

Monte Carlo on Manifolds: Sampling Densities and Integrating Functions

EMILIO ZAPPA
Courant Institute

MIRANDA HOLMES-CERFON
Courant Institute

AND

JONATHAN GOODMAN
Courant Institute

Abstract

We describe and analyze some Monte Carlo methods for manifolds in euclidean space defined by equality and inequality constraints. First, we give an MCMC sampler for probability distributions defined by unnormalized densities on such manifolds. The sampler uses a specific orthogonal projection to the surface that requires only information about the tangent space to the manifold, obtainable from first derivatives of the constraint functions, hence avoiding the need for curvature information or second derivatives. Second, we use the sampler to develop a multistage algorithm to compute integrals over such manifolds. We provide single-run error estimates that avoid the need for multiple independent runs. Computational experiments on various test problems show that the algorithms and error estimates work in practice. The method is applied to compute the entropies of different sticky hard sphere systems. These predict the temperature or interaction energy at which loops of hard sticky spheres become preferable to chains. © 2018 Wiley Periodicals, Inc.

1 Introduction

Many physical and engineering systems involve motion with equality constraints. For example, in physics and chemistry, bonds between atoms or colloidal particles may be modeled as having a fixed length [17], while in robotics, joints or hinges connecting moving components may be written as constraints on distances or angles [36]. Constraints also arise in statistics on the set of parameters in parametric models [8]. In such problems the space of accessible configurations is lower-dimensional than the space of variables that describe the system, often forming a manifold embedded in the full configuration space. One may then be interested in sampling a probability distribution defined on the manifold or calculating an integral over the manifold such as its volume.

This paper presents Monte Carlo methods for computing a d -dimensional integral over a d -dimensional connected manifold M embedded in a d_a -dimensional euclidean space, with $d_a \geq d$. The manifold is defined by sets of constraints. We first describe an MCMC sampler for manifolds defined by constraints in this way. Then we describe a multiphase procedure that uses the sampler to estimate integrals. It should go without saying (but rarely does) that deterministic algorithms to compute integrals are impractical except in very low dimensions or for special problems (e.g., [32]).

Our MCMC sampler has much in common with other algorithms for constraint manifolds, including other samplers, optimizers, differential algebraic equation (DAE) solvers (see, e.g., [14]), etc. A move from a point $x \in M$ starts with a move in the tangent space at x , which is $v \in T_x$. This is followed by a projection back to $y \in M$, which we write as $y = x + v + w$. The sampler simplifies if we require $w \perp T_x$, so the projection is perpendicular to the tangent space *at the original point*. This idea was first suggested in [24].¹ This is different from other natural choices, such as choosing $y \in M$ to minimize the projection distance $\|y - (x + v)\|$. The choice of $w \perp T_x$ makes the sampler easy to implement as it requires only first derivatives of the constraint surfaces. Neither solving for y using Newton's method nor the Metropolis Hastings detailed balance condition require any higher derivatives.

Several other surface sampling methods require second derivatives, such as the explicit parametrization method of [8], the Hamiltonian method such as of [2], and geodesic methods such as [3, 23]. Methods based on discretizing SDEs (e.g., [4, 5]) are exact samplers only in continuous time. There is time discretization error (inexact sampling) when implemented with a nonzero time step. It is possible to use Metropolis rejection to remove time discretization error, but the volume ratio involved would involve second derivatives of the constraint functions if the projection is to the closest point on the constraint surface. Exact sampling without time discretization error is desirable because it allows larger time steps.

It is common in our computational experiments that there is no $w \perp T_x$ so that $x + v + w \in M$, or that one or more w -values exist but the Newton solver fails to find one. Section 2 explains how we preserve detailed balance even in these situations.

Our goal is to compute integrals of the form

$$(1.1) \quad Z = \int_M f(x) \sigma(dx),$$

where $\sigma(dx)$ is the d -dimensional surface area measure (Hausdorff measure) and f is a positive smooth function. Writing $B \subset \mathbb{R}^{d_a}$ for the ball centered at a point $x_0 \in M$, we also estimate certain integrals of the form

$$(1.2) \quad Z_B = \int_{M \cap B} f(x) \sigma(dx).$$

¹ We learned of this work after we submitted the original version of this paper for publication.

For each B , we define a probability distribution on $M \cap B$:

$$(1.3) \quad \rho_B(dx) = \frac{1}{Z_B} f(x) \sigma(dx).$$

The sampling algorithm outlined above, and described in more detail in Section 2, allows us to draw samples from ρ_B .

To calculate integrals as in (1.1) we use a multiphase strategy related to nested sampling and thermodynamic integration (e.g., [12]). Similar strategies were applied to volume estimation problems by the Lovasz school; see, e.g., [26]. Let $B_0 \supset B_1 \supset \cdots \supset B_k$ be a nested contracting family of balls with a common center. With an abuse of notation, we write Z_i for Z_{B_i} . We use MCMC sampling in $M \cap B_i$ to estimate the ratio

$$(1.4) \quad R_i = \frac{Z_i}{Z_{i+1}}.$$

We choose the last ball B_k small enough that we may estimate Z_k by direct Monte Carlo integration. We choose B_0 large enough that $M \subset B_0$ so $Z = Z_0$. Our estimate of Z is

$$(1.5) \quad \hat{Z} = \hat{Z}_k \prod_{i=0}^{k-1} \hat{R}_i,$$

where hatted quantities such as \hat{Z} are Monte Carlo estimates. Section 3 describes the procedure in more detail.

Error estimates should be a part of any Monte Carlo computation. Section 3.2 describes our procedure for estimating

$$\sigma_{\hat{Z}}^2 = \text{var}(\hat{Z}),$$

a procedure adapted from [18]. If we can estimate $\sigma_{\hat{R}_i}^2$, and if these are small enough, then we can combine them to estimate $\sigma_{\hat{Z}}^2$. We explore how $\sigma_{\hat{Z}}^2$ depends on parameters of the algorithm via several simplified model problems in Section 5. We specifically analyze how the error depends on the amount by which we shrink the balls for each ratio, and make predictions that could lead to good systematic choices of this parameter.

We illustrate our methods on a number of examples. We apply the sampler to the surface of a torus and cone in 3D, as well as the special orthogonal group $SO(n)$ for $n = 11$, which is a manifold of dimension 55 in an ambient space of dimension 121 (Section 2.3). All of these examples have certain marginal distributions that are known analytically, so we verify the sampler is working correctly.

We verify the integration algorithm is correct by calculating the surface areas of a torus and $SO(n)$ for $2 \leq n \leq 7$ (Sections 4.1 and 4.2).

Finally, we apply our algorithm to study clusters of sticky spheres (Section 4.3), a model that is studied partly to understand how colloidal particles self-assemble

into various configurations [16, 17, 30]. A simple question in that direction concerns a linear chain of n -spheres—whether it is more likely to be closed in a loop or open in a chain. According to equilibrium statistical mechanics, the ratio of the probabilities of being open versus closed depends on a parameter (the “sticky parameter”) characterizing the binding energy and temperature, and the entropies, which are surface areas of the kind we are computing. We compute the entropies for open and closed chains of lengths 4 through 10. This data can be used to predict the relative probabilities of being in a loop or a chain if the sticky parameter is known, but if it is not known our calculations can be used to measure it by comparing with data.

Notation and assumptions. Throughout the paper, we will denote by M a d -dimensional connected manifold embedded in an ambient space \mathbb{R}^{d_a} , which is defined by equality and inequality constraints. There are m equality constraints, $q_i(x) = 0$, $i = 1, \dots, m$, where the q_i are smooth functions $q_i : \mathbb{R}^{d_a} \rightarrow \mathbb{R}$. There are l inequality constraints of the form $h_j(x) > 0$ for $j = 1, \dots, l$. The manifold M is

$$(1.6) \quad M = \{x \in \mathbb{R}^{d_a} : q_i(x) = 0, i = 1, \dots, m, h_j(x) > 0, j = 1, \dots, l\}.$$

If M has several connected components, then the sampler may sample only one component, or it may hop between several components. It is hard to distinguish these possibilities computationally.

We let Q_x be the matrix whose columns are the gradients $\{\nabla q_i(x)\}_{i=1}^m$. This makes Q_x the transpose of the Jacobian of the overall constraint function $q : \mathbb{R}^{d_a} \rightarrow \mathbb{R}^m$. The entries of Q_x are

$$(1.7) \quad (Q_x)_{ij} = \frac{\partial q_i(x)}{\partial x_j}.$$

We assume that Q_x has full rank m everywhere on M . By the implicit function theorem, this implies that the dimension of M is $d = d_a - m$ and that the tangent space $T_x \equiv T_x M$ at a point $x \in M$ is well-defined. In this case $\{\nabla q_i(x)\}_{i=1}^m$ form a basis of the orthogonal space $T_x^\perp \equiv T_x M^\perp$. Note that M inherits the metric from the ambient space \mathbb{R}^{d_a} by restriction. The corresponding d -dimensional volume element is d -dimensional Hausdorff measure, which we denote by $\sigma(dx)$.

