Simulating Canonical Ensemble with Hamiltonian Monte-Carlo Method

Sayema Lubis

May 2025

1 Introduction

This paper aims to explore the successes and limitations of the Hamiltonian Monte Carlo (HMC), which has made its path into modern statistical computing. The story begins by exploring the geometry of high-dimensional probability distributions and how it presents challenges for efficient statistical computation. We then examine a common statistical computation method, Markov Chain Monte Carlo (MCMC), highlighting its essential features and limitations. This leads us to HMC and how it addresses the challenges of MCMC, while also highlighting its broad applications and efficiency.

2 Limitations of Traditional MCMC Methods

As background, we first consider the Markov Chain Monte Carlo (MCMC) algorithm, which uses Markov chains to sample from a target probability distribution through stochastic processes [3]. The basic idea is to construct a Markov chain whose steady-state (or equilibrium) distribution approximates the desired target distribution. As the Markov chain evolves, the samples generated along its trajectory provide representative draws from the target distribution.

One of the earliest MCMC algorithms is the Metropolis-Hastings algorithm, which only requires a function proportional to the target probability density. This algorithm uses a proposal distribution $g(x)$ to suggest a new state and an acceptance distribution,

$$A(y|x) = min(1, \frac{f(y)g(x|y)}{f(x)g(y|x)})$$

to decide whether to move to the proposed state. The original Markov chain Monte Carlo algorithm and what is commonly used today uses the Gaussian distribution as its proposal mechanism[1, pg. 15],

$$g(y|x) = \mathcal{N}(y|x, \Sigma)$$

The algorithm iteratively proposes a new state, accepts or rejects it, and repeats this process to generate samples. Often, a number of initial samples are discarded due to correlation between sequential samples. Due to its symmetry under the exchange of the initial and proposed points, the proposal density cancels in the acceptance probability, simplifying it to

$$A(y|x) = min(1, \frac{f(y)}{f(x)})$$

2.1 Inefficiencies in High Dimensions

A characteristic property of high-dimensional spaces is that there is significantly more volume outside any given neighborhood than inside of it [1, pg. 6]. We can simply visualize this fact by considering a partitioned parameter space, whose partitions are rectangular boxes centered around the mode. Our claim is that as we progressively increase the dimension, we expect the volume in the center partition becomes negligible

compared to the neighboring partitions. As shown in Figure 1, a 1-D representation shows that there are two neighboring partitions. Then, in 2-D we increase to eight neighboring partitions, while 3-D has 26. As we continue with the pattern, we can conclude that our claim remains true.

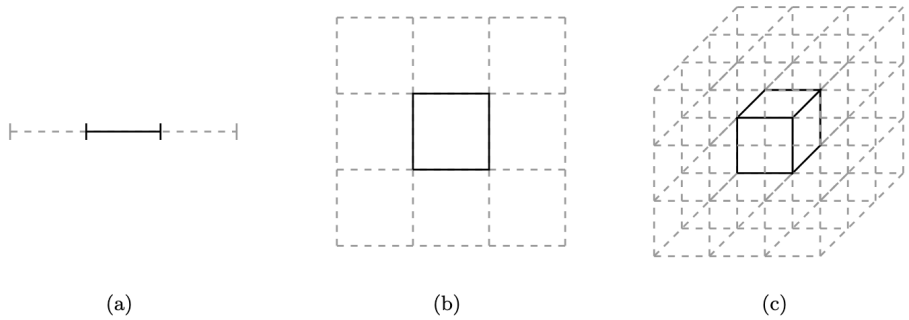


Figure 1: *A rectangular partitioning centered around a distinguished point, such as a mode, under increasing dimensions. (a) The relative weight in one dimension is 1/3 (b) in two dimensions, 1/9 and finally, (c) in three dimensions, 1/27. We can see that the center partition becomes quickly becomes negligible as we go to higher dimensions. Figure taken from [1, pg. 6]*

In high-dimensional spaces, the significant contributions to expectation values come from a narrow region called the *typical set*, which balances high probability density with sufficient volume [1, pg. 7]. Exploring this typical set efficiently is crucial for statistical computing. Simple methods like the Metropolis-Hastings struggle because the volume outside the typical set overwhelms the volume inside, leading most random proposals to fall into low-density regions and be rejected.

