadow Hamiltonian), a large system, or a poorly behaved target density leads to greater numerical error and thus to lower sample acceptance, which induces heavy autocorrelation, necessitating a larger sample size and thus higher computational costs.

One approach to ease this burden is to exploit the structure of the numerical integrator error and instead target the density corresponding to a modified, *shadow Hamiltonian*. This leads to higher sample acceptance rate, at the cost of some induced bias. This bias is usually well-quantified, and we can compensate for this induced bias.

4.7 No U-Turn Sampler (NUTS)

5 Langevin Dynamics Inspired Samplers and Integrators

In addition to Hamiltonian dynamics, we can model the dynamics of molecular systems with Langevin dynamics. This model relies on the fact that a real world molecular system is unlikely to be present in a vacuum (there may be friction, jostling, etc.). There are two types of Langevin dynamics: overdamped and underdamped. The **Gibbs measure** mentioned below is an invariant distribution of this random process, similar to a stationary distribution of a Markov chain. That is, if we ran the model for an infinite amount of time, the Gibbs measure would be the density representing the probability of finding that particle at a certain location at any point in time.

1. The overdamped Langevin equation does not rely on the momenta.

$$\dot{\mathbf{q}} = -\nabla U(\mathbf{q}) + \sqrt{2\beta^{-1}}\dot{W} \tag{97}$$

   Its Gibbs measure (invariant distribution) is

$$\pi(\boldsymbol{\theta}) = \frac{1}{Z}\exp\big(-\beta U(\mathbf{q})\big), \text{ where } Z = \int \exp\big(-\beta U(\mathbf{q})\big)\, dq \tag{98}$$

   More precisely, given that the path $\mathbf{q}(t)$ at time t is distributed according to (parameterized) density ρ_t, $\rho_t \to \frac{1}{Z}e^{-\beta U(\mathbf{q})}$ as $t \to +\infty$.

2. The underdamped Langevin equation can be interpreted as a Hamiltonian model, with the additional $-\gamma\mathbf{p} + \sqrt{2\gamma\beta^{-1}}\dot{W}$ term representing the interaction of the Hamiltonian system with an outside environment (called a heat bath or thermostat).

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{M}^{-1}\mathbf{p} \\ \dot{\mathbf{p}} &= -\nabla U(\mathbf{q}) - \gamma\mathbf{p} + \sqrt{2\gamma\beta^{-1}}\mathbf{M}^{1/2}\dot{W}\end{aligned}$$

   The γ is the damping constant and β is the inverse temperature. We can think of the term $-\gamma\mathbf{p}$ as the damping/dissipative term which "drags" the momentum $\mathbf{p}$ to 0. The higher the γ, the stronger this drag. As γ grows, the system spans from the inertial all the way to the diffusive (aka Brownian) regime. The term $\sqrt{2\gamma\beta^{-1}}\dot{W}$ is the random term, which increases as temperature increases. It has invariant distribution

$$\pi(\boldsymbol{q}, \boldsymbol{p}) = \frac{1}{Z}\exp\Big(-\beta\big[U(\mathbf{q}) + \frac{1}{2}|\mathbf{p}|^2\big]\Big) \tag{99}$$

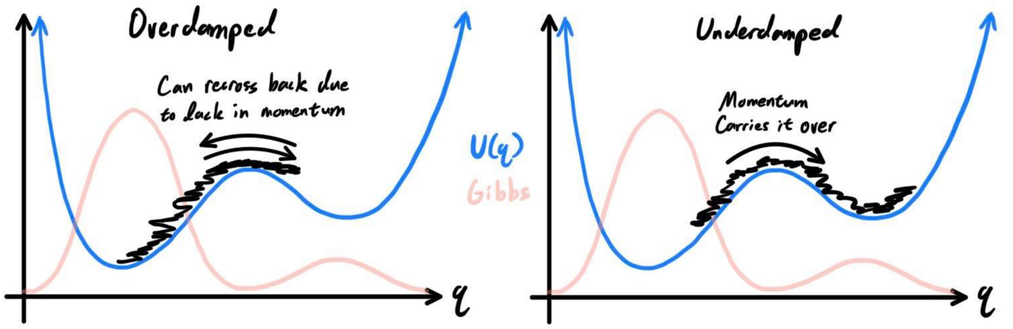
To understand the relationship between the overdamped and underdamped Langevin equations and the physical systems that they represent, we can think of the overdamped equation as a limit of the underdamped one. As we set $\gamma \to +\infty$ (followed by an appropriate time scale), the underdamped Langevin equation would converge to the overdamped because the friction term would become very large, causing the momenta to dissipate instantaneously. Another way to describe this limit is to incorporate a mass matrix $\mathbf{M}$ into the underdamped:

