Molecular dynamics (MD) simulation is a deterministic procedure to integrate the equations of motion based on the classical mechanics principles (Hamiltonian equations). This method was first proposed by Alder and Wainwright (1959) and has become one of the most widely used research tools for complex physical systems. In a typical MD simulation study, one first sets up the quantitative system (model) of interest under a given condition (e.g., fixed number of particles and constant total energy). Then, successive configurations of the system, as a function of time, are generated by following Newton’s laws of motion. After a period of time for “equilibration,” one can start to collect “data” from this computer experiment the data consist of a sequence of snapshots that record the positions and velocities of the particles in the system during a period of time. Based on these records, one can estimate “typical characteristics,” which can often be expressed as the time average of a function of the realized configurations, of the simulated physical system.

Ideally, one would like to generate continuous trajectories (video shots instead of snapshots), but in computer realizations, one is forced to use discrete time steps (a discretization of the Hamiltonian equations). Since the time step used in MD simulations is constrained by the need to conserve the total energy (at least approximately), how to choose a good step size has always been an art in the field. On one hand, the system will evolve very slowly if the time step is too small, whereas on the other hand, the simulation result will be very inaccurate if the time step used in simulation is too large.
Molecular dynamics simulation is a deterministic procedure and has to move on a hyper-surface where a certain Hamiltonian is conserved. In contrast, the goal of a Monte Carlo simulation is to sample "typical configurations" from a Boltzmann distribution which is determined by the system's potential energy and temperature. Because the "time average" will converge to the "configuration average" in a large system (the ergodicity theorem\(^1\)), estimations from MC simulations often correspond very well with those from MD simulations. Although many problems can be treated by both means, it is also the case that in some problems one method is easier to implement than the other.

It is noted in Duane, Kennedy, Pendleton and Roweth (1987) that, besides the ergodicity theorem, there is also a very close technical connection between the MD and the MC simulations. The basic idea behind the new technique they proposed, hybrid Monte Carlo (HMC), is that one can use MD to generate trial moves in a MCMC sampler. Thus, in a certain sense, a bad MD move can be a good MCMC proposal. The advantage of the MD proposal is that the resulting MCMC moves follow the dynamics of the target distribution more closely. As a consequence, much of the randomness in an "unbiased" random-walk proposal is suppressed. Even for a non-physical system, using an artificial Hamiltonian equation to generate MD proposals is very helpful in making good MCMC transitions (Neal 1996). This chapter will illustrate the basic idea behind the HMC and discuss some useful generalizations.

### 9.1 Basics of Newtonian Mechanics

Let \( \mathbf{x}(t) \) denote the \( d \)-dimensional position vector of a body of particles at time \( t \) (e.g., \( d = 3N \) for an \( N \)-particle system in a three-dimensional space). We also assume that there is a \( d \)-dimensional mass vector \( \mathbf{m} = (m_1, \ldots, m_d) \). Let \( \mathbf{v}(t) \equiv \dot{\mathbf{x}}(t) \) denote the speed vector of the particles and let \( \ddot{\mathbf{v}}(t) \) be its acceleration vector. Suppose \( \mathbf{F} \) is the force exerted on the particle; then, the Newton's law of motion states that

\[
\mathbf{F} = m \ddot{\mathbf{v}}(t),
\]

\(^1\)The ergodicity theorem states that the average over a period of time of a function of the system configuration, as the time period goes to infinity, is equal to the average of that function over all configurations weighted by the Boltzmann factor \( \exp(-U(\mathbf{x})/\beta T) \), where \( U(\mathbf{x}) \) is the potential energy of the system and \( T \) is the temperature; that is,

\[
\lim_{t \to \infty} \frac{1}{t} \int_0^t h(\mathbf{x}_s) \, ds = \int h(\mathbf{x}) \exp(-U(\mathbf{x})/\beta T) \, d\mathbf{x}.
\]
where the product between two vectors is assumed to be component-wise:

\[ \mathbf{m} \mathbf{v} = (m_1v_1, \ldots, m_dv_d). \]

Instead of using the velocity vector \( \mathbf{v} \), a more convenient but equivalent variable, the *momentum* vector, defined as

\[ \mathbf{p} = \mathbf{mv} \equiv (m_1v_1, \ldots, m_dv_d), \]

is most frequently seen in classical mechanics. The *kinetic energy* of the system is usually defined as

\[ k(\mathbf{p}) = \frac{1}{2} \sum_{i=1}^{d} m_i v_i^2 = \frac{1}{2} \sum_{i=1}^{d} \frac{p_i^2}{m_i} = \frac{1}{2} \left\| \frac{\mathbf{p}}{\sqrt{\mathbf{m}}} \right\|^2. \]

Here, the ratio between two vectors (e.g., \( \mathbf{p}/\sqrt{\mathbf{m}} \)) is a component-wise operation which results in a new vector \( (p_1/m_1, \ldots, p_d/m_d) \). The *phase space* of a system is defined as the product space of \( \mathbf{x} \) and \( \mathbf{p} \), and vector \( (\mathbf{x}, \mathbf{p}) \) is often referred to as a point in the phase space.

Let \( U(\mathbf{x}) \) be the potential energy field of the system. Then the total energy of the particle system at a given time is

\[ H(\mathbf{x}, \mathbf{p}) = U(\mathbf{x}) + k(\mathbf{p}). \quad (9.1) \]

The law of the conservation of energy says that the total energy remains constant in a closed system. Differentiating both sides of (9.1) with respect to \( t \), we derive the Newton’s law of motion from the law of energy conservation. It is mathematically more convenient to write Newton’s equation in the form of *Hamiltonian equations*:

\[ \dot{\mathbf{x}}(t) = \frac{\partial H(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}} \quad (9.2) \]

\[ \dot{\mathbf{p}}(t) = -\frac{\partial H(\mathbf{x}, \mathbf{p})}{\partial \mathbf{x}} \quad (9.3) \]

Clearly, the first equation describes the definition of \( \mathbf{p} \) and the second equation is essentially Newton’s law. This classical formulation through Hamiltonian equations can be generalized to quantum mechanics, in which the energy function is replaced by a Hamiltonian operator.

