proceedings

Proceedings

Galilean and Hamiltonian Monte Carlo †

John Skilling *

Maximum Entropy Data Consultants Ltd., CB4 1XE Kenmare, Ireland

* Correspondence: skilling@eircom.net

† Presented at the 39th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Garching, Germany, 30 June–5 July 2019.

Published: 5 December 2019

Abstract: Galilean Monte Carlo (GMC) allows exploration in a big space along systematic trajectories, thus evading the square-root inefficiency of independent steps. Galilean Monte Carlo has greater generality and power than its historical precursor Hamiltonian Monte Carlo because it discards second-order propagation under forces in favour of elementary force-free motion. Nested sampling (for which GMC was originally designed) has similar dominance over simulated annealing, which loses power by imposing an unnecessary thermal blurring over energy.

Keywords: Nested sampling; simulated annealing; Hamiltonian Monte Carlo; Galilean Monte Carlo

PACS: 02.50.Ng

1. Introduction

Question: How does a mathematician find a needle in a haystack?
Answer: Keep halving the haystack and discarding the "wrong" half.

This trick relies on having a test for whether the needle is or is not in the chosen half. With that test in hand, the mathematician expects to find the needle in $\log_2 N$ steps instead of the $O(\frac{1}{2}N)$ trials of direct point-by-point search.

The programmer is faced with a similar problem when trying to locate a small target in a large space. We do not generally have volume-wide global tests available to us, being instead restricted to point-wise evaluations of some quality function $Q(\mathbf{x})$ at selected locations $\mathbf{x}$. A successful algorithm should have two parts.

One part uses quality differences to drive successive trial locations towards better (larger) quality values. This iteration reduces the available possibilities by progressively eliminating bad (low quality) locations. *Nested sampling* [1] accomplishes this without needing to interpret Q as energy or anything else. By relying only on comparisons ($>$ or $=$ or $<$) it's invariant to any monotonic regrade, thereby preserving generality. Its historical precursor was simulated annealing [2], in which $\log Q$ was restrictively interpreted as energy in a thermal equilibrium.

The other part of a successful algorithm concerns how to move location without decreasing the quality attained so far. Here, it will often be more efficient to move systematically for several steps in a chosen direction, rather than diffuse slowly around with randomly directed individual steps. After n steps, the aim is to have moved $\Delta\mathbf{x} \propto n$, not just $\sqrt{n}$. *Galilean Monte Carlo* (GMC) accomplishes this with steady ("Galilean") motion controlled by quality value. Its historical precursor was Hamiltonian Monte Carlo (HMC) [3], in which motion was controlled by "Hamiltonian" forces restrictively defined by a quality gradient which sometimes doesn't exist.

In both parts, nested sampling *compression* and GMC *exploration*, generality and power are retained by avoiding presentation in terms of physics. After all, elementary ideas underlie our understanding of physics, not the other way round, and discarding what isn't needed ought to be helpful.

2. Compression by Nested Sampling

Question: How does a programmer find a small target in a big space?
Answer: ……

There may be very many (N) possible "locations" $\mathbf{x}$ to choose from. For clarity, assume these have equal status *a priori*—there is no loss of generality because unequal status can always be modelled by retreating to an appropriate substratum of equivalent points. For the avoidance of doubt, we are investigating practical computation so N is finite, though with no specific limit.

As the first step, the programmer with no initial guidance available can at best select a random location $\mathbf{x}_1$ for the first evaluation $Q_1 = Q(\mathbf{x}_1)$. The task of locating larger values of Q carries no assumption of geometry or even topology. Locations could be shuffled arbitrarily and the task would remain just the same. Accordingly, we are allowed to shuffle the locations into decreasing Q-order without changing the task (Figure 1). If ordering is ambiguous because different locations have equal quality, the ambiguity can be resolved by assigning each location its own (random) key-value to break the degeneracy.