2 Sampling on a Constraint Manifold

2.1 The Algorithm

We describe the MCMC algorithm we use to sample a probability distribution of the form

$$(2.1) \quad \rho(dx) = \frac{1}{Z} f(x) \sigma(dx).$$

MCMC surface sampling algorithm

```

Begin:  $x = X_n$ ,  $f(x)$  is known {
  Proposal: generate  $y \in M$  or declare failure
  Calculate  $Q_x$  using (1.7)
  Find orthonormal bases for  $T_x$  and  $T_x^\perp$  using the QR decomposition of  $Q_x$ 
  Generate  $v \in T_x$  with  $v \sim p(v)$  using the orthonormal basis of  $T_x$ 
  Projection to  $M$ :
     $[a, \text{flag}] = \text{project}(q, x + v, Q_x)$  // solve  $q(x + v + Q_x a) = 0$  for  $a$ 
    if  $\text{flag} == 0$  {
       $X_{n+1} = x$ ; done // projection failure: reject, back to Begin.
    }
     $y = x + v + Q_x d$  // successful projection
  Inequality check:
    if  $h_i(y) < 0$  for some  $y$  {
       $X_{n+1} = x$ ; done // inequality failure: reject; back to Begin.
    }
  Metropolis-Hastings acceptance/rejection step:
    Generate  $Q_y$  and find orthonormal bases for  $T_y$  and  $T_y^\perp$ 
    Find  $v' \in T_y$  and  $w' \in T_y^\perp$  so that  $x = y + v' + w'$ 
     $P_a = \min\left(1, \frac{f(y)p(v')}{f(x)p(v)}\right)$  // acceptance probability
    Generate  $U \sim \text{unif}(0, 1)$ 
    if  $U > P_a$  {
       $X_{n+1} = x$ ; done //  $\text{Pr}(\text{rej}) = 1 - P_a$ 
      // rejection; back to Begin.
    }
  Reverse Projection, try to recover  $x = y + v' + Q_y a$ 
     $[a, \text{flag}] = \text{project}(q, y + v', Q_y)$ 
    if  $\text{flag} == 0$  {
       $X_{n+1} = x$ ; done // reverse projection failure: reject, back to Begin.
    }
  Accept move:  $X_{n+1} = y$  // successful recovery; accept the proposal
}

 $[a, \text{flag}] = \text{project}(q, z, Q)$  // Newton's method to solve  $q(z + Qa) = 0$  for  $a$ 
Initialization:  $a = 0, i = 0, \text{flag} = 1$ 
While  $|q(z + Qa)| > \text{tol}$  {
  Solve  $(\nabla q(z + Qa))^T Q \Delta a = -q(z + Qa)$  for  $\Delta a$  // see (2.6)
   $a = a + \Delta a$ 
   $i = i + 1$ 
  if  $i > \text{nmax}$  {
     $\text{flag} = 0$ ; return // projection failure: reject; return.
  }
}

```

FIGURE 2.1. Summary of the MCMC algorithm in pseudocode.

Here σ is the d -dimensional surface measure on M given by (1.6). As usual, the algorithm requires repeated evaluation of f but does not use the unknown normalization constant Z . The algorithm is described in pseudocode in Figure 2.1.

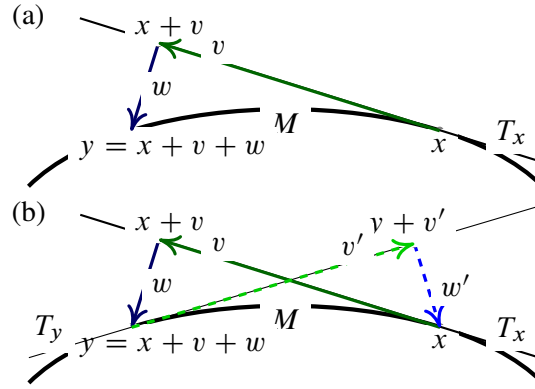


FIGURE 2.2. Illustration of the sampler. (a) The vector $v \in T_x$ is projected normal to T_x to $y = x + v + w \in M$. (b) The reverse step, needed for detailed balance, has $v' \in T_y$, $w' \in T_y^\perp$, and $x = y + v' + w'$. The angle between w and T_x is the same as the angle between w' and T_x .

Our Metropolis style MCMC algorithm generates a sequence $X_k \in M$ with the property that

$$(2.2) \quad \text{if } X_k \sim \rho \text{ then } X_{k+1} \sim \rho.$$

Once $X_k = x$ is known, the algorithm makes a proposal $y \in M$. If the proposal fails (see below) or if it is rejected, then $X_{k+1} = X_k$. If the algorithm succeeds in generating a y , and if y is accepted, then $X_{k+1} = y$. If M is connected, compact, and smooth, then the algorithm is geometrically ergodic. See, e.g., [25] for an efficient summary of the relevant MCMC theory.

The proposal process begins with a tangential move $x \rightarrow x + v$ with $v \in T_x$. We generate v by sampling a proposal density $p(v|x)$ defined on T_x . All the numerical experiments reported here use an isotropic d -dimensional Gaussian with width s centered at x :

$$(2.3) \quad p(v|x) = \frac{1}{(2\pi)^{d/2} s^d} e^{-\frac{|v|^2}{2s^2}}.$$

We generate v using an orthonormal basis for T_x , which is the orthogonal complement of the columns of the constraint gradient matrix Q_x ; see (1.7). This orthonormal basis is found as the last d columns of the $d_a \times d_a$ matrix Q in the QR decomposition of Q_x , which we find using dense linear algebra.²

Given x and v , the projection step looks for $w \in T_x^\perp$ with $y = x + v + w \in M$. See Figure 2.2(a) for an illustration. It does this using an m -component column

² Please be aware of the conflict of notation here. The $d_a \times d_a$ orthogonal Q of the QR decomposition is not the $d_a \times d$ gradient Q_x .

vector $a = (a_1, \dots, a_m)^\top$ and

$$(2.4) \quad w = \sum_{j=1}^m a_j \nabla q_j(x) = Q_x a.$$

The unknown coefficients a_j are found by solving the nonlinear equations

$$(2.5) \quad q_i(x + v + Q_x a) = 0, \quad i = 1, \dots, m.$$

This can be done using any nonlinear equation solver. In our code, we solve (2.5) using simple Newton's method (no line search, regularization, etc.) and initial guess $a = 0$. The necessary Jacobian entries are

$$(2.6) \quad J_{ij} = \partial_{a_j} q_i(x + v + Q_x a) = (\nabla q_i(x + v + Q_x a))^\top \nabla q_j(x).$$

We iterate until the first time a convergence criterion is satisfied,

$$(2.7) \quad |q(x + v + Q_x a)| \leq \mathbf{tol},$$

where $|q|$ is some norm. In our implementation, we used the l_2 norm. When this criterion is satisfied, the projection is considered a success and $y = x + v + w$ is the proposal. If the number of iterations reaches **nmax** before the convergence criterion is satisfied, then the projection phase is considered a failure. It is possible to change these details, but if you do you must also make the corresponding changes in the detailed balance check below.

There are other possible projections from $x + v$ to M , each with advantages and disadvantages. It would seem natural, for example, to find y by computing

$$(2.8) \quad \arg \min_{y \in M} |x + v - y|_2.$$

This produces a projection $w \in T_y^\perp$ rather than our $w \in T_x^\perp$. It also has the advantage that such a y always exists and is almost surely unique. The optimization algorithm that computes (2.8) may be more robust than our Newton-based nonlinear equation solver. The disadvantage of (2.8) is that there are curvature effects in the relative surface area computation that are part of detailed balance. Computing these curvatures requires second derivatives of the constraint functions q_i . This is not an intrinsic limitation, but requires more computation and is not as straightforward to implement as our algorithm.

The proposal algorithm just described produces a random point $y \in M$. There are two cases where we must reject it immediately so the detailed balance relation may hold (which we verify in Section 2.2). First, we must check whether an inequality constraint is violated, so $h_i(y) \leq 0$ for some i . If so, then y is rejected. Second, we must check whether the reverse step is possible, i.e., whether it is possible to propose x starting from y . To make the reverse proposal, the algorithm would have to choose $v' \in T_y$ so that $x = y + v' + w'$ with $w' \perp T_y$. Figure 2.2(b) illustrates this. Since x and y are known, we may find v' and w' by the requirement that $x - y = v' + w'$, $v' \in T_y$, $w' \in T_y^\perp$. This is always possible

by projecting $x - y$ onto T_y and T_y^\perp , which are found using the QR decomposition of Q_y . However, we must additionally determine whether the Newton solver would find x starting from $y + v'$, a step that was neglected in earlier studies, e.g., [24]. Even though we know that x is a solution to the equations, we must still verify that the Newton solver produces x . With that aim, we run the Newton algorithm from initial guess $y + v'$ and see whether the converge criterion (2.7) is satisfied within **nmax** iterations. If not, the proposal y is rejected. Whenever y is rejected, we set $X_{k+1} = x$.