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{M}^{-1}\mathbf{p} \\ \dot{\mathbf{p}} &= -\nabla U(\mathbf{q}) - \gamma\mathbf{M}^{-1}\mathbf{p} + \sqrt{2\gamma\beta^{-1}}\dot{W}\end{aligned}$$

If we let $\mathbf{M} \to \mathbf{0}$, then we can see that the dissipative term $-\gamma\mathbf{M}^{-1}\mathbf{p}$ will grow very large, which leads to convergence to the overdamped equation.

The overdamped Langevin equation is usually used to represent Brownian motion, similar to a random walk, in which there is no memory of the momenta from one time to another. The underdamped Langevin equation incorporates the momenta $\mathbf{p}$, and so the trajectory would be a lot smoother.

The underdamped equation has a lot nicer properties that allows us to sample efficiently. For example, when we have a double well potential $U(q)$ with the associated Gibbs measure, sampling from this potential with an overdamped integrator can cause problems. The overdamped integrator does not remember momentum, and so when crossing the energy barrier it tends to go over and recross back due to the random term.

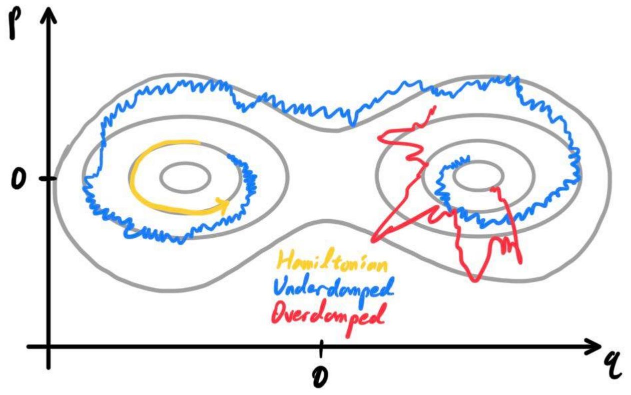


For the underdamped, the momentum is remembered and so when we reach over the barrier, the momentum that accelerates the particle across the well, along with the momentum that accelerates it down the well as soon as it is across, carries the particle into the other well. The choice of our friction coefficient γ will determine how often we transition one stable state (well) to the other stable state. Choosing the right γ is very important when sampling.

1. If γ is too large, we will have very similar dynamics to the overdamped Langevin equation (lots of randomness and potential recrossings), which is not ideal.
2. If γ is too small, it will be very similar to Hamiltonian dynamics, with a very small dissipative and random forces. It will end up just crossing back and forth smoothly and deterministically.

To compare Hamiltonian, underdamped, and overdamped dynamics, let us take a look at the phase space of the double well, with equi-Hamiltonian level sets.

1. A Hamiltonian flow will precisely be along the level sets, since the Hamiltonian is conserved.
2. An underdamped Langevin flow (with γ not too large) will move slowly between level sets. It is important not to set γ to small since then our flow would transition very slowly between level sets and not explore our phase space very quickly.
3. An overdamped Langevin flow (or underdamped with large γ) will move very quickly between level sets, leading to a random walk behavior.



5.1 SGLD

Now if we add Gaussian noise to SGD, then we get *Stochastic Gradient Langevin Dynamics (SGLD)* sampler, which is the discretized form of the overdamped Langevin equation

$$\dot{\mathbf{q}} = -M^{-1}\nabla_{\mathbf{q}} U(\mathbf{q}) + \sqrt{2\beta^{-1}} M^{-1/2} \dot{W} \tag{100}$$

where M is the mass matrix, β the inverse temperature, and $\dot{W}$ a Weiner process. If the gradient computations are exact, then SGLD reduces to the *Langevin Monte Carlo* algorithm. This algorithm is also a reduction of Hamiltonian Monte Carlo, consisting of a single leapfrog step proposal rather than a series of steps. Since SGLD can be formulated as a modification of both SGD and MCMC methods, it lies at the intersection between optimization and sampling algorithms. The method maintains SGD's ability to quickly converge to regions of low cost while providing samples to facilitate posterior inference.