Figure 2: *The green region indicates the Metropolis-Hastings proposal density, which we can see is strongly biased towards the outside of the typical set, hence the Metropolis acceptance probability vanishes. (a) The proposals stray too far away from the typical set if the proposal variances are large, ultimately getting rejected. (b) If the proposal variances are smaller, then it stays within the typical set, however the resulting transition density concentrates tightly around the initial point. We end up with a Markov chain that explores the typical set in an extremely slow manner. Figure taken from [1, pg. 17]*

The Metropolis-Hastings algorithm is intuitive because the proposal distribution tends to favor regions with larger volume, which often correspond to the tails of the target distribution [1, pg. 16]. However, the Metropolis correction step filters out proposals that land in areas of very low density. Together, this process biases the sampling toward regions with large volumes, effectively focusing on the typical set as intended. Consequently, it is ineffective in higher dimensions for the very reason that the typical set gets overwhelmed by the volume outside of itself.

As we move to higher dimensional spaces, the typical set sits in a narrow region. But random proposals from the Metropolis-Hastings algorithm often land outside of this region, in the low-probability tails. These probabilities almost always get rejected, and thus the chain barely moves. We could make smaller proposals to stay within the typical set and get more accepted steps, but then the chain moves very slowly, taking more time to explore the space [1, pg. 17].

2.2 Pathological Behavior

Problems arise when the target distribution exhibits "pathological behavior" [1, pg. 13]. One key issue mentioned is when the typical set pinches into regions of high curvature. Most basic Markov transitions are not designed to "resolve these details" or "maneuver into these tight regions" [1, pg. 13]. As a result, the Markov chains may simply ignore these pathological regions, leading to incomplete exploration. Since the Markov chain must asymptotically recover the exact expectations, it tries to compensate for missing these regions. This typically manifests as the chain getting "stuck" near the boundary of the pathological region.

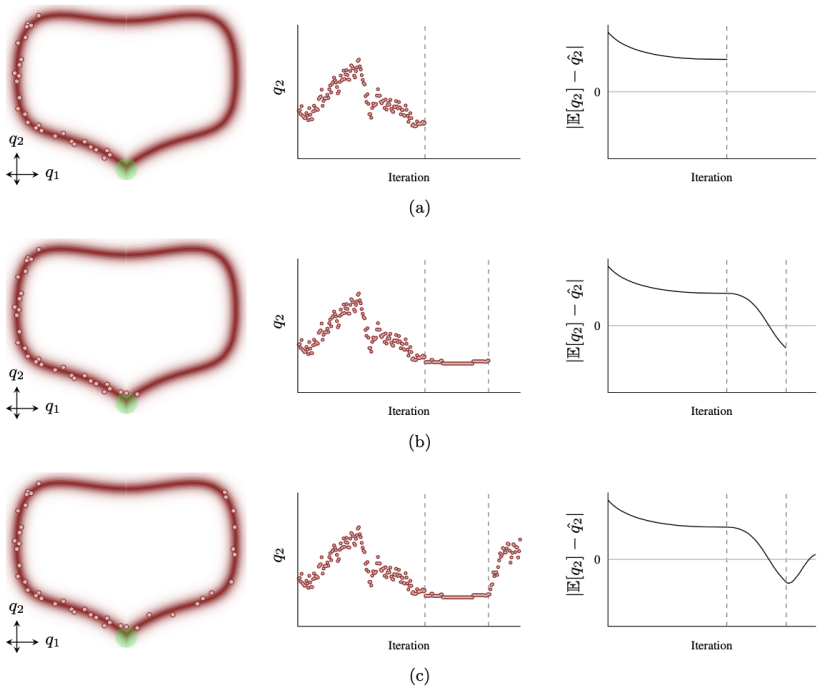


Figure 3: *Pathological regions usually cause Markov chains to get "stuck". (a) The Markov chain starts by exploring well-behaved regions of the typical set, avoiding the pathological neighborhood entirely which biases the Markov chain Monte Carlo estimators. (b) Giving the Markov chain time to approach the boundary of the pathological region, it will slowly correct the Markov chain Monte Carlo estimators. (c) It will eventually explore the rest of the typical set. This process repeats, causing the resulting estimators to oscillate around the true expectations, and inducing strong biases. Figure taken from [1, pg. 14]*

While hovering near this boundary, the estimators are drawn towards the values from the pathological region, sometimes even overcorrecting and inducing bias in the opposite direction before eventually escaping. This process repeats, causing estimators to strongly oscillate around the true expectations. While these oscillations average out asymptotically, stopping the chain after finite time (which is always the case in practice) almost always destroys this balance, resulting in substantial biases. We can visually see this process in Figure 3.