If we come to this point, we compute an acceptance probability $a(y|x)$ using the Metropolis Hastings formula

$$(2.9) \quad a(y|x) = \min\left(1, \frac{f(y)p(v'|y)}{f(x)p(v|x)}\right).$$

The reader might expect that we should use the more complicated formula

$$a(y|x) = \min\left(1, \frac{f(y)p(v'|y)J(x|y)}{f(x)p(v|x)J(y|x)}\right),$$

where $J(y|x)$ is the inverse of the determinant of the linearization of the projection $v \rightarrow y$, formally, $|\frac{\partial v}{\partial y}|$. This would account for how the area of a small patch of surface near x is distorted upon mapping it to y . However, it turns out that the Jacobian factors cancel:

$$(2.10) \quad J(y|x) = J(x|y).$$

This is evident in part (b) of Figure 2.2, where the angle between w and T_y is the same as the angle between w' and T_x . The proof in two dimensions is an exercise in euclidean geometry. In more dimensions, $J(y|x)$ and $J(x|y)$ are products of cosines of principal angles between T_x and T_y ; see [1].

To prove (2.10), let U_x and U_y be matrices whose columns are orthonormal bases for T_x and T_y , respectively. Column i of U_x will be denoted $u_{x,i}$, and similarly for U_y . Consider a vector $\xi \in T_x$, which may be written $\xi = \sum a_i u_{x,i}$. Let $\eta \in T_y$ be the projection of ξ to T_y normal to T_x , and write $\eta = \sum b_j u_{y,j}$. The orthogonality condition is $u_{x,i}^\top \eta = u_{x,i}^\top \xi = a_i$. This leads to the equations

$$\sum_j u_{x,i}^\top u_{y,j} b_j = a_i.$$

These take the form $Pb = a$, or $b = P^{-1}a$, where $P = U_x^\top U_y$. Since U_x and U_y are orthonormal bases, the volume element going from $x \rightarrow y$ is expanded by a factor

$$(2.11) \quad J(y|x)^{-1} = \det(P^{-1}) = \det((U_x^\top U_y)^{-1}).$$

Similar reasoning shows that the projection from T_y to T_x perpendicular to T_y expands volumes by a factor of

$$(2.12) \quad J(x|y)^{-1} = \det((U_y^\top U_x)^{-1}).$$

These factors are equal because the determinants of a matrix and its transpose are the same, proving (2.10).

We close this section with a general remark concerning the choice of parameters. A key part of the algorithm is the projection step, which requires using Newton's method or some other algorithm to solve a nonlinear system of equations. This algorithm is called twice at every step of the sampler, and so making it efficient is critical to the overall efficiency of the sampler. Profiling reveals that most of the computing time is spent in the **project** part (see Figure 2.1). We found that setting the parameter **nmax** low, or otherwise terminating the solver rapidly, made a significant difference in the overall efficiency. Unlike most problems that require solving equations, we are *not* interested in guaranteeing that we can find a solution if one exists—we only need to find certain solutions rapidly; our reverse projection check and subsequent rejection ensures that we correct for any asymmetries in the solver and still satisfy detailed balance.

2.2 Verifying the Balance Relation

In this section we verify what may already be clear, that the algorithm described above satisfies the desired balance relation (2.2). In the calculations below, we assume that the constraints hold *exactly*, an assumption that will not be true when they are solved for numerically, as we discuss in the remark at the end of the section.

To start, let us express the probability distribution of the proposal point y as the product of a surface density and d -dimensional surface measure,

$$(2.13) \quad P_x(dy) = h(y|x)\sigma(dy) + \xi(x)\delta_x(dy).$$

Here $\delta_x(dy)$ is the unit point mass probability distribution at the point x , $\xi(x)$ is the probability that the projection step failed; i.e., Newton's method failed to produce a y or it lay outside the boundary, and $h(y|x)$ is the proposal density, i.e., the probability density of successfully generating $y \in M$ from x . The proposal density is the product of three factors:

$$h(y|x) = (1 - \mathbf{1}_{F_x}(y)) \cdot p(v|x) \cdot J(y|x),$$

where the set $F_x \subset M$ consists of those points $y \in M$ that cannot be reached using our Newton solver starting from x , and $\mathbf{1}_{F_x}(y)$ denotes the characteristic function of the set F_x , which equals 1 if $y \in F_x$ and 0 if $y \notin F_x$. Note that we must have $\int_M h(y|x)d\sigma(y) + \xi(x) = 1$.

If the acceptance probability is $a(y|x)$, then the overall probability distribution of y takes the form

$$(2.14) \quad R_x(dy) = a(y|x)h(y|x)\sigma(dy) + \xi(x)\delta_x(dy).$$

Here, $\xi(x)$ is the overall probability of rejecting a move when starting at x , equal to the sum of the probability of not generating a proposal move, and the probability of successfully generating a proposal move that is subsequently rejected. Since

$1 - a(y|x)$ is the probability of rejecting a proposal at y , the overall rejection probability is

$$\xi(x) = \zeta(x) + \int_M (1 - a(y|x))h(y|x)\sigma(dy).$$

We quickly verify that the necessary balance relation (2.2) is satisfied if $a(y|x)$ is chosen to satisfy the detailed balance formula

$$(2.15) \quad f(x)a(y|x)h(y|x) = f(y)a(x|y)h(x|y).$$

For this, suppose X_k has probability distribution on M with density $g_k(x)$ with respect to surface area measure:

$$X_k \sim \rho_k(dx) = g_k(x)\sigma(dx).$$

We check that if (2.15) is satisfied, and $g_k = f$, then $g_{k+1} = f$. We do this in the usual way, which is to write the integral expression for g_{k+1} and then simplify the result using (2.15). To start,

$$g_{k+1}(x) = \int_M g_k(y)a(x|y)h(x|y)d\sigma(y) + \xi(x)g_k(x).$$

The integral on the right represents jumps to x from $y \neq x$ on M . The second term on the right represents proposals from x that were unsuccessful or rejected. If $g_k = f$, when using the detailed balance relation (2.15) this becomes

$$(2.16) \quad g_{k+1}(x) = \int_M f(x)a(y|x)h(y|x)d\sigma(y) + \xi(x)f(x).$$

The integral on the right now represents the probability of accepting a proposal from x :

$$\text{Pr}(\text{accept from } x) = \int_M a(y|x)h(y|x)d\sigma(y).$$

When combined with the second term, the right side of (2.16) is equal to $f(x)$, as we wished to show.

We use the standard Metropolis Hastings formula to enforce (2.15):

$$(2.17) \quad a(y|x) = \min\left(1, \frac{f(y)h(x|y)}{f(x)h(y|x)}\right).$$

If we succeed in proposing y from x , then we know $y \notin F_x$, so we are able to evaluate $h(y|x)$. However, it remains possible that $x \notin F_y$, in which case $h(x|y) = 0$. This is why we must apply the Newton solver to the reverse move, to see whether it would succeed in proposing x from y . If not, we know $h(x|y) = 0$ and therefore $a(y|x) = 0$ and we must reject y .

Remark 2.1. We have assumed so far the constraints hold exactly, which is not true when we solve for y numerically. Rather, the proposal move will lie within some tolerance tol , so we are sampling a “fattened” region

$$M_{\text{tol}} = \{y : |q_i(y)| \leq \text{tol}, i = 1, \dots, m, h_j(x) > 0, j = 1, \dots, l\}.$$

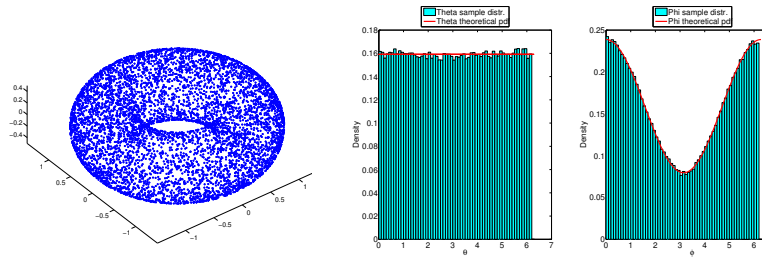


FIGURE 2.3. Left: Points generated by our MCMC sampler on a torus. We generated 10^6 points total and plot 10^4 of these, one every 100 MCMC steps. Center and right: empirical and theoretical distribution of the torus angles θ (center) and ϕ (right). The theoretical ϕ density (see, e.g., [8]) is $\frac{1}{2\pi}(1 + \frac{r}{R} \cos(\phi))$.

It is well-known (e.g., [11, 15]) that if the fattened region is uniformly sampled, the distribution near M will not be the uniform distribution, but rather will differ by an $O(1)$ amount that does not vanish as $\tau_{01} \rightarrow 0$.

We do not know the distribution by which our numerical solver produces a point in $M_{\tau_{01}}$. However, since we observe the correct distributions on M in the examples below (for small enough τ_{01}), we infer that we are *not* sampling the fattened region uniformly, but rather in a way that is consistent with the surface measure on the manifold in the limit when $\tau_{01} \rightarrow 0$. Moreover, it is practical to take τ_{01} to be extremely small, on the order of machine precision, because of the fast local convergence of Newton's method.

2.3 Examples

In this section we illustrate the MCMC sampler with three examples: a torus, a cone, and the special orthogonal group $SO(n)$.