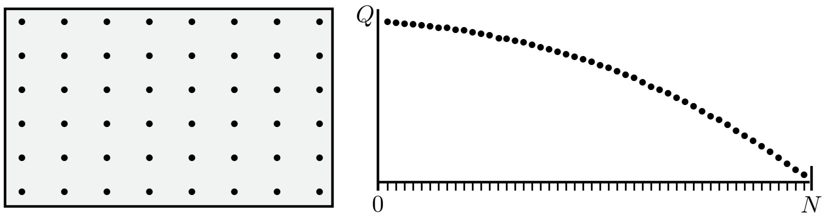

Figure 1. N locations (left) ordered (right) by quality Q.

Being chosen at random, $\mathbf{x}_1$'s shuffled rank Nu_1 marks an equally random fraction of the ordered N locations. Our knowledge of u_1 is uniform: $u_1 \sim \texttt{Uniform}(0,1)$. We can encode this knowledge as one or (better) more samples that simulate what the position might actually have been. If the programmer deems a single simulation too crude and many too bothersome, the mean and standard deviation

$$\log u_1 = -1 \pm 1 \tag{1}$$

often suffice.

The next step is to discard the "wrong" points with $Q < Q_1$ and select a second location $\mathbf{x}_2$ randomly from the surviving Nu_1 possibilities. Being similarly random, $\mathbf{x}_2$'s rank Nu_1u_2 marks a random fraction of those Nu_1, with $u_2 \sim \texttt{Uniform}(0,1)$ (Figure 2).

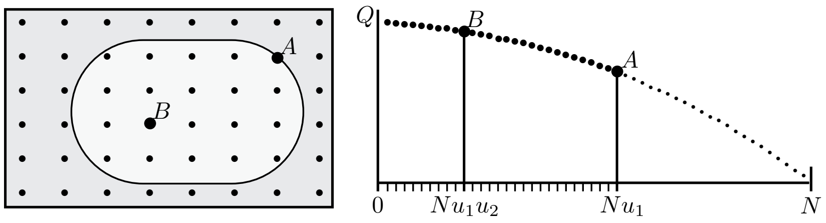

Figure 2. Selection of second location B after discarding domain outside A.

And so on. After k steps of increasing quality $Q_1 < Q_2 < \cdots < Q_k$, the net compression ratio $X_k = u_1u_2 \ldots u_k$ can be simulated as several samples from

$$X_k \sim \underbrace{\texttt{Uniform}(0,1) \cdot \texttt{Uniform}(0,1) \cdot \ldots \cdot \texttt{Uniform}(0,1)}_{k} \tag{2}$$

to get results fully faithful to our knowledge or simply abbreviated as mean and standard deviation

$$\log X_k = \underbrace{(-1 \pm 1) + (-1 \pm 1) + \ldots (-1 \pm 1)}_{k} = -k \pm \sqrt{k} \tag{3}$$

Compression proceeds exponentially until the user decides that Q has been adequately maximised. At that stage, the evaluated sequence $Q_1 < Q_2 < \cdots < Q_k$ of qualities Q has been paired with the corresponding sequence $X_1 > X_2 > \ldots X_k$ of compressions X (Figure 3), either severally simulated or abbreviated in mean.

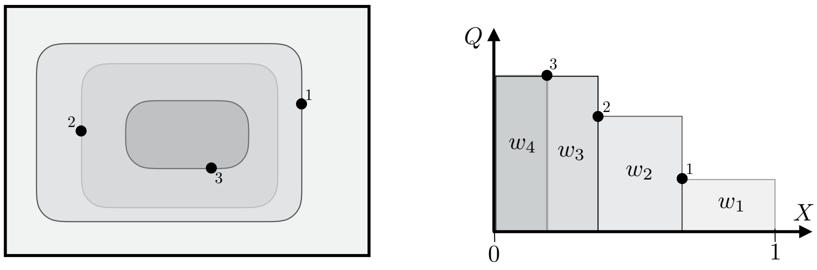

Figure 3. Nested sampling produces the relationship $Q(X)$.