3 Hamiltonian Monte Carlo: A Physics-Inspired Alternative

Hamiltonian Monte Carlo (HMC) is a Markov chain Monte Carlo (MCMC) method that was first proposed by Simon Duane, Anthony Kennedy, Brian Pendelton and Duncan Roweth in 1987. Their paper, titled "Hybrid Monte Carlo," which was its original name, described a new way to conduct a Monte Carlo simulation by introducing a fictitious time and Hamiltonian dynamics [2, p. 217]. Furthermore, the Hamiltonian plays two important roles [2, p. 220]. First, the acceptance Hamiltonian H defines the equilibrium distribution and

enters the expression for the acceptance probability, which depends on the change,

$$\delta H = H(O') - H(O)$$

where O is the original configuration, and O' is the proposed new configuration. Second, the guidance Hamiltonian H' governs the equations of motion that generate proposal moves. Importantly, the algorithm does not require H and H' to be identical; allowing $H \neq H'$ introduces flexibility that can be exploited to optimize the acceptance rate. As long as the acceptance probability is correctly based on the change in H, the algorithm ensures sampling from the desired distribution.

3.1 Key Advantages Over MCMC

Efficient Exploration of the Typical Set HMC is designed to exploit the geometry of the probability distribution it is sampling from [1, pg. 16]. Unlike diffusive methods like Random Walk Metropolis which concentrate around the initial point and linger in small neighborhoods, HMC generates trajectories that "coherently glide through the typical set" by following contours of high probability mass [1, pg. 17-18]. The Hamiltonian dynamics generate trajectories that rapidly move through phase space while being constrained to the typical set, projecting down to efficient exploration of the target distribution's typical set [1, pg. 25]. This effective exploration generally leads to faster convergence compared to random walk-based MH algorithms.

Better Performance in High Dimensions Traditional random walk methods scale poorly with increasing dimensionality. In high dimensions, the volume of parameter space concentrates away from the mode. Random Walk Metropolis proposals either stray too far from the typical set, leading to low acceptance rates, or take tiny steps, resulting in extremely slow movement. HMC is uniquely suited to the high-dimensional problems of applied interest because its Hamiltonian trajectories effectively navigate this complex geometry. The acceptance probability in HMC deteriorates only negligibly as the dimension increases, unlike RWM [1, pg. 39-40].

Higher Acceptance Rates (Ideally 100%) In theory, if Hamiltonian dynamics could be simulated exactly, the energy of the system (the Hamiltonian) would be conserved [2, pg. 218]. This conservation would lead to a Metropolis-Hastings acceptance rate of exactly 1. While practical implementations use discrete numerical integrators (like Leapfrog), which introduce errors that necessitate a Metropolis-Hastings correction step, the acceptance rate often remains high in practice. This contrasts with random walk methods where acceptance rates can be very low, especially in high dimensions or for complex distributions. A higher acceptance rate means fewer proposed steps are rejected, leading to more efficient sampling.

4 Structure of the HMC Algorithm

4.1 Algorithm Overview and Pseudocode

The HMC algorithm follows essentially the same structure as the Metropolis-Hastings algorithm: propose a new sample, accept with some probability. The key difference is that HMC uses Hamiltonian dynamics to generate the proposal, which leads to more informed moves. Additionally, because the proposal distribution is symmetric, the acceptance rule becomes simpler. Here's a breakdown of the main steps:

1. Draw a new value of momentum p from the distribution, $e^{-\beta H}$ where $H = T + U$, where T is the kinetic energy and U is the potential energy.

2. Starting in state (q, p) (where q is a position variable), run Hamiltonian dynamics for L steps with the step size ϵ using the Leapfrog method. L and ϵ are the hyperparameter of our algorithm.

