Nested basin-sampling

Matthew Griffiths and David J. Wales*

Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, United Kingdom

E-mail: dw34@cam.ac.uk

Abstract

We report an embarrassingly parallel method for the evaluation of thermodynamic properties over an energy landscape exhibiting broken ergodicity, *nested basin-sampling* (NBS). We also introduce the No Galilean U-Turn Sampler (NoGUTS), a new sampling scheme based on the No U-Turn Sampler (NUTS) introduced by Hoffman and Gelman (2014) that works with the Galilean Monte Carlo scheme introduced by Betancourt (2012) to aid the efficient generation of new live points. NoGUTS can be thought of as a form of reflective slice sampling with an automatic stopping criterion. We apply this approach to a benchmark atomic cluster of 31 Lennard-Jones atoms, which exhibits a low temperature solid-solid heat capacity peak. The calculated heat capacity is compared with results generated by parallel tempering (PT), basin-sampling parallel tempering (BSPT), and standard nested sampling (NS) simulations. NBS reproduces the full heat capacity curve predicted by PT and BSPT, whilst the NS calculation with similar computational cost fails to resolve the low temperature solid-solid phase transition.

1 Introduction

Evaluating the thermodynamic properties of an energy landscape requires evaluation of integrals of the form

$$\mathcal{I}_{\Phi_0}[f] = \int_{\Phi_0} f(V(\boldsymbol{R}))\,\mathrm{d}\boldsymbol{R}, \tag{1}$$

where Φ_0 is the domain of the integral and $\mathcal{I}_{\Phi_0}[f]$ is a functional of the integral of f over Φ_0. $V(\boldsymbol{R})$ is the potential energy function, and $\boldsymbol{R}$ is the $3N$ dimensional vector of Cartesian coordinates for N atoms. Such integrals generally have to be estimated numerically using stochastic methods, which fall into two broad classes, *thermal* or *athermal*. Thermal methods directly generate samples from probability distributions related to $f(V(\boldsymbol{R}))$, often using Markov chain Monte Carlo (MCMC), and usually sampling from the canonical distribution. More advanced thermal methods may sample from a set of related probability distributions simultaneously.[3–11]

The vast majority of configuration space will be extremely high in energy, and to generate statistically valid samples thermal methods must obey detailed balance. Almost all large moves will land in high energy regions and so will be rejected. Hence these sampling methods must take short local moves to have a reasonable acceptance rate, increasing the time taken to simulate large-scale rearrangements. The convergence is dominated by the time taken to simulate such rearrangements, which can lead to broken ergodicity.

One effective technique for generating thermal samples through MCMC is Hamiltonian Monte Carlo (HMC).[12] In HMC the state space is doubled, incorporating a momentum into the Monte Carlo (MC) simulation, which enables the algorithm to make directed moves away from the starting point, which can be much more effective than a simple random walk.[12,13]

A key challenge for HMC is choosing an appropriate simulation length; too short and the trajectory will not travel far enough away from the starting point, too long and the trajectory will begin to return to its starting point. The No U-turn sampler (NUTS) is an

algorithm that enables HMC to automatically detect when the trajectory begins returning to its starting point (the U-turn),[1] avoiding the need to set a trajectory length. NUTS also eliminates sampling bias in standard HMC caused by symplectic integrator errors.[13]

Athermal methods attempt to determine the density of states,

$$\Omega(V) = \frac{d\Phi(V)}{dV}, \tag{2}$$

where

$$\Phi(V) = \int_{V(\boldsymbol{R})<V} \mathrm{d}\boldsymbol{R} \tag{3}$$

is the configuration volume, so eq. (1) can then be expressed as,

$$\mathcal{I}_{\Phi_0}[f] = \int_{-\infty}^{\infty} f(V)\Omega(V)\,\mathrm{d}V = \int_{0}^{\Phi(\infty)} f(V)\,\mathrm{d}\Phi(V). \tag{4}$$

The density of states can be determined by discretising into a set of energy bins, as in transition matrix Monte Carlo[14] and Wang–Landau sampling,[15] or a set of 'temperatures' as in statistical temperature Monte Carlo.[16–18] The key challenge associated with these methods is that the relative size of adjacent histogram bins can be extremely large if the bin ranges are not carefully chosen, so the probability of any move landing in the smaller bins becomes too low and the simulation will not converge.

Nested sampling (NS) presents an alternative approach that effectively chooses the optimal bin-width during the course of the simulation dynamically.

1.1 Nested sampling

NS is an approach that was developed by Skilling[19] to efficiently calculate the evidence in Bayesian inference (see appendix D), which is equivalent to the evaluation of eq. (1). Skilling's insight was to reformulate the density of states integral, eq. (4), as a Lebesgue

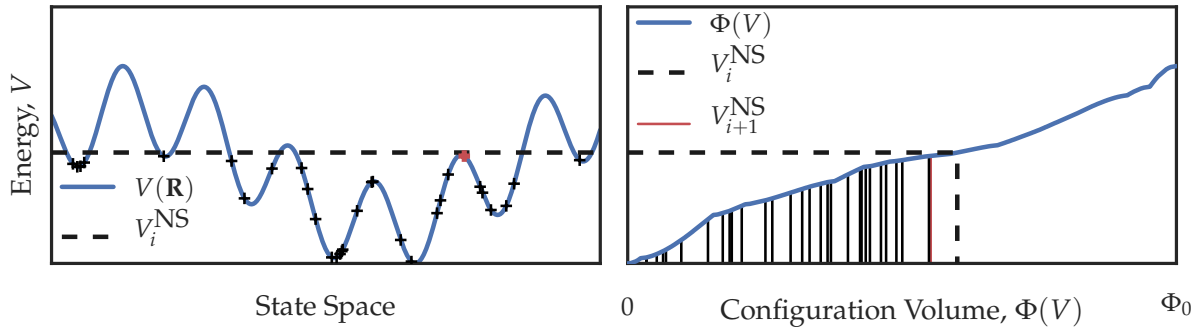


Figure 1: A schematic representation of how nested sampling can calculate the relative difference in configuration volume between two energy thresholds. n^{live} live points are randomly distributed over $\boldsymbol{R}$ such that they all have energy less than V_i^{NS}. The live point with the highest energy, V_{i+1}^{NS} is highlighted in red. As the live points are randomly distributed over $\boldsymbol{R}$ (and so $\Phi(V)$) we know that $\Phi(V_i^{\mathrm{NS}})/\Phi(V_{i+1}^{\mathrm{NS}}) \equiv t_i^{\mathrm{NS}} \sim \mathcal{B}(n^{\mathrm{live}}, 1)$.

integral,

$$\Pr(D) = \int \Pr\left(D|M(\boldsymbol{\theta})\right) \Pr\left(M(\boldsymbol{\theta})\right) \, \mathrm{d}\boldsymbol{\theta} = \int_0^\infty \lambda \, \mathrm{d}X^{\mathrm{NS}}(\lambda), \tag{5}$$

where $\Pr(D|M(\boldsymbol{\theta})$ is the *likelihood* of the observed data D given a model M with parameters $\boldsymbol{\theta}$, $\Pr(M(\boldsymbol{\theta}))$ is the *prior* probability of the model having a parameter value $\boldsymbol{\theta}$, $\Pr(D)$ is the *evidence*, and $X^{\mathrm{NS}}(\lambda)$ is the *prior volume* enclosed within likelihood contour $\Pr\left(D|M(\boldsymbol{\theta})\right) > \lambda$, and

$$X^{\mathrm{NS}}(\lambda) = \int_{\Pr(D|M(\boldsymbol{\theta}))>\lambda} \Pr\left(M(\boldsymbol{\theta})\right) \, \mathrm{d}\boldsymbol{\theta}. \tag{6}$$

With the correspondence $\boldsymbol{\theta} \equiv \boldsymbol{R}$, $\Pr\left(D|M(\boldsymbol{\theta})\right) \equiv f(V(\boldsymbol{R}))$, and taking $\Pr\left(M(\boldsymbol{\theta})\right)$ to be uniform over all of the available configuration space, eq. (4) is recovered exactly, and the prior volume becomes proportional to the configuration volume, eq. (3). Hence it is possible to define nested sampling in terms of configuration volumes.

Nested sampling works by determining the ratio in configuration volume for decreasing thresholds $\boldsymbol{V}^{\mathrm{NS}} = \{V_1^{\mathrm{NS}}, \cdots, V_{N^{\mathrm{NS}}}^{\mathrm{NS}}\}$. This list is generated by maintaining a set of n^{live} independent replicas (also known as *live points*) and then iteratively removing the highest energy replica (whereupon it becomes a *dead point*).

A new live point is generated randomly and uniformly within the configuration vol-

ume enclosed by the energy contour of the most recently removed dead point. This process means that the *configuration volume ratios* enclosed by the energy contours of successively removed dead points can be modelled by a set of independent beta-distributed variables (see appendix C.1 and eq. (71) for more details),

$$t_j^{\mathrm{NS}} = \frac{\Phi(V_j^{\mathrm{NS}})}{\Phi(V_{j+1}^{\mathrm{NS}})} \sim \mathcal{B}(n^{\mathrm{live}}, 1). \tag{7}$$

A beta distributed variable, $t_{\mathcal{B}} \sim \mathcal{B}(\alpha_{\mathcal{B}}, \beta_{\mathcal{B}})$, has a probability density

$$\Pr\left(t_{\mathcal{B}}\right) = \mathcal{B}(t_{\mathcal{B}}|\alpha_{\mathcal{B}}, \beta_{\mathcal{B}}) = \frac{\Gamma(\alpha_{\mathcal{B}} + \beta_{\mathcal{B}})}{\Gamma(\alpha_{\mathcal{B}})\Gamma(\beta_{\mathcal{B}})} t_{\mathcal{B}}^{\alpha_{\mathcal{B}}-1} \left(1 - t_{\mathcal{B}}\right)^{\beta_{\mathcal{B}}-1}, \tag{8}$$

for $0 \leq t_{\mathcal{B}} \leq 1$, where Γ is the gamma function. The moments of the beta distribution are

$$\mathbb{E}[t_{\mathcal{B}}^a (1 - t_{\mathcal{B}})^b] = \int_0^1 t_{\mathcal{B}}^a (1 - t_{\mathcal{B}})^b \mathcal{B}(t_{\mathcal{B}}|\alpha_{\mathcal{B}}, \beta_{\mathcal{B}}) \, \mathrm{d}t_{\mathcal{B}} = \frac{\Gamma(\alpha_{\mathcal{B}} + \beta_{\mathcal{B}})}{\Gamma(\alpha_{\mathcal{B}})\Gamma(\beta_{\mathcal{B}})} \frac{\Gamma(\alpha_{\mathcal{B}} + a)\Gamma(\beta_{\mathcal{B}} + b)}{\Gamma(\alpha_{\mathcal{B}} + a + \beta_{\mathcal{B}} + b)}. \tag{9}$$

For a more detailed overview of the beta distribution see appendix C.1.

An overview of the basic nested sampling algorithm is shown in algorithm 1, and a schematic for one step of a nested sampling run is shown in fig. 1.

Algorithm 1 Basic nested sampling

Input: n^{live}
Output: $\boldsymbol{V}^{\mathrm{max}}$
Initialise empty list $\boldsymbol{V}^{\mathrm{max}} = \{\}$
generate n^{live} independent replicas of live points with sorted energies $\boldsymbol{V} = \{V_1 < V_2 < \cdots < V_{n^{\mathrm{live}}}\}$
repeat
    Remove dead point $V_{n^{\mathrm{live}}}$ from $\boldsymbol{V}$ and append to $\boldsymbol{V}^{\mathrm{max}}$
    Generate new replica uniformly from prior with energy $V_{\mathrm{new}} < V_{n^{\mathrm{live}}}$
    Insert V_{new} into $\boldsymbol{V}$
until Convergence

For nested sampling, convergence is normally considered to have occurred when the value of the evidence integral (configuration integral) contained by the live points is less than some specified fraction of the total evidence (the partition function), or the energy difference between the highest and lowest live points is less than some tolerance.

If the likelihood is set to be the canonical probability of a configuration, and $\boldsymbol{\theta}$ is the configuration, then the evaluation of the evidence in eq. (5) is equivalent to the calculation of the partition function. Formulating the integral as a Lebesgue integral enables NS to effectively choose the optimal bin width during the course of a NS calculation.

Dynamic nested sampling During nested sampling it is possible to dynamically change the number of replicas being sampled by increasing the number of replicas in the energy/likelihood ranges that contribute most to the observables, which can be useful to improve the accuracy. The ratio of volumes enclosed by successive energies/likelihoods will still be beta distributed, but the number of live points can now change. This process is known as dynamic nested sampling.[20] Dynamic nested sampling runs can be combined by merging all the energies/likelihoods into a single sorted list. The ratio of volumes enclosed will again be beta distributed, with n^{live} equal to the sum of the live points in all the nested sampling runs considered.

In this framework every step of a nested sampling run can be viewed as the removal of the highest energy live point, followed by the addition of some number of live points sampled uniformly below the energy of the point removed. If no live points are added their number will decrease as the live point with the highest energy is successively removed, which is equivalent to removing multiple points at the same time. Here we will assume that in every step of a nested sampling run exactly one live point is removed, but there can be a dynamic number of live points.

1.1.1 Challenges

The key computational challenge associated with nested sampling is generating replicas uniformly from the configuration space (the prior) subject to the constraint that the energy/likelihood of the replicas is less/greater than a given cut-off.[21,22] Three key factors complicate this process.

- The configuration (prior) volume of interest is normally a tiny fraction of the total.
- The potential energy landscape has an exponentially large number of local minima, all corresponding to maxima of the likelihood.
- As the potential energy constraint decreases during the nested sampling simulation, regions in configuration space will become disconnected, and sampling across them becomes challenging.

A variety of approaches have been developed to tackle these problems.

- A hard constraint variant of HMC known as Galilean sampling, which exploits isolikelihood contours/potential gradient information[23,24] to permit long-range directed moves.
- MULTINEST[25] fits a set of intersecting ellipsoidal contours to the set of live points and then performs rejection sampling within the contours, although the efficiency of this method decreases with the dimensionality of the system.
- POLYCHORD[21] extends the slice sampling algorithms to multimodal distributions.
- Superposition enhanced nested sampling (SENS)[26] uses a population of low energy minima (corresponding to high likelihood) obtained using a global optimisation algorithm, such as basin-hopping (BH),[27–29] to propose moves to cross barriers that the MCMC walks cannot overcome. In exact-SENS, replicas are generated via Hamiltonian replica exchange;[3,8] in inexact-SENS new samples are generated at low energies

by approximating the potential function as a set of harmonic wells. Both of these approaches enable SENS to significantly improve the accuracy of the calculated density of states at lower energies, whilst needing fewer replicas than in a standard NS simulation.