Q can then be integrated as

$$Z = \int Q(\mathbf{x})\, d\mathbf{x} = \int_0^1 Q(X)\, dX \approx \sum_{i=1}^{k} w_i\,, \quad w_i = Q_i\, \Delta X_i \tag{4}$$

so that quantification is available for Bayesian or other purposes. Any function of Q can be integrated too from the same run, changing Q_i to some related Q'_i while leaving the X_i fixed. And the statistical uncertainty in any integral Z is trivially acquired from the repeated simulations (2) of what the compressions X might have been according to their known distributions.

That's nested sampling. It requires two user procedures additional to the $Q(\mathbf{x})$ function. The first is to sample an initially random location to start the procedure. The second—which we next address—is to move to a new random location obeying a lower bound on Q. Note that there is no modulation within a constrained domain. Locations are either acceptable, $Q(\mathbf{x}) \geq Q^*$, or not, $Q(\mathbf{x}) < Q^*$.

3. Exploration by Galilean Monte Carlo

The obvious beginners' MCMC procedure for moving from one acceptable location to another, while obeying detailed balance but not moving so far that Q always disobeys the lower bound Q^*, is:

```
Start at x with acceptable quality $Q(\mathbf{x}) \geq Q^*$
Repeat for length of trajectory
    Set v = isotropically random velocity
        $\mathbf{x}' = \mathbf{x} + \mathbf{v}$ = trial location
    if( $Q(\mathbf{x}) \geq Q^*$ )    accept new $\mathbf{x} = \mathbf{x}'$
    else                            reject $\mathbf{x}'$ by keeping x
```
(5)

However, randomising $\mathbf{v}$ every step is diffusive and slow, with net distance travelled increasing only as the square root of the number of steps.

All locations within the constrained domain are equally acceptable, so the program might better try to proceed in a straight line, changing velocity only when necessary in an attempt to reflect specularly off the boundary (Figure 4, left). The user is asked to ensure that the imposed geometry makes sense in the context of the particular application, otherwise there will be no advantage.

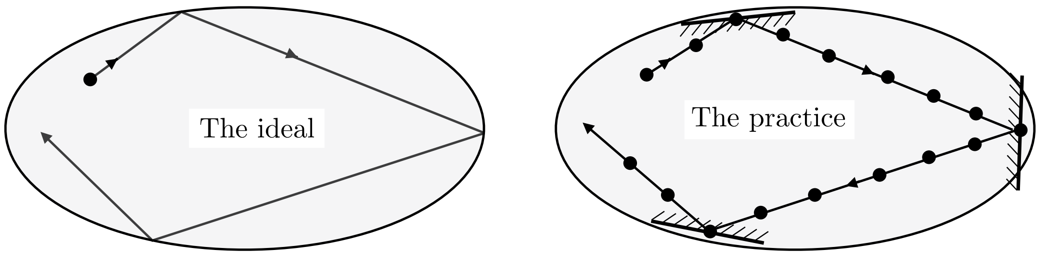

Figure 4. The motivation behind Galilean Monte Carlo (GMC).

With finite step lengths, it's not generally possible to hit the boundary exactly whilst simultaneously being sure that it had not already been encountered earlier, so the ideal path is impractical. Instead, we take a current location **x** and suppose that a corresponding unit vector **n** can be defined there as a proxy for where the surface normal would be if that surface was close at hand (Figure 4, right). Again, it is the user's responsibility to ensure that **n** makes sense in the context of the particular application: exploration procedures cannot anticipate the quirks of individual applications.

Reflection from a plane orthogonal to **n** (drawn horizontally in Figure 5) modifies an incoming velocity **v** (in northward direction from the South) to $\mathbf{v}' = \mathbf{v} - 2\mathbf{n}(\mathbf{n}^T\mathbf{v})$. Depending on the circumstances, the incoming particle may proceed straight on to the North ($+\mathbf{v}$), or be reflected to the East ($+\mathbf{v}'$), or back-reflected to the West ($-\mathbf{v}'$), or reversed back to the South ($-\mathbf{v}$).
Mistakenly, the author's earlier introduction of GMC in 2011 [4] *reduced the possibilities by eliminating West, but at the cost of allowing the particle to escape the constraint temporarily, which damaged the performance and cancelled its potential superiority.*

If the potential destination North is acceptable (bottom left in Figure 5), the particle should move there and not change its direction (so **n** need not be computed). Otherwise, the particle needs to change its direction but not its position.