### 9.2 Molecular Dynamics Simulation

The main task of MD simulation is the integration of the equations of motion over a given period of time and to study the physical and chemical properties of the system during a particular period (such as the effect of
water in the process of protein folding. Because one can only operate discretely on a computer, the continuous-time equations of motion have to be discretized and a difference method has to be used. By standard Taylor expansion, the Hamiltonian equations can be approximated as

\[
x(t + dt) = x(t) + \frac{p(t)}{m} dt + \frac{\dot{p}(t)}{2m} dt^2 + \cdots, \quad (9.4)
\]

\[
p(t + dt) = p(t) + p(t) dt + \frac{\ddot{p}(t)}{2} dt^2 + \cdots. \quad (9.5)
\]

This type of approximation forms the basis for all finite-difference methods. One of the most widely used algorithms for integrating the equations of motion is the so-called Verlet algorithm (Verlet 1967), which is the simplest and perhaps the best. The Verlet algorithm is based on the observation that

\[
x(t + dt) + x(t - dt) = 2x(t) + \frac{\dot{p}(t)}{m} dt^2 + O(dt^4), \quad (9.6)
\]

\[
x(t + dt) - x(t - dt) = -\frac{\ddot{p}(t)}{m} dt + O(dt^3). \quad (9.7)
\]

For a chosen small time increment \(\Delta t\), Equation (9.6) gives rise to the position update

\[
x(t + \Delta t) = 2x(t) - x(t - \Delta t) - \frac{1}{m} \left. \frac{\partial H}{\partial x} \right|_t (\Delta t)^2,
\]

and Equation (9.7) gives rise to the momentum or, equivalently, the speed update

\[
p(t + \Delta t) = m \frac{x(t + \Delta t) - x(t - \Delta t)}{2\Delta t}.
\]

All the foregoing vector operations involving \(m\) are component-wise.

Another commonly used method for the MD simulation is the leapfrog method (Hockney 1970), which is equivalent to the Verlet algorithm (i.e., giving identical trajectories). The distinctive feature of the leap-frog algorithm is that it updates the momentum variable \(p(t)\) at half-time intervals:

\[
x(t + \Delta t) = x(t) + \Delta t \frac{p(t + \frac{1}{2}\Delta t)}{m}, \quad (9.8)
\]

\[
p \left( t + \frac{1}{2} \Delta t \right) = p \left( t - \frac{1}{2} \Delta t \right) + \frac{\partial H}{\partial x} \left| t \right. \Delta t. \quad (9.9)
\]

Note that \(\partial H / \partial x\) is a function of \(x\) alone and does not depend on the value of \(p\). The momentum at time \(t\) can be computed as \([p(t + \frac{1}{2}\Delta t) + p(t - \frac{1}{2}\Delta t)]/2\) afterward.
To illustrate, we consider a small “ball” with mass 1 in a one-dimensional space, with Hamiltonian \( H(x, p) = U(x) + k(p) \), where

\[
U(x) = x^2 + a^2 - \log[\cosh(ax)], \quad k(p) = p^2 / 2.
\]

For \( a = 1.5 \), the potential function \( U(x) \) is shown in Figure 9.1.

\[\text{FIGURE 9.1. The shape of the potential energy } U(x), \text{ showing that two energetically favorable positions are at } x = a \text{ and } x = -a.\]

We let the initial speed \( p(0) \) be 2.0, 1.5, 1.1, 0.7, and 0.1, respectively, and let \( x(0) = 1 \) for all the five cases. The leap-frog algorithm was performed for 200 time steps with step size \( \Delta t = 0.1 \). Figure 9.2 shows that the total energy in all the cases is not exactly preserved (which is due to the discrete approximation), whereas the trajectories in the phase space are well behaved. As one can see from Figure 9.2(b), the particle travels across the two energy wells when the initial speed \( p(0) \) is large (greater than 1.1), and the particle is “trapped” in the mode it is started from when the initial speed is small.

An important property of the Verlet or leap-frog algorithm is, as in the exact Hamiltonian equations, that the volume is preserved from one step to another; that is, suppose \( V(0) \) is a subset in the phase space \( \mathcal{X} \times \mathcal{P} \). One can define

\[
V(t) = \{(x(t), p(t)) : (x(0), p(0)) \in V(0)\}.
\]

Then, the “volume preservation” property says that

\[
|V(t)| \overset{\text{def}}{=} \int \int_{V(t)} dx dp = \int \int_{V(0)} dx dp = |V(0)|.
\]

For the Hamiltonian equation, this property is called Liouville’s theorem (Arnold 1989). To see why the volume preservation property holds in a
Hamiltonian system, we define the phase flow mapping

\[ g^t : (x(0), p(0)) \mapsto (x(t), p(t)). \]

Then, \( \{g^t : t \in (-\infty, \infty)\} \) is a group of transformations. A simple application of the Taylor expansion in combination with the Hamiltonian equations shows that, for a small \( t \),

\[
g'(x, p) = (x, p) + (\dot{x}, \dot{p})t + O(t^2)
= (x, p) + \left( \frac{\partial H}{\partial p} - \frac{\partial H}{\partial x} \right) t + O(t^2).
\]

Let \( y = (x, p) \) and \( f(y) = \left( \frac{\partial H}{\partial p} - \frac{\partial H}{\partial x} \right) t + O(t^2) \). Then, the Hamiltonian mapping becomes

\[
g'(y) = y + f(y)t + O(t^2).
\]

A key argument in proving Liouville’s theorem is the observation that for small \( t \),

\[
|V(t)| = \int_{V(0)} \det \left( \frac{\partial g'(y)}{\partial y} \right) dx dp
= \int_{V(0)} \det \left( I + \frac{\partial f(y)}{\partial y} t + O(t^2) \right).
\]

By Lemma 2 of Arnold (1989), we have

\[
\det(I + At) = 1 + t \text{ tr}(A) + O(t^2).
\]

Thus,

\[
|V(t)| = \int_{V(0)} \left( 1 + \text{ div} \left( \frac{\partial H}{\partial p}, -\frac{\partial H}{\partial x} \right) t + O(t^2) \right) dx dp
= |V(0)| + O(t^2).
\]
The last equation follows from the fact that the divergence
\[
\text{div} \left( \frac{\partial H}{\partial p} - \frac{\partial H}{\partial \dot{x}} \right) = \frac{\partial}{\partial x} \left( \frac{\partial H}{\partial p} \right) + \frac{\partial}{\partial p} \left( -\frac{\partial H}{\partial \dot{x}} \right) = 0.
\]
Thus, we have shown that \( \lim_{t \to 0} \frac{|V(t)| - |V(0)|}{t} \to 0 \), which implies that the volume as a function of \( t \) is constant.