1.1.2 Integration

Using nested sampling we can calculate an estimate of eq. (1) and its associated uncertainty[22] by calculating the first and second moments of $\mathcal{I}_{\Phi_0}[f]$. Suppose we have performed nested sampling and generated N^{NS} nested sampling points, where the jth point was sampled with n_j^{NS} live points present. We can approximate the integral eq. (4) by the sum

$$\mathcal{I}_{\Phi_0}[f] \approx \Phi_0 \sum_j^{N^{\mathrm{NS}}} V_j^{\mathrm{NS}}(\Phi(V_{j+1}^{\mathrm{NS}}) - \Phi(V_j^{\mathrm{NS}})) = \Phi_0 \sum_j^{N^{\mathrm{NS}}} f_j(1-t_j)\prod_{k=1}^{j-1} t_k, \tag{10}$$

where $f_j = f(V_j^{\mathrm{NS}})$. Assuming the quadrature error is negligible[19] and as all the volume ratios are independent, the expected value of $\mathcal{I}_{\Phi_0}[f]$ can be calculated straightforwardly from eq. (9),

$$\mathbb{E}_{\mathrm{NS}}[\mathcal{I}_{\Phi_0}[f]] = \Phi_0 \sum_{j=1}^{N^{\mathrm{NS}}} f_j \frac{1}{n_j^{\mathrm{NS}}} \prod_{k=1}^{j} \frac{n_k^{\mathrm{NS}}}{n_k^{\mathrm{NS}}+1}. \tag{11}$$

$\mathbb{E}_{\mathrm{NS}}[\mathcal{I}_{\Phi_0}[f]^2]$ can be found by adapting the method used by Keeton[22] to approxiate the uncertainty of estimates obtained using nested sampling runs with a fixed number of live points,

$$\mathbb{E}_{\mathrm{NS}}[\mathcal{I}_{\Phi_0}[f]^2] = \sum_{l=1}^{N^{\mathrm{NS}}} \left[\frac{2f_l}{n_l^{\mathrm{NS}}} \left(\prod_{k=1}^{l} \frac{n_k^{\mathrm{NS}}}{n_k^{\mathrm{NS}}+1} \right) \left(\sum_{j=1}^{l} \left(\frac{f_j}{n_j^{\mathrm{NS}}+1} \prod_{k'=1}^{j} \frac{n_{k'}^{\mathrm{NS}}+1}{n_{k'}^{\mathrm{NS}}+2} \right) \right) \right], \tag{12}$$

which can be calculated in $O(N^{\mathrm{NS}})$ operations. The statistical uncertainty for the estimate of $\mathcal{I}_{\Phi_0}[f]$ can be calculated as,

$$\sigma^2_{\mathcal{I}_{\Phi_0}[f]} = \mathbb{E}_{\mathrm{NS}}[\mathcal{I}_{\Phi_0}[f]^2] - \mathbb{E}_{\mathrm{NS}}[\mathcal{I}_{\Phi_0}[f]]^2. \tag{13}$$

1.2 Principle of superposition

It is common practice to split the configuration space into different regions and then tackle the integral eq. (1) in each region independently. Often different regions are associated with different minima, using the basin of attraction defined by a minimisation algorithm.[30–32] as the set of points that lead to the same minimum. Basins of attraction are only guaranteed to be contiguous when steepest-descent minimisation is used.[31,32] The boundary between two contiguous basins of attraction is a watershed[30] or transition surface.

The total density of states can then be described by the sum over basins of attraction of the minima to give the superposition partition function:[32–37]

$$\Omega(V) = \sum_{\mu \in \boldsymbol{R}_{\min}} P_\mu \Omega_\mu(V), \tag{14}$$

where P_μ is the number of distinguishable permutation-inversion isomers of minimum μ, and $\Omega_\mu(V)$ is the density of states for the corresponding basin of attraction.

This approach can be useful, because in many situations determining $\Omega_\mu(V)$ is more straightforward than determining the full density of states. At energies close to the minimum, the potential function can be well approximated by a harmonic potential, with an analytic density of states. Additionally, there are no barriers within a basin of attraction.

1.2.1 Harmonic superposition approximation

Given a database of minima, the fastest method for estimating the density of states is the harmonic superposition approximation (HSA)[37–40] where the density of states,

$$\Omega_\mu(V^I) \propto \theta(V^I - V_\mu^Q) \frac{(V^I - V_\mu^Q)^{\kappa/2-1}}{\overline{\nu}_\mu}, \tag{15}$$

and configuration volume,

$$\Phi_\mu(V^I) \propto \theta(V^I - V_\mu^Q)\frac{(V^I - V_\mu^Q)^{\kappa/2}}{\overline{\nu}_\mu}, \tag{16}$$

of each individual minimum μ are approximated by a harmonic potential with known analytic form, where θ is the Heaviside step function, V_μ^Q is the energy of minimum μ, κ is the number of vibrational degrees of freedom (the number of non-zero eigenvalues of the Hessian), and $\overline{\nu}_\mu$ is the geometric mean of the vibrational normal modes. The harmonic superposition partition function is simple to calculate and accurate at low temperatures, but at high temperatures anharmonic vibrational effects can introduce systematic errors.[37]

This approach has been combined with Wang–Landau sampling in the first Basin-Sampling scheme (BS)[9] and with parallel-tempering (PT) in the basin-sampling/parallel tempering (BSPT) method.[10]

2 Motivation

Standard nested sampling must be run with a minimum number of live points when simulating systems exhibiting broken ergodicity, so that there are a sufficient number of points in each basin once they become disconnected to ensure uniform sampling across the disconnected basins. Choosing the correct number of live points poses a challenge, as it is not obvious *a priori* what will be sufficient, and simulating a large number of live points is expensive and tricky to parallelise.

NBS tackles this problem by performing NS simulations with a single live point, with each new live point being spawned by a random walk originating from the previous live point. These simulations will be called nested optimisations (NOpts), as each simulation is guaranteed to finish in a minimum. This approach means that a given NOpt will never jump out the basin it is currently in, so the NOpts that are in the same basin can be combined together to provide an estimate of the configuration volume of that specific

basin. We describe NOpts in more detail in section 3.

Instead of inferring the volumes of disconnected basins by the number of live points present in a given basin, the volume of a given basin can be estimated from the fraction of NOpts that fall into it and the statistics of the aggregated NOpts associated with the basin. To ensure sufficient sampling across disconnected regions enough NOpts must be done to ensure sufficient statistics to estimate the probability of a NOpt landing in a specific basin, and there must be enough NOpts in each basin so that their aggregated statistics are sufficiently accurate. This approach has the advantage of being highly parallelisable, and does not require the number of live points to be chosen before beginning the simulation. The computational details for inferring the basin volumes and integrals over the potential energy surface (PES) are discussed below in section 4.

In this superposition based approach basins are considered and sampled separately. In contrast SENS uses the harmonic approximation to the potential to seed low energy replicas into the simulation.[26] Similar superposition-based approaches are used by MULTINEST and POLYCHORD, although in these codes it is assumed that it is possible to cluster the configuration space into either disconnected regions or a set of hyperellipsoids, which is not always straightforward for an arbitrary PES or a high-dimensional system.

The behaviour of regions in configuration space becoming mutually inaccessible can be visualised using a disconnectivity graph (DG),[41–44] which shows the energy level above which minima become *connected* and is illustrated in fig. 2. The exact definition of connectivity can vary, leading to alternative DG representations. For the standard definition, two minima are said to be connected at a given energy threshold if there exists a sequence of transition states connecting them that are all below the threshold. The structure of a DG can provide insight into the dynamical behaviour of the system being studied.[45]

In the NBS sampling considered here any random walk constrained to stay below the energy of a given *node* will only be able to visit the volume of space associated with whichever *branch* the random walk begins in. An *edge* of the correspinding NBS DG

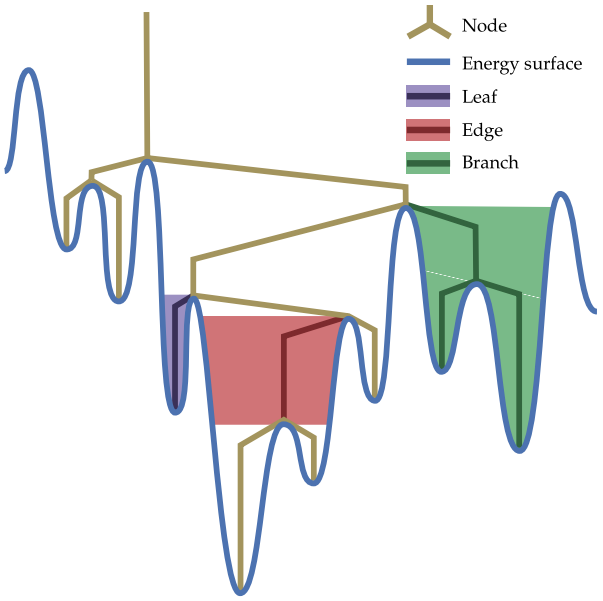


Figure 2: Classification scheme for an NBS disconnectivity graph. Two points in space are considered to be connected if it is possible to move between them without exceeding the energy of the higher point. The graph describes the topology of the connectivity induced by NBS for an energy landscape. Nodes in the graph indicate the energy threshold at which different *child* branches in the energy landscape become connected. A *branch* volume of a node corresponds to a region that becomes connected to other branches above the node.

represents a region in space where all pairs of points in the region are connected by a barrierless path, so the energy does not exceed the higher value of the pair. We will refer to different regions or volumes of configuration space as leaves, edges or branches, defined below. In NBS it is these regions that are considered.

The DG we employ for NBS differs from standard DGs, where nodes correspond to the minimum energy transition state that connects two branches descending from that node. In NBS graphs the nodes correspond to the energy level at which MC walks to generate new replicas in the two regions do not cross the barrier between them, which happens once the probability of the MC walk moving between the regions becomes too small. These barriers will be termed *lazy* barriers to differentiate them from true potential energy barriers, because the MC walk has not been run for 'long enough' to cross them. For brevity, we will henceforth refer to lazy basins simply as basins. An approach to

self-consistently infer the resulting DG from just the NOpt results without needing a PES specific similarity metric is described in section 8.2.

NBS has some advantages over standard nested sampling,

- the NOpt simulations are embarrassingly parallel,
- it is easy to perform additional simulations to increase the accuracy,
- by modelling the volumes of the basins independently it is possible to enhance the accuracy at low energies using the harmonic superposition approximation to calculate the configuration volume at low energies, in a similar manner to basin-sampling.[10]

Unfortunately, these advantages come at the cost of creating more stringent requirements for the generation of new live points. Ensuring that the new live points are sufficiently decorrelated from the previous point becomes more important as the number of live points in the simulation decreases.

To alleviate these issues we introduce NoGUTS in section 6 to facilitate more efficient generation of new replicas within a basin. In addition, the selection of an appropriate step size during the simulation is important to ensure that each new live point is generated efficiently. However naively changing the step size can induce bias.[46,47]

In section 7 we describe a scheme to enable selection of an appropriate step size during the course of a NOpt, which mostly nullifies the sampling bias that can occur when the step size is adjusted during a MC simulation. A logistic model relating the acceptance rate to the cut-off energy and step size is used to choose an efficient step size, with a delay to reduce history dependence.

3 Nested optimisation

The NBS DG can be sampled by *nested optimisation* (NOpt), defined as NS with only a single live point. Performing NS with a single live point means that new replicas are always spawned by MC walks of length $N_{\mathrm{MC}}^{\mathrm{opt}}$ starting from the location of the last live point, which means that the NS will never jump across a lazy barrier. As the nested optimisation run continues it will therefore descend the NBS DG, sampling all the edges connected from the starting edge, to the minimum that it finishes in (see fig. 2).

A single NOpt run will not provide good statistics about the configuration volume of the edges it samples, but as more nested optimisation runs are completed and merged (see section 1.1) a more accurate picture of the configuration volume of the edges can be built. Furthermore, at any given node, the relative volumes of each of the child edges of the node can be estimated by analysing the statistics of the number of NOpt runs that fall into each edge, which is discussed in detail in appendices A.1 and A.2.

Each nested optimisation run is completely independent, so these calculations are embarrassingly parallel.

3.1 Local sampling close to a minimum

The chance of a random nested optimisation run finishing in a specific minimum will be extremely small for most of the minima, which means that the density of states will be rather uncertain before the basin has merged with other basins, as estimated by the NS at energies close to the minimum. To decrease this uncertainty the local basin of a minima can be sampled by performing traditional NS or NOpts, except that all new replicas are generated by random walks beginning at the minimum itself. This process may only work up to a certain energy level, as the random walks to generate new live points may cease to be ergodic. We will refer to this process as local sampling, to indicate that it samples only the section of the disconnectivity tree local to the minimum.

3.2 Stopping criterion

Many nested sampling algorithms use the statistics of the live points to determine a stopping criterion for the simulation, commonly when the energy difference between the highest and lowest energy replica decreases below some energy tolerance. In NBS, this approach would not work, as there is only ever one live point. Instead the statistics of the dead points can be considered. For example, the expected difference in energy, $V_{\text{tol}}^{\text{opt}}$, between the current live point and the one sampled $N_{\text{stop}}^{\text{opt}}$ iterations ago, will be approximately equal to the energy difference in a $2^{N_{\text{stop}}^{\text{opt}}}$ live point standard NS simulation at the same energy cut-off, which makes this comparison an effective termination criterion for an NOpt.

4 Nested basin-sampling calculations

Suppose we have performed a set of nested optimisation runs for a PES, and we know the path that every run took during the simulation. Using these results we can proceed with a calculation similar to that described in section 1.1.2 to calculate the global properties of the PES, as in eq. (4).

4.1 Notation

An edge volume, $\Phi_{\beta_{\delta+1}}^{\beta_\delta}$, can be defined with respect to the NBS DG by its parent node, β_δ, and child node, $\beta_{\delta+1}$; where δ is the depth of the node on the NBS DG, and β indexes the siblings of the node. The *branch volume* associated with a node, β_δ, can be defined as $\Phi_{\beta_\delta} = \Phi_{\beta_\delta}^{\beta_{\delta-1}} + \sum_{\beta_{\delta+1}}(\Phi_{\beta_{\delta+1}})$, see figs. 2 and 3. The edge $\Phi_{\beta_\delta}^{\beta_{\delta-1}}$ has $N_{\text{NS}}^{\beta_\delta}$ total dead points and the jth point of the aggregated runs has energy $V_j^{\beta_\delta}$ and $n_j^{\beta_\delta}$ live points present. $M_{\beta_{\delta+1}}^{\beta_\delta}$ runs fall from branch Φ_{β_δ} into $\Phi_{\beta_{\delta+1}}$. $\Phi_{\beta_{\delta+1}}(V)$ is the configuration volume in the branch $\Phi_{\beta_{\delta+1}}$ with energy less than V, so $\Phi_{\beta_{\delta+1}}(V_0^{\beta_\delta}) \equiv \Phi_{\beta_{\delta+1}}$. The configuration volume ratio of the branch is $t_j^{\beta_\delta} = \Phi_{\beta_\delta}(V_{j-1}^{\beta_\delta})/\Phi_{\beta_\delta}(V_j^{\beta_\delta})$.

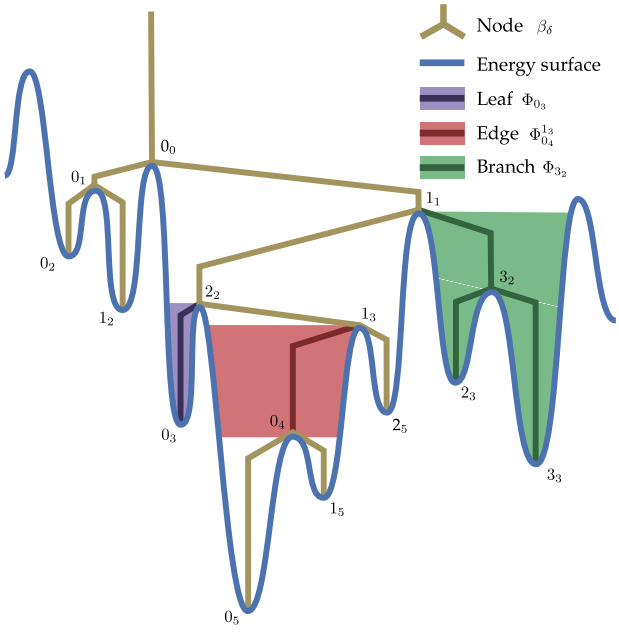


Figure 3: The notation scheme for a NBS disconnectivity graph.

4.2 Estimating basin configuration volumes

If the NBS DG is known and there is a set of NOpt runs, the configuration volume of the edges can be estimated by two complementary methods.

1. The configuration volumes of each edge of the NBS DG can be calculated using NS, with the additional slight complication that the relative configuration volumes of edges connected to any given node are modelled by the Dirichlet distribution, the multinomial generalisation of the beta distribution (see appendix C.2). We assume that the process by which the NOpts fall into different edges can be modelled as a multinomial process with the multinomial probabilities proportional to the volume of the child edge at the energy at which the edges become disconnected during a NOpt. This method is termed the *top down* approach.

2. Because the configuration volume has been separated into different regions it is also possible to estimate the configuration volume near a minimum basin (leaf) using the harmonic superposition approximation, eq. (16). The relative volumes of *higher* energy levels can then also be estimated from the aggregated NOpt runs, calculating

the configuration volumes *bottom-up*. At each node the volume of the higher edge will equal the sum of the volumes of the child edges. This calculation enhances the relative basin size estimates at low energy. This method is termed the *bottom up* approach.