3. After running L steps, negate the momentum variables, giving a proposed state of (q^*, p^*). The negation makes the proposal distribution symmetric so if we run L steps again, will get back to the original state. Although this will not affect the process

4. The proposed state (q^*, p^*) is accepted as the next state using a Metropolis-Hastings acceptance probability:

$$A(p^*, q^*) = min[1, \frac{f(q^*, p^*)g(q, p|q^*, p^*)}{f(q, p)g(q^*, p^*|q, p)}] \quad \text{which simplifies (due to symmetry) to}$$

$$= min[1, \frac{f(q^*, p^*)}{f(q, p)}]$$

$$= min[1, \frac{e^{-\beta H(q^*, p^*)}}{e^{-\beta H(q,p)}}]$$

$$= min[1, e^{\beta[-T(p^*)-U(q^*)+T(p)+U(q)]}]$$

If the next state is rejected, then the current state becomes the next state. We can expect an acceptance probability of 1 because the Hamiltonian is conserved.

4.2 Interpretation of Parameters

The two key hyperparameters are the step size ϵ (sometimes denoted $\delta\tau$) and the number of integration steps L. HMC simulates Hamiltonian dynamics in phase space (position q, momentum p) to propose new states, allowing for efficient exploration of complex distributions.

The step size ϵ determines how finely the continuous dynamics are discretized during simulation, typically via the Leapfrog method. Smaller ϵ values yield more accurate trajectories but require more steps to cover the same distance. Larger values reduce computational cost but may introduce significant errors, reducing the Metropolis acceptance rate.

The parameter L specifies how many steps are taken per trajectory. Together, $L \cdot \epsilon$ gives the trajectory length. Longer trajectories (larger L) allow for broader exploration of the parameter space, especially the typical where most probability mass resides. However, increasing L also accumulates numerical errors, which may lower acceptance rates and waste computation.

Both ϵ and L must be tuned to balance accuracy, exploration, and efficiency. Their interplay directly affects the performance and reliability of the HMC algorithm.

4.3 Parallelizing for Particle Simulations

To improve computational efficiency, we parallelized the Leapfrog integration step across particles. Each particle evolves independently under Hamiltonian dynamics, making the computation well-suited for parallel execution. We utilize Julia's `Threads.@threads` macro to distribute the particles among the available CPU threads. The set of particles is partitioned into chunks, and each thread processes one chunk. For every particle in its chunk, a thread applies the Leapfrog algorithm: starting with a half-step update to the momentum, followed by a loop over L steps in which the position is updated via a full step, and the momentum is updated with either a full or half-step, depending on the iteration. Once the integration is complete, the updated position and momentum are stored. This parallelization allows all particles to be updated simultaneously, significantly accelerating the simulation while preserving the correctness of each particle's trajectory.

5 The Canonical Ensemble

The canonical ensemble describes a system in thermal contact with a large temperature reservoir, which fixes the system's temperature (T) while its energy is allowed to fluctuate. The probability of finding the system in a specific microstate r with energy E_r is given by the Boltzmann distribution, proportional to $e^{-\beta E_r}$ where $\beta = 1/(kT)$. The normalization constant for this probability is the partition function (Z), defined as the sum over all possible microstates:

$$Z \equiv \sum_r e^{-\beta E_r}$$

The partition function is a central quantity as it encapsulates all equilibrium statistical information** about the system, and its calculation is often easier than counting microstates at fixed energy. A key macroparameter derived from Z is the average energy ($\bar{E}$), which is obtained through the relationship $\bar{E} = -\frac{\partial \ln Z}{\partial \beta}$. Fluctuations in energy, like the variance, can also be calculated from derivatives of $\ln Z$.

6 Key Results

6.1 Ideal Gas

We begin with a simple model: the **ideal gas**, composed of N non-interacting monoatomic particles in a box of volume $V = L^3$. The only energy contribution is kinetic, making the total energy the sum of each particle's kinetic energy.