It is much easier to show that the leap-frog algorithm is volume-preserving as well. From (9.9), the half-step update for the momentum gives rise to a map in the phase space:
\[
(x, p) \mapsto (x, p + f(x) \Delta t),
\]
which results in an identity Jacobian. Therefore, this mapping is volume-preserving. The other half-step update for the position variable \( x \), following (9.8), results in a mapping
\[
(x, p) \mapsto (x + g(p) \Delta t, p),
\]
which is also volume-preserving.

### 9.3 Hybrid Monte Carlo

A major advantage of molecular dynamics simulation in physical systems is its reliance on basic physics principles (e.g. Newton’s equation), which has been shown by nature to work well. Typically, the trajectory of a MD simulation in the position space follows the dynamics of the potential function. As we have seen in Figure 9.2, however, the total energy cannot be exactly preserved in a MD simulation. The fluctuation can be rather substantial if the step size of MD moves is large in comparison with the initial momentum given to the ball. Therefore, a main problem with MD simulation is the stringent requirement of a small time-step size. For example, the protein folding process takes about \( 10^{-3} \) seconds in nature. A proper MD simulation of such a process needs a step size of order \( 10^{-12} \) and will take about \( 10^6 \) days using a current computer. Another potential problem with the MD simulation is its unquantified error due to discretization of the Hamiltonians.

In contrast, in a standard Metropolis-type Monte Carlo simulation, the proposal distribution cannot be easily adapted to “local dynamics” of the target distribution. For example, if the system of interest consists of closely packed particles, a random proposal for moving a particle is most likely rejected because the proposed new position has been partially occupied by others. Or, if the target distribution \( \pi \) is “banana shaped” (or “snake shaped”), a random proposal in the configuration space tends to waste a lot of effort poking around in wrong directions. To overcome some of these difficulties, Duane et al. (1987) introduced the method of hybrid Monte Carlo...
(HMC) which combines the basic idea of MD (i.e., proposing new positions based on Hamiltonian equations) and the Metropolis acceptance-rejection rule to produce Monte Carlo samples from a given target distribution.

Suppose of interest is to draw samples from \( \pi(x) \propto \exp\{-U(x)\} \). Two basic observations are important for the HMC method: (a) If we can simulate \((x, p)\) from the distribution \( \pi(x, p) \propto \exp\{-H(x, p)\} \), then, marginally, \( x \sim \pi \) and \( p \) follows the Gaussian distribution \( \phi(p) \propto \exp(-\|p/\sqrt{m}\|^2/2) \); (b) the Hamiltonian path is “time reversible.” Property (b) needs some more explanation: if we run the leap-frog algorithm, starting from \((x, p)\) for \( t \) steps to reach \((x', p')\), then we can start from \((x', -p')\) and run \( t \) steps of the same algorithm to get back to the starting position but with opposite momentum [i.e., to \((x, -p)\)]. This property is easy to check for the leap-frog algorithm because each half-step can be reversed by negating the momentum. The importance of property (b) will made clearer later in this section.

Duane et al. (1987) introduced a “fictitious” momentum variable conjugate to the configuration variable \( x \) and a guide Hamiltonian

\[
H'(x, p) = U'(x) + k(p),
\]

where \( p \) is an auxiliary variable with the same dimensionality as \( x \), \( k(p) = \sum_{i=1}^{d} \frac{p_i^2}{m_i} \), and the \( m_i \) are positive quantities representing the “masses” of the components. Vector \( p \) plays the role of “momentum variable.” The function \( U'(x) \) is allowed to be different from the target one, \( U(x) \). In the next section, we generalize this formulation to accommodate a larger class of “kinetic energy” functions, \( k(p) \). The guide Hamiltonian is used to generate a proposal state. Another Hamiltonian, the acceptence Hamiltonian, is

\[
H(x, p) = U(x) + k(p),
\]

which is used to decide acceptance and rejection. The reasons that one might use two different Hamiltonians, \( H' \) and \( H \), are two: (i) since one can only run a discrete time process on a computer, the resulting discretization of the “exact” Hamiltonian can be treated as a “guide Hamiltonian” and (ii) one sometimes wants to be flexible in proposing the new position in the phase space. For example, if the original potential energy \( U(x) \) has steep energy barriers, having a smoother or “tempered” potential energy function [e.g., \( U'(x) = U(x)/T \) with \( T > 1 \)] may help the sampler get out of local energy wells.

The HMC algorithm is an iterative procedure and can be implemented as follows. Suppose at time \( t \) we are at position \( x \) (i.e., \( x^{(t)} = x \)) of the configuration space \( \mathcal{X} \) (one needs not record the momentum information). Then, at time \( t+1 \), we do the following:

- generate a new momentum vector \( p \) from the Gaussian distribution [i.e., from \( \phi(p) \propto \exp\{-k(p)\} \)];
run the leap-frog algorithm (or any deterministic time-reversible and volume-preserving algorithm), starting from \((x, p)\), for \(L\) steps, to obtain a new state in the phase space, \((x', p')\);

- accept the proposed state \((x', p')\) (i.e., let \(x^{(t+1)} = x'\)) with probability

\[
\min\{1, \exp(-H(x', p') + H(x, p))\},
\]

and let \(x^{(t+1)} = x\) with the remaining probability.