The mathematical details for computing these top-down and bottom-up estimates of the configuration volume are described in section 4, given a known NBS DG and aggregated NS results for each edge. The volume inside a branch can be calculated either in terms of the parent node branch and edge volumes in a top down manner, analogous to a standard NS simulation, or in terms of the child node branch volumes in a bottom up approach, where the HSA is used to calculate the configuration volume at low energies.

The bottom up approach can be seen as been broadly equivalent to the strategy used by BS[9] and BSPT[10] to connect the results of a higher temperature simulation to the low-temperature accuracy of the HSA.

4.2.1 Top down approach

The configuration volume inside a branch can be calculated from the configuration volume ratios,

$$\Phi_{\beta_{\delta+1}}(V_j^{\beta_{\delta+1}}) = \Phi_{\beta_{\delta+1}} \prod_{k=1}^{j} t_k^{\beta_{\delta+1}}. \tag{17}$$

The branch volume can be recursively calculated *top down* from its parent branch,

$$\Phi_{\beta_{\delta+1}} = p_{\beta_{\delta+1}}^{\beta_\delta} \left(\Phi_{\beta_\delta} - \Phi_{\beta_\delta}^{\beta_{\delta-1}} \right), \tag{18}$$

where $p_{\beta_{\delta+1}}^{\beta_\delta}$ is the *branch probability* of a NS run going from branch Φ_{β_δ} to $\Phi_{\beta_{\delta+1}}$. The branch probability will be Dirichlet distributed over the indexes, $\beta_{\delta+1}$,

$$p_{\beta_{\delta+1}}^{\beta_\delta} \sim \text{Dir}(M_{\beta_{\delta+1}}^{\beta_\delta}). \tag{19}$$

The edge volume for a node can be calculated from eq. (17),

$$\Phi^{\beta_\delta}_{\beta_{\delta+1}} = \Phi_{\beta_{\delta+1}} - \Phi_{\beta_{\delta+1}}(V^{\beta_{\delta+1}}_{N^{\beta_\delta}_{\mathrm{NS}}}). \tag{20}$$

The precise procedure by which top down estimates of these volumes can be calculated is given in appendix A.1.

4.2.2 Bottom up approach

The volume of a branch, Φ_{β_δ}, can be expressed in terms of the sum of its child branches (which are themselves calculated bottom up from their children) and the configuration volume ratios:

$$\Phi_{\beta_\delta}(V^{\beta_\delta}_j) = \left(\sum_{\beta_{\delta+1}} \Phi_{\beta_{\delta+1}}\right) \prod_{k=j}^{N^{\beta_{\delta+1}}_{\mathrm{NS}}} \frac{1}{t^{\beta_{\delta+1}}_k}. \tag{21}$$

We can calculate bottom up estimates of this volume from eq. (7), as described in appendix A.2.

To perform this calculation the configuration volume at the bottom of the leaves needs to be known. These volumes can be calculated using the HSA for the minimum configuration volume in eq. (16) for each of the leaves. The HSA will be most accurate for energies close to the minimum, and we show below in section 5.2 how this range can be determined.

4.2.3 Integration

The integral of $f(V(\boldsymbol{x}))$ over Φ_0 will be equal to the sum of the integrals over all the edges of the disconnectivity graph,

$$\mathcal{I}_{\Phi_0}[f] = \sum_{\beta_\delta, \beta'_{\delta+1}} \mathcal{I}_{\Phi^{\beta_\delta}_{\beta'_{\delta+1}}}[f]. \tag{22}$$

As the derivation is somewhat involved we demonstrate how to compute the first and second moments of $\mathcal{I}_{\Phi_0}[f]$ for the top down approach in appendix A.1, following a similar process to that described in section 1.1.2.

In section 4.3 we show how estimates of the basin volumes from both approaches can be combined to create a more accurate overall estimate of the integral.

4.3 Interpolating between the top-down and bottom-up calculations

The second moments calculated for the top-down and bottom-up procedures in eqs. (47) and (64) can be used to obtain a weighted sum of the two results that naturally incorporates the associated uncertainty with either calculation to produce the best overall estimate of the basin volumes. As the configuration volumes for both procedures are calculated from the product of a set of independently distributed variables, the overall configuration volume was calculated by a weighted sum of logarithms,

$$\mathbb{E}_{\mathrm{I}}\left[\ln\left(\Phi^{\beta_\delta}_{\beta_{\delta+1}}(V_j^{\beta_{\delta+1}})\right)\right] \approx \frac{\left(\dfrac{\ln\left(\mathbb{E}_{\mathrm{D}}\left[\Phi^{\beta_\delta}_{\beta_{\delta+1}}(V_j^{\beta_{\delta+1}})\right]\right)}{w_{\mathrm{D}}(V_j^{\beta_{\delta+1}})} + \dfrac{\ln\left(\mathbb{E}_{\mathrm{U}}\left[\Phi^{\beta_\delta}_{\beta_{\delta+1}}(V_j^{\beta_{\delta+1}})\right]\right)}{w_{\mathrm{U}}(V_j^{\beta_{\delta+1}})}\right)}{\left(\dfrac{1}{w_{\mathrm{D}}(V_j^{\beta_{\delta+1}})} + \dfrac{1}{w_{\mathrm{U}}(V_j^{\beta_{\delta+1}})}\right)}, \tag{23}$$

where $\mathbb{E}_{\mathrm{D/U}}\left[\Phi^{\beta_\delta}_{\beta_{\delta+1}}(V_j^{\beta_{\delta+1}})\right]$ is the expected configuration volume [calculated top down (D) or bottom up (U)] of all the basins connected to $\beta_{\delta+1}$ up to an energy of $V_j^{\beta_{\delta+1}}$, and $\mathbb{E}_{\mathrm{I}}$ is the expectation value of the interpolation.

The chosen weighting was the logarithmic ratio of the second moment to the square of the first moment,

$$w_{\mathrm{U/D}}(V_j^{\beta_{\delta+1}}) = \ln\left(\frac{\mathbb{E}_{\mathrm{U/D}}\left[\Phi^{\beta_\delta}_{\beta_{\delta+1}}(V_j^{\beta_{\delta+1}})^2\right]}{\mathbb{E}_{\mathrm{U/D}}\left[\Phi^{\beta_\delta}_{\beta_{\delta+1}}(V_j^{\beta_{\delta+1}})\right]^2}\right), \tag{24}$$

which is an effective approximation to the uncertainty in the logarithmic volume.

First, the relative scale factor between the bottom-up and top down calculations needs to be determined. In theory this scale factor could be estimated by calculating the exact configuration volume corresponding to vibrational degrees of freedom for each harmonic basin. However, in this implementation of NBS it was found that instead, the relative factor could be determined by minimising the logarithmic difference between the top-down and bottom-up basin volumes for each of the leaves, at the harmonic energy, as calculated in section 5.2, scaled by the sum of the weights, as calculated in eq. (24).

This scheme unfortunately does not smoothly connect the configuration volumes above and below a node, so to ensure a smooth interpolation, the value of $\mathbb{E}_{\mathrm{U}}\left[\Phi_{\beta_\delta}\right]$ can be adjusted slightly to ensure that

$$\mathbb{E}_{\mathrm{I}}\left[\Phi_{\beta_\delta}^{\beta_{\delta-1}}(V_{N_{\mathrm{NS}}^{\beta_\delta}}^{\beta_\delta})\right] = \sum_{\beta_{\delta+1}} \mathbb{E}_{\mathrm{I}}\left[\Phi_{\beta_{\delta+1}}^{\beta_\delta}(V_{N_{\mathrm{NS}}^{\beta_\delta}}^{\beta_\delta})\right]. \tag{25}$$

With the above caveats the trapezium rule can then be used to evaluate eq. (4) using eq. (23) on each edge and then summing the results.

5 Determining the NBS disconnectivity graph

To calculate the configuration volume, as described in section 4, the NBS DG first needs to be known. While it may be possible to adapt the procedure used by Pártay et al.[48] to generate landscape charts, this approach to detecting disconnecting regions would require the configurations generated by the NS to be saved, which significantly increases the storage demands of the method, and requires an appropriate similarity metric specific to the problem at hand.

An alternative approach was developed for this work, where different basins are merged together at the energy level above which the configuration volume estimated by NS for each basin looks identical, which avoids storing the configurations of the dead

points and the specification of a problem specific metric.

This approach does not guarantee that the DG generated will accurately represent the true NBS DG, as it only merges basins when the configuration volumes appear identical. However, the overall density of states produced should not be affected by the merge, ensuring that the method produces self-consistent results. The mergers primarily serve to decrease the *uncertainty* of the configuration volume estimates.

5.1 Comparing basin volumes

The configuration volume of two different basins can be compared by Bayesian model comparison, where the evidence [see eq. (5)] of two models is compared. The hypothesis that the basin volumes are the same can be compared against the probability that they are different. Suppose there are two different basins, β_δ, β'_δ, connected to the same parent basin, $\beta_{\delta-1}$. The basins should merge at the energy above which the density of states appears identical for each basin. To find this energy threshold the configuration volume ratio at two different energy levels, V_j, V_{j+1}, was modelled as a beta-distributed variable [see eq. (68)],

$$\frac{\Phi_{\beta_\delta}^{\beta_{\delta-1}}(V_j)}{\Phi_{\beta_\delta}^{\beta_{\delta-1}}(V_{j+1})} = t_{\beta_\delta}^{\beta_{\delta-1}}(V_j) \sim \mathcal{B}\left(a_{\beta_\delta}^{\beta_{\delta-1}}(V_j), b_{\beta_\delta}^{\beta_{\delta-1}}(V_j)\right), \tag{26}$$

whose parameters can be estimated from the first and second moments of the configuration volume using eqs. (72) and (73).

We can interpret $a_{\beta_\delta}^{\beta_{\delta-1}}(V_j)$ and $b_{\beta_\delta}^{\beta_{\delta-1}}(V_j)$ as binomial pseudocounts of uniformly sampled points in $\Phi_{\beta_\delta}^{\beta_{\delta-1}}$ with an energy cut-off of V_j, where $a_{\beta_\delta}^{\beta_{\delta-1}}(V_j)$ are the number of points observed to have energy greater than V_{j+1} and $b_{\beta_\delta}^{\beta_{\delta-1}}(V_j)$ less than. It is possible to generate true count data from the individual results from the NOpts runs in this region, but the statistical properties of these count results would not be as good.

The *maximum a posteriori* (*map*) estimate of the merge energy,

$$V_{\text{merge}}^{\beta_\delta=\beta'_\delta} = \arg\max_{V_j} \left[\prod_{V_{j'}<=V_j} \Pr\left(t_{\beta_\delta}^{\beta_{\delta-1}}(V_{j'}) = t_{\beta'_\delta}^{\beta_{\delta-1}}(V_{j'})\right) \prod_{V_{j''}>V_j} \Pr\left(t_{\beta_\delta}^{\beta_{\delta-1}}(V_{j''}) \neq t_{\beta'_\delta}^{\beta_{\delta-1}}(V_{j'})\right) \right], \tag{27}$$

of two branches, Φ_{β_δ} and $\Phi_{\beta'_\delta}$ can be obtained by modelling the fitted beta parameters of $t_{\beta_\delta}^{\beta_{\delta-1}}(V_j)$ and $t_{\beta'_\delta}^{\beta_{\delta-1}}(V_j)$ from eq. (26) as binomial pseudocounts. The evidence of the models is calculated as follows,

$$\Pr\left(t_{\beta_\delta}^{\beta_{\delta-1}} = t_{\beta'_\delta}^{\beta_{\delta-1}}\right) = \int_0^1 \text{Bin}\left(a_{\beta_\delta}^{\beta_{\delta-1}} | a_{\beta_\delta}^{\beta_{\delta-1}} + b_{\beta_\delta}^{\beta_{\delta-1}}, t_{\beta_\delta}^{\beta_{\delta-1}}\right) \text{Bin}\left(a_{\beta_{\delta'}}^{\beta_{\delta-1}} | a_{\beta_{\delta'}}^{\beta_{\delta-1}} + b_{\beta_{\delta'}}^{\beta_{\delta-1}}, t_{\beta_\delta}^{\beta_{\delta-1}}\right) \mathcal{B}(t_{\beta_\delta}^{\beta_{\delta-1}} | a_{\text{prior}}, b_{\text{prior}})\, \mathrm{d}t_{\beta_\delta}^{\beta_{\delta-1}} \tag{28}$$

$$\begin{aligned}\Pr\left(t_{\beta_\delta}^{\beta_{\delta-1}} \neq t_{\beta'_\delta}^{\beta_{\delta-1}}\right) = &\int_0^1 \text{Bin}\left(a_{\beta_\delta}^{\beta_{\delta-1}} | a_{\beta_\delta}^{\beta_{\delta-1}} + b_{\beta_\delta}^{\beta_{\delta-1}}, t_{\beta_\delta}^{\beta_{\delta-1}}\right) \mathcal{B}(t_{\beta_\delta}^{\beta_{\delta-1}} | a_{\text{prior}}, b_{\text{prior}})\, \mathrm{d}t_{\beta_\delta}^{\beta_{\delta-1}} \\ &\times \int_0^1 \text{Bin}\left(a_{\beta_{\delta'}}^{\beta_{\delta-1}} | a_{\beta_{\delta'}}^{\beta_{\delta-1}} + b_{\beta_{\delta'}}^{\beta_{\delta-1}}, t_{\beta'_\delta}^{\beta_{\delta-1}}\right) \mathcal{B}(t_{\beta'_\delta}^{\beta_{\delta-1}} | a_{\text{prior}}, b_{\text{prior}})\, \mathrm{d}t_{\beta'_\delta}^{\beta_{\delta-1}}, \end{aligned} \tag{29}$$

where for brevity we have dropped the argument of V_j from the pseudocounts, $\text{Bin}(m|n,p)$ is the binomial distribution of observing m successes from n trials with probability of success of p, and a_{prior} and b_{prior} are the parameters of the prior beta distribution over the volume ratio. The maximum entropy uninformative prior is $a_{\text{prior}} = b_{\text{prior}} = 1/2$. A full analytic derivation of these results is given by eqs. (81) and (82) in appendix D.

This approach allows us to self-consistently merge different edges on the NBS DG, as the edges only merge when their densities of states are sufficiently similar. Using eq. (84) it is also possible to consider merging multiple edges simultaneously.

The above procedures for merging the different basins require an ordered list of energy levels to compare all the separate basins. This list was generated by choosing energy

levels evenly spaced in the logarithm of the configuration volume of the aggregated runs for all the separate basins, so the configuration volume ratio would be approximately constant as the energy levels decreased. In this work a volume ratio of 0.5 was chosen, so V_j corresponds to the expected energy of a new live point generated with energy cut-off V_{j+1}.

5.2 Determining the harmonic energy range

To start the bottom-up procedure we must calculate the energy range over which a minimum, μ, is treated as harmonic. Here a similar procedure was performed as described above. A set of energy levels is defined, $V_j < V_{j+1}$, evenly spaced by the logarithm of the *harmonic configuration volume*, $t_{\text{harm}} = (V_j - V_\mu^Q)^{\kappa/2} \,/ (V_{j+1} - V_\mu^Q)^{\kappa/2}$. The probability that the NBS volume ratio and harmonic volume ratio are the same can be calculated,

$$\Pr\left(t_{\beta_\delta}^{\beta_{\delta-1}}(V_j) = t_{\text{harm}}\right) = \text{Bin}(a(V_j)|a(V_j) + b(V_j), t_{\text{harm}}), \tag{30}$$

and the probability they are different,

$$\Pr\left(t_{\beta_\delta}^{\beta_{\delta-1}}(V_j) \neq t_{\text{harm}}\right) = \int_0^1 \text{Bin}\left(a_{\beta_\delta}^{\beta_{\delta-1}}|a_{\beta_\delta}^{\beta_{\delta-1}} + b_{\beta_\delta}^{\beta_{\delta-1}}, t_{\beta_\delta}^{\beta_{\delta-1}}\right) \mathcal{B}(t_{\beta_\delta}^{\beta_{\delta-1}}|a_{\text{prior}}, b_{\text{prior}})\, \mathrm{d}t_{\beta_\delta}^{\beta_{\delta-1}}. \tag{31}$$

The full analytic derivation of these results is given in eqs. (81) and (82). The *map* estimate of the harmonic energy level,

$$V_\mu^{\text{harm}} = \arg\max_{V_j} \left[\prod_{V_{j'} \leq V_j} \Pr\left(t_{\beta_\delta}^{\beta_{\delta-1}}(V_{j'}) = t_{\text{harm}}\right) \prod_{V_{j''} > V_j} \Pr\left(t_{\beta_\delta}^{\beta_{\delta-1}}(V_{j''}) \neq t_{\text{harm}}\right) \right], \tag{32}$$

can then be found.