6.2 Theoretical expectation

For a single particle in a 3D infinite potential well (box), the stationary states are:

$$V(x,y,z) = \begin{cases} 0 & 0 \le x,y,z \le L \\ \infty & \text{otherwise} \end{cases}, \quad \psi(x,y,z) \propto \sin(k_x x)\sin(k_y y)\sin(k_z z)$$

with $k_i = \frac{\pi n_i}{L}$ and energy eigenvalues:

$$E = \frac{h^2}{8mL^2}(n_x^2 + n_y^2 + n_z^2)$$

In a thermal reservoir (canonical ensemble), the partition function for N distinguishable particles is:

$$Z = \left(\sum_{n_x,n_y,n_z} e^{-\beta \frac{h^2}{8mL^2}(n_x^2+n_y^2+n_z^2)} \right)^N = \xi^N$$

For indistinguishable particles, divide by $N!$: $Z = \frac{\xi^N}{N!}$.

In the classical (large n) limit, sums become integrals:

$$\xi = \left(\int_0^\infty e^{-\beta \frac{h^2}{8mL^2} n^2} dn \right)^3 = \left(\frac{1}{2}\frac{L}{h}\sqrt{8\pi mkT} \right)^3$$

So the total partition function becomes:

$$Z(N,V,T) = \frac{1}{N!}\frac{V^N}{h^{3N}}(2\pi mkT)^{3N/2}$$

The average energy follows from:

$$\overline{E} = -\frac{\partial}{\partial \beta}\ln Z = \frac{3}{2}NkT$$

This result is consistent with the classical equipartition theorem for an ideal monatomic gas.

6.2.1 Comparison with simulation

Since the only energy present in an ideal gas system is kinetic energy, our Hamiltonian is $H = \frac{\vec{p}^2}{2m}$ where p is our momentum variable and m is the mass of the particle. We set $m = 1$ and $\beta = 1$ for our simulation. From Figure 4, we can see that over the course of 1000 samples, our average energy converges to what we theoretically expect.

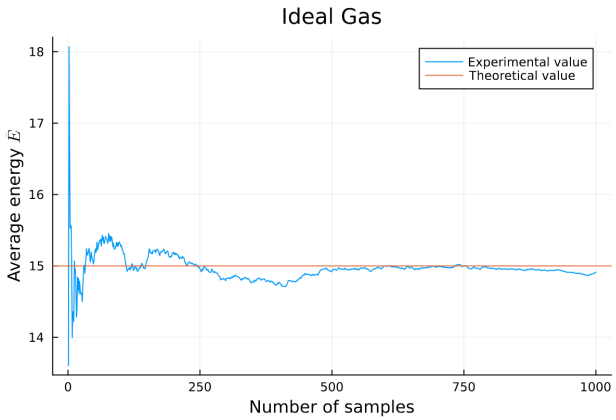


Figure 4: *We simulated* 10 *particles over* 1000 *samples with a* β *of 1. The leapfrog parameters are* 0.1 *for step size* ϵ *and* 100 *for number of steps* L*. It ultimately converges to* $E = 15$ *although the simulated energy initially fluctuates before stabilizing near the predicted value. Late-stage samples exhibit minor thermal variations around equilibrium, confirming proper sampling. The agreement between simulation and theory validates the numerical implementation for this classical system.*

6.3 Springy Molecules

After considering the ideal gas, we can also consider a system where its not purely kinetic energy contributing to the Hamiltonian but also a spring potential.

6.3.1 Theoretical expectation

Consider a classical system of S very weakly interacting monoatomic molecules in a container of volume V at temperature T. The energy of a single molecule is given by

$$E = \frac{1}{2m}\vec{p}^2 + \frac{1}{2}k\vec{r}^2$$

where $k > 0$ is a spring constant.

Recalling the **equipartition theorem**, each degree of freedom that appears only quadratically contributes an average energy of $\frac{1}{2}kT$ to the total energy of the system [5]. In this model, both the kinetic and the potential are both quadratic – each contributing three degrees of freedom in a three-dimensional system (one per spatial axis). As a result, each particle contributes a total average energy of $3 \times \frac{1}{2}kT$ from the kinetic part and an additional $3 \times \frac{1}{2}kT$ from the potential part:

$$\bar{E}_{\text{particle}} = \frac{3}{2}kT + \frac{3}{2}kT = 3kT$$

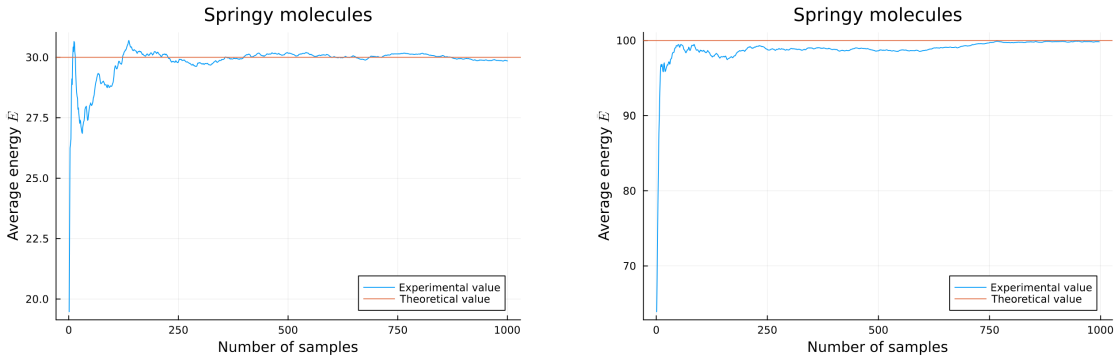
Thus, the total average energy for N particles is:

$$\bar{E} = 3NkT$$

6.3.2 Comparison with simulation

Figure 5a illustrates the case where the system is in three dimensions. As expected, the average energy converges to $\bar{E} = 30$, which is exactly twice the value observed for the ideal gas in the same dimensional setting. This result aligns with theoretical predictions, since the inclusion of a harmonic potential adds three additional quadratic degrees of freedom. This doubles the energy per particle compared to a purely kinetic system.

To further understand the system's behavior in higher-dimensional spaces, Figure 5b presents the case where the number of spatial dimensions is increased to 10. In this scenario, we observe a significantly higher average energy. This increase is consistent with the equipartition theorem: each additional spatial dimension introduces two more quadratic degrees of freedom (one for kinetic and one for potential energy), and each of these contributes $1/2kT$. Thus, in d dimensions, the total average energy per particle becomes $\bar{E}_{\text{particle}} = d{\cdot}kT$, and the total energy scales linearly with both the number of particles and the dimensionality. Consequently, higher-dimensional systems naturally exhibit larger average energies.



(a) d = 3 (b) d = 10

Figure 5: *Using the same parameters, we simulated* 10 *particles over* 1000 *samples with a* β *of 1 with a spring constant* k *of* 0.3. *The leapfrog parameters are* 0.1 *for step size* ϵ *and* 100 *for number of steps* L. *Furthermore, we also varied the number of dimensions* d *from 3 to 10.*

6.4 The "Nasty Gas"

Inspired by a homework question given in the Statistical Mechanics course, we decided to find the results of the following problem (HW6.5):
Consider a gas of N non-interacting atoms in a d-dimensional box of volume V, with energy

$$E = \sum_{i=1}^{N} A|\vec{p_i}|^s$$

where s and A are some constants.

6.4.1 Theoretical expectation

We can construct our canonical partition function,

$$Z = \frac{V^N}{N!h^{dN}} \left(\int d^d p \, e^{-\beta A|\vec{p}|^s} \right)^N$$

In d-dimensional spherical coordinates, we can write:

$$\int d^d p = S_d \int_0^\infty dp \, p^{d-1}$$

where S_d is the surface area of a unit d-dimensional sphere.

Changing variables to $x = (\beta A)^{1/s} p$, we get:

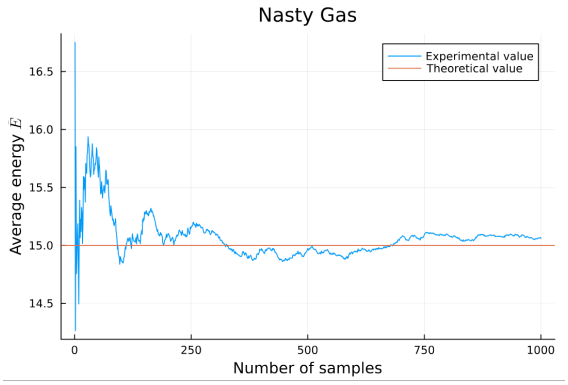
$$\int_0^\infty dp \, p^{d-1} e^{-\beta A p^s} = \frac{1}{s} (\beta A)^{-d/s} \Gamma\left(\frac{d}{s}\right)$$