Torus

Consider a torus \mathbb{T}^2 embedded in \mathbb{R}^3 , implicitly defined by

$$(2.18) \quad \mathbb{T}^2 = \left\{ (x, y, z) \in \mathbb{R}^3 : \left(R - \sqrt{x^2 + y^2} \right)^2 + z^2 - r^2 = 0 \right\},$$

where R and r are real positive numbers, with $R > r$. Geometrically, \mathbb{T}^2 is the set of points at distance r from the circle of radius R in the (x, y) -plane centered at $(0, 0, 0)$. An explicit parametrization of \mathbb{T}^2 is given by

$$(2.19) \quad \mathbb{T}^2 = \left\{ (R + r \cos(\phi) \cos(\theta), R + r \cos(\phi) \sin(\theta), r \sin(\phi)) : \theta, \phi \in [0, 2\pi] \right\}.$$

We ran $N = 10^6$ MCMC steps to sample uniform measure ($f(x) = 1$) on \mathbb{T}^2 with toroidal radius $R = 1$, poloidal radius $r = 0.5$, and step size scale $s = 0.5$. Figure 2.3 shows our empirical marginal distributions and the exact theoretical

distributions. The empirical distributions of the toroidal angle θ and the poloidal angle ϕ are correct to within statistical sampling error. Around 6% of proposed moves were rejected because of failure in the reverse projection step. When we didn't implement this step, the marginals were not correct.

Cone

We consider the circular right-angle cone $\mathcal{C} \subset \mathbb{R}^3$ with vertex $(0, 0, 0)$ given by

$$(2.20) \quad \mathcal{C} = \left\{ (x, y, z) \in \mathbb{R}^3 : z - \sqrt{x^2 + y^2} = 0, \ x^2 + y^2 < 1, \ z > 0 \right\}.$$

We study this example in order to determine how our algorithm might behave near singular points of a hypersurface. We have written strict inequalities to be consistent with the general formalism (1.6). In principle, the algorithm should do the same thing with nonstrict inequalities, since the set of points with equality (e.g., $z = 0$) has probability 0. If we allowed $z = 0$, then \mathcal{C} would not be a differentiable manifold because of the singularity at the vertex.

We suspected that the projection steps (see Figure 2.1) would be failure prone near the vertex because of high curvature, and that the autocorrelation times would be large or even unbounded. Nevertheless, Figure 2.4 shows satisfactory results, at least for this example. We ran for $N = 10^6$ MCMC steps with step size parameter $s = 0.9$. The theoretical marginal densities for X , Y , and Z are easily found to be

$$g_X(x) = \frac{2}{\pi} \sqrt{1 - x^2}, \quad g_Y(y) = \frac{2}{\pi} \sqrt{1 - y^2}, \quad g_Z(z) = 2z.$$

Special Orthogonal Group

We apply our surface sampler to the manifold $SO(n)$, which is the group of $n \times n$ orthogonal matrices with determinant equal to 1. We chose $SO(n)$ because these manifolds have high dimension, $\frac{1}{2}n(n-1)$, and high codimension, $\frac{1}{2}n(n+1)$ [22]. We view $SO(n)$ as the set of $n \times n$ matrices, $x \in \mathbb{R}^{n \times n}$, that satisfy the row orthonormality constraints for $k = 1, \dots, n$ and $l > k$:

$$g_{kk}(x) = \sum_{m=1}^n x_{km}^2 = 1, \quad g_{kl}(x) = \sum_{m=1}^n x_{km}x_{lm} = 0.$$

This is a set of $\frac{1}{2}n(n+1)$ equality constraints. The gradient matrix Q_x defined by (1.7) may be seen to have full rank for all $x \in SO(n)$. Any x satisfying these constraints has $\det(x) = \pm 1$. The set with $\det(x) = 1$ is connected. It is possible that our sampler would propose an x with $\det(x) = -1$, but we reject such proposals. We again chose density $f(x) = 1$, which gives $SO(n)$ a measure that is, up to a constant, the same as the natural Haar measure on $SO(n)$.

One check of the correctness of the samples is the known large- n distribution of $T = \text{Tr}(x)$, which converges to a standard normal as $n \rightarrow \infty$ [7]. This is surprising in view of the fact that $\text{Tr}(x)$ is the sum of n random numbers, each of which has $\text{Var}(x_{kk}) = 1$. If the diagonal entries were independent with this variance, then

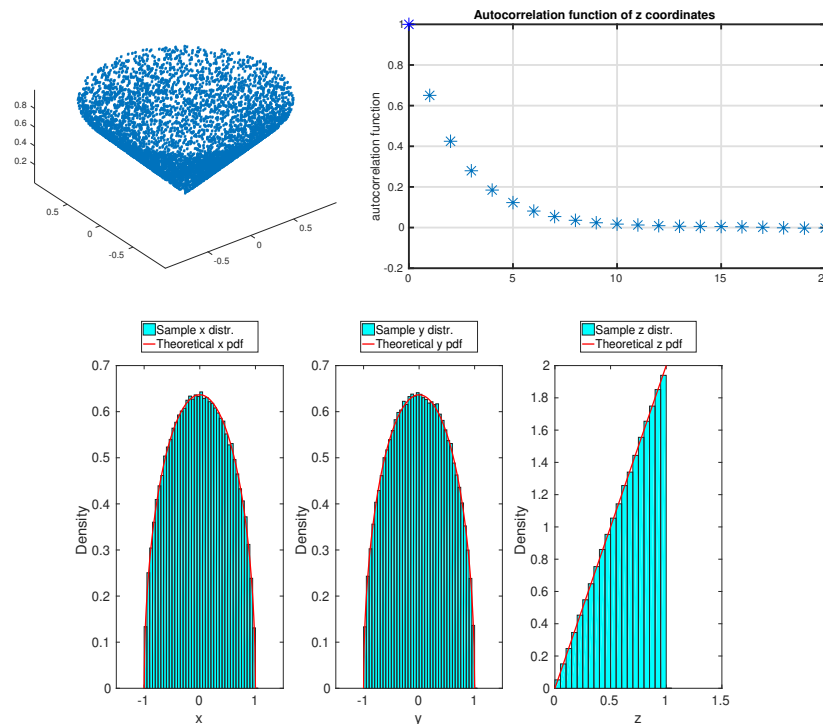


FIGURE 2.4. An MCMC run of length 10^6 for uniform sampling on a cone. Top left: plot of 10^4 points of the MCMC chain, over every 100 steps. Top right: autocorrelation function of the coordinates Z . Bottom: empirical and theoretical distributions for the coordinates (X, Y, Z) .

$\text{Var}(T) = n^{1/2}$ instead of the correct $\text{Var}(T) = 1$. This makes the correctness of the T -distribution an interesting test of the correctness of the sampler.

Figure 2.5 presents results for $n = 11$ and $N = 10^6$ MCMC steps, and proposal length scale $s = 0.28$. The distribution of T seems correct but there are nontrivial autocorrelations. The acceptance probability was about 35%.

Note that there are better ways to sample $SO(n)$ than to use our sampler. For example, one can produce independent samples by choosing $y \in \mathbb{R}^{n \times n}$ with i.i.d. entries from $\mathcal{N}(0, 1)$ and use the QR decomposition $y = Rx$, where R is upper triangular and x is orthogonal.

3 Integrating over a Manifold

3.1 Algorithm

In this section we describe an algorithm to calculate integrals over a manifold M of the form (1.1). We assume that M given in (1.6) is bounded. The strategy as outlined in the Introduction is to consider a sequence of d_a -dimensional balls

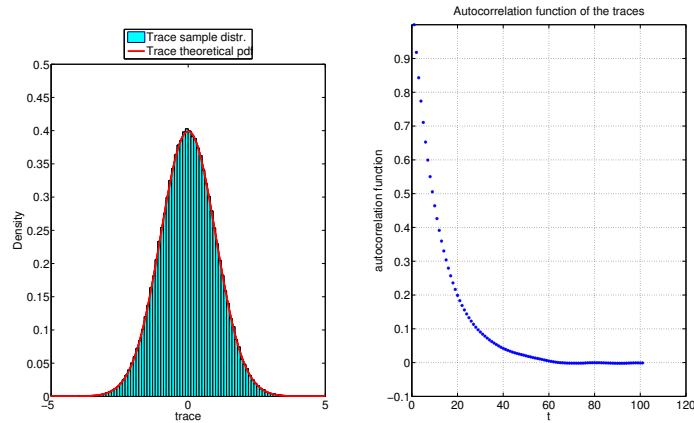


FIGURE 2.5. An MCMC run of length 10^6 for uniform sampling from $SO(11)$. Left: the theoretical and empirical distribution of $T = \text{Tr}(x)$. Right, the empirical autocorrelation of T .

$B_0 \supset B_1 \supset \cdots \supset B_k$, centered at $x_0 \in M$ with radii r_i for $i = 1, \dots, k$, where $B_0 \supset M$ entirely contains the manifold, and B_k is “small enough” in a way we describe momentarily. We *assume* that each $M \cap B_i$ is connected, an assumption that is critical for our algorithm to work. We define the collection of integrals (see (1.2))

$$Z_i = \int_{B_i \cap M} f(x) d\sigma(x).$$

The final estimate is $Z \approx \hat{Z}$ according to (1.5).