To see why this algorithm works, we define the leap-frog move at the second step by a mapping \(g^L\) on the phase space:

\[
g^L: (x(0), p(0)) \mapsto (x(L), p(L)).
\]

A heuristic argument (Duane et al. 1987, Neal 1993) goes as follows: (a) If we let \((x', p') = g^L(x, p)\), we have \((x, -p) = g^L(x', -p')\); (b) \(\pi(x, p) = \pi(x, -p)\) for any pair of \((x, p)\); and (c) because of volume preservation, we have \(dx dp = dx' dp'\). These three properties suggest that the proposal is “symmetric,” as required by the Metropolis algorithm, implying that the algorithm is valid.

Here, we provide a more rigorous mathematical proof of invariance. Suppose \(x\) follows the target distribution \(\pi(x)\) at time \(0\); after generating a new \(p\), making the MD transition, and deciding on rejection-acceptance, at time \(1\), we have a new point \(x'\) with density \(f(x')\). To show that \(f(\cdot\cdot\cdot)\) is in fact the same as \(\pi(\cdot\cdot\cdot)\), we will show that for any square integrable function \(h(\cdot\cdot\cdot)\), the equality \(E_h(x) = E_f h(x)\) holds. To proceed, we let \(g^f\) be the Hamiltonian mapping on the phase space [i.e., \((x', p') = g^f(x, p)\)] and let \(g^{-L}\) denote the inverse mapping [i.e., \((x, p) = g^{-L}(x', p')\)]. Then

\[
E_f h(x^*) = \int h(x^*) \left[ \frac{\pi(g^{-L}(x^*, p))}{\pi(g^f(x^*, p))} \min\left\{1, \frac{\pi(x^*, p)}{\pi(g^{-L}(x^*, p))}\right\} |J_{g^{-L}}| + \pi(x^*, p) \left(1 - \min\left\{1, \frac{\pi(g^f(x^*, p))}{\pi(x^*, p)}\right\}\right\} dx' dp
\]

where the Jacobian term \(|J_{g^{-L}}|\) is equal to 1 because of the volume preservation property of the Hamiltonian map \(g^f\) and \(g^{-L}\). The first part of the right-hand side comes from acceptance and the second part comes from rejection.

Continuing with the equation,

\[
E_f h(x^*) = \int h(x^*) \pi(x^*, p) dx'^* dp + \int h(x^+) \pi(x^*, p) dp - \min\{\pi(x^*, p), \pi(g^f(x^*, p))\} dx'^* dp = \int h(x^*) \pi(x^*) dx^*,
\]

(9.10)
The first equality follows from the volume-preserving property of the mapping $g^L$ (so that the Jacobian term disappears). Finally, because of symmetry, we have

$$\pi(x, -p) = \pi(x, p). \quad (9.12)$$

If we let $(x', p') = g^L(x, p)$, then time-reversibility means that

$$g^L(x', -p') = (x, p). \quad (9.13)$$

Combining (9.12) and (9.13), we have

$$\pi(x, -p) = \pi(x, p) = \pi(g^{-L}(x', p')) = \pi(g^L(x', -p')).$$

Thus, the equality between (9.10) and (9.11) follows from

$$\int h(x') \min\{\pi(g^{-L}(x', p')), \pi(x', p')\} dx' dp'$$

$$= \int h(x') \min\{\pi(g^{-L}(x', -p')), \pi(x', -p')\} dx' dp'$$

$$= \int h(x') \min\{\pi(g^{-L}(x', -p')), \pi(x', p')\} dx' dp'$$

$$= \int h(x) \min\{\pi(x, p), g^L(x, p)\} dx dp.$$

At this point, we have proved that $f(x') = \pi(x')$, meaning that the HMC transition leaves the target distribution $\pi(x)$ invariant. Besides sharing the same volume-preserving and time-reversible properties as the guide Hamiltonian, the actual Hamiltonian also preserves the value of $H(x, p)$, thus incurring no rejections. It is also clear from the discussion in Section 9.2 that the leap-frog moves are valid for proposing moves in HMC (i.e., volume-preserving and time-reversible).

### 9.4 Algorithms Related to HMC

#### 9.4.1 Langevin-Euler moves

As in the previous subsection, we let $\pi(x) \propto \exp\{-U(x)\}$ be the target distribution of interest. The Langevin diffusion refers to the following result: The solution of the stochastic differential equation

$$dx_t = -\frac{1}{2} \frac{\partial U(x_t)}{\partial x} dt + dW_t,$$

where $W_t$ the standard Brownian motion, follows the target distribution $\pi$. Thus, a discretization of the equation is of the form

$$x_{t+1} = x_t - \frac{1}{2} \frac{\partial U(x_t)}{\partial x} h + \sqrt{h} Z_t,$$
where \( Z_t \) follows a standard Gaussian distribution. It we let \( \sqrt{\kappa} = dt \) in Equation (9.4) and let \( \mathbf{p}_t \) be a refreshed momentum drawn from the Gaussian distribution \( \phi(\mathbf{p}) \), the Langevin update is exactly the same as the second order Taylor expansion of the Newton’s law. Thus, a Langevin update is equivalent to a single-step hybrid Monte Carlo move Neal (1993).