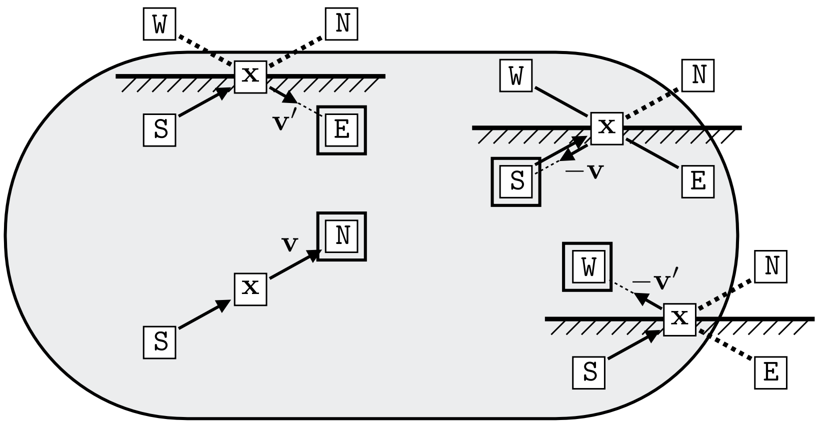

Figure 5. North – East – West – South, the four Galilean outcomes.

For a North-South oriented velocity to divert into East-West, either East or West must be acceptable, but not both because East-West particles would then pass straight through without interacting with North-South, so the proposed diversion would break detailed balance. Likewise, for an East-West velocity to divert North-South, either North or South must be acceptable but not both. These conditions yield the following procedure:

```
Start at $\mathbf{x}$ with acceptable quality $Q(\mathbf{x}) \geq Q^*$
Set $\mathbf{v} =$ isotropically random velocity
Repeat for length of trajectory
    Set N = “$Q(\mathbf{x}+\mathbf{v}) \geq Q^*$”
    if( N )                      exit with $\mathbf{x} = \mathbf{x}+\mathbf{v}$       [go North]
    Set $\mathbf{v}' = \mathbf{R}\mathbf{v} =$ reflection velocity
    Set E = “$Q(\mathbf{x}+\mathbf{v}') \geq Q^*$”, W = “$Q(\mathbf{x}-\mathbf{v}') \geq Q^*$”, S = “$Q(\mathbf{x}-\mathbf{v}) \geq Q^*$”
    if( S & (E but not W) )      exit with $\mathbf{v} = \mathbf{v}'$          [aim East]
    if( S & (W but not E) )      exit with $\mathbf{v} = -\mathbf{v}'$         [aim West]
                   otherwise     exit with $\mathbf{v} = -\mathbf{v}$          [aim South]
```
(6)

Any self-inverse reflection operator $\mathbf{R}$ will do, though the reflection idea suggests $\mathbf{R} = \mathbf{I} - 2\mathbf{n}\,\mathbf{n}^T$.

That's Galilean Monte Carlo. The trajectory is explored uniformly, with each step yielding an acceptable (though correlated) sample.

4. Compression and Exploration

GMC was originally designed for nested-sampling compression, from which probability distributions can be built up after a run by identifying quality as likelihood L in the weighted sequence (4) of successively compressed locations. However, GMC can also be used when exploring a weighted distribution directly.

For *compression* (standard nested sampling, Figure 6, left), only the domain size X is iterated, albeit under likelihood control.