5.3 Comparing local sampling to aggregated NOpts

Before aggregating the results from local sampling as described in section 3.1 with the results from standard aggregated NOpts, it is important to check the energy range over which both methods produce similar estimates of the density of states. The *map* estimate can be calculated as in eq. (27), except that inequality signs are swapped, as we expect the density of states to diverge as the energy *increases*.

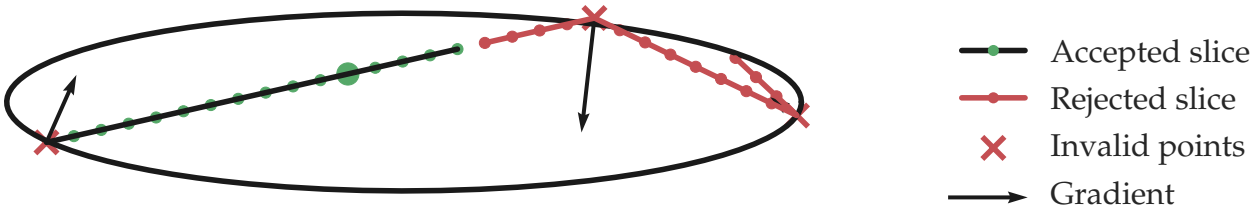


Figure 4: A graphic showing how NoGUTS generates slices. The large green point shows the starting point. The red slice represents a proposed slice to double the length of the current slice. However, because the red slice has a u-turn present, the proposal to extend the slice is rejected and the algorithm quits.

6 The No Galilean U-Turn Sampler

Here we describe the No Galilean U-Turn Sampler (NoGUTS), which is a modification of the No U-Turn Sampler (NUTS)[1,13] to work with Galilean sampling (see section 1.1).[2,23,24] NoGUTS can be viewed as a form of multivariate reflective slice sampling[49] with an automatic stopping criterion, and as such we will refer to the trajectories generated by the algorithm as *slices*.

Several modifications have been made to the NUTS algorithm to make it work for the problem at hand.

- The leapfrog integration step has been replaced with the Galilean sampling equivalent step, which is described in algorithm 4.
- Each point, $\boldsymbol{R}'$, on a slice has a forward $\boldsymbol{p}'_+$ and a backwards velocity $\boldsymbol{p}'_-$ associated with it, to account for reflections in direction at invalid points. This modifica-

tion primarily affects the Galilean step, as described in algorithm 4, but also introduces some additional bookkeeping in the BuildTree and NoGUTS algorithms, as described in algorithms 2 and 3, compared to NUTS.

- The ability to include constraints in the simulation has been incorporated, described in algorithm 5. Constraints can be most straightforwardly incorporated by rejecting any slice that violates the constraints with a simple test function TestConstraint($\boldsymbol{R}'$), which returns true if $\boldsymbol{R}'$ satisfies the constraints, and false otherwise. However, by introducing a 'constraint potential', $V_{\text{constraint}}(\boldsymbol{R})$, which is 0 for all configurations that satisfy the constraint and positive for invalid configurations, the NoGUTS simulation can reflect off the constraint boundaries in addition to the energy cut-off boundaries. If the configuration encountered violates both the potential cut-off and the constraint, the algorithm as described reflects off the sum of the normalised gradient and constraint gradient at that point. The NoGUTS simulation could also reflect off just the gradient or the constraint gradient and maintain detailed balance. It is straightforward to construct continuous constraint potentials for hard sphere constraints, by summing the excess radius of all the points that exceed the radius of the hard sphere.

- In NUTS the points to include in the trajectory are determined by a 1-D slice sampling process, however in the case of sampling from a hard constraint, all points that are valid can be straightforwardly included in the slice generated.

- The stopping criterion also needs to be slightly modified to account for the multivalued velocities, so for the positive direction the positive velocity must be used and vice versa for the negative direction.

- Here we consider atomic clusters with zero angular and linear overall momentum, so before the stopping criterion was calculated any net linear and angular momentum was removed.

6.1 Overview

Galilean sampling can be a very efficient method for exploring a hard constraint space as it allows long-range directed moves away from the starting point. The choice of simulation length for Galilean sampling is extremely important, as the reflective nature of the movement can cause the replica to start moving back to its starting point, significantly reducing the efficiency.

NoGUTS (and NUTS) enable the simulation to detect when this process occurs and stop, whilst *maintaining detailed balance*. The algorithm works by recursively doubling a slice of points, or equivalently building a binary tree, until it reaches a stopping criterion, as shown in fig. 4. For each iteration the algorithm selects forwards or backwards directions randomly and then attempts to build a slice of equal length in that direction. If at any point of building the new slice the algorithm detects that the stopping criterion would have been satisfied then NoGUTS stops and rejects the new slice. It rejects this new slice because the probability of moving from it to the current slice would be zero, as a NoGUTS simulation starting from the new slice would terminate before adding the current slice. If the new slice is successfully added to the current slice then the stopping criterion can be tested again on the new combined slice, and then the process can be repeated. This procedure ensures that each valid point on the slice has an equal probability of generating an identical slice, preserving detailed balance. To ensure that the algorithm terminates in a reasonable time a maximum recursion depth, j_{depth}, can be specified.

The algorithm does not need to store all the valid points as the slice is generated. Instead it maintains for every sub-slice its associated selected point, which has been randomly chosen uniformly out of the valid points in that sub-slice. When joining two sub-slices the algorithm will randomly pick a selected point from one of the sub-slices with probability equal to the number of valid points in that sub-slice, ensuring that the selected point of the new slice has been selected uniformly from the union of valid points of for the pair.[1]

This process of implicitly building the slice is performed by the recursive BuildTree function described in algorithm 3, which is called by the NoGUTS algorithm, described in algorithm 2, to progressively double a slice until the stopping criterion is reached.

7 Adapting the step size

During the course of a NOpt run the energy cut-off and valid region of space will vary drastically, so the optimal step size to maintain an efficient acceptance rate will tend to decrease by several orders of magnitude during the course of the run. To cope with this variation we can define a simple logistic model for predicting the acceptance probability, p_{acc}, for a given step size, δ, at a given energy cut-off, V_{cut},

$$p_{\text{acc}}(\delta, V_{\text{cut}}) = \frac{1}{1 + \exp\left(m_{\text{acc}} V_{\text{cut}} + c_{\text{acc}}\right)\delta}. \tag{33}$$

Rearranging we find

$$m_{\text{acc}} V_{\text{cut}} + c_{\text{acc}} = \operatorname{logit} p_{\text{acc}} + \ln \delta, \tag{34}$$

where $\operatorname{logit} x = \ln x - \ln(1 - x)$. This result suggests that the appropriate step size for a NoGUTS simulation can be chosen by performing an appropriate linear fit to the previous simulation results. Suppose during a simulation with cut-off energy V_t, and step size δ_t, that n_t^{acc} moves finish below V_t and n_t^{rej} finish above V_t. Then we can calculate the expected value,

$$\mathbb{E}\left[\operatorname{logit} p_{\text{acc}}(\delta_t, V_t)\right] = \psi_0(n_t^{\text{acc}}) - \psi_0(n_t^{\text{rej}}), \tag{35}$$

by modelling $p_{\text{acc}} \sim \mathcal{B}(n_{\text{accept}}, n_{\text{reject}})$, where $\psi_0(x) = d \ln(\Gamma(x))/\, dx$ is the digamma function. This model enables us to predict an appropriate step size for a given energy cut-off.

7.1 Avoiding non-Markovian dynamics

If the step size of a MCMC simulation is adjusted without due care the simulation may cease to be Markovian.[46,47] However, during an NOpt the step size must be adjusted quite drastically, as the energy cut-off decreases to maintain an efficient acceptance rate.

One method to significantly reduce any sampling artifacts generated by adapting the step size is to introduce a delay, $N_{\text{delay}}^{\text{opt}}$, for incorporating the accept/reject statistics to the above model, so that the replica in NBS has time to completely move away from the regions used to determine the optimal step size.

Additionally, to avoid biasing this model with high energy points, a rolling window of length $N_{\text{window}}^{\text{opt}}$ can be applied, so that only the last $N_{\text{window}}^{\text{opt}}$ dead points generated (and the associated lag introduced by the delay) are used to choose the step size for the NoGUTS simulation.

8 Results

The Lennard-Jones (LJ) potential is a simple representation for the energy of a pair of atoms:[50]

$$V_{\text{LJ}}(r) = 4\,\epsilon_{\text{LJ}}\left[\left(\frac{\sigma_{\text{LJ}}}{r}\right)^{12} - \left(\frac{\sigma_{\text{LJ}}}{r}\right)^{6}\right], \tag{36}$$

where ϵ_{LJ} and $2^{1/6}\sigma_{\text{LJ}}$ are respectively the pair equilibrium depth of the potential well and separation. When applied to homoatomic systems both ϵ_{LJ} and σ_{LJ} can be set equal to unity to make the potential dimensionless without loss of generality. The heat capacity has been investigated for LJ clusters at a range of sizes,[3,8,10,26,51–55] which makes them useful model systems for benchmarking.

Clusters of 31 LJ atoms (LJ_{31}) have been studied by a variety of different approaches, as this is the smallest LJ cluster to exhibit a solid-solid heat capacity peak at low temperatures and a solid-liquid peak at higher temperature. To accurately reproduce both

thermodynamic features requires effective sampling, both at the lowest energies to accurately reproduce the solid-solid peak, and at higher energies to reproduce the solid-liquid peak.

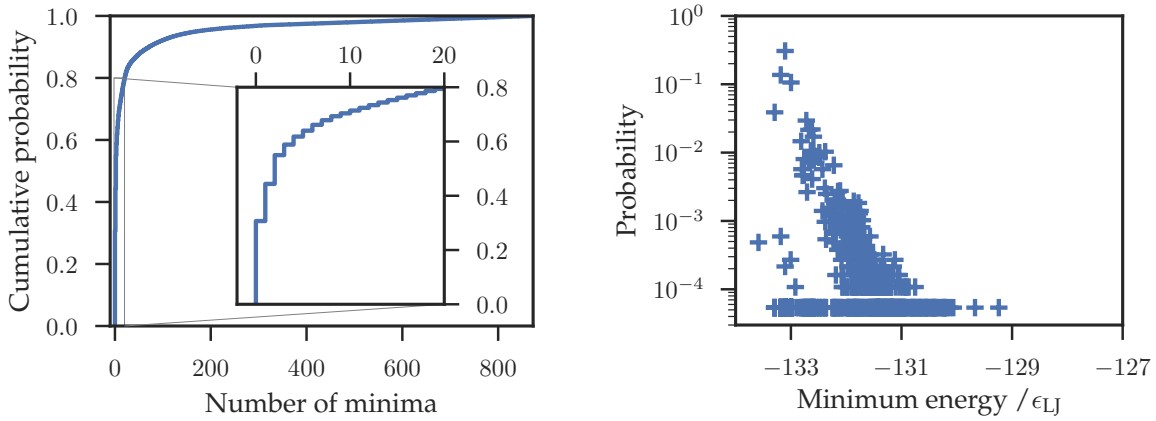
To avoid evaporation of the cluster the atoms were constrained to stay within a sphere of radius 2.5 σ_{LJ}, as in previous studies.[10,26] 20,000 independent NOpts were performed. In addition, for each of the four lowest energy minima, local NS, as described in section 3.1, was performed with 1000 live points. Identical minima were detected by performing 100 iterations of the Go-PERMDIST algorithm.[56]

Each new live point was generated by 20 iterations of NoGUTS with a max tree depth of $j_{\mathrm{max}} = 8$. Overall angular and linear momentum were removed from the velocities before evaluating the NoGUTS stopping criterion. The target acceptance ratio was chosen to be $p_{\mathrm{acc}} = 0.5$, the optimal step size was determined over a window of $N^{\mathrm{opt}}_{\mathrm{window}} = 100$ points, with a delay of $N^{\mathrm{opt}}_{\mathrm{delay}} = 20$ iterations to avoid non-Markovian behaviour. The simulation was stopped when the energy difference between the live point and the dead point from $N^{\mathrm{opt}}_{\mathrm{stop}} = 10$ iterations previously was less than $V^{\mathrm{opt}}_{\mathrm{tol}} = 0.1\,\epsilon_{\mathrm{LJ}}$.

For comparison, a calculation of the heat capacity using standard NS with 20,000 live points was also performed. The live points were generated using NoGUTS with the same parameters as for the NBS calculation. This NS simulation was performed using the software developed by Martiniani et al.[26]. The simulation was stopped when the energy difference of the live points was less than 0.1ϵ.

The NBS simulation overall generated 11×10^6 live points using 3×10^{10} energy gradient calculations. The NS simulation generated 10×10^6 live points using 3.5×10^{10} energy gradient calculations.

The runs generated by the local NBS were found to be indistinguishable from the runs generated by standard NBS at all energies when performing the *maximum a posteriori* calculation of the merge energy.



(a) Cumulative probability

(b) Probability versus energy

Figure 5: In (a) the cumulative probability of a nested optimisation ending in a given range of the most likely minima is shown for LJ_{31}. The inset shows a magnification of the cumulative probabilities of the 20 most likely minima. In (b) the probability of landing in a specific minimum is plotted against the energy of that minimum in NS runs for LJ_{31}.

8.1 Distribution of minima

It is interesting to analyse the distribution of minima generated by 20,000 NOpts, as shown in figs. 5a and 5b. Only 9 of the runs landed in the global minimum, whereas 30.6% of the NOpt runs landed in just a single minimum ($V_\mu = -133.1\,\epsilon_{LJ}$); 79.4% of the runs landed in just 20 of the minima; and during 20,000 minimisations the nested optimisations, only 873 distinct minima were found, whilst the actual number of distinct minima for LJ_{31} has been estimated as approximately 10^{15}, excluding permutation-inversion isomers.[10]

This structure is remarkably different from performing standard minimisations on LJ_{31}, and suggests there might be ways of associating most of the LJ_{31} PES of interest with a very small number of minima. It is not immediately obvious what drives this difference, but it seems that the leaves associated with most minima do not make a meaningful contribution to the overall configuration volume, as opposed to the branches associated with the minima.

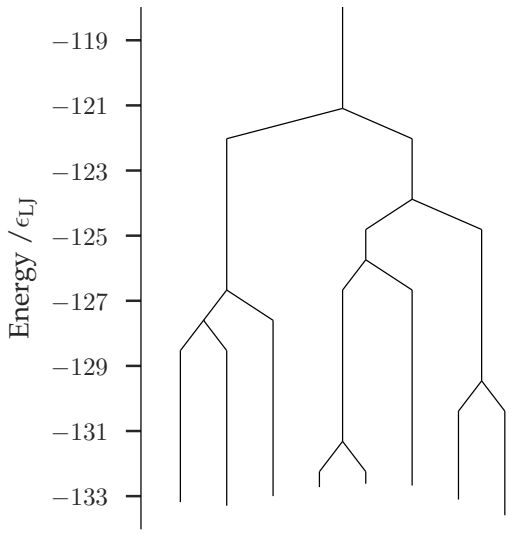


Figure 6: The NBS disconnectivity graph for LJ_{31} with 20,000 NOpts.