Thus, the partition function becomes:

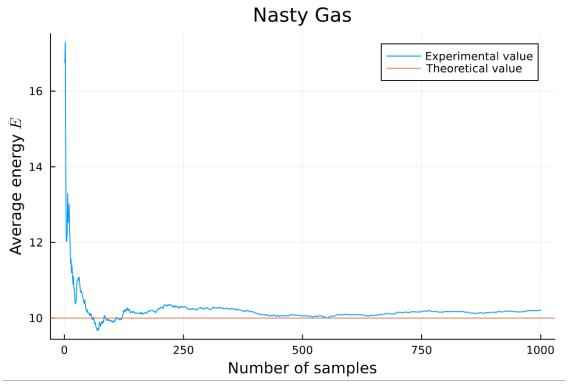
$$Z = \frac{V^N S_d^N}{N! h^{dN}} \left(\frac{1}{s} (\beta A)^{-d/s} \Gamma\left(\frac{d}{s}\right) \right)^N = \frac{V^N}{N!} \left(\frac{S_d}{s h^d} \left(\frac{kT}{A} \right)^{d/s} \Gamma\left(\frac{d}{s}\right) \right)^N$$

For the average energy, we compute:

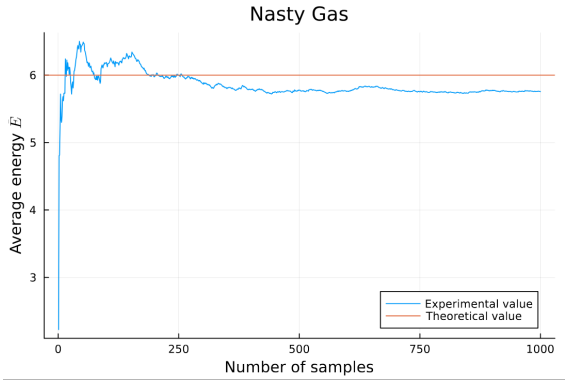
$$\bar{E} = -\left(\frac{\partial \ln Z}{\partial \beta} \right)_V = \frac{d}{s} NkT$$



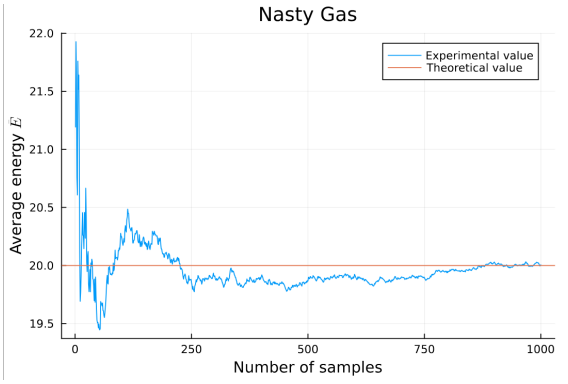
(a) d = 3, s = 2



(b) d = 3, s = 3



(c) d = 3, s = 5



(d) d = 10, s = 5

Figure 6: *The simulated results and their convergence to the theoretical expectation when varying the s and d variables over 1000 samples with a β of 1. The leapfrog parameters are* 0.1 *for step size ϵ and* 100 *for number of steps L.*

6.4.2 Comparison with simulation

For the case where $A = 0.5$ and $s = 2$, the system reduces to a classical ideal gas with kinetic energy $E = \frac{|\vec{p}|^2}{2m}$. The behavior of the average energy aligns with the familiar result $\bar{E} = \frac{d}{2} NkT$. Figure 6a is shown to be similar to the previous ideal gas graph, where the average energy converges to $\bar{E} = \frac{3}{2} NkT = 15$ for $N = 10$ particles in three dimensions. This agreement confirms the consistency of our generalized Hamiltonian formulation with the standard kinetic theory in the case $s = 2$.