To estimate the ratio $R_i = Z_i / Z_{i+1}$, we generate n_i points in $B_i \cap M$ with the probability distribution ρ_i as in (2.11). This is done using the MCMC algorithm given in Section 2 by including the extra inequality $|x - x_0|^2 < r_i^2$. An estimator for R_i is then given by

$$(3.1) \quad \hat{R}_i = \frac{n_i}{N_{i,i+1}},$$

where, for $i \leq j$, we define

$$(3.2) \quad N_{i,j} = \# \text{ points generated in } B_i \cap M \text{ that also lie in } B_j \cap M.$$

We could use additional elements $N_{i,j}$ with $j > i + 1$ to enhance the estimate of the ratios, but we leave exactly how to do this as a question for future research. Note that $N_{i,i} = n_i$.

The integral over the smallest set $M \cap B_k$ is computed in a different way. If M had no equality constraints, so that the dimension of M were equal to the dimension of B_k , then we could choose $B_k \subset M$ to lie entirely inside M and so the integral (for constant f) would be known analytically. Indeed, this idea was used

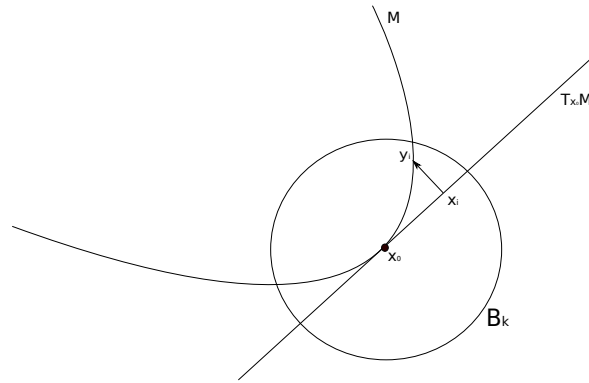


FIGURE 3.1. Illustration of the projection step for the integration over the smallest intersection $B_k \cap M$ for a one-dimensional manifold: given a point x_i uniformly generated in the ball $B_k \cap T_{x_0}M$, we find the unique point $y_i \in M$ by moving perpendicular to $T_{x_0}M$, starting from x_i .

in several studies that developed efficient algorithms to calculate the volume of a convex body [26, 32]. With the equality constraints, in general there is no B_k with radius $r_k > 0$ for which the integral Z_k is known in closed form exactly.

Instead, we choose r_k small enough that there is an easily computed one-to-one correspondence between points in $M \cap B_k$ and points in the tangent plane $D_k = T_{x_0} \cap B_k$. This happens, for example, when the manifold $M \cap B_k$ is a graph above D_k . We use the coordinates in the flat disk D_k to integrate over $M \cap B_k$, which is simply the surface integral as illustrated in Figure 3.1:

$$(3.3) \quad Z_k = \int_{M \cap B_k} f(y) d\sigma(y) = \int_{D_k} f(y(x)) J^{-1}(x) dx.$$

This uses the following notation. The integral on the right is over the d -dimensional disk. For any $x \in D_k$, the projection to M that is perpendicular to T_{x_0} is called $y(x) \in M$. For small enough r_k , the projection is easily calculated using the projection algorithm of Figure 2.1. The Jacobian factor $J(x) = \det(U_{x_0}^T U_{y(x)})$ is known from (2.11), where as before U_x is a matrix whose columns are an orthonormal basis of T_x .

The integral over D_k may be computed deterministically if d is small enough, but otherwise we estimate it by direct Monte Carlo integration. We choose n_k independent points $x_i \in D_k$, which is an easy task for a d -dimensional disk. We compute the corresponding projections and Jacobian weight factors, $y_i = y(x_i)$ and $J_i = J(x_i)$. Some of the y_i will be outside of $M \cap B_k$, so we need an extra indicator function factor, $\mathbf{1}_{B_k}(y_i)$, that is equal to 1 if $y_i \in M \cap B_k$. The d -dimensional volume of D_k is known, so we have the direct Monte Carlo estimate

$$(3.4) \quad \hat{Z}_k = \frac{\text{vol}_d(D_k)}{n_k} \sum_{i=1}^{n_k} \mathbf{1}_{B_k}(y_i) f(y_i) J_i^{-1}, \quad \text{vol}_d(D_k) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)} r_k^d.$$

The estimators (3.1) and (3.4) are substituted in (1.5) to give our overall estimate of the integral Z .

There is one additional consideration that is important to make this work in practice. That is, the projection from D_k to $M \cap B_k$ must *never fail*. If it does, the estimate (3.4) will not be convergent. Empirically we found the projection always worked when r_k was small enough, which proved to be a more limiting criterion than the criterion that $M \cap B_k$ be a graph above D_k .

3.2 Variance Estimate

This section serves two purposes. One is to present the method we use to estimate the variance σ_Z^2 of the estimator (1.5). This is similar to the strategy of [18], but adapted to this setting. This estimate is accurate only if the MCMC runs are long enough so that Z_k and the R_i are estimated reasonably accurately. Our computational tests below show that the estimator can be useful in practice. The other purpose is to set up notation for some heuristics for choosing parameters, a question dealt with in more depth in Section 5.

We start with the simple estimator \hat{Z}_k in (3.4) for the integral over the smallest ball. It is convenient to write the estimator as the sum of the exact quantity and a statistical error, and then to work with relative error rather than absolute error. We neglect bias, as we believe that error from bias is much smaller than statistical error in all our computations. In the present case, we write

$$(3.5) \quad \hat{Z}_k = Z_k + \sigma_k \zeta_k = Z_k(1 + \rho_k \zeta_k),$$

where ζ_k is a zero mean random variable with $\text{Var}(\zeta_k) = 1$, σ_k is the absolute standard deviation of \hat{Z}_k , and $\rho_k = \sigma_k/Z_k$ is the relative standard deviation. Since the samples in (3.4) are independent, the standard deviation satisfies

$$\sigma_k^2 = \frac{\text{vol}_d^2(D_k)}{n_k} \text{Var}(G(y)), \quad G(y) = \mathbf{1}_{B_k}(y) f(y) J^{-1}(y).$$

We estimate $\text{Var}(G(y))$ with the sample variance and obtain, for the absolute and relative variances,

$$(3.6) \quad \sigma_k^2 \approx \frac{\text{vol}_d^2(D_k)}{n_k^2} \sum_{i=1}^{n_k} (G(y_i) - \bar{G})^2, \quad \rho_k^2 \approx \frac{\sigma_k^2}{\hat{Z}_k^2}.$$

Here $\bar{G} = \sum_{i=1}^{n_k} \frac{1}{n_k} G(y_i)$ is the sample mean.

Now consider the estimator \hat{R}_i in (3.1) for the i^{th} volume ratio. This involves the number count $N_{i,i+1}$. We start by approximating $\text{Var}(N_{i,i+1})$ and then incorporate this into the approximation for $\text{Var}(\hat{R}_i)$. The number count may be written

$$(3.7) \quad N_{i,i+1} = \sum_{j=1}^{n_i} \mathbf{1}_{B_{i+1}}(X_j),$$

where $X_j \in M \cap B_i$ is our sequence of samples. In the invariant probability distribution of the chain, the probability of $X_j \in B_{i+1}$ is

$$(3.8) \quad p_i = \mathbb{E}[\mathbf{1}_{B_{i+1}}(X_j)] = \frac{1}{R_i} = \frac{Z_{i+1}}{Z_i}.$$

Since $\mathbf{1}_{B_{i+1}}(X_j)$ is a Bernoulli random variable, we have (in the invariant distribution)

$$(3.9) \quad \text{Var}(\mathbf{1}_{B_{i+1}}(X_j)) = p_i(1 - p_i).$$

We will call this the static variance. It alone is not enough to determine $\text{Var}(N_{i,i+1})$, since the steps of the Markov chain are correlated. To account for this correlation, we use the standard theory (see, e.g., [33]) of error bars for MCMC estimators of the form (3.7), which we now briefly review.

Consider a general sum over a function of a general MCMC process

$$S = \sum_{j=1}^n F(X_j).$$

The equilibrium lag t autocovariance is (assuming X_j is in the invariant distribution and $t \geq 0$)

$$C_t = \text{cov}[F(X_j), F(X_{j+t})].$$

This is extended to be a symmetric function of t using $C_t = C_{|t|}$. It is estimated from the MCMC run using

$$(3.10) \quad \bar{F} \approx \frac{1}{n_i} \sum_{j=1}^{n_i} F(X_j), \quad C_t \approx \frac{1}{n_i - t} \sum_{j=1}^{n_i-t} (F(X_j) - \bar{F})(F(X_{j+t}) - \bar{F}).$$

The Einstein Kubo sum is

$$D = \sum_{t=-\infty}^{\infty} C_t.$$

The autocorrelation time is

$$(3.11) \quad \tau = \frac{D}{C_0} = 1 + \frac{2}{\text{Var}(F(X_j))} \sum_{t=1}^{\infty} C_t.$$

The sum may be estimated from data using the self-consistent window approximation described in [33]. The static variance C_0 in (3.11) may be estimated using (3.9) rather than (3.10). The point of all this is the large- n variance approximation

$$\text{Var}(S) \approx nD = nC_0\tau.$$

For our particular application, this specializes to

$$(3.12) \quad \text{Var}(N_{i,i+1}) \approx np_i(1 - p_i)\tau_i,$$

where τ_i is the correlation time for the indicator function $\mathbf{1}_{B_{i+1}}(X_j)$, assuming points are generated in $M \cap B_i$. It may be calculated using (3.11).