8.2 Disconnectivity graph

The NBS results were used to construct a NBS DG, which is shown in fig. 6. The maximum entropy prior, $\alpha_{\text{prior}} = 0.5$, was chosen when determining the merge energies. Only minima that more than 100 NOpts finished in, or had local NOpts, were included in the calculation. The NOpts of the other minima were aggregated together and treated as a single effective minimum.

8.3 Heat capacity

Using the DG illustrated in section 8.2 the heat capacity of LJ_{31} was calculated using NBS and is shown in fig. 7. For comparison, results generated by the equivalent calculation for the standard NS simulation with 20,000 live points and a previous study[10] using BSPT and PT are also illustrated.

The NBS and NS simulations exhibit very similar high temperature solid-liquid heat capacity peaks, though both are slightly lower than the peaks calculated by PT and BSPT. The NBS results closely match the low temperature solid-solid peak calculated by PT and BSPT. The standard NS simulation failed to find the lowest energy minimum and so fails

to reproduce the lowest temperature peak. At higher temperatures the NBS and PT heat capacity curves match extremely well.

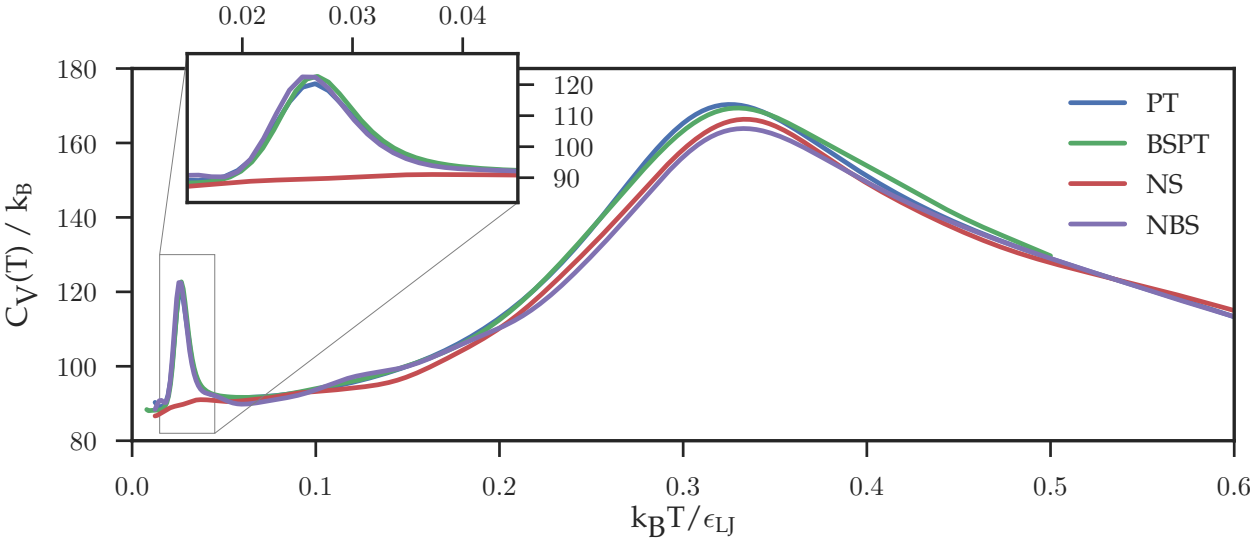


Figure 7: The heat capacity of LJ_{31}, as calculated by NBS with 20,000 NOpts and 1,000 local NOpts on each of the four lowest energy minima. For comparison, results from a standard NS simulation with 20,000 live points and a previous study using BSPT and PT[10] are shown. Each new live point for the NS and NBS simulation was generated using NoGUTS with $j_{\max} = 8$. Inset is a magnification of the low temperature solid-solid peak.

9 Conclusions

Many schemes that have been developed to calculate equilibrium thermodynamic properties require parallelisation to function efficiently.[3,8,10,26,55] In this work we present a new method, nested basin-sampling (NBS), which proceeds by performing a set of embarrassingly parallel nested optimisations (NOpts), which can be combined after the simulations end.

By splitting the configuration volume into separate regions, the calculation can provide a more detailed understanding of the structure of the energy landscape, and how the global thermodynamic properties are encoded. The harmonic approximation can be employed to enhance the accuracy at low temperatures, by separating the configuration volume into disconnected regions.

NBS was used to calculate the heat capacity of LJ_{31}, a benchmark system exhibiting broken ergodicity.[10] The heat capacity calculated using NBS agrees well with other methods, and compares favourably with NS. It was able to successfully resolve the low temperature solid-solid heat capacity peak, which standard NS missed, when performed with a comparable number of live points and energy gradient calculations.

The close agreement with the previous results suggests that the step size adjustment scheme, combined with NoGUTS, is sufficient to ensure that the results generated by the NOpts generate sufficiently unbiased samples for this test case.

There are several possibilities for future work.

- Due to the embarrassingly parallel nature of the NBS calculation, this approach can tackle much larger systems, where equilibrium is usually difficult to achieve.

- The method in its current form is still fairly inefficient compared to BSPT.[10] However, there are many avenues that could be explored to increase its efficiency, particularly when combined with its local sampling scheme. It is likely that good results can still be achieved with a smaller number of NOpts, and some preliminary work suggests that reasonable heat capacity curves can be generated using just local sampling.

- It should be possible to enhance the results generated by local sampling using configurations generated by previous local sampling NOpts as starting points, instead of the minimum.

- It should also be possible to assign contributions to the heat capacity from specific parts of the NBS disconnectivity graph by extending the scheme that was recently applied to results obtained with BSPT.[57]

- We could relate the NBS DG to the DG obtained by generating a transition state network.

- Since NBS partitions the PES into a set of separate regions it is possible to quantify which regions in configuration space have been poorly sampled, further prioritise sampling in those regions, and also provide better measures of convergence for the simulation.

- There are a variety of parameters, p_{acc}, j_{depth}, $N_{\text{delay}}^{\text{opt}}$, $N_{\text{MC}}^{\text{opt}}$, $N_{\text{stop}}^{\text{opt}}$, $N_{\text{window}}^{\text{opt}}$, and $V_{\text{tol}}^{\text{opt}}$, that need to be chosen before beginning a NBS calculation. It is important to quantify how the choice of these hyperparameters affects the overall results and efficiency of the NBS simulation.

- The properties of the NoGUTS sampler could be explored in more detail, in particular how the target acceptance rate affects the overall efficiency of the method.

Acknowledgement

This work was supported by the EPSRC Cambridge NanoDTC, EP/G037221/1. The authors are grateful to Prof. Gábor Csányi and Prof. Florent Calvo for their helpful comments and suggestions. A python implementation of nested basin-sampling can be found on Github, `https://github.com/matthewghgriffiths/nestedbasinsampling`, which extends PELE.[58] The graphs were made using Matplotlib (`https://matplotlib.org/`)[59] and Seaborn (`http://seaborn.pydata.org/`). The diagrams were produced using Inkscape (`https://inkscape.org/en/`).

A Nested sampling calculations

Here we describe the computational details for deriving estimators for integrals over the NBS disconnectivity graph. For clarity we have summarised the notation used in table 1.

Table 1: Summary of notation for NBS.

Variable	Description
$\Phi^{\beta_\delta}_{\beta_{\delta+1}}$	edge volume
$\Phi_{\beta_{\delta+1}}$	branch volume
β	node label
δ	node depth
$N^{\beta_\delta}_{\text{NS}}$	total number of dead points in edge $\Phi^{\beta_{\delta-1}}_{\beta_\delta}$
$V^{\beta_\delta}_j$	The energy of the jth point edge $\Phi^{\beta_{\delta-1}}_{\beta_\delta}$
$n^{\beta_\delta}_j$	The number of live points present for the jth point of edge $\Phi^{\beta_{\delta-1}}_{\beta_\delta}$
$M^{\beta_\delta}_{\beta_{\delta+1}}$	Number of runs falling from Φ_{β_δ} into $\Phi_{\beta_{\delta+1}}$
$p^{\beta_\delta}_{\beta_{\delta+1}}$	The branch probability of falling from Φ_{β_δ} into $\Phi_{\beta_{\delta+1}}$, distributed by $\text{Dir}(M^{\beta_\delta}_{\beta_{\delta+1}})$
$t^{\beta_\delta}_j$	The configuration volume ratio, $\Phi_{\beta_\delta}(V^{\beta_\delta}_{j-1})/\Phi_{\beta_\delta}(V^{\beta_\delta}_j)$

A.1 Top-down calculations

As from section 4.2.1 we can express the basin volume,

$$\Phi_{\beta_{\delta+1}} = p^{\beta_\delta}_{\beta_{\delta+1}} \left(\Phi_{\beta_\delta} - \Phi^{\beta_{\delta-1}}_{\beta_\delta} \right), \tag{37}$$

in terms of its parent node basin and edge volumes, and its branch probability. The volume within the basin can be calculated from its configuration volume ratios,

$$\Phi_{\beta_{\delta+1}}(V^{\beta_{\delta+1}}_j) = \Phi_{\beta_{\delta+1}} \prod_{k=1}^{j} t^{\beta_{\delta+1}}_k. \tag{38}$$

The moments of the branch probabilities are

$$\mathbb{E}_{\text{B}} \left[p^{\beta_\delta}_{\beta_{\delta+1}} \right] = \frac{M^{\beta_\delta}_{\beta_{\delta+1}}}{M^{\beta_\delta}}, \tag{39}$$

$$\mathbb{E}_{\mathrm{D}}\left[{p^{\beta_\delta}_{\beta_{\delta+1}}}^2\right] = \frac{M^{\beta_\delta}_{\beta_{\delta+1}}(M^{\beta_\delta}_{\beta_{\delta+1}}+1)}{M^{\beta_\delta}(M^{\beta_\delta}+1)}, \tag{40}$$

$$\mathbb{E}_{\mathrm{D}}\left[p^{\beta_\delta}_{\beta_{\delta+1}}p^{\beta_\delta}_{\beta'_{\delta+1}}\right] = \frac{M^{\beta_\delta}_{\beta_{\delta+1}}M^{\beta_\delta}_{\beta'_{\delta+1}}}{M^{\beta_\delta}(M^{\beta_\delta}+1)}, \tag{41}$$

where $\mathbb{E}_{\mathrm{D}}$ indicates that this is the expectation of the top down volume. The *edge ratio* $X^{\beta_{\delta-1}}_{\beta_\delta} = \left(\Phi_{\beta_\delta} - \Phi^{\beta_{\delta-1}}_{\beta_\delta}\right)/\Phi_{\beta_\delta}$ can be calculated using NS, as

$$X^{\beta_{\delta-1}}_{\beta_\delta} = \prod_{j=1}^{N^{\beta_\delta}_{\mathrm{NS}}} t^{\beta_\delta}_j, \tag{42}$$

which has moments

$$\mathbb{E}_{\mathrm{D}}\left[X^{\beta_{\delta-1}}_{\beta_\delta}\right] = \prod_{j=1}^{N^{\beta_\delta}_{\mathrm{NS}}} \frac{n^{\beta_\delta}_j}{n^{\beta_\delta}_j+1}, \tag{43}$$

$$\mathbb{E}_{\mathrm{D}}\left[{X^{\beta_{\delta-1}}_{\beta_\delta}}^2\right] = \prod_{j=1}^{N^{\beta_\delta}_{\mathrm{NS}}} \frac{n^{\beta_\delta}_j}{n^{\beta_\delta}_j+2}. \tag{44}$$

We can define the basin volume in terms of the edge ratio,

$$\Phi_{\beta_\delta} = p^{\beta_{\delta-1}}_{\beta_\delta} X^{\beta_{\delta-1}}_{\beta_\delta} \Phi_{\beta_{\delta-1}}, \tag{45}$$

so we can define top down estimates of the top down basin volume,

$$\mathbb{E}_{\mathrm{D}}\left[\Phi_{\beta_\delta}(V^{\beta_\delta}_j)\right] = \mathbb{E}_{\mathrm{D}}\left[p^{\beta_{\delta-1}}_{\beta_\delta}\right]\mathbb{E}_{\mathrm{D}}\left[X^{\beta_{\delta-1}}_{\beta_\delta}\right]\mathbb{E}_{\mathrm{D}}\left[\Phi_{\beta_{\delta-1}}\right]\prod_{k=1}^{j}\frac{n^{\beta_\delta}_j}{n^{\beta_\delta}_j+1}, \tag{46}$$

$$\mathbb{E}_{\mathrm{D}}\left[\Phi_{\beta_\delta}(V^{\beta_\delta}_j)^2\right] = \mathbb{E}_{\mathrm{D}}\left[{p^{\beta_{\delta-1}}_{\beta_\delta}}^2\right]\mathbb{E}_{\mathrm{D}}\left[{X^{\beta_{\delta-1}}_{\beta_\delta}}^2\right]\mathbb{E}_{\mathrm{D}}\left[\Phi^2_{\beta_{\delta-1}}\right]\prod_{k=1}^{j}\frac{n^{\beta_\delta}_j}{n^{\beta_\delta}_j+2}. \tag{47}$$

We can define the edge volume in terms of the edge ratios and branch probabilities,

$$\Phi_{\beta_{\delta'+1}}^{\beta_{\delta'}} = \Phi_0 \, p_{\beta_{\delta'+1}}^{\beta_{\delta'}} (1 - X_{\beta_{\delta'+1}}^{\beta'_\delta}) \prod_{\delta=0}^{\delta'-1} p_{\beta_{\delta+1}}^{\beta_\delta} X_{\beta_{\delta+1}}^{\beta_\delta}. \tag{48}$$

The different edge ratios will be uncorrelated, so calculating the moments of $\Phi_{\beta_{\delta'+1}}^{\beta_{\delta'}}$ can be done by substituting the appropriate moments into eq. (44).

As $X_{\beta_{\delta+1}}^{\beta_\delta}$ and $\mathcal{I}_{\Phi_{\beta_{\delta+1}}^{\beta_\delta}}[f]$ both depend on $t_j^{\beta_{\delta+1}}$, $X_{\beta_{\delta+1}}^{\beta_\delta}$ will be correlated with $\mathcal{I}_{\Phi_{\beta_{\delta+1}}^{\beta_\delta}}[f]$, so we need to calculate the moments of the product $X_{\beta_{\delta+1}}^{\beta_\delta} \mathcal{I}_{\Phi_{\beta_{\delta+1}}^{\beta_\delta}}[f]$ to make an unbiased estimate of $\bar{g}$,

$$\mathbb{E}_{\mathrm{D}}\left[X_{\beta_{\delta+1}}^{\beta_\delta} \mathcal{I}_{\Phi_{\beta_{\delta+1}}^{\beta_\delta}}[f]\right] = \mathbb{E}_{\mathrm{D}}\left[\Phi_{\beta_{\delta+1}}^{\beta_\delta}\right] \left(\prod_{l=1}^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \frac{n_l^{\beta_{\delta+1}}}{n_l^{\beta_{\delta+1}}+1}\right) \sum_j^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \left(\frac{g_j^{\beta_{\delta+1}}}{n_j^{\beta_{\delta+1}}} \prod_{k=1}^{j} \frac{n_k^{\beta_{\delta+1}}+1}{n_k^{\beta_{\delta+1}}+2}\right), \tag{49}$$

$$\mathbb{E}_{\mathrm{D}}\left[{X_{\beta_{\delta+1}}^{\beta_\delta}}^2 \mathcal{I}_{\Phi_{\beta_{\delta+1}}^{\beta_\delta}}[f]\right] = \mathbb{E}_{\mathrm{D}}\left[\Phi_{\beta_{\delta+1}}^{\beta_\delta}\right] \left(\prod_{l=1}^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \frac{n_l^{\beta_{\delta+1}}}{n_l^{\beta_{\delta+1}}+2}\right) \sum_j^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \left(\frac{g_j^{\beta_{\delta+1}}}{n_j^{\beta_{\delta+1}}} \prod_{k=1}^{j} \frac{n_k^{\beta_{\delta+1}}+2}{n_k^{\beta_{\delta+1}}+3}\right), \tag{50}$$

$$\begin{aligned}\mathbb{E}_{\mathrm{D}}\left[X_{\beta_{\delta+1}}^{\beta_\delta} \mathcal{I}_{\Phi_{\beta_{\delta+1}}^{\beta_\delta}}[f^2]\right] = \mathbb{E}_{\mathrm{D}}\left[{\Phi_{\beta_{\delta+1}}^{\beta_\delta}}^2\right] &\left(\prod_{l=1}^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \frac{n_l^{\beta_{\delta+1}}}{n_l^{\beta_{\delta+1}}+1}\right) \\ &\times \sum_{l=1}^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \left[\frac{2g_l^{\beta_{\delta+1}}}{n_l^{\beta_{\delta+1}}+1} \left(\prod_{k=1}^{j} \frac{n_k^{\beta_{\delta+1}}+1}{n_k^{\beta_{\delta+1}}+2}\right) \sum_{j=1}^{l} \left(\frac{g_j^{\beta_{\delta+1}}}{n_j^{\beta_{\delta+1}}+2} \prod_{k=1}^{j} \frac{n_k^{\beta_{\delta+1}}+2}{n_k^{\beta_{\delta+1}}+3}\right)\right],\end{aligned} \tag{51}$$

$$\begin{aligned}\mathbb{E}_{\mathrm{D}}\left[{X_{\beta_{\delta+1}}^{\beta_\delta}}^2 \mathcal{I}_{\Phi_{\beta_{\delta+1}}^{\beta_\delta}}[f^2]\right] = \mathbb{E}_{\mathrm{D}}\left[{\Phi_{\beta_{\delta+1}}^{\beta_\delta}}^2\right] &\left(\prod_{l=1}^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \frac{n_l^{\beta_{\delta+1}}}{n_l^{\beta_{\delta+1}}+2}\right) \\ &\times \sum_{l=1}^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \left[\frac{2g_l^{\beta_{\delta+1}}}{n_l^{\beta_{\delta+1}}+2} \prod_{k=1}^{j} \frac{n_k^{\beta_{\delta+1}}+2}{n_k^{\beta_{\delta+1}}+3} \sum_{j=1}^{l} \left(\frac{g_j^{\beta_{\delta+1}}}{n_j^{\beta_{\delta+1}}+3} \prod_{k=1}^{j} \frac{n_k^{\beta_{\delta+1}}+3}{n_k^{\beta_{\delta+1}}+4}\right)\right].\end{aligned} \tag{52}$$