If we set the mass matrix to be I, we can update $\mathbf{q}$ according to the following discretization.

$$\mathbf{q}_{t+1} = \mathbf{q}_t - \nabla_{\mathbf{q}} U(\mathbf{q}_t) + \sqrt{2\beta^{-1}}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{101}$$

Algorithm 4 SGLD

```
Require: Initial $\boldsymbol{\theta}_0$, Stepsize function $h(t)$, Minibatch size $m$
  for $t = 0$ to $T$ do
    $\hat{g}(\theta_t) \leftarrow \nabla_\theta \log p(\theta_t \mid M_m(\mathcal{D}))$
    $\epsilon_t \sim \mathcal{N}(0, I)$
    $\theta_{t+1} \leftarrow \theta_t + h(t) \cdot \hat{g}(\theta_t) + \sqrt{2h(t)\beta^{-1}}\,\epsilon_t$
  end for
```

We can incorporate the mass matrix, which is approximated by the precision of the log posterior ($M^{-1} = \Sigma$), for adapting (along with preconditioning if needed). This would result in the discretized step:

$$\mathbf{q}_{t+1} = \mathbf{q}_t - M^{-1} \nabla_{\mathbf{q}} U(\mathbf{q}_t) + \sqrt{2\beta^{-1}} M^{-1}\,\boldsymbol{\epsilon} \tag{102}$$

Algorithm 5 Adaptive SGLD

```
Require: Initial $\boldsymbol{\theta}_0$, Stepsize function $h(t)$, Minibatch size $m$, Adaptation burn-in $B$, Adaptation frequency
  $U$
  $\mu_0^{\text{emp}} \leftarrow 0$
  $\Sigma_0 \leftarrow I$
  $\Sigma_0^{\text{emp}} \leftarrow I$
  for $t = 0$ to $T$ do
    $\hat{g}(\theta_t) \leftarrow \nabla_\theta \log p(\theta_t \mid M_m(\mathcal{D}))$
    $\epsilon_t \sim \mathcal{N}(0, \Sigma^t)$
    $\theta_{t+1} \leftarrow \theta_t + h(t)\,\Sigma_t\,\hat{g}(\theta_t) + \sqrt{2h(t)\beta^{-1}}\,\epsilon_t$
    $\Sigma_{t+1}^{\text{emp}} \leftarrow \frac{1}{t+1}\left[(\theta^{t+1} - \mu_t)(\theta^{t+1} - \mu_t)^T - \Sigma_t^{\text{emp}}\right]$
    $\mu_{t+1}^{\text{emp}} \leftarrow \mu_t + \frac{1}{t+1}[\theta_{t+1} - \mu_t]$
    if $t > B$ and $t$ is divisible by $U$ then
      $\Sigma_{t+1} \leftarrow \Sigma_{t+1}^{\text{emp}}$
    end if
  end for
```

5.2 MALA

We can slightly modify SGLD to get the *Metropolis Adjusted Langevin Algorithm (MALA)* sampler, which has two differences from SGLD:

1. SGLD uses a minibatch approximation of the gradient (hence the name stochastic), while MALA always uses the entire dataset.
2. MALA has an additional Metropolis accept/reject step on the proposal state, while SGLD always "accepts" the new state.

For the sake of conciseness, we will provide the adaptive MALA algorithm.

Algorithm 6 Adaptive MALA

```
Require: Initial $\boldsymbol{\theta}_0$, Stepsize function $h(t)$, Minibatch size $m$, Adaptation burn-in $B$, Adaptation frequency $U$
  $\mu_0^{\text{emp}} \leftarrow 0$
  $\Sigma_0 \leftarrow I$
  $\Sigma_0^{\text{emp}} \leftarrow I$
  for $t = 0$ to $T$ do
      $\hat{g}(\theta_t) \leftarrow \nabla_\theta \log p(\theta_t \mid \mathcal{D})$
      $\epsilon_t \sim \mathcal{N}(0, \Sigma^t)$
      $P_{t+1} \leftarrow \theta_t + h(t)\,\Sigma_t\,\hat{g}(\theta_t) + \sqrt{2h(t)\beta^{-1}}\,\epsilon_t$
      if $\log p(P_{t+1} \mid \mathcal{D}) \geq \log p(\theta_t \mid \mathcal{D})$ then
          $\theta_{t+1} \leftarrow P_{t+1}$
      else
          $\delta \sim \text{Uniform}[0,1]$
          if $\delta < \log p(P_{t+1} \mid \mathcal{D}) / \log p(\theta_t \mid \mathcal{D})$ then
              $\theta_{t+1} \leftarrow P_{t+1}$
          else
              $\theta_{t+1} \leftarrow \theta_t$
          end if
      end if
      $\Sigma_{t+1}^{\text{emp}} \leftarrow \frac{1}{t+1}\left[(\theta^{t+1} - \mu_t)(\theta^{t+1} - \mu_t)^T - \Sigma_t^{\text{emp}}\right]$
      $\mu_{t+1}^{\text{emp}} \leftarrow \mu_t + \frac{1}{t+1}[\theta_{t+1} - \mu_t]$
      if $t > B$ and $t$ is divisible by $U$ then
          $\Sigma_{t+1} \leftarrow \Sigma_{t+1}^{\text{emp}}$
      end if
  end for
```