### 9.4.2 Generalized hybrid Monte Carlo

Suppose we can augment the sample space of \( \mathbf{x} \) to include a “momentum” variable, say \( \mathbf{p} \). So now the pseudo-phase space consists of \( \phi = (\mathbf{x}, \mathbf{p}) \) and the target distribution \( \pi(\mathbf{x}) \) is augmented to \( \pi(\phi) \). Furthermore, suppose we can find an irreducible transition rule on the phase space, \( T(\phi, \phi') \), so that the detail balance \( \pi(\phi')T(\phi, \phi') = \pi(\phi)T(\phi', \phi) \) is maintained for some non-negative function \( \pi^*(\phi) \) (unique up to a normalizing constant). Let the current position in the original space be \( \mathbf{x}^{(0)} \). Then we can supplement it with a momentum realization \( \mathbf{p}^{(0)} \) in a suitable way and then evolve the phase space point \( \phi^{(0)} = (\mathbf{x}^{(0)}, \mathbf{p}^{(0)}) \) according to the surrogate transition \( T(\cdot, \cdot) \). After \( k \) steps, we accept the new point with probability

\[
\min \left\{ 1, \frac{\pi(\phi^{(1)})/\pi^*(\phi^{(1)})}{\pi(\phi^{(0)})/\pi^*(\phi^{(0)})} \right\}.
\]

In hybrid Monte Carlo, one augment the original target distribution \( \pi(\mathbf{x}) \) to a distribution defined on the phase space; that is, \( \pi(\phi) = \pi(\mathbf{x}) \times f(\mathbf{p}) \), where \( \log f(\mathbf{p}) = k(\mathbf{p}) = -||\mathbf{p}/\sqrt{\mathbf{m}}||^2/2 \). The leap-frog algorithm leads to a uniform \( \pi^* \). From the proof in the previous section and the construction of the leap-frog moves, we see that the kinetic energy function \( k(\mathbf{p}) \) can be rather arbitrary. The basic requirements for \( k(\mathbf{p}) \) are that (a) it is symmetric in \( \mathbf{p} \) [i.e., \( k(\mathbf{p}) = k(-\mathbf{p}) \)] and (b) it is bounded from below. In this case, the leap-frog algorithm becomes

\[
\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \frac{\partial H}{\partial \mathbf{p}} \bigg|_{t+\phi},
\]

\[
\mathbf{p} \left( t + \frac{\Delta t}{2} \right) = \mathbf{p} \left( t - \frac{\Delta t}{2} \right) + \Delta t \frac{\partial H}{\partial \mathbf{x}} \bigg|_{t+\phi}.
\]

By the same argument as in Section 9.2, we can clearly see that this generalized leap-frog move is volume-preserving. To see why it is time reversible when the kinetic energy function \( k(\mathbf{p}) \) is symmetric, we let \( \mathbf{x'}(t) = \mathbf{x}(t+\Delta t) \), \( \mathbf{p'}(t - \Delta t/2) = -\mathbf{p}(t + \Delta t/2) \), and apply one leap-frog step. Then, we will end at position \( \mathbf{x}(t) \) with the negated momentum \( -\mathbf{p}(t - \Delta t/2) \). It is not clear, though, how to choose an effective kinetic energy function.

An interesting special case of the generalized hybrid Monte Carlo (GHMC) for the one-dimensional sampler was proposed by Gustafson (1998). First, he augmented the original state space \( x \) to \( (x, P) \) where \( P \) is either \(-1\) or
1 and has a marginal uniform distribution. Suppose that currently we are at \((x^{(t)}, p^{(t)})\); then, the proposed new configuration is \(y = x^{(t)} + p^{(t)}|Z|\), where \(Z \sim N(0, \sigma^2)\), and let

\[
(x^{(t+1)}, p^{(t+1)}) = \begin{cases} 
(y, p^{(t)}) & \text{with probability } r \\
(x^{(t)}, -p^{(t)}) & \text{with probability } 1 - r,
\end{cases}
\]

where \(r\) is the usual Metropolis ratio. It is very easy to generalize the algorithm to higher-dimensional cases: The auxiliary variable \(P\) is a uniformly distributed random direction in a \(d\)-dimensional space and \(Z = \sqrt{X^2}\), where \(X^2\) has a \(\chi^2\)-distribution with \(d\) degrees of freedom. Although effective for one-dimensional problems, this method may not be useful in multidimensional cases. The reason is that negating direction \(P\) in a high-dimensional space typically will not lead to a better search direction.

### 9.4.3 Surrogate transition method

In some Monte Carlo simulation problems (e.g., simulation of polarized liquid water), evaluation of the energy function \(h(x) = -\log \pi(x)\) involves expensive computation (such as inverting a 2000 \(\times\) 2000 matrix to compute the polarization vector). It is often very inexpensive, however, to obtain a reasonably good approximation \(h^*(x)\) of \(h(x)\). For example, instead of solving a large linear equation completely, one can opt to perform a few rounds of iterative updates. Then, the Metropolis algorithm needs to be adjusted to accommodate this variation.

Mathematically, we assume that one can conduct a reversible Markov transition \(S(x, y)\) (surrogate) which leaves \(\pi^*(x) \propto \exp(-h^*(x))\) invariant; that is, the detailed balance

\[
\pi^*(x) S(x, y) = \pi^*(y) S(y, x)
\]

is satisfied. A valid surrogate transition can be devised by making use of the Metropolis principle on \(\pi^*(x)\).

Suppose our current sample is \(x^{(t)}\). We let \(y_0 = x^{(t)}\) and recursively sample \(y_i \sim S(y_{i-1}, \cdot)\) for \(i = 1, \ldots, k\). Then, we update \(x^{(t+1)} = y_k\) with probability

\[
\min \left\{ 1, \frac{\pi(y_k)/\pi^*(y_k)}{\pi(x^{(t)})/\pi^*(x^{(t)})} \right\}
\]

and let \(x^{(t+1)} = x^{(t)}\) with the remaining probability.

To show that the foregoing procedure is valid, we see that the proposal transition function from \(y_0\) to \(y_k\) can be formally written as

\[
S^{(k)}(y_0, y_k) = \int \cdots \int S(y_0, y_1) \cdots S(y_{k-1}, y_k) dy_1 \cdots dy_{k-1}.
\]
In words, \( S^{(k)}(\cdot, \cdot) \) is the \( k \)-step transition function for the surrogate Markov chain defined by \( S \). It is easily seen that \( \pi^*(x)S^{(k)}(x, y) = \pi^*(y)S^{(k)}(y, x) \). Thus, the actual transition function from \( x(t) = x \) to \( x(t+1) = y \neq x \) has the form

\[
A(x, y) = S^{(k)}(x, y) \min \left\{ 1, \frac{\pi(y)/\pi^*(y)}{\pi(x)/\pi^*(x)} \right\}.
\]