As we increase the exponent to $s = 3$ and $s = 5$, the average energy decreases. This is consistent with the inverse dependence of $\bar{E}$ on s. A higher s causes the energy to grow more rapidly with momentum, making

high-energy states thermally less accessible. Consequently, it lowers the average energy at fixed temperature. Figure 6b and 6c show an increase in the expected value of $\bar{E}$ compared to the $s = 2$ case. This is because the Hamiltonian,

$$E = A|\vec{p}|^s$$

becomes more sensitive to high-momentum particles as s increases. The energy contribution grows more rapidly with momentum, leading to a higher energy density in the tails of the momentum distribution. Consequently, even though the temperature is fixed, the average energy increases with larger s, since more weight is given to higher-momentum states in the Boltzmann distribution.

Since the HMC method allows an effective computation in higher dimensions, we can see analyze the behaviour as we increase the number of dimensions. In Figure 6d, we investigate the effect of increasing the spatial dimension from $d = 3$ to $d = 10$, keeping s fixed. As the number of dimensions increases, the average energy increases proportionally. This is expected because each additional dimension introduces a new momentum degree of freedom, providing more ways for the system to distribute energy—consistent with the equipartition theorem.

7 Conclusion

In this study, we investigated the thermodynamic behavior of classical systems using the Hamiltonian Monte Carlo (HMC) method. We began with the ideal gas model, where only kinetic energy contributes to the Hamiltonian, and successfully recovered the well-known result $\bar{E} = \frac{3}{2}NkT$, consistent with the equipartition theorem. We then generalized the kinetic energy term to $\sum |\vec{p}_i|^s$, allowing us to study how the exponent s influences the average energy. As expected, increasing s led to a reduction in average energy due to the suppression of high-momentum contributions.

Our simulations further examined how increasing spatial dimensionality affects thermodynamic properties. We observed that higher-dimensional systems yielded larger average energies, which aligns with the prediction that each degree of freedom contributes $\frac{1}{2}kT$ to the total energy.

To expand on this, we introduced a harmonic (spring) potential to the Hamiltonian and found that the average energy doubled compared to the ideal gas case, once again matching theoretical expectations. These findings were obtained using the HMC method, which efficiently samples from the canonical ensemble by simulating Hamiltonian dynamics. We also implemented a parallelized version of leapfrog integration to compute the trajectories of multiple particles simultaneously, significantly accelerating the simulation process.

Overall, our results confirm the validity of the HMC framework for simulating thermodynamic systems and demonstrate how both the form of the Hamiltonian and system dimensionality influence energy distribution. This approach lays the groundwork for studying more complex interacting systems in future work.

7.1 Future work

A natural extension of this work is to reproduce the results of Tagawa et al. on the efficiency of Hybrid Monte Carlo (HMC) applied to the multicanonical ensemble. Their study used fluid argon as a model system with $N = 108$ particles interacting via the Lennard-Jones potential, using parameters $\sigma = 3.41$ Å, $\epsilon/k_B = 120$ K, and density $\rho = 0.02237$ Å^{-3} [4, pg. 3-4]. The system was simulated at $T_0 = 180$ K under a multicanonical effective potential E_{mc}, where $\partial E_{\mathrm{mc}}/\partial E$ was computed using Lagrangian cubic interpolation. Each HMC run consisted of 10^6 molecular dynamics steps.

At present, we are unable to fully replicate this due to computational constraints—most notably, the requirement of a small timestep ($\sim$30 fs) in the leapfrog integrator, which demands significant computational power. In future work, we plan to optimize our codebase and utilize high-performance computing resources to perform large-scale multicanonical HMC simulations. This will allow us to assess the method's efficiency in systems with rugged energy landscapes and verify the findings of Tagawa et al.

References

[1] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[2] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[3] Brian Keng. Markov chain monte carlo methods, rejection sampling and the metropolis-hastings algorithm, December 2015. https://bjlkeng.io/posts/markov-chain-monte-carlo-mcmc-and-the-metropolis-hastings-algorithm/.

[4] Tsugumichi Tagawa, Toshihiro Kaneko, and Shinichi Miura. On computational efficiency of the hybrid monte carlo method applied to the multicanonical ensemble. *Molecular Simulation*, 43(13-16):1291–1294, 2017.

[5] Sudarson S Sinha Zane Sterkewolf. 18.11: The equipartition principle. https://chem.libretexts.org/Bookshelves/Physical$_a$nd$_T$heoretical$_C$hemistry$_T$extbook$_M$aps/Physical$_C$hemistry$_($LibreTe