Compression:
```
Enter with $X$ and $L$
    Set constraint $L^* = L$ defining $X^*$
    Sample within $L^*$ to get $X' = uX^*$
        Exit with $X'$ and $L'$
```
(7)

For *exploration* (standard reversible MCMC, Figure 6, right), the likelihood is relaxed as well through a preliminary random number $u' \sim \texttt{Uniform}(0, 1)$.

Exploration:
```
Enter with $X$ and $L$
    Set constraint $L^* = u'L$ defining $X^*$
    Sample within $L^*$ to get $X' = uX^*$
        Exit with $X'$ and $L'$
```
(8)

This is equivalent to standard Metropolis balancing “Accept $\mathbf{x}'$ if $L(\mathbf{x}') \geq u'L(\mathbf{x})$”, the only difference being that the lower bound $u'L$ is set beforehand instead of checked afterwards.

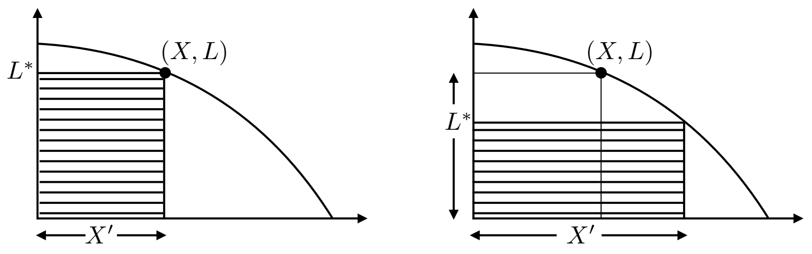

Figure 6. GMC for compression (**left**) and exploration (**right**).

5. Exploration by Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) [3] uses a physical analogy with kinetic theory of gases in which a thermal distribution of moving particles, whose position/velocity probability distribution factorises into space and velocity parts

$$\Pr(\mathbf{x},\mathbf{v}) \propto e^{-E(\mathbf{x},\mathbf{v})}\,, \quad E(\mathbf{x},\mathbf{v}) = V(\mathbf{x}) + T(\mathbf{v}) \tag{9}$$

with potential energy V defining the spatial target distribution $\Pr(\mathbf{x}) \propto e^{-V(\mathbf{x})}$ and kinetic energy $T = \frac{1}{2}|\mathbf{v}|^2$ distributed as the Boltzmann thermal equilibrium $\Pr(\mathbf{v}) \propto e^{-T(\mathbf{v})}$.
The usual dynamics (Figure 7)

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}\,, \quad \frac{d\mathbf{v}}{dt} = -\nabla V(\mathbf{x}) \tag{10}$$

relaxes an initial setting towards the joint equilibrium (9) under occasional collisions which reset $\mathbf{v}$ according to $\Pr(\mathbf{v})$, leaving $\mathbf{x}$ as a sample from the target $\Pr(\mathbf{x})$.

Between collisions, the force field is necessarily digitised into impulses at discrete time intervals δt, so the computation obeys

$$\delta\mathbf{x} = \mathbf{v}\,\delta t\,, \quad \delta\mathbf{v} = -\nabla V(\mathbf{x})\,\delta t \tag{11}$$

To make the trajectory reversible and increase the accuracy order, the impulses are halved at the start $\mathbf{x}$ and end $\mathbf{x}'$, but even this does not ensure full accuracy because the dynamics has been approximated (Figure 8).

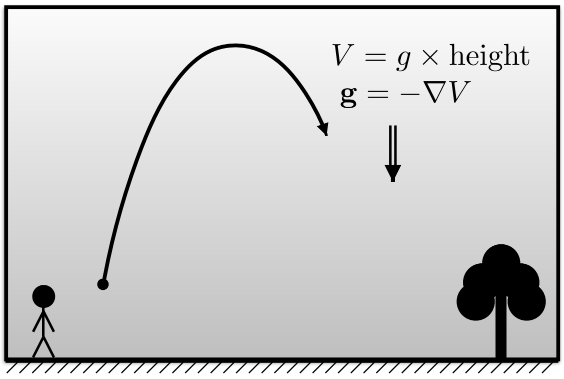

Figure 7. The Hamiltonian Monte Carlo idea.