Hence,

\[
\pi(x)A(x, y) = \pi^*(x)S^{(k)}(x, y) \min \left\{ \frac{\pi(x)}{\pi^*(x)}, \frac{\pi(y)}{\pi^*(y)} \right\}
\]

\[
= \pi^*(y)S^{(k)}(y, x) \min \left\{ \frac{\pi(x)}{\pi^*(x)}, \frac{\pi(y)}{\pi^*(y)} \right\}
\]

\[
= \pi(y)A(y, x),
\]

which is the detailed balance condition. A surrogate transition procedure with \( k = 1 \) is given by Liu and Chen (1998) under the sequential importance sampling framework. We find that the surrogate procedure can be more generally applicable in Monte Carlo simulation of complicated systems. At a conceptual level, the HMC can be seen as a surrogate transition method in which a discretized Hamiltonian is used as a surrogate to guide for the dynamical moves in the phase space (joint space of positions and momentums). Because of the time reversibility and volume preservation properties, the corresponding \( \pi^* \) in HMC is the uniform distribution.

9.5 Multipoint Strategies for Hybrid Monte Carlo

9.5.1 Neal’s window method

In a standard HMC algorithm, each Monte Carlo update involves \( L \) (often between 40 and 70) steps of deterministic Hamiltonian leap-frog moves. The acceptance-rejection decision is made, however, based only on the “energy” comparison between the starting and the ending configurations of the leap-frog trail. Neal (1994) suggested that some middle steps in the trail can also be used in order to increase the acceptance rate in HMC. Suppose the current position vector is \( x^{(0)} \) and we generate a renewed momentum vector \( p^{(0)} \) from its Gaussian distribution. Then, the “window algorithm” can be stated as follows:

- Choose a window size \( W < L \) (either deterministically or from a fixed distribution).
- Draw \( K \) from \( \{0, 1, \ldots, W - 1\} \) uniformly.
Starting from the current point in the phase space $\phi(0) = (x^{(0)}, p^{(0)})$, run the leap-frog steps backward for $K$ steps and forward for $L - K$ steps to result in a trajectory

$$\phi(-K), \ldots, \phi(-1), \phi(0), \phi(1), \ldots, \phi(L - K),$$

- Place the "acceptance window" of size $W$ at the end of the trajectory:

$$A = \{\phi(L - K - W + 1), \ldots, \phi(L - K)\};$$

and the "rejection window" at the beginning

$$R = \{\phi(-K), \ldots, \phi(-K + W - 1)\}.$$

- Define the "free energy" of a window $W$ as

$$F(W) = -\log \left( \sum_{\phi(j) \in W} \exp \{-H(\phi(j))\} \right).$$

Go to the acceptance window $A$ with probability

$$\min \{1, \exp \{F(A) - F(R)\}\};$$

and stay in the rejection window $R$ with the remaining probability.

- Having decided on the window $W$, a particular state $\phi = \phi(j)$ within that window is selected according to probability

$$P(\phi(j)) = \exp \{-H(\phi(j)) + F(W)\}.$$

The new state is updated as $(x^{(t+1)}, p^{(t+1)}) = \phi$.

A graphical illustration is given in Figure 9.3. (Neal 1994) showed by examples that this approach, not surprisingly, can improve the acceptance rate of the HMC algorithm. However, it is less clear if the window method actually improves the computational efficiency of the algorithm (i.e., resulting in a more rapidly mixing Markov chain).
9.5.2 Multipoint method

The multipoint Metropolis method described in Section 5.5 allows one to choose a good candidate among multiple random proposals. Its basic principle can be further extended to accommodate the multiple leap-frog steps in the HMC algorithm.

Similar to the description of the window method in the previous section, we define \( \phi(0) = (x_0^{(t)}, p_0^{(t)}) \), where \( p_0^{(t)} \) is a “renewed” momentum vector, and conduct the following procedures:

**Multipoint HMC**

- From the starting states, \( \phi(0) \), run \( L \) leap-frog iterations to obtain \( \phi(1), \ldots, \phi(L) \).

- Select one candidate \( \phi' \) from the last \( M \) configurations [i.e., \( \phi' \) is chosen from \( \phi(L - M + 1), \ldots, \phi(L) \)] according to their respective Boltzmann probabilities; that is,

\[
\Pr(\phi' = \phi(L - M + k)) \propto w_k \exp\{-H(\phi(L - M + k))\},
\]

for \( k = 1, \ldots, M \). Suppose we have selected \( \phi' = \phi(L - K) \).

- Run \( K \) reversed leap-frog steps from \( \phi(0) \) (by using the negated momentum) to get \( \phi(-1), \ldots, \phi(-K) \).

- Let \( (x, p) = \phi' \) with probability

\[
p = \min \left\{ 1, \frac{\sum_{j=1}^{K} w_j \exp\{-H(\phi(L - K - j))\}}{\sum_{j=1}^{K} w_j \exp\{-H(\phi(L - M + j))\}} \right\},
\]

and let \( (x, p) = \phi(0) \) with probability \( 1 - p \).

- Let \( x^{(t+1)} = x \) and sample (renew) \( p^{(t+1)} \) from \( N(0, \Sigma) \), where \( \Sigma = \text{diag}(m_1^{-1}, \ldots, m_2^{-1}) \).