We use this to write $N_{i,i+1}$ in terms of a random variable ξ_i with unit variance and mean zero as

$$(3.13) \quad N_{i,i+1} \approx n_i p_i \left(1 + \sqrt{\frac{(1-p_i)\tau_i}{n_i p_i}} \xi_i \right).$$

Substituting this approximation into (3.1) gives the approximation

$$(3.14) \quad \hat{R}_i \approx \frac{1}{p_i} \left(1 - \sqrt{\frac{(1-p_i)\tau_i}{n_i p_i}} \xi_i \right),$$

where we expanded the term $1/N_{i,i+1}$ assuming that n_i was large. Continuing, we substitute into (1.5) to get

$$\hat{Z} \approx Z \left[(1 + \rho_k \zeta_k) \prod_{i=0}^{k-1} \left(1 - \sqrt{\frac{(1-p_i)\tau_i}{n_i p_i}} \xi_i \right) \right].$$

If the relative errors are all small, then products of them should be smaller still. This suggests that we get a good approximation of the overall relative error keeping only terms linear in ζ_k and the ξ_i . This gives

$$\hat{Z} \approx Z \left[1 + \rho_k \zeta_k + \sum_{i=0}^{k-1} \sqrt{\frac{(1-p_i)\tau_i}{n_i p_i}} \xi_i \right].$$

Finally, we assume that the random variables ζ_k and ξ_i are all independent. As with the earlier approximations, this becomes valid in the limit of large n_i . In that case, we get

$$(3.15) \quad \text{Var}(\hat{Z}) = \sigma_{\hat{Z}}^2 \approx Z^2 \left(\rho_k^2 + \sum_i \frac{(1-p_i)\tau_i}{n_i p_i} \right).$$

The one standard deviation error bar for \hat{Z} is $Z\sigma_r$, where the relative standard deviation is

$$(3.16) \quad \sigma_r \approx \sqrt{\rho_k^2 + \sum_{i=0}^{k-1} \frac{(1-p_i)\tau_i}{n_i p_i}}.$$

All the quantities on the right in (3.16) are estimated from a single run of the sampler in each $M \cap B_i$, with p_i estimated as $1/\hat{R}_i$ (see (3.1)), τ_i estimated from the sequence $\mathbf{1}_{B_{i+1}}(X_j)$ using a self-consistent window approximation to (3.11), and ρ_k estimated as in (3.6).

3.3 Algorithm Parameters

This section describes how we choose the base point, x_0 , and the sequence of ball radii r_0, r_1, \dots, r_k . A guiding motivation is the desire to minimize the variance of \widehat{Z} , estimated by (3.15), for a given amount of work. Our parameter choices are heuristic and certainly may be improved. But they lead to an overall algorithm that is “correct” in that it gives convergent estimates of the integral Z .

Our aim is to choose r_0 as small as possible, and r_k as large as possible, to have a smaller overall ratio to estimate. Additionally, we expect the estimate of the final integral \widehat{Z}_k to be significantly faster and more accurate than each ratio estimate \widehat{R}_i , since \widehat{Z}_k is estimated by independently chosen points, so we wish r_k to be large. With this aim in mind, the first step is to learn about M through sampling. We generate n points $x_i \in M$ by sampling the surface measure $\rho(x) = \frac{1}{Z} f(x) d\sigma(x)$, using n MCMC steps of the surface sampling algorithm of Section 2. Then we choose x_0 from among these samples. Usually we would like it to be near the center of M , to reduce r_0 or increase r_k . Unless otherwise noted, for our implementation we choose it far from an inequality constraint boundary using

$$(3.17) \quad x_0 = \arg \max_{x_i=x_1, \dots, x_n} \min_j h_j(x_i).$$

The h_j are the functions that determine the boundary of M (cf. (1.6)). This choice may allow the smallest ball B_k to be large, or make it so that most points of B_k project to points that satisfy the inequality constraints. This is heuristic in that there is no quantitative relation between $\min_j h_j(x_i)$ and the distance (euclidean or geodesic or whatever) from x_i to the boundary of M , if M has a boundary.

This is not the only possibility for choosing x_0 . One could also choose it randomly from the set of sampled points. Another possibility would be to center x_0 by minimizing the “radius” of M about x_0 , which is the maximum distance from x_0 to another $x \in M$. We could do this by setting $x_0 = \arg \min_{x_i} \max_j |x_i - x_j|$. Having a smaller radius may mean that we need fewer balls B_j in the integration algorithm.

Once x_0 is fixed, the radius r_0 of the biggest ball B_0 can be found using the sample points:

$$(3.18) \quad r_0 = \max\{|x_0 - x_i| : i = 1, \dots, n\}.$$

Next consider the minimum radius r_k . We need this to be small enough that (a) $B_k \cap M$ is single-valued over T_{x_0} , and (b) the projection from $D_k \rightarrow M \cap B_k$ never fails. We test (b) first because it is the limiting factor. We choose a candidate radius \tilde{r} , sample $n \approx 10^5$ points uniformly from D_k , and project to $M \cap B_k$. If any single point fails, we shrink \tilde{r} by a factor of 2 and test again. When we have a radius \tilde{r} where all of the 10^5 projections succeed, our error in estimating (3.4) should be less than about 0.001%.

Then, we proceed to test (a) approximately as follows. We sample n points x_1, \dots, x_n uniformly from $B_{\tilde{r}}(x_0) \cap M$, where $B_{\tilde{r}}(x_0)$ is the ball of radius \tilde{r}

centered at x_0 . We then consider every pair (x_i, x_j) and check the angle of the vector $v_{ij} = x_i - x_j$ with the orthogonal complement $T_{x_0}^\perp$. Specifically, we construct an orthogonal projection matrix onto the normal space $T_{x_0}^\perp$, for example, as $P = U_{x_0} U_{x_0}^\top$. Then, if there exists a pair for which $|v - Pv| < \text{tol}$, where tol is a tolerance parameter, we shrink \tilde{r} and repeat the test. In practice, we shrink \tilde{r} by a constant factor each time (we used a factor $C = 2$). We expect this test to detect regions where $B_{\tilde{r}}(x_0) \cap M$ is multiple-valued as long as it is sampled densely enough. Of course, if the overlapping regions are small, we may miss them, but then these small regions will not contribute much to the integral Z_k .

An alternative test (not implemented here) would be to sample $B_{\tilde{r}}(x_0) \cap M$ and check the sign of the determinant of the projection to T_{x_0} , namely $\text{sgn}(\det(U_{x_0}^\top U_y))$ where $y \in B_{\tilde{r}}(x_0) \cap M$. If $B_{\tilde{r}}(x_0) \cap M$ is multiple-valued, we expect there to be open sets where this determinant is negative, and therefore we would find these sets by sampling densely enough.

Having chosen x_0 , r_0 , and r_k , it remains to choose the intermediate ball radii r_i . We use a fixed $\nu > 1$ and take the radii so the balls have fixed d -dimensional volume ratios (note d is typically different from the ball's actual dimension d_a), as

$$(3.19) \quad \left(\frac{r_i}{r_{i+1}} \right)^d = \nu.$$

Such fixed ν strategies are often used for volume computation [26]. Given (3.19), the radii are $r_i = r_0 \nu^{-i/d}$. Since we have already chosen the smallest radius, this gives a relationship between ν and k as $\nu = (r_0/r_k)^{d/k}$. Note that we choose r_k , the smallest radius, before we choose k , the number of stages.

It is not clear what would be the best choice for ν . If ν is too large, then the ratios R_i (see (3.1)) will not be estimated accurately—indeed, they will be large so p_i in (3.16) will be small, increasing the relative standard deviation of the estimator. If ν is too small, then k will be large and the errors at each stage will accumulate. In addition, the error at each single stage will be large if we fix the total number of MCMC points, since there will be fewer points per stage. Clearly, these issues are important for the quality of the algorithm in practice, but they are issues that are seriously underexamined in the literature. We return to them in Section 5.

4 Integration: Examples

This section presents some experiments with the integration algorithm of Section 3. There are three sets of computations. We first compute the surface area of the torus \mathbb{T}^2 (Section 4.1). This computation is cheap enough that we can repeat it many times and verify that the error bar formula (3.16) is reasonably accurate. We then calculate the volume of $SO(n)$ (Section 4.2) to verify the algorithm can work on surfaces with high dimension and high codimension. Finally, we apply the algorithm to study clusters of sticky spheres (Section 4.3) and make new predictions that could be tested experimentally.