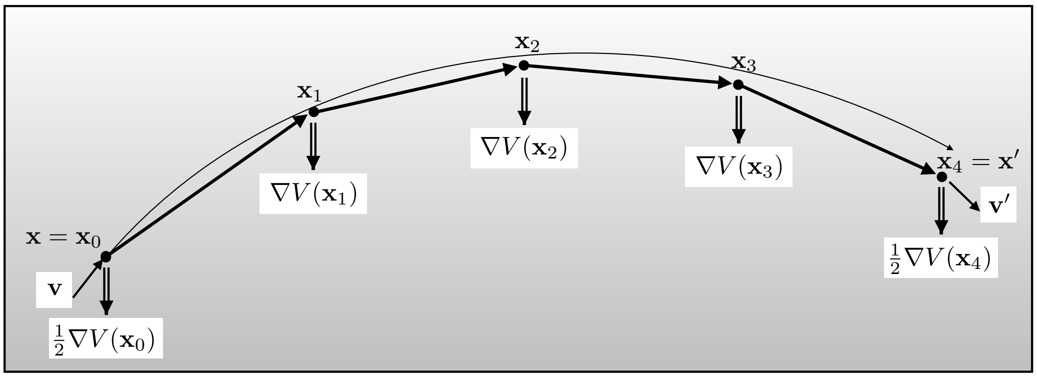

Figure 8. Hamiltonian Monte Carlo path approximates the ideal continuous path.

To correct this, the destination $\mathbf{x}'$, whose total energy $E' = V' + T'$ ought to agree with the initial $E = V + T$, is subjected to the usual Metropolis balancing.

$$\text{Accept } \mathbf{x}' \quad \text{iff} \quad e^{-E'} \geq e^{-E} \cdot \texttt{Uniform}(0,1). \tag{12}$$

In practice, the correction is often ignored because the (reversible) algorithm explores "level sets" whose contours are often an adequately good approximation to the true Hamiltonian provided the fixed timestep is not too large.

That's Hamiltonian Monte Carlo. The trajectory is explored non-uniformly, with successive steps being closer at altitude where the particles are moving slower, so that sampling is closer where the density is smaller—a mismatch which needs to be overcome by the equilibrating collisions. And, of course, HMC requires the potential $V(\mathbf{x})$ (*a.k.a.* log-likelihood) to be differentiable and generally smooth.

6. Compression versus Simulated Annealing

Simulated annealing uses a physical analogy to thermal equilibrium to compress from a prior probability distribution to a posterior. As in HMC, though without the complication of kinetic energy, the likelihood (or quality) is identified as the exponential $L = e^{-E}$ of an energy. In annealing, the energy is scaled by a parameter β so that the quality becomes $Q = L^{\beta} = e^{-\beta E}$ of this scaled energy, with β used to connect posterior (where $\beta = 1$) with prior (where $\beta = 0$).

This "simulates" thermal equilibrium at coolness (inverse temperature) β, and "annealing" refers to sufficiently gradual cooling from prior to posterior that equilibrium is locally preserved. A few lines of algebra, familiar in statistical mechanics, show that the evidence (or partition function) can be accumulated from successive thermal equilibria as

$$\log Z = \int_0^1 \langle \log L \rangle_{\beta} \, d\beta \tag{13}$$

where $\langle \log L \rangle_{\beta}$ is the average log-likelihood as determined by sampling the equilibrium appropriate to coolness β. Equilibrium is defined by weights L^{β} and can be explored either by GMC or (traditionally) by HMC. There is seldom any attempt to evaluate the statistical uncertainty in $\log Z$, the necessary fluctuations being poorly defined in the simulations.