5.3 Langevin Numerical Integrators

5.3.1 Euler-Mayurama Method

The Euler-Mayurama integrator models Brownian dynamics/overdamped Langevin dynamics. We update the position vector $\mathbf{q}$ with a single timestep:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + h\mathbf{M}^{-1}\nabla U(\mathbf{q}_k) + \sqrt{2hk_BT}\mathbf{M}^{-1/2}\mathbf{R}_k \tag{103}$$

where $\mathbf{R}_k$ are vectors of standard independent Gaussian $\mathcal{N}(0, I)$ variables, resampled at each step. Since k_B is the Boltzmann constant, we can set $\beta = (k_BT)^{-1}$ to be the **inverse temperature** parameter, reducing the above to

$$\mathbf{q}_{k+1} = \mathbf{q}_k + h\mathbf{M}^{-1}\nabla U(\mathbf{q}_k) + \sqrt{2h\beta^{-1}}\mathbf{M}^{-1/2}\mathbf{R}_k \tag{104}$$

This EM discretized scheme has an invariant measure $\hat{\pi}_h$ that is also an approximation of the true Gibbs measure π of the original Langevin equation. We subscript it with the step size h since convergence will be dependent on h.

$$\hat{\pi}_h(\mathbf{q}) = \pi(\mathbf{q}) + \mathcal{O}(h) \tag{105}$$

We can interpret the $\mathcal{O}(h)$ term as a term $\rho(\mathbf{q})h$ (where ρ is some density) that vanishes linearly as $h \to 0$.

5.3.2 Leimkuhler-Matthews Method

The Leimkuhler-Matthews method also finds discretized solutions to Brownian dynamics, with position update of

$$\mathbf{q}_{k+1} = \mathbf{q}_k + h\mathbf{M}^{-1}\nabla U(\mathbf{q}_k) + \sqrt{2h\beta^{-1}}\mathbf{M}^{-1/2}\left(\frac{\mathbf{R}_k + \mathbf{R}_{k-1}}{2}\right) \tag{106}$$

5.3.3 BAOAB Method

The BAOAB method is a symplectic integrator that models an undampened Langevin flow, with the following steps per timestep:

$$
\begin{aligned}
\mathbf{p}_{k+1/2} &= \mathbf{p}_k - \frac{h}{2}\nabla U(\mathbf{q}_k) \\
\mathbf{q}_{k+1/2} &= \mathbf{q}_k + \frac{h}{2}\mathbf{M}^{-1}\mathbf{p}_{k+1/2} \\
\hat{\mathbf{p}}_{k+1/2} &= e^{-h\gamma}\mathbf{p}_{k+1/2} + \sqrt{\beta^{-1}(1-e^{-2\gamma h})}\mathbf{M}^{1/2}\mathbf{R}_k \\
\mathbf{q}_{k+1} &= \mathbf{q}_{k+1/2} + \frac{h}{2}\mathbf{M}^{-1}\hat{\mathbf{p}}_{k+1/2} \\
\mathbf{p}_{k+1} &= \hat{\mathbf{p}}_{k+1/2} - \frac{h}{2}\nabla U(\mathbf{q}_{k+1})
\end{aligned}
$$

where $\mathbf{R}_k$ are vectors of standard independent Gaussian $\mathcal{N}(0, I)$ variables, resampled at each step. Notice that the O step incorporates the randomness within this integrator. BAOAB is a discretization of an underdamped Langevin flow, but BAOAB with an extremely large γ would be similar to a discretization of an overdamped Langevin flow. There are other BAO splitting schemes, such as OBABO, OABAO, and ABOBA, but BAOAB is the best.