To simplify this calculation, we define the *branch integral*,

$$\mathcal{I}_{\Phi_{\beta_\delta}}[f] = \sum_{\delta'=\delta}^{\arg\max_\delta \beta_\delta} \mathcal{I}_{\Phi_{\beta_\delta}^{\beta_{\delta-1}}}[f], \tag{53}$$

over the configuration space, so we can define the branch integral in terms of the edge integral and daughter branch integral,

$$\mathcal{I}_{\Phi_{\beta_\delta}}[f] = \mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta_\delta}}[f] + \sum_{\beta'} \mathcal{I}_{\Phi_{\beta'_{\delta+1}}}[f]. \tag{54}$$

Hence the first moment can be calculated as

$$\mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi_{\beta_\delta}}[f]\right] = \mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta'_\delta}}[f]\right] + \sum_{\beta'} \mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi_{\beta'_{\delta+1}}}[f]\right], \tag{55}$$

and the second moment will be

$$\begin{aligned}\mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi_{\beta_\delta}}[f]^2\right] = \mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta_\delta}}[f]^2\right] + \sum_{\beta'} \mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta'_\delta}}[f]\mathcal{I}_{\Phi_{\beta'_{\delta+1}}}[f]\right] \\ + \sum_{\beta',\beta''} \mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi^{\beta_\delta}_{\beta'_{\delta+1}}}[f]\mathcal{I}_{\Phi^{\beta_\delta}_{\beta''_{\delta+1}}}[f]\right],\end{aligned} \tag{56}$$

and the moments can be calculated as

$$\begin{aligned}\mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta_\delta}}[f^2]\right] = \mathbb{E}_{\mathrm{D}}\left[p^{\beta_{\delta-1}}_{\beta_\delta}\right] \\ \times \left(\mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta_{\delta+1}}}[f^2]\right] - 2\mathbb{E}_{\mathrm{D}}\left[X^{\beta_{\delta-1}}_{\beta_\delta}\mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta_\delta}}[f^2]\right] + \mathbb{E}_{\mathrm{D}}\left[\left(X^{\beta_{\delta-1}}_{\beta_\delta}\mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta_\delta}}[f^2]\right)\right]\right),\end{aligned} \tag{57}$$

$$\begin{aligned}\mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta_\delta}}[f]\mathcal{I}_{\Phi_{\beta'_{\delta+1}}}[f]\right] = \frac{\mathbb{E}_{\mathrm{D}}\left[{\Phi_{\beta_{\delta-1}}}^2\right]}{\mathbb{E}_{\mathrm{D}}\left[\Phi_{\beta_\delta}\right]\mathbb{E}_{\mathrm{D}}\left[\Phi^{\beta_{\delta-1}}_{\beta_\delta}\right]} \\ \times \left(\mathbb{E}_{\mathrm{D}}\left[X^{\beta_{\delta-1}}_{\beta_\delta}\mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta_\delta}}[f]\right] - \mathbb{E}_{\mathrm{D}}\left[{X^{\beta_{\delta-1}}_{\beta_\delta}}^2\mathcal{I}_{\Phi^{\beta_{\delta-1}}_{\beta_\delta}}[f]\right]\right)\mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi_{\beta'_{\delta+1}}}[f]\right],\end{aligned} \tag{58}$$

$$\mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi_{\beta'_{\delta+1}}}[f]\mathcal{I}_{\Phi_{\beta''_{\delta+1}}}[f]\right]$$

$$= \frac{\mathbb{E}_{\mathrm{D}}\left[{\Phi_{\beta_\delta}}^2\right]}{\mathbb{E}_{\mathrm{D}}\left[\Phi_{\beta_\delta}\right]^2} \frac{\mathbb{E}_{\mathrm{D}}\left[p_{\beta'_{\delta+1}}^{\beta_\delta} p_{\beta''_{\delta+1}}^{\beta_\delta}\right]}{\mathbb{E}_{\mathrm{D}}\left[p_{\beta'_{\delta+1}}^{\beta_\delta}\right] \mathbb{E}_{\mathrm{D}}\left[p_{\beta''_{\delta+1}}^{\beta_\delta}\right]} \mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi_{\beta'_{\delta+1}}}[f]\right] \mathbb{E}_{\mathrm{D}}\left[\mathcal{I}_{\Phi_{\beta''_{\delta+1}}}[f]\right], \quad (59)$$

where the branch volume moments are

$$\mathbb{E}_{\mathrm{D}}\left[\Phi_{\beta_\delta}\right] = \mathbb{E}_{\mathrm{D}}\left[\Phi_0\right] \left(\prod_{\delta'=0}^{\delta} \mathbb{E}_{\mathrm{D}}\left[p_{\beta_{\delta'}}^{\beta_{\delta'-1}}\right] \mathbb{E}_{\mathrm{D}}\left[X_{\beta_{\delta'}}^{\beta_{\delta'-1}}\right] \right), \quad (60)$$

$$\mathbb{E}_{\mathrm{D}}\left[\Phi_{\beta_\delta}^2\right] = \mathbb{E}_{\mathrm{D}}\left[\Phi_0^2\right] \left(\prod_{\delta'=0}^{\delta} \mathbb{E}_{\mathrm{D}}\left[{p_{\beta_{\delta'}}^{\beta_{\delta'-1}}}^2\right] \mathbb{E}_{\mathrm{D}}\left[{X_{\beta_{\delta'}}^{\beta_{\delta'-1}}}^2\right] \right). \quad (61)$$

A.2 Bottom-up calculations

From section 4.2.2 the basin volume can be defined, bottom up as,

$$\Phi_{\beta_\delta}(V_j^{\beta_\delta}) = \left(\sum_{\beta_{\delta+1}} \Phi_{\beta_{\delta+1}} \right) \prod_{k=j}^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \frac{1}{t_k^{\beta_{\delta+1}}}. \quad (62)$$

The moments for the basin volumes can be calculated for the bottom up calculation,

$$\mathbb{E}_{\mathrm{U}}\left[\Phi_{\beta_{\delta+1}}(V_j^{\beta_{\delta+1}})\right] = \sum_{\beta_\delta} \left(\mathbb{E}_{\mathrm{U}}\left[\Phi_{\beta_\delta}\right]\right) \left(\prod_{k=j}^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \frac{n_k^{\beta_{\delta+1}}}{n_k^{\beta_{\delta+1}} - 1} \right), \quad (63)$$

$$\mathbb{E}_{\mathrm{U}}\left[\Phi_{\beta_{\delta+1}}(V_j^{\beta_{\delta+1}})^2\right] = \mathbb{E}_{\mathrm{U}}\left[\left(\sum_{\beta_\delta} \Phi_{\beta_\delta} \right)^2 \right] \left(\prod_{k=j}^{N_{\mathrm{NS}}^{\beta_{\delta+1}}} \frac{n_k^{\beta_{\delta+1}}}{n_k^{\beta_{\delta+1}} - 2} \right), \quad (64)$$

as all the child branch volumes of $\Phi_{\beta_{\delta+1}}$ are independent in the bottom up calculation, $\mathbb{E}_{\mathrm{U}}\left[\left(\sum_{\beta_\delta} \Phi_{\beta_\delta}\right)^2\right]$ is straightforward to calculate.

B NoGUTS

Here we explicitly describe the algorithm and its associated sub-algorithms for sampling new points using the NoGUTS algorithm. Here the function unif(a, b) uniformly gener-

ates a random real number between a and b.

Algorithm 2 The NoGUTS algorithm

Function: NoGUTS
Input: $\boldsymbol{R}$, δ, V, V_{cut}, j_{max}
Output: $\boldsymbol{R}, V', n_{\text{accept}}, n_{\text{reject}}$ ▷ The new live point generated by NoGUTS
$j_{\text{depth}} = 0$
$n_{\text{accept}} = 0, n_{\text{reject}} = 0$
$s_{\text{valid}} = \text{True}$
$\boldsymbol{R}^+ = \boldsymbol{R}^+ = \boldsymbol{R}$
$\boldsymbol{p}_+^+ = \boldsymbol{p}_+^- = \boldsymbol{p}_-^+ = \boldsymbol{p}_-^- \sim \mathcal{N}(0, \boldsymbol{I})$ ▷ Initialise random velocity
while s_{valid} AND $j_{\text{depth}} < j_{\text{max}}$ **do**
  if unif$(0,1) < 0.5$ **then** ▷ Make proposal to double slice
    $\_, \_, \_, \boldsymbol{R}^-, \boldsymbol{p}_+^-, \boldsymbol{p}_-^-, \boldsymbol{R}', \boldsymbol{p}_+', \boldsymbol{p}_-', V', n_{\text{accept}}', n_{\text{reject}}', s_{\text{valid}}'$
      $=$ BuildTree$(\boldsymbol{R}^-, \boldsymbol{p}_+^-, \boldsymbol{p}_-^-, V_{\text{cut}}, -1, \delta, j_{\text{depth}})$ ▷ See algorithm 3
  else
    $\boldsymbol{R}^+, \boldsymbol{p}_+^+, \boldsymbol{p}_-^+, \_, \_, \_, \boldsymbol{R}', \boldsymbol{p}_+', \boldsymbol{p}_-', V', n_{\text{accept}}', n_{\text{reject}}', s_{\text{valid}}'$
      $=$ BuildTree$(\boldsymbol{R}^+, \boldsymbol{p}_+^+, \boldsymbol{p}_-^+, V_{\text{cut}}, 1, \delta, j_{\text{depth}})$
  end if
  if s_{valid}' AND unif$(0,1) < n_{\text{accept}}'/n_{\text{accept}}$ **then** ▷ take point selected by new slice
    $\boldsymbol{R}_{\text{NoGUTS}} = \boldsymbol{R}'$
    $V_{\text{NoGUTS}} = V'$
  end if
  $n_{\text{accept}} = n_{\text{accept}} + n_{\text{accept}}', n_{\text{reject}} = n_{\text{reject}} + n_{\text{reject}}'$
  $s_{\text{valid}} = s_{\text{valid}}'$ AND StopCriterion$(\boldsymbol{R}^+, \boldsymbol{p}_+^+, \boldsymbol{p}_-^+, \boldsymbol{R}^-, \boldsymbol{p}_+^-, \boldsymbol{p}_-^-)$
    ▷ Test whether proposal is valid or whether the new slice has performed a u-turn
  $j = j + 1$
end while

Algorithm 3 BuildTree

Function: BuildTree
Input: $\boldsymbol{R}, \boldsymbol{p}_+, \boldsymbol{p}_-, V_{\text{cut}}, v_{\text{dir}}, \delta, j_{\text{depth}}, V$
Optional: $V_{\text{constraint}}$, TestConstraints
Output: $\boldsymbol{R}^+, \boldsymbol{p}_+^+, \boldsymbol{p}_-^+, \boldsymbol{R}^-, \boldsymbol{p}_+^-, \boldsymbol{p}_-^-, \boldsymbol{R}', \boldsymbol{p}_+, \boldsymbol{p}'_-, V', n_{\text{accept}}, n_{\text{reject}}, s_{\text{valid}}$
if $j_{\text{depth}} = 0$ **then**
  if using constraint potential **then** ▷ See algorithm 5
    $\boldsymbol{R}', \boldsymbol{p}'_+, \boldsymbol{p}'_-, V', n' = \text{ConsGalileanStep}(\boldsymbol{R}, \boldsymbol{p}_+, \boldsymbol{p}_-, V_{\text{cut}}, v_{\text{dir}}, \delta, V, V_{\text{constraint}})$
  else ▷ See algorithm 4
    $\boldsymbol{R}', \boldsymbol{p}'_+, \boldsymbol{p}'_-, V', n' = \text{GalileanStep}(\boldsymbol{R}, \boldsymbol{p}_+, \boldsymbol{p}_-, V_{\text{cut}}, v_{\text{dir}}, \delta, V)$
  end if
  $\boldsymbol{R}^+, \boldsymbol{p}_+^+, \boldsymbol{p}_-^+ = \boldsymbol{R}', \boldsymbol{p}'_+, \boldsymbol{p}'_-$
  $\boldsymbol{R}^-, \boldsymbol{p}_+^-, \boldsymbol{p}_-^- = \boldsymbol{R}', \boldsymbol{p}'_+, \boldsymbol{p}'_-$
  $s_{\text{valid}} = \text{TestConstraints}(\boldsymbol{R}')$ ▷ Stop if constraint broken
else ▷ Recursively build binary tree
  $\boldsymbol{R}^+, \boldsymbol{p}_+^+, \boldsymbol{p}_-^+, \boldsymbol{R}^-, \boldsymbol{p}_+^-, \boldsymbol{p}_-^-, \boldsymbol{R}', \boldsymbol{p}'_+, \boldsymbol{p}'_-, V', n'_{\text{accept}}, n'_{\text{reject}}, s'_{\text{valid}}$
    $= \text{BuildTree}(\boldsymbol{R}, \boldsymbol{p}_+, \boldsymbol{p}_-, V_{\text{cut}}, v_{\text{dir}}, \delta, j_{\text{depth}} - 1)$
  if s'_{valid} **then**
    if $v_{\text{dir}} = -1$ **then**
      $\text{-}, \text{-}, \text{-}, \boldsymbol{R}^-, \boldsymbol{p}_+^-, \boldsymbol{p}_-^-, \boldsymbol{R}'', \boldsymbol{p}''_+, \boldsymbol{p}''_-, V'', n''_{\text{accept}}, n''_{\text{reject}}, s''_{\text{valid}}$
        $= \text{BuildTree}(\boldsymbol{R}^-, \boldsymbol{p}_+^-, \boldsymbol{p}_-^-, V_{\text{cut}}, v_{\text{dir}}, \delta, j_{\text{depth}} - 1)$
    else
      $\boldsymbol{R}^+, \boldsymbol{p}_+^+, \boldsymbol{p}_-^+, \text{-}, \text{-}, \text{-}, \boldsymbol{R}'', \boldsymbol{p}''_+, \boldsymbol{p}''_-, V'', n''_{\text{accept}}, n''_{\text{reject}}, s''_{\text{valid}}$
        $= \text{BuildTree}(\boldsymbol{R}^+, \boldsymbol{p}_+^+, \boldsymbol{p}_-^+, V_{\text{cut}}, v_{\text{dir}}, \delta, j_{\text{depth}} - 1)$
    end if
    $n_{\text{accept}} = n'_{\text{accept}} + n''_{\text{accept}}$
    $s_{\text{valid}} = s''_{\text{valid}}$ AND $\text{StopCriterion}(\boldsymbol{R}^+, \boldsymbol{p}_+^+, \boldsymbol{R}^-, \boldsymbol{p}_-^-)$
    if $\text{unif}(0, 1) < \frac{n''_{\text{accept}}}{n_{\text{accept}}}$ **then** ▷ Implicitly join slices together
      $\boldsymbol{R}', V', \boldsymbol{p}'_+, \boldsymbol{p}'_- = \boldsymbol{R}'', V'', \boldsymbol{p}''_+, \boldsymbol{p}''_-$
    end if
  end if
end if