At coolness β, the equilibrium distribution of locations $\mathbf{x}$, initially uniform over the prior, is modulated by L^{β} so that the samples have probability distribution $\Pr(\mathbf{x}) \propto L(\mathbf{x})^{\beta}$ which corresponds to

$$\Pr(X) \propto L(X)^{\beta} \tag{14}$$

in terms of compression. Consequently, samples cluster around the maximum of $\beta \log L + \log X$, where the $\log L(\log X)$ curve has slope $-1/\beta$ (Figure 9, left). Clearly this only works properly if $\log L(\log X)$ is concave ($\frown$). Any zone of convexity ($\smile$) is unstable, with samples heading toward either larger L at little cost to X or toward larger X at little cost to L. A simulated-annealing program cannot enter a convex region, and the steady cooling assumed in (13) cannot occur.

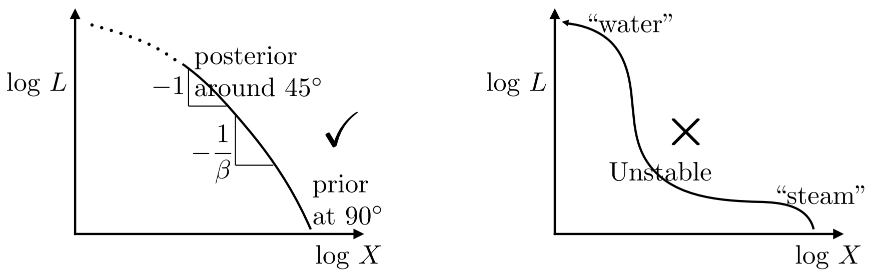

Figure 9. Simulated annealing without (**left**) and with (**right**) phase change.

In the physics analogy, this behaviour is a *phase change* and can be exemplified by the transition from steam to water at 100 °C (Figure 9, right). Because of the different volumes (*exponentially* different in large problems), a simulated-annealing program will be unable to place the two phases in algorithmic contact, so will be unable to model the latent heat of the transition. Correspondingly, computation of evidence Z will fail. *Yet our world is full of interesting phase changes, and a method that cannot cope with them cannot be recommended for general use.*

Nested sampling, on the other hand, compresses steadily with respect to the abscissa $\log X$ regardless of the (monotonic) behaviour of $\log L$, so is impervious to this sort of phase change. Contrariwise, simulated annealing cools through the slope, which need not change monotonically. By using thermal equilibria which average over $e^{-\beta E}$, simulated annealing views the system through the lens of a Laplace transform, which is notorious for its ill-conditioning. Far better to deal with the direct situation.

7. Conclusions

The author suggests that, just as nested sampling dominates simulated annealing, …

Nested sampling	**Simulated Annealing**
Steady compression	Arbitrary cooling schedule for β
Invariant to relabelling Q	Q has fixed form L^β
Can deal with phase changes	Cannot deal with phase changes
Evidence $Z = \int L\,dX$ $\underline{\text{with uncertainty}}$	Evidence $Z = \exp \int_0^1 \langle \log L \rangle_\beta \, d\beta$

… so does Galilean Monte Carlo dominate Hamiltonian.

Galilean Monte Carlo	**Hamiltonian Monte Carlo**
No rejection	Trajectories can be rejected
Any metric is OK	Riemannian metric required
Invariant to relabelling Q	Trajectory explores nonuniformly
Quality function $Q(\mathbf{x})$ is arbitrary	Quality $Q(\mathbf{x})$ must be differentiable
Step functions OK (nested sampling)	Can not use step functions
Can sample any probability distribution	Probability distribution must be smooth
Needs 2 work vectors	Needs 3 work vectors

In each case, reverting to elementary principles by discarding physical analogies enhances generality and power.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Skilling, J. Nested Sampling for general Bayesian computation. *J. Bayesian Anal.* **2006**, *1*, 833–860.
2. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680.
3. Duane, S.; Kennedy, A.D.; Pendleton, B.J.; Roweth, D. Hybrid Monte Carlo. *Phys. Lett. B* **1987**, *195*, 216–222.
4. Skilling, J. Bayesian computation in big spaces—Nested sampling and Galilean Monte Carlo. *AIP Conf. Proc.* **2012**, *1443*, 145–156.