The \( w_j \) in the algorithm are non-negative numbers are completely controlled by the user. The role of this weighting vector to give a prior preference for certain points on the leap-frog trajectory. For example, one may wish to emphasize those points that are farther to the end of the leap-frog trajectory more. In this case, we may choose \( w_j \propto \sqrt{j} \). A graphical view of the multipoint method is given in Figure 9.4. One can see that the multipoint approach is very similar to the window method, but is more flexible and more efficient in general (Qin and Liu 2000). The correctness of this algorithm can be shown by combining a similar argument in validating the general multipoint approach with the volume preservation and time reversibility properties of the leap-frog moves.
9.6 Application of HMC in Statistics

Using the HMC to solve statistical inference problems was first introduced by Neal (1996). This effort was only 10 years behind that in physics and theoretical chemistry. In contrast, statisticians were 40 years late in using the Metropolis algorithm. The connection between statistical problems, especially Bayesian inference problems, and physics problems is, in fact, quite simple: Since all unknowns in a probabilistic model, be they tuning parameters, missing observations, latent structures, or observed data, can be treated as joint random variables in a Bayesian framework, all the inference tasks can be reduced to the evaluation of certain expectation with respect to the posterior distribution of unknown variables. This target posterior distribution can always be written out explicitly, up to a normalizing constant, as

$$\pi(\theta) \propto f(y | \theta) \pi_0(\theta) \equiv c \exp\{-U(\theta)\},$$

where $f(\cdot)$ is the probabilistic model that connects data with unknown parameters, $\pi_0(\cdot)$ is the prior distribution of $\theta$, and

$$U(\theta) = -\log f(y | \theta) - \log \pi_0(\theta).$$

In order to use the HMC to sample from this posterior distribution, we need to introduce an auxiliary “momentum” variable $p$ and construct the “guide Hamiltonian” $H(\theta, p) = U(\theta) + k(p)$.

Although it is conventional to let

$$k(p) = \sum_{i=1}^{d} \frac{p_i^2}{2m_i},$$

this pseudo-energy function can be chosen more flexibly (Section 9.4.2). Even with the conventional choice of $k(p)$, a tricky question is how to choose appropriate $m_i$’s so as to make the algorithm more efficient. Intuitively, $m_i$ is the “mass” of the $i$th component. Thus, the larger an $m_i$ is,
the “slower” that component moves. The efficiency of HMC can be improved by setting $m_i$ differently for each $p_i$ according to the properties of component $x_i$. A related important issue is the step-size choice for the leap-frog moves in HMC. In the following subsections, we describe a few statistical inference problems in which the HMC has an obvious edge over other MCMC strategies. We hope that these expository descriptions will motivate other researchers to study theoretical properties of HMC and to investigate strategies for tuning the algorithm.

### 9.6.1 Indirect observation model

Consider the following statistical inference problem. Let $\theta$ be a parameter vector in some parameter space $\Omega$ and let $x(\theta)$ be a random vector whose distribution is known completely if $\theta$ is given. Suppose, however, that we cannot directly observe $x$, but observe, instead, a vector $y$ whose relationship with $x$ and $\theta$ can be described as

$$y = g(x(\theta), \theta), \quad (9.15)$$

where the functional form of $g(\cdot)$ is known. Of interest is the Bayesian inference on the parameter vector $\theta$. We further assume that inverting the function $g$ is difficult; thus, obtaining the likelihood function of $\theta$ is infeasible. The standard approach in literature is the simulated method of moment (McFadden 1989), but it may be inefficient when the moment functionals are not chosen properly. We take a likelihood-based approach here.

To resolve the difficulty that the likelihood function cannot be evaluated easily, we modify model (9.15) by introducing a Gaussian noise and pretend that $y$ is drawn from the modified model

$$y = g(x(\theta), \theta) + \epsilon \quad (9.16)$$

where $\epsilon \sim N(0, \sigma^2 I)$, $I$ is the identity matrix, and $\sigma$ is a tuning parameter controlled by the user. Therefore, under model (9.16), the joint posterior distribution of $x$ and $\theta$ can be derived as follows:

$$\pi_\sigma(x, \theta \mid y) \propto f_\sigma(y \mid x, \theta; \sigma^2) f(x \mid \theta) \pi_0(\theta), \quad (9.17)$$

where $\pi_0$ is the prior for $\theta$. When the data vector $(x, y)$ can be decomposed as $(x_1, y_1), \ldots, (x_n, y_n)$ and they are i.i.d. given $\theta$, then we have

$$\pi_\sigma(x, \theta \mid y) \propto \prod_{i=1}^n [f_\sigma(y_i \mid x_i, \theta; \sigma^2)] f(x_i \mid \theta) \pi_0(\theta),$$

where $f_\sigma$ stands for the density function of the modified model (9.16). The posterior distribution $\pi_\sigma(\theta \mid y)$ is then a marginal of $\pi_\sigma(x, \theta \mid y)$. It can
be shown that under mild conditions, \( \pi_\sigma(\theta \mid y) \) converges to the posterior density \( \pi(\theta \mid y) \) almost surely as \( \sigma \to 0 \).

Our strategy (Chen, Qin and Liu 2000) for the estimation in indirect observation model is to choose a sequence of \( \sigma \)'s:

\[
\sigma_1 > \sigma_2 > \cdots > \sigma_l.
\]

and generate Monte Carlo samples from all the \( \pi_{\sigma_j} \), \( j = 1, \ldots, l \). If we are interested in the posterior mean \( \hat{\theta} \) of \( \theta \), for example, we compute the posterior means \( \hat{\theta}_j \) under the \( j \)th modified distribution and then fit a quadratic function

\[
\hat{\theta}_j = \beta_0 + \beta_1 \sigma_j + \beta_2 \sigma_j^2 + e_j.
\]

The estimate \( \hat{\beta}_0 \) of \( \beta_0 \) then corresponds to the unmodified model and serves as our final estimate of \( \hat{\theta} \).

A technical difficulty remains in our approach: Monte Carlo sampling from (9.17) is generally difficult to do. In particular, as \( \sigma \to 0 \), the posterior distribution of \( \pi_\sigma \) lives on an almost degenerate subspace, making the random-walk-type Metropolis algorithm very inefficient. For example, a trivial example is

\[
y = \theta x, \quad x \sim N(\theta, 1).
\]

Under the modified model, we have

\[
\pi_\sigma(x, \theta) \propto \exp \left\{ \frac{(y - x\theta)^2}{2\sigma^2} \right\} \pi_0(\theta).
\]

When \( \sigma \) is small, the sample is forced to lie around the curve \( x\theta = y \). See the contour plots in Figure 9.5. Thus, a random directional proposal will almost surely fail. To overcome this computational difficulty, we can apply the HMC method to draw samples from each \( \pi_{\sigma_j} \).