Recall that the true invariant measure of underdamped Langevin equations is $\pi(\mathbf{q}, \mathbf{p}) = \frac{1}{Z}\exp\big(-U(\mathbf{q}) + \frac{1}{2}|\mathbf{p}|^2\big)$. BAOAB is a second-order scheme, meaning that the invariant measure $\hat{\pi}_h$ of this discretized scheme is a second order approximation of π. With some analysis, we can see that $\hat{\pi}$ is of order 2 (in fact, the BAO methods are all of order 2).

$$
\begin{aligned}
\hat{\pi}_h(\mathbf{q}, \mathbf{p}) &= \pi(\mathbf{q}, \mathbf{p}) + Ch^2 f_2(\mathbf{q}, \mathbf{p})\pi(\mathbf{q}, \mathbf{p}) + \mathcal{O}(h^4) \\
&= \pi(\mathbf{q}, \mathbf{p}) + \mathcal{O}(h^2)
\end{aligned}
$$

where f_2 is some function such that $\mathbb{E}_{\mathbf{p}}[f_2] = 0$. But we are typically interested in just $\mathbf{q}$ when looking at the Gibbs density of a system, so we look at the marginal measure of $\mathbf{q}$: $\hat{\pi}_h(\mathbf{q}) = \int_{\mathbf{p}} \hat{\pi}_h(\mathbf{q}, \mathbf{p})\, d\mathbf{p}$, leading us to rewrite the above as

$$\hat{\pi}_h(\mathbf{q}) = \pi(\mathbf{q}) + Ch^2 f_2(\mathbf{q})\pi(\mathbf{q}) + \mathcal{O}(h^4) \tag{107}$$

Furthermore, the constant $C \in \mathcal{O}(1/\gamma)$, where γ is the friction constant seen in the underdamped Langevin equation. This means that as γ increases, C decreases, and so for sufficiently big γ, the second term vanishes and we have an order 4 approximation. Depending on what specific scheme (BAOAB, ABOBA, etc.), the constant C would be different. Therefore, the BAOAB scheme is of order 2 in underdamped dynamics and of order 4 in overdamped dynamics.

5.4 Splitting Methods for Langevin Dynamics

Just like how to split Hamiltonians into components to build symplectic integrators, we can split an SDE (specifically, the undamped Langevin equations) as such

$$\begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{M}^{-1}\mathbf{p} \\ \mathbf{0} \end{pmatrix}}_{A} + \underbrace{\begin{pmatrix} \mathbf{0} \\ -\nabla U(\mathbf{q}) \end{pmatrix}}_{B} + \underbrace{\begin{pmatrix} \mathbf{0} \\ -\gamma\mathbf{p} + \sqrt{2\gamma\beta^{-1}}\mathbf{M}^{1/2}\dot{W} \end{pmatrix}}_{O} \tag{108}$$

where each of the three parts A, B, O may be solved exactly with discretizations given as

$$
\begin{aligned}
\hat{\Phi}_h^A(\mathbf{q}_k, \mathbf{p}_k) &= \big(\mathbf{q}_k + h\mathbf{M}^{-1}\mathbf{p}_k, \mathbf{p}_k\big) \\
\hat{\Phi}_h^B(\mathbf{q}_k, \mathbf{p}_k) &= \big(\mathbf{q}_k, \mathbf{p}_k - h\nabla U(\mathbf{q}_k)\big) \\
\hat{\Phi}_h^O(\mathbf{q}_k, \mathbf{p}_k) &= \big(\mathbf{q}_k, e^{-\gamma h}\mathbf{p}_k + \sqrt{\beta^{-1}(1-e^{-2\gamma h})}\mathbf{M}^{1/2}\mathbf{R}\big)
\end{aligned}
$$

and therefore, composing them with each other gives specific schemes. For example, the ABO scheme is

$$\hat{\Phi}_h^{[ABO]} = \hat{\Phi}_h^O \circ \hat{\Phi}_h^B \circ \hat{\Phi}_h^A \tag{109}$$

We can also symmetrically split these steps down further (must it be symmetric? since we don't have to worry about order of shadow Hamiltonian). For example,

$$\begin{aligned}
\hat{\Phi}_h^{[BABO]} &= \hat{\Phi}_h^O \circ \hat{\Phi}_{h/2}^B \circ \hat{\Phi}_h^A \circ \hat{\Phi}_{h/2}^B \\
\hat{\Phi}_h^{[BAOAB]} &= \hat{\Phi}_{h/2}^B \circ \hat{\Phi}_{h/2}^A \circ \hat{\Phi}_h^O \circ \hat{\Phi}_{h/2}^A \circ \hat{\Phi}_{h/2}^B
\end{aligned}$$

There are much more Langevin integrators that we can construct from the A, B, O blocks.