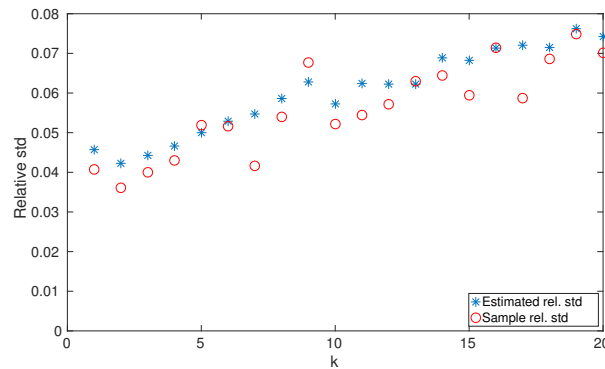


FIGURE 4.1. Relative standard deviations σ_r for the volume estimator on a torus, calculated using a total of 10^5 MCMC points with other parameters given in Section 4.1. The standard deviation was estimated using our formula (3.16) applied to one run (blue stars), and also as the sample standard deviation of 50 independent runs (red circles).

4.1 Surface of a Torus

Consider the torus of Section 2.3 with $R = 1$ and $r = 0.5$. The exact area is $Z = 4\pi^2 r R \approx 19.7392$. The geometry of the manifold is simple enough that we can choose the parameters of the algorithm analytically. We take the initial point to be $x_0 = (R + r, 0, 0)$. The smallest ball B_0 centered at x_0 such that $\mathbb{T}^2 \subset \mathbb{T}^2 \cap B_0$ has radius $r_0 = 2(R + r)$. We take the radius of the smallest ball to be $r_k = r$, for which the projection $T_{x_0} \cap B_k \rightarrow \mathbb{T}^2 \cap B_k$ is clearly single-valued. For each computation we generate n points at each stage, with the total number of points fixed to be $n_t = kn = 10^5$.

We calculated the volume for each number of stages $k = 1, \dots, 20$. We repeated each calculation 50 times independently. This allowed us to get a direct estimate of the variance of the volume estimator \hat{Z} . Figure 4.1 shows that the standard deviation estimated from a single run using (3.16) is in reasonable agreement with the direct estimate from 50 independent runs.³ It also shows that for this problem $k = 2$ seems to be optimal. It does not help to use a lot of stages, but it is better to use more than one.

4.2 The Special Orthogonal Group

The volume of $SO(n)$ as described in Section 2.3 is (see, e.g., [37])

$$\text{vol}(SO(n)) = 2^{\frac{n(n-1)}{4}} \prod_{i=1}^{n-1} \text{vol}(S^i) \quad \text{where} \quad \text{vol}(S^i) = \frac{2\pi^{\frac{i+1}{2}}}{\Gamma(\frac{i+1}{2})}$$

³ We do not strive for very precise agreement. An early mentor of one of us (JG), Malvin Kalos, has the wise advice: “Don’t put error bars on error bars.”

n	d	V_t	\bar{V}	$\widetilde{s_V}$	ϵ_r
2	1	8.89	8.9	0.0038	0.0047
3	3	223.3	223	0.0078	0.0024
4	6	$1.24 \cdot 10^4$	$1.2 \cdot 10^4$	0.016	0.012
5	10	$1.31 \cdot 10^6$	$1.3 \cdot 10^6$	0.022	0.024
6	15	$2.30 \cdot 10^8$	$2.2 \cdot 10^8$	0.0354	0.0519
7	21	$6.09 \cdot 10^{10}$	$6.0 \cdot 10^{10}$	0.0547	0.0094

TABLE 4.1. Numerical vs. analytical results for the volume of the special orthogonal group $SO(n)$ for $2 \leq n \leq 7$. Here d is the dimension of $SO(n)$, \bar{V} is the sample mean obtained from a sample of size 50, and the relative standard deviation $\widetilde{s_V}$ and relative error ϵ_r are given in (4.1).

is the volume of the i^{th} -dimensional sphere. This formula may be understood as follows. An element of $SO(n)$ is a positively oriented orthonormal basis of \mathbb{R}^n . The first basis element may be any element of the unit sphere in \mathbb{R}^n , which is S^{n-1} . The next basis element may be any element orthogonal to the first basis element, so it is in S^{n-2} , and so on. The prefactor $2^{n(n-1)/4}$ arises from the embedding of $SO(n)$ into $\mathbb{R}^{n \times n}$. For example, $SO(2)$ is a circle that would seem to have length 2π , but in \mathbb{R}^4 it is the set $(\cos(\theta), \sin(\theta), -\sin(\theta), \cos(\theta))$, $0 \leq \theta < 2\pi$. This curve has length $2\sqrt{2}\pi$. The difference is the ratio $2^{2 \times 1/4} = \sqrt{2}$.

We applied the integration algorithm to compute the volume of $SO(n)$ for $2 \leq n \leq 7$. For each n , we ran the algorithm $N = 50$ independent times and computed the mean of the estimates \bar{V} and the sample standard deviation s_V . We used a total of $n_t = 10^5$ points and $k = 4$ steps for each n . Since the true value V_t is known, we can compute the relative standard deviation $\widetilde{s_V}$ and the relative error ϵ_r via

$$(4.1) \quad \widetilde{s_V} = \frac{s_V}{\sqrt{N}V_t}, \quad \epsilon_r = \left| \frac{\bar{V} - V_t}{V_t} \right|.$$

We give the results in Table 4.1, which shows our relative errors are small, and close to the estimated standard deviations. We notice that the errors increase as the dimension gets higher.

4.3 Sticky-Sphere Clusters

Next we consider a system of particles interacting with a so-called sticky potential, which can be thought of as a delta-function when the surfaces of two particles are in contact. This is a model for particles whose range of interaction is very short compared to the particles' diameters, as is the case for many kinds of colloidal particles. The probabilities of finding the system in different states in equilibrium can be calculated using statistical mechanics [16], and we briefly summarize the relevant ideas.

Consider a system of N unit spheres in \mathbb{R}^3 , that is represented as a vector $x = (x_1, \dots, x_N)^T \in \mathbb{R}^{3N}$, where $x_i \in \mathbb{R}^3$ is the center of the i^{th} sphere. We suppose

there are m pairs of spheres in contact $E = \{(i_1, j_1), \dots, (i_m, j_m)\}$. For each pair in contact there is a constraint

$$(4.2) \quad |x_i - x_j|^2 - 1 = 0, \quad (i, j) \in E,$$

and for each pair not in contact there is an inequality that says the spheres cannot overlap:

$$(4.3) \quad |x_i - x_j| > 1, \quad (i, j) \notin E.$$

To remove the translational degrees of freedom of the system, we fix the center of mass of the cluster at the origin. We do this by imposing three extra constraints, which together are written in vector form as

$$(4.4) \quad \sum_{i=1}^N x_i = 0.$$

We define M to be the set of points x satisfying the constraints (4.2) and (4.4), and the inequalities (4.3). It is possible that M has singular points where the equality constraints (4.2) are degenerate (have gradients that are not linearly independent), but we ignore this possibility (note that the 1 in Section 2.3 shows our algorithm may work even near singular points.) Therefore M is a manifold of dimension $d = 3N - m - 3$ embedded in an ambient space of dimension $d_a = 3N$, with surface area element $d\sigma$. If M is not connected, our algorithm samples a single connected component.

The equilibrium probability to find the spheres in a particular cluster defined by contacts E is proportional to the partition function Z . In the sticky-sphere limit, and where the cluster is in a container much larger than the cluster itself, the partition function is calculated (up to constants that are the same for all clusters with the same number of spheres) as [16, 17]

$$(4.5) \quad Z = \kappa^m z \quad \text{where } z = \int_M f(x) d\sigma(x).$$

Here κ is the “sticky parameter,” a single number depending on both the strength of the interaction between the spheres and the temperature. This parameter is something that must be known or measured for a given system. The factor z is called the “geometrical partition function” because it does not depend on the nature of the interaction between the spheres nor the temperature. The function $f(x)$ is given by

$$(4.6) \quad f(x) = \prod_{i=1}^{3N-m-3} \lambda_i(x)^{-1/2},$$

which is the product of the nonzero eigenvalues λ_i of $R^T R$, where R is half the Jacobian of the set of contact constraints (4.2). In the mathematical theory of rigidity, the matrix R is called the rigidity matrix [6].

Manifold	z	z_t
8	6.99 ± 0.12	7.014
9	44.4 ± 0.33	44.402
10	57.1 ± 0.26	57.07
11	18.9 ± 0.20	19.18
12	3.30 ± 0.020	3.299
13	62.4 ± 0.23	62.34
14	3.35 ± 0.039	3.299
15	27.7 ± 0.14	27.78
16	11.7 ± 0.058	11.67
17	28.5 ± 0.31	28.41
18	8.44 ± 0.088	8.238
19	69.2 ± 0.33	69.43
20	50.9 ± 0.27	50.86

TABLE 4.2. Geometrical partition functions z of the configuration spaces of clusters of $N = 6$ unit spheres with $m = 10$ contacts (the numbering is as in [17], where the manifolds are called “modes.”) Each computation is done using a single MCMC run of $n_t = 10^8$ points. The error bars (such as ± 0.12 for manifold 8) are one standard deviation, estimated using (3.16). We compare the calculations with the value z_t obtained in [17].

As a first check that our algorithm is working, we calculate geometrical partition functions z that are already known by considering the 13 different clusters of 6 spheres with $m = 10$ contacts. All manifolds formed in this way are five-dimensional. For each manifold we compute z as in (4.5) using $n_t = 10^8$ points and $k = 4$ steps. For each estimate we compute its error bar using formula (3.16). We compare these values with the values z_t calculated in [17], which were obtained by parametrizing and triangulating each manifold and calculating the integrals using finite elements.⁴ Table 4.2 shows that the two calculations agree.