Having to simulate from multiple related systems is also a common problem in physics and chemistry where it is often of interest to estimate a certain system property at different temperatures. Several “tempering” strategies (Chapter 10) have been developed in the physics and statistics literature for improving MCMC efficiencies in these problems. We found that the parallel tempering method (Geyer 1991) is especially useful for our problem. The basic idea of parallel tempering is to run the \( l \) sampling processes (each corresponding to a \( \pi_{\sigma_i} \)) in parallel and to allow the sampler to switch between different configurations corresponding to different distributions.

More precisely, suppose \( \varphi_i \) and \( \varphi_j \) are the current states for two HMC chains corresponding to \( \pi_{\sigma_i} \) and \( \pi_{\sigma_j} \) (\( i \neq j \)). An exchange of the two
configurations is proposed with a small probability $\delta$. If the exchange move is proposed, it is accepted with probability

$$
\min \left\{ 1, \frac{\exp\left(-H(\phi_0; \sigma_0^2) - H(\phi_1; \sigma_1^2)\right)}{\exp\left(-H(\phi_1; \sigma_1^2) - H(\phi_0; \sigma_0^2)\right)} \right\}.
$$

(9.19)

Since the detailed balanced condition is never violated, each HMC chain converges to the stationary distribution with its particular value of $\sigma$.

### 9.6.2 Estimation in the stochastic volatility model

The stochastic volatility (SV) model is a nonlinear state-space model (Sections 1.6, 3.3, and 4.5) and can be considered as a generalization of the celebrated Black-Scholes formula (Hull and White 1987). A simple stochastic model has the form

$$
y_t = \epsilon_t \beta \exp(x_t / 2), \quad x_{t+1} = \phi x_t + \eta_t, \quad t = 1, \ldots, T,
$$

(9.20)

where $\epsilon_t \sim N(0, 1)$ and $\eta_t \sim N(0, \sigma^2)$. One can see that the log of $\text{var}(y_t)$ in (9.20) follows an AR(1) process. Because of its nonlinear nature, the model parameters are difficult to estimate. Shephard and Pitt (1997) suggested a way to use the Gibbs sampler to obtain Bayes estimates. They note, however, that the usual Gibbs sampler converges extremely slowly. They
developed an improved MCMC algorithm based on conditional sampling of a block of variables (Liu and Sabatti 2000). We recently reported some promising results for using a HMC-based algorithm to compute the Bayes estimates in a SV model (Chen, Qin and Liu 2000). Our dataset consists of daily exchange rates of pound/dollar from 10/1/1981 to 6/28/1985 (a total of $T=946$ observations). Let $r_t$ be the daily exchange rate and let $dr_t = \log r_{t+1} - \log r_t$. Since the mean of $y_t$ is zero in the SV model, we define $y_t$ as
\[
y_t = 100 \left( dr_t - \sum dr_t / T \right),
\]
for $t = 1, \ldots, T$ and fit a SV model (9.20) with these $y_t$.

Let $\mathbf{x} = (x_1, \ldots, x_T)$ and $\mathbf{y} = (y_1, \ldots, y_T)$, and let the prior for $\beta$ be $p(\beta^2) \propto \beta^{-2}$ (improper); for $\sigma^2$, Inv-$\chi^2(10, 0.05)$; and for $(\phi + 1)/2$, a beta prior with shape parameters 20 and 1.5. Then the following conditional distributions can be easily sampled from:
\[
\beta^2 | \mathbf{y}, \mathbf{x} \sim \text{Inv-}\chi^2 \left( T, \frac{1}{T} \frac{y_t^2}{\exp(x_t)} \right),
\]
\[
\sigma^2 | \phi, \mathbf{x} \sim \text{Inv-}\chi^2(T + 10, V),
\]
\[
\pi(\phi | \sigma^2, \mathbf{x}) \propto \exp \left\{ -\frac{x_t^2(1 - \phi^2) + \sum_{t=2}^{T} (x_t - \phi x_{t-1})^2}{2\sigma^2} \right\} (1 + \phi)^{0.5} (1 - \phi),
\]
where
\[
V = \frac{1}{T + 10} \left[ 0.5 + \frac{x_t^2(1 - \phi^2) + \sum_{t=2}^{T} (x_t - \phi x_{t-1})^2}{2\sigma^2} \right].
\]

Once the parameter values are given, the negative log-density is
\[
U(\mathbf{x}) = \sum_{t=1}^{T} \left\{ \frac{x_t^2}{2} + \frac{y_t^2}{2\beta^2 \exp(x_t)} \right\} + \frac{x_t^2(1 - \phi^2)}{2\sigma^2} + \sum_{t=1}^{T} \frac{(x_{t+1} - \phi x_t)^2}{2\sigma^2}.
\]
The posterior density of $\mathbf{x}$, given the parameter values, is proportional to $\exp(-U(\mathbf{x}))$.

We implemented the following iterative sampling algorithm: Given $\mathbf{x}$, we drew the parameters $\beta, \phi,$ and $\sigma^2$ from the above conditional distributions; whereas given the realized values of the parameters, we drew the state variable $\mathbf{x}$ by the HMC. The step sizes for the HMC moves of $x_2, \ldots, x_T$ were chosen as 0.03 and that for $x_1$ was 0.06.

This Gibbs-HMC method was run for 28,000 iterations and the results from the last 20,000 iterations are reported in Figure 9.6 and Table 9.1. It can be seen from Figure 9.6 that the efficiency of the Gibbs-HMC algorithm is comparable to that of the “grouping” method in Shephard and
Pitt (1997). A relative advantage of the HMC approach is that no special Gaussian approximations of certain likelihood function is needed, which makes our method more automatic.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>Standard Error</th>
<th>Covariance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta$</td>
<td>0.6433</td>
<td>0.0058</td>
<td>9.1e−03</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.1481</td>
<td>0.0273</td>
<td>-2.16e−04</td>
</tr>
<tr>
<td>$\phi$</td>
<td>0.9801</td>
<td>0.0003</td>
<td>1.48e−04</td>
</tr>
</tbody>
</table>

TABLE 9.1. Bayes estimates of the parameters in the stochastic volatility model.