Algorithm 4 Galilean step

Function: GalileanStep
Input: $\boldsymbol{R}, \boldsymbol{p}_+, \boldsymbol{p}_-, V_{\text{cut}}, v_{\text{dir}}, \delta, V$
▷ Start point, forwards velocity, backwards velocity, energy cut-off, direction, step size, potential
Output: $\boldsymbol{R}', \boldsymbol{p}'_+, \boldsymbol{p}'_-, V', n'$
▷ New point, forwards velocity, backwards velocity, energy, and validity
if $v_{\text{dir}} = -1$ **then**
▷ Selecting appropriate velocity vector
$\boldsymbol{p} = -\boldsymbol{p}_-$
else
$\boldsymbol{p} = \boldsymbol{p}_+$
end if
$\boldsymbol{R}' = \boldsymbol{R} + \delta\boldsymbol{p}$
$V' = V(\boldsymbol{R}')$
$\boldsymbol{G}' = \nabla_{\boldsymbol{R}} V(\boldsymbol{R}')$
$r_{\text{energy}} = V' > V_{\text{cut}}$
if r_{energy} **then**
▷ New point not valid
$n' = 0$
▷ Reflect velocity
$\boldsymbol{p}' = \boldsymbol{p} - 2\dfrac{\boldsymbol{G}' \cdot \boldsymbol{p}}{\boldsymbol{G}' \cdot \boldsymbol{G}'}\boldsymbol{G}'$
else
▷ New point is valid
$n' = 1$
$\boldsymbol{p}' = \boldsymbol{p}$
end if
if $v_{\text{dir}} = -1$ **then**
▷ Set new velocity vectors
$\boldsymbol{p}'_+ = \boldsymbol{p}_-$
$\boldsymbol{p}'_- = -\boldsymbol{p}'$
else
$\boldsymbol{p}'_+ = \boldsymbol{p}'$
$\boldsymbol{p}'_- = \boldsymbol{p}_+$
end if

Algorithm 5 Constrained Galilean step

Function: ConsGalileanStep
Input: $\boldsymbol{R}, \boldsymbol{p}_+, \boldsymbol{p}_-, V_{\text{cut}}, v_{\text{dir}}, \delta, V, V_{\text{constraint}}$
Output: $\boldsymbol{R}', \boldsymbol{p}'_+, \boldsymbol{p}'_-, V', n'$
if $v_{\text{dir}} = -1$ **then**
$\boldsymbol{p} = -\boldsymbol{p}_-$
else
$\boldsymbol{p} = \boldsymbol{p}_+$
end if
$\boldsymbol{R}' = \boldsymbol{R} + \delta\boldsymbol{p}$
$V' = V(\boldsymbol{R}')$
$\boldsymbol{G}' = \nabla_{\boldsymbol{R}} V(\boldsymbol{R}')$
$V'_{\text{constraint}} = V_{\text{constraint}}(\boldsymbol{R}')$
$\boldsymbol{G}'_{\text{constraint}} = \nabla_{\boldsymbol{R}} V_{\text{constraint}}(\boldsymbol{R}')$
$r_{\text{energy}} = V' > V_{\text{cut}}$
$r_{\text{constraint}} = V'_{\text{constraint}} > 0$
if r_{energy} OR $r_{\text{constraint}}$ **then**
if r_{energy} AND NOT $r_{\text{constraint}}$ **then**
$\boldsymbol{G}'' = \boldsymbol{G}'$
else if NOT r_{energy} AND $r_{\text{constraint}}$ **then**
$\boldsymbol{G}'' = \boldsymbol{G}'_{\text{constraint}}$
else if r_{energy} AND $r_{\text{constraint}}$ **then**
▷ Reflect off both gradients
$\boldsymbol{G}'' = \dfrac{\boldsymbol{G}'}{|G'|} + \dfrac{\boldsymbol{G}'_{\text{constraint}}}{|G'_{\text{constraint}}|}$
end if
$n' = 0$
$\boldsymbol{p}' = \boldsymbol{p} - 2\dfrac{\boldsymbol{G}'' \cdot \boldsymbol{p}}{\boldsymbol{G}'' \cdot \boldsymbol{G}''}\boldsymbol{G}''$
else
$n' = 1$
$\boldsymbol{p}' = \boldsymbol{p}$
end if
if $v_{\text{dir}} = -1$ **then**
$\boldsymbol{p}'_+ = \boldsymbol{p}_-$
$\boldsymbol{p}'_- = -\boldsymbol{p}'$
else
$\boldsymbol{p}'_+ = \boldsymbol{p}'$
$\boldsymbol{p}'_- = \boldsymbol{p}_+$
end if

C Probability distributions

C.1 Beta distribution

To model a binomial event, such as flipping a coin, we need to be able to specify a probability distribution of the *probability* of obtaining heads, a probability of a *probability*. The probability of observing a certain number of heads, N^{heads}, after a fixed number of flips, N^{flips}, with a fixed probability of obtaining heads, p_{heads}, is modelled by the binomial distribution,

$$N^{\text{heads}} \sim \text{Bin}(N^{\text{flips}}, p_{\text{heads}}) \tag{65}$$

with probability,

$$\Pr(N^{\text{heads}}|N^{\text{flips}}, p^{\text{heads}}) = \text{Bin}(N^{\text{heads}}|N^{\text{flips}}, p_{\text{heads}}) = \binom{N^{\text{flips}}}{N^{\text{heads}}} {p_{\text{heads}}}^{N^{\text{heads}}} (1-p_{\text{heads}})^{N^{\text{flips}}-N^{\text{heads}}}, \tag{66}$$

where $\binom{n}{m} = \frac{n!}{m!(n-m)!}$. The beta distribution is the *conjugate prior* to the binomial distribution (see appendix D), so it is the most straightforward way to define a distribution over the binomial probability

$$p_{\text{heads}} \sim \mathcal{B}(\alpha_{\mathcal{B}}, \beta_{\mathcal{B}}), \tag{67}$$

with probability density,

$$\begin{aligned} \Pr(p_{\text{heads}}) &= \begin{cases} \dfrac{\Gamma(\alpha_{\mathcal{B}} + \beta_{\mathcal{B}})}{\Gamma(\alpha_{\mathcal{B}})\Gamma(\beta_{\mathcal{B}})} p_{\text{heads}}^{\alpha_{\mathcal{B}}-1} (1 - p_{\text{heads}})^{\beta_{\mathcal{B}}-1} & \text{when } 0 < p_{\text{heads}} < 1 \\ 0 & \text{otherwise} \end{cases} \\ &\equiv \mathcal{B}(p_{\text{heads}}|\alpha_{\mathcal{B}}, \beta_{\mathcal{B}}), \end{aligned} \tag{68}$$

where $\alpha_{\mathcal{B}}$, $\beta_{\mathcal{B}}$ are the shape parameters of the beta distribution. The normalisation constant can also be written as a beta function,

$$B(\alpha_{\mathcal{B}}, \beta_{\mathcal{B}}) = \frac{\Gamma(\alpha_{\mathcal{B}} + \beta_{\mathcal{B}})}{\Gamma(\alpha_{\mathcal{B}})\Gamma(\beta_{\mathcal{B}})}. \tag{69}$$

The moments of the beta distribution are

$$\int_0^1 t_{\mathcal{B}}^a (1 - t_{\mathcal{B}})^{b'} \mathcal{B}(t_{\mathcal{B}}|\alpha_{\mathcal{B}}, \beta_{\mathcal{B}}) \, \mathrm{d}t_{\mathcal{B}} = \frac{\Gamma(\alpha_{\mathcal{B}} + \beta_{\mathcal{B}})}{\Gamma(\alpha_{\mathcal{B}})\Gamma(\beta_{\mathcal{B}})} \frac{\Gamma(\alpha_{\mathcal{B}} + a)\Gamma(\beta_{\mathcal{B}} + b')}{\Gamma(\alpha_{\mathcal{B}} + a + \beta_{\mathcal{B}} + b')}, \tag{70}$$

so in the case of eqs. (11) and (12) $t_k = t_{\mathcal{B}}$, $\alpha_{\mathcal{B}} = 1$ and $\beta_{\mathcal{B}} = n_k^{\mathrm{NS}}$, so we would find that $\mathbb{E}[t_{\mathcal{B}}] = n_k^{\mathrm{NS}}/(n_k^{\mathrm{NS}} + 1)$, $\mathbb{E}[t_{\mathcal{B}}^2] = n_k^{\mathrm{NS}}/(n_k^{\mathrm{NS}} + 2)$, and $\mathbb{E}[(1 - t_{\mathcal{B}})] = 1/(n_k^{\mathrm{NS}} + 1)$.

The beta distribution can also be used to model the *order statistics* of samples from the uniform distribution. If $n^{\mathcal{U}}$ samples have been drawn independently from the uniform distribution, then the $k^{\mathcal{U}}$th order statistic,

$$\mathcal{U}_{k^{\mathcal{U}}} \sim \mathcal{B}(k^{\mathcal{U}}, n^{\mathcal{U}} - k^{\mathcal{U}} + 1), \tag{71}$$

the distribution of the $k^{\mathcal{U}}$th highest value of $n^{\mathcal{U}}$ independent samples from the uniform distribution, which derives naturally from considering the binomial likelihood of observing $k^{\mathcal{U}}$ successes out of $n^{\mathcal{U}}$ trials with probability $\mathcal{U}_{k^{\mathcal{U}}}$. It is in this sense that NS uses the beta distribution as the set of live points can be modelled as uniformly distributed samples over $\Phi(V)$.

C.1.1 Fitting beta distributions

It is possible to fit a beta distribution, $\mathcal{B}(a_{\mathrm{fit}}, b_{\mathrm{fit}})$, to some other random variable, $t_{\mathrm{fit}} \sim q_{\mathrm{fit}}$, by matching the first and second moments, because the beta distribution has only two

degrees of freedom,

$$a_{\text{fit}} = \mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}\right]\left(\frac{\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}\right]\left(\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}\right]-1\right)}{\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}^2\right]-\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}\right]^2}-1\right), \tag{72}$$

$$b_{\text{fit}} = \left(\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}\right]-1\right)\left(\frac{\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}\right]\left(\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}\right]-1\right)}{\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}^2\right]-\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}\right]^2}-1\right), \tag{73}$$

where $\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}\right]$ is the first moment of t_{fit} and $\mathbb{E}_{q_{\text{fit}}}\left[t_{\text{fit}}^2\right]$ is the second moment of t_{fit} with respect to q_{fit}.

C.2 Dirichlet distribution

The binomial distribution and the beta distribution can be generalised to model multiple probabilities, for example, studying the rolling of a k^{die}-sided die. The results, $\boldsymbol{N}^{\text{die}} = \{N_1^{\text{die}}, \cdots, N_{k^{\text{die}}}^{\text{die}}\} \sim \text{Mult}(\boldsymbol{p}^{\text{die}})$ of rolling the die with probabilities, $\boldsymbol{p}^{\text{die}} = \{p_1^{\text{die}}, \cdots, p_{k^{\text{die}}}^{\text{die}}\}$ will be distributed according to the multinomial distribution, $\boldsymbol{N}^{\text{die}} \sim \text{Mult}(\boldsymbol{p}^{\text{die}})$, with probability density,

$$\text{Mult}(\boldsymbol{N}^{\text{die}}|\boldsymbol{p}^{\text{die}}) = \frac{\left(\sum_{j=1}^{k^{\text{die}}} N_j^{\text{die}}\right)!}{\prod_{k=1}^{N} N_k^{\text{die}}!}\prod_{k=1}^{N} {p_k^{\text{die}}}^{N_k^{\text{die}}}. \tag{74}$$

The conjugate prior of the multinomial distribution is the Dirichlet distribution, parametrised by an N-dimensional parameter vector, $\boldsymbol{\alpha}^{\text{die}} = (\alpha_1^{\text{die}}, \cdots, \alpha_N^{\text{die}})$. For $\boldsymbol{p}^{\text{die}} \sim \text{Dir}(\boldsymbol{\alpha}^{\text{die}})$ the probability distribution of $\boldsymbol{p}^{\text{die}}$ will be

$$\Pr(\boldsymbol{p}^{\text{die}}|\boldsymbol{\alpha}^{\text{die}}) = \frac{\Gamma\left(\sum_{j=1}^{N}\alpha_j^{\text{die}}\right)}{\prod_{j=1}^{N}\Gamma(\alpha_j^{\text{die}})}\prod_{k=1}^{N} {p_k^{\text{die}}}^{\alpha_k^{\text{die}}-1}, \tag{75}$$

where $\sum_{j=1}^{N} p_j^{\text{die}} = 1$.

D Bayesian inference

In Bayesian statistics, probabilities are used to encode beliefs about some phenomenon. At the heart of Bayesian statistics is Bayes' rule,

$$\Pr(M(\boldsymbol{\theta})|D) = \frac{\Pr(D|M(\boldsymbol{\theta}))\Pr(M(\boldsymbol{\theta}))}{\Pr(D)}, \tag{76}$$

which allows the *prior* belief/distribution, $\Pr\left(M(\boldsymbol{\theta})\right)$, for the parameters, $\boldsymbol{\theta}$, of some model, $M(\boldsymbol{\theta})$, to be updated to give a *posterior* belief/distribution, $\Pr(M(\boldsymbol{\theta})|D)$, for the parameters, given some observed data, D. This update is performed by taking the product of the *likelihood*, $\Pr(D|M(\boldsymbol{\theta}))$, of observing the data given the model with the prior. This product is normalised by the *evidence*, $\Pr(D)$, the probability of observing the data given all possible instances of the model, calculated by integrating the product of the likelihood and prior or *marginalising* over all possible parameters,

$$\Pr(D) = \int \Pr\left(D|M(\boldsymbol{\theta})\right)\Pr\left(M(\boldsymbol{\theta})\right)\,\mathrm{d}\boldsymbol{\theta}. \tag{77}$$

When using the beta distribution to model p_{heads}, we can specify an uninformative prior on the coin toss, $p_{\text{heads}} \sim \mathcal{B}(\alpha_p, \beta_p)$, where α_p and β_p specify our prior belief of p_{heads}. For an uninformative prior $\alpha_p = \beta_p = 1/2$. We can use Bayes' rule to update our belief of p_{heads},

$$\begin{aligned}\Pr(p_{\text{heads}}|N^{\text{heads}}, N^{\text{flips}}) &= \frac{\text{Bin}(N^{\text{heads}}|N^{\text{flips}}, p_{\text{heads}})\mathcal{B}(p_{\text{heads}}|\alpha_p, \beta_p)}{\Pr(N^{\text{heads}}|N^{\text{flips}})} \\ &= \mathcal{B}(p_{\text{heads}}|N^{\text{heads}} + \alpha_p, N^{\text{flips}} - N^{\text{heads}} + \beta_p).\end{aligned} \tag{78}$$

Here we see why the parameters of the beta distribution are commonly viewed as pseudocounts, since they can be viewed as representing the number of observations of the event happening or not happening.

D.1 Model comparison

The evidence is useful as it allows different models to be compared. Given two models, M_1 and M_2, and some data, the probability of the hypothesis that the first model is correct, $\mathcal{H}_1$, is

$$\Pr(\mathcal{H}_1) = \frac{\Pr(M_1|D)\Pr(M_1)}{\Pr(M_1|D)\Pr(M_1) + \Pr(M_2|D)\Pr(M_2)} = \frac{K^{\text{BF}}}{1 + K^{\text{BF}}}, \tag{79}$$

where the relative evidence between the two models is known as the *Bayes factor*,

$$K^{\text{BF}} = \frac{\Pr(M_1|D)}{\Pr(M_2|D)} \frac{\Pr(M_1)}{\Pr(M_2)}, \tag{80}$$

and $\Pr(M_1)$ and $\Pr(M_2)$ are the prior belief of whether $\mathcal{H}_1$ or $\mathcal{H}_2$ is true. If *a priori* both models are viewed equally likely then $\Pr(M_1) = \Pr(M_2) = 1/2$.

When $K^{\text{BF}} > 1$, $\mathcal{H}_1$ is more likely. Conversely if $K^{\text{BF}} < 1$, $\mathcal{H}_2$ is more likely given the data. In the next section it will be shown how to compare binomial distributions using Bayesian model comparison.

D.1.1 Comparing binomial distributions

Consider an example where we want to compare examination pass rates, p_a and p_b, between two different departments, a and b. Suppose we observe that there are n_a^{pass} passes and n_a^{fail} fails from department a and the equivalent for b, and $n_{a/b}^{\text{students}} = n_{a/b}^{\text{pass}} + n_{a/b}^{\text{fail}}$. We are interested in testing the hypothesis that the pass rates are the same, $\mathcal{H}_0 : p_a = p_b$, or different, $\mathcal{H}_1 : p_a \neq p_b$. We can compare these hypotheses by Bayesian model comparison. So we can calculate the evidence of model 0,

$$\begin{aligned}
&\Pr(p_a = p_b | n_a^{\text{pass}}, n_a^{\text{fail}}, n_b^{\text{pass}}, n_b^{\text{fail}}) \\
&\quad = \int \text{Bin}(n_a^{\text{pass}} | n_a^{\text{students}}, p_a) \text{Bin}(n_b^{\text{pass}} | n_b^{\text{students}}, p_a) \mathcal{B}(p_a | \alpha^{\text{pass}}, \beta^{\text{pass}}) \, \mathrm{d}p_a \\
&\quad = \binom{n_a^{\text{students}}}{n_a^{\text{pass}}} \binom{n_b^{\text{students}}}{n_b^{\text{pass}}} B(n_a^{\text{pass}} + n_b^{\text{pass}} + \alpha^{\text{pass}}, n_a^{\text{fail}} + n_b^{\text{fail}} + \alpha^{\text{fail}}),
\end{aligned} \tag{81}$$

and model 1,

$$\begin{aligned}
&\Pr(p_a \neq p_b | n_a^{\text{pass}}, n_a^{\text{fail}}, n_b^{\text{pass}}, n_b^{\text{fail}}) \\
&= \iint \text{Bin}(n_a^{\text{pass}} | n_a^{\text{students}}, p_a) \text{Bin}(n_b^{\text{pass}} | n_b^{\text{students}}, p_b) \mathcal{B}(p_a | \alpha^{\text{pass}}, \beta^{\text{pass}}) \mathcal{B}(p_b | \alpha^{\text{pass}}, \beta^{\text{pass}}) \, \mathrm{d}p_a \, \mathrm{d}p_b \\
&\quad = \binom{n_a^{\text{students}}}{n_a^{\text{pass}}} \binom{n_b^{\text{students}}}{n_b^{\text{pass}}} B(n_a^{\text{pass}} + \alpha^{\text{pass}}, n_a^{\text{fail}} + \alpha^{\text{fail}}) B(n_b^{\text{pass}} + \alpha^{\text{pass}}, n_b^{\text{fail}} + \alpha^{\text{fail}}), \quad (82)
\end{aligned}$$

where α^{pass} and α^{fail} encode the prior pseudocounts on the pass rate. The Bayes factor comparing the two hypotheses can be calculated

$$\begin{aligned}
&K_{\mathcal{B}}^{\text{BF}}(n_a^{\text{pass}}, n_a^{\text{fail}}, n_b^{\text{pass}}, n_b^{\text{fail}}) \\
&\qquad = \frac{B(n_a^{\text{pass}} + n_b^{\text{pass}} + \alpha^{\text{pass}}, n_a^{\text{fail}} + n_b^{\text{fail}} + \alpha^{\text{fail}})}{B(n_a^{\text{pass}} + \alpha^{\text{pass}}, n_a^{\text{fail}} + \alpha^{\text{fail}}) B(n_b^{\text{pass}} + \alpha^{\text{pass}}, n_b^{\text{fail}} + \alpha^{\text{fail}})}. \quad (83)
\end{aligned}$$

In addition we can generalise eq. (81) to more than two departments, with pass rates, $\boldsymbol{p}^{\text{pass}} = \{p_1, \cdots, p_{N_{\text{departments}}}\}$, observed passes counts $\boldsymbol{n}^{\text{pass}} = \{n_1^{\text{pass}}, \cdots, n_{N_{\text{departments}}}^{\text{pass}}\}$ and fail counts $\boldsymbol{n}^{\text{fail}} = \{n_1^{\text{fail}}, \cdots, n_{N_{\text{departments}}}^{\text{fail}}\}$ then we can calculate the Bayes factor for them having the same pass rate,

$$\begin{aligned}
&\Pr\left((p_j = p_k) \forall p_j, p_k \in \boldsymbol{p}^{\text{pass}} | \boldsymbol{n}^{\text{pass}}, \boldsymbol{n}^{\text{fail}}\right) \\
&\qquad = B\left(\alpha^{\text{pass}} + \sum_{j=1}^{N_{\text{departments}}} n_j^{\text{pass}}, \alpha^{\text{fail}} + \sum_{j=1}^{N_{\text{departments}}} n_j^{\text{fail}}\right) \prod_{j=1}^{N_{\text{departments}}} \binom{n_j^{\text{pass}} + n_j^{\text{fail}}}{n_j^{\text{pass}}}. \quad (84)
\end{aligned}$$

References

(1) Hoffman, M. D.; Gelman, A. *J. Mach. Learn. Res.* **2014**, *15*, 1593–1623.

(2) Betancourt, M. Nested sampling with constrained Hamiltonian Monte Carlo. *AIP Conf. Proc.* **2010**, *1305*, 165–172.

(3) Sharapov, V. A.; Mandelshtam, V. A. Solid-solid structural transformations in

lennard-jones clusters: Accurate simulations versus the harmonic superposition approximation. *J. Phys. Chem. A* **2007**, *111*, 10284–10291.

(4) Swendsen, R. H.; Wang, J.-S. Replica Monte Carlo simulation of spin glasses. *Phys. Rev. Lett.* **1986**, *57*, 2607–2609.

(5) Sugita, Y.; Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.* **1999**, *314*, 141–151.

(6) Earl, D. J.; Deem, M. W. Parallel tempering: Theory, applications, and new perspectives. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3910.

(7) Fukunishi, H.; Watanabe, O.; Takada, S. *J. Chem. Phys.* **2002**, *116*, 9058.

(8) Mandelshtam, V. A.; Frantsuzov, P. A.; Calvo, F. Structural transitions and melting in LJ 74-78 Lennard-Jones clusters from adaptive exchange Monte Carlo simulations. *J. Phys. Chem. A* **2006**, *110*, 5326–5332.

(9) Bogdan, T. V.; Wales, D. J.; Calvo, F. Equilibrium thermodynamics from basin-sampling. *J. Chem. Phys.* **2006**, *124*, 44102.

(10) Wales, D. J. Surveying a complex potential energy landscape: Overcoming broken ergodicity using basin-sampling. *Chem. Phys. Lett.* **2013**, *584*, 1–9.

(11) Frenkel, D.; Smit, B. *Understanding molecular simulation, second edition*; Academic Press: London, 2002.

(12) Duane, S.; Kennedy, A.; Pendleton, B. J.; Roweth, D. Hybrid Monte Carlo. *Phys. Lett. B* **1987**, *195*, 216–222.

(13) Betancourt, M. A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv Prepr. arXiv1701.02434* **2017**,

(14) Wang, J.-s.; Swendsen, R. H. Transition Matrix Monte Carlo Method. *J. Stat. Phys.* **2002**, *106*, 245.

(15) Wang, F.; Landau, D. P. Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States. *Phys. Rev. Lett.* **2001**, *86*, 2050–2053.

(16) Kim, J.; Straub, J. E.; Keyes, T. Statistical-temperature Monte Carlo and Molecular dynamics algorithms. *Phys. Rev. Lett.* **2006**, *97*, 1–4.

(17) Kim, J.; Keyes, T.; Straub, J. E. Replica exchange statistical temperature Monte Carlo. *J. Chem. Phys.* **2009**, *130*, 1–10.

(18) Kim, J.; Straub, J. E.; Keyes, T. Replica exchange statistical temperature molecular dynamics algorithm. *J. Phys. Chem. B* **2012**, *116*, 8646–8653.

(19) Skilling, J. Nested sampling for general Bayesian computation. *Bayesian Anal.* **2006**, *1*, 833–860.

(20) Higson, E.; Handley, W.; Hobson, M.; Lasenby, A. Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation. *Prepr.* **2017**,

(21) Handley, W. J.; Hobson, M. P.; Lasenby, A. N. POLYCHORD: Nested sampling for cosmology. *Mon. Not. R. Astron. Soc. Lett.* **2015**, *450*, L61–L65.

(22) Keeton, C. R. On statistical uncertainty in nested sampling. *Mon. Not. R. Astron. Soc.* **2011**, *414*, 1418–1426.

(23) Feroz, F.; Skilling, J. Exploring multi-modal distributions with nested sampling. AIP Conf. Proc. 2013; pp 106–113.

(24) Skilling, J. Bayesian computation in big spaces - Nested sampling and Galilean Monte Carlo. *AIP Conf. Proc.* **2012**, *1443*, 145–156.

(25) Feroz, F.; Hobson, M. P.; Bridges, M. MultiNest: An efficient and robust Bayesian inference tool for cosmology and particle physics. *Mon. Not. R. Astron. Soc.* **2009**, *398*, 1601–1614.

(26) Martiniani, S.; Stevenson, J. D.; Wales, D. J.; Frenkel, D. Superposition enhanced nested sampling. *Phys. Rev. X* **2014**, *4*, 31034.

(27) Li, Z.; Scheraga, H. A. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. *Proc. Natl. Acad. Sci. USA* **1987**, *84*, 6611.

(28) Li, Z.; Scheraga, H. A. Structure and free energy of complex thermodynamic systems. *J. Mol. Struct. THEOCHEM* **1988**, *179*, 333–352.

(29) Wales, D.; Doye, J. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms. *J. Phys. Chem. A* **1998**, *101*, 5111–5116.

(30) Maxwell, J. C. On Hills and Dales. *London, Edinburgh, Dublin Philos. Mag. J. Sci. 4th Ser.* **1870**, *40*, 421–426.

(31) Mezey, P. G. *Potential Energy Hypersurfaces*; Elsevier: Amsterdam, 1987.

(32) Wales, D. J. *Energy landscapes*; Cambridge University Press, 2004.

(33) Stillinger, F. H.; Weber, T. A. packing structures and transitions in liquids and solids. *Science* **1984**, *225*, 983.

(34) Stillinger, F. H. A Topographic View of Supercooled Liquids and Glass Formation. *Science* **1995**, *267*, 1935.

(35) Strodel, B.; Wales, D. J. Free energy surfaces from an extended harmonic superposition approach and kinetics for alanine dipeptide. *Chem. Phys. Lett.* **2008**, *466*, 105–115.

(36) Sharapov, V. A.; Meluzzi, D.; Mandelshtam, V. A. Low-temperature structural transitions: Circumventing the broken-ergodicity problem. *Phys. Rev. Lett.* **2007**, *98*, 105701.

(37) Wales, D. J. Coexistence in small inert gas clusters. *Mol. Phys.* **1993**, *78*, 151.

(38) Stillinger, F. H.; Weber, T. A. Hidden structure in liquids. *Phys. Rev. A* **1982**, *25*, 978–989.

(39) Franke, G.; Hilf, E. R.; Borrmann, P. The structure of small clusters: Multiple normal-modes model. *J. Chem. Phys.* **1993**, *98*, 3496–3502.

(40) Calvo, F.; Doye, J. P. K.; Wales, D. J. Characterization of Anharmonicities on Complex Potential Energy Surfaces: Perturbation Theory and Simulation. *J. Chem. Phys.* **2001**, *115*, 9627–9636.

(41) Becker, O. M.; Karplus, M. The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics. *J. Chem. Phys.* **1997**, *106*, 1495–1517.

(42) Wales, D. J.; Miller, M. a.; Walsh, T. R. Archetypal energy landscapes. *Nature* **1998**, *394*, 758–760.

(43) Krivov, S. V.; Karplus, M. Free energy disconnectivity graphs: Application to peptide models. *J. Chem. Phys.* **2002**, *117*, 10894–10903.

(44) Evans, D. A.; Wales, D. J. Free Energy landscapes of model peptides and proteins. *J. Chem. Phys.* **2003**, *118*, 3891–3897.

(45) Wales, D. J. Energy landscapes: calculating pathways and rates. *Int. Rev. Phys. Chem.* **2006**, *25*, 237–282.

(46) Miller, M. A.; Amon, L. M.; Reinhardt, W. P. Should one adjust the maximum step size in a Metropolis Monte Carlo simulation? *Chemical Physics Letters* **2000**, *331*, 278 – 284.

(47) Swendsen, R. H. How the maximum step size in Monte Carlo simulations should be adjusted. *Phys. Procedia* **2011**, *15*, 81–86.

(48) Pártay, L. B.; Bartók, A. P.; Csányi, G. Efficient sampling of atomic configurational spaces. *J. Phys. Chem. B* **2010**, *114*, 10502–10512.

(49) Neal, R. M. Slice sampling. *Ann. Statist.* **2003**, *31*, 705–767.

(50) Jones, J. E.; Ingham, A. E. On the calculation of certain crystal potential constants, and on the cubic crystal of least potential energy. *Proc. R. Soc. A* **1925**, *107*, 636–653.

(51) Labastie, P.; Whetten, R. L. ? *Phys. Rev. Lett.* **1990**, *65*, 1567.

(52) Calvo, F.; Neirotti, J. P.; Freeman, D. L.; Doll, J. D. Phase changes in 38 atom Lennard-Jones clusters. II: A parallel tempering study of equilibrium and dynamic properties in the molecular dynamics and microcanonical. *J. Chem. Phys.* **2000**, *112*, 10350–10357.

(53) Calvo, F.; Neirotti, J. P.; Freeman, D. L.; Doll, J. D. Phase changes in 38 atom Lennard-Jones clusters. II: A parallel tempering study of equilibrium and dynamic properties in the molecular dynamics and microcanonical. *J. Chem. Phys.* **2000**, *112*, 10350–10357.

(54) Noya, E. G.; Doye, J. P. K. Structural transitions in the 309-atom magic number Lennard-Jones cluster (6 pages). *J. Chem. Phys.* **2006**, *124*, 104503.

(55) Ballard, A. J.; Martiniani, S.; Stevenson, J. D.; Somani, S.; Wales, D. J. Exploiting the potential energy landscape to sample free energy. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2015**, *5*, 273–289.

(56) Griffiths, M.; Niblett, S. P.; Wales, D. J. Optimal Alignment of Structures for Finite and Periodic Systems. *J. Chem. Theory Comput.* **2017**, *13*, 4914–4931, PMID: 28841314.

(57) Wales, D. J. Decoding heat capacity features from the energy landscape. *Phys. Rev. E* **2017**, *95*, 030105(R).

(58) PELE: Python Energy Landsape Explorer. `https://github.com/pele-python/pele`.

(59) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95.