Next, we consider a cluster of N identical spheres that may form either a chain (C) or a loop (L), as in Figure 4.2. We are interested in the conditions under which chains are preferable to loops. Let $M_{C,N}$ and $M_{L,N}$ denote the manifolds corresponding to a chain and loop, respectively; these have dimensions $2N - 2$ and

⁴In [17], the calculations were done on the quotient manifold obtained by modding out the rotational degrees of freedom. We originally tried working in a quotient space here by fixing six coordinates; this worked for these low-dimensional manifolds but we had problems with higher-dimensional ones, particularly chains. We presume this was because of the singularity in the quotient space. To be consistent, all calculations in this paper are performed on the original manifold M , which contains rotational degrees of freedom. Therefore, we compare our results to the value $z_t = 8\pi^2 \tilde{z}$, where \tilde{z} is the value reported in [17]. The constant $8\pi^2$ is the volume of the rotational space $SO(3)$.

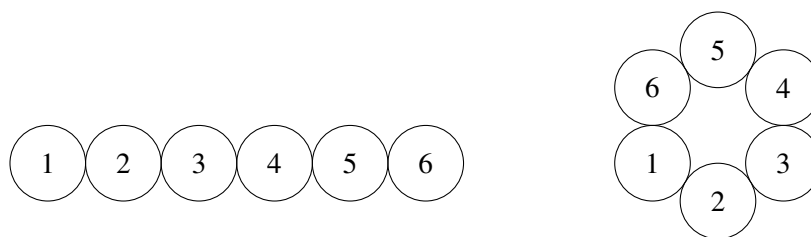


FIGURE 4.2. Two-dimensional view of two clusters of 6 spheres: a chain (left) and a loop (right).

N	z_C/z_L	$(n_L z_L)/(n_C z_C)$
4	6.02 ± 0.032	24.1 ± 0.13
5	13.5 ± 0.13	67.5 ± 0.65
6	22.9 ± 0.41	138 ± 2.5
7	35.1 ± 1.0	246 ± 7.0
8	49.0 ± 2.6	392 ± 21
9	60.3 ± 5.0	543 ± 45
10	83.6 ± 9.5	836 ± 95

TABLE 4.3. Ratios between the geometrical partition functions of a chain and a loop of N particles, for $4 \leq N \leq 10$. We report both the ratio z_C/z_L of a single chain vs. loop, and the ratio $(n_C z_C)/(n_L z_L)$ when spheres are treated as indistinguishable. The latter may be used to estimate the sticky parameter κ as in (4.8).

$2N - 3$. For example, for $N = 10$ spheres the dimensions are 18 and 17. Integrals over these manifolds are not easily calculated by deterministic parametrizations.

For each $N = 4 - 10$ and for each $i \in \{L, C\}$, we compute the volume V and the geometrical partition function z of the manifold $M_{i,N}$, using $n_t = 10^8$ points and $k = 4$ steps each. Because we don't have other results to compare to, we check that our calculations are correct by comparing our calculated value of the average of f , namely $\bar{h} = z/V$, to that estimated by our sampling algorithm, \tilde{h} . The values of \tilde{h} and \bar{h} agree within statistical error as shown in Table 4.4.

With the values of z_C and z_L we can make experimental predictions. Indeed, the ratio of the probabilities of finding a system of N sticky spheres in a chain versus a loop in equilibrium is

$$(4.7) \quad \frac{P(\text{chain})}{P(\text{loop})} = \kappa^{-1} \frac{n_C z_C}{n_L z_L},$$

where n_C and n_L are the number of distinct copies of each manifold that one obtains by considering all permutations of spheres. If the spheres are indistinguishable, then one can calculate that $n_C = N!/2$ and $n_L = (N-1)!/2$.⁵ From the calculated ratios $(n_C z_C)/(n_L z_L)$, shown in Table 4.3, and the value of the sticky parameter, which must be measured or inferred in some other way, we obtain a prediction for the fraction of chains versus loops, which could be compared to experimental data. Of course, the spheres could assemble into other configurations, whose entropies we could in principle calculate, but (4.7) still holds without calculating these additional entropies because the normalization factors cancel.

If we don't know the sticky parameter κ for the system, then (4.7) can actually be used to *measure* it for a given system. Indeed, an estimate for κ using a fixed value of N would be

$$(4.8) \quad \kappa \approx \hat{\kappa} = \frac{\# \text{ of loops of length } N}{\# \text{ of chains of length } N} \cdot \frac{n_C z_C}{n_L z_L}.$$

Such a measurement could be useful, for example, in systems of emulsion droplets whose surfaces have been coated by strands of sticky DNA [10]. The DNA attaches the droplets together, like Velcro®, but the sticky patch can move around on the surface of a droplet, so a chain of attached droplets can sample many configurations. Understanding the strength of the droplet interaction from first principles, or even measuring it directly, is a challenge because it arises from the microscopic interactions of many weak DNA bonds as well as small elastic deformation of the droplet. But (4.7) could be used to measure the stickiness simply by counting clusters in a microscope.

Finally, we note that the ratio $(n_C z_C)/(n_L z_L)$ is related to the loss of entropy as a chain forms a bond to become a loop. Even though a loop has more bonds, hence lower energy, it still might be less likely to be observed in equilibrium because it has lower entropy. The ratio $(n_C z_C)/(n_L z_L)$ is exactly the value of stickiness κ above which a loop becomes more likely than a chain, which corresponds to a threshold temperature or binding energy. Note that as N increases the ratio does too, so we expect fewer loops for longer polymers than for shorter ones.

⁵ The number of distinct copies is $N!/o$, where o is the symmetry number of the clusters, i.e., the number of permutations that are equivalent to an overall rotation. For the chain, the symmetry group is C_2 , the cyclic group of order 2, and therefore the symmetry number is 2. For the loop, the symmetry group is C_N , the rotational group of a regular N -gon of order N , so the symmetry number is $2N$. It is also possible that the spheres are distinguishable, for example, if the bonds forming the chain are unbreakable and only one bond can break and reform, in which case we would have $n_C = n_L = 1$.

Note that for three-dimensional clusters, the number of distinct clusters of N spheres should usually be computed as $2N!/o$, where o is the number of permutations that are equivalent to a rotation of the cluster (e.g., [35].) In our case, however, the clusters can be embedded into a two-dimensional plane, so a reflection of the cluster is equivalent to a rotation, hence we do not need to include the factor 2.

	V	z	\bar{h}	\tilde{h}
4 Chain	$3.51 \cdot 10^2 \pm 1.03$	$1.32 \cdot 10^2 \pm 0.38$	0.374 ± 0.0022	$0.379 \pm 1.5 \cdot 10^{-5}$
4 Loop	$8.27 \cdot 10^1 \pm 0.20$	$2.18 \cdot 10^1 \pm 0.054$	0.264 ± 0.0013	$0.261 \pm 7.6 \cdot 10^{-5}$
5 Chain	$3.03 \cdot 10^3 \pm 16$	$8.31 \cdot 10^2 \pm 4.3$	0.274 ± 0.0029	$0.279 \pm 1.1 \cdot 10^{-4}$
5 Loop	$3.19 \cdot 10^2 \pm 4.3$	$6.19 \cdot 10^1 \pm 0.85$	0.194 ± 0.0053	$0.194 \pm 2.8 \cdot 10^{-4}$
6 Chain	$2.70 \cdot 10^4 \pm 2.7 \cdot 10^2$	$5.33 \cdot 10^3 \pm 52$	0.197 ± 0.0039	$0.206 \pm 1.9 \cdot 10^{-4}$
6 Loop	$1.66 \cdot 10^3 \pm 13$	$2.32 \cdot 10^2 \pm 1.9$	0.143 ± 0.0024	$0.147 \pm 3.1 \cdot 10^{-5}$
7 Chain	$2.42 \cdot 10^5 \pm 1.3 \cdot 10^4$	$3.84 \cdot 10^4 \pm 1.9 \cdot 10^3$	0.158 ± 0.0017	$0.151 \pm 1.7 \cdot 10^{-4}$
7 Loop	$1.01 \cdot 10^4 \pm 1.2 \cdot 10^2$	$1.09 \cdot 10^3 \pm 13$	0.109 ± 0.0026	$0.110 \pm 2.6 \cdot 10^{-5}$
8 Chain	$2.19 \cdot 10^6 \pm 6.4 \cdot 10^4$	$2.77 \cdot 10^5 \pm 8.3 \cdot 10^3$	0.127 ± 0.0075	$0.111 \pm 9.6 \cdot 10^{-4}$
8 Loop	$6.76 \cdot 10^4 \pm 1.6 \cdot 10^3$	$5.65 \cdot 10^3 \pm 1.3 \cdot 10^2$	0.0836 ± 0.0039	$0.0824 \pm 2.5 \cdot 10^{-5}$
9 Chain	$2.12 \cdot 10^7 \pm 1.1 \cdot 10^6$	$1.73 \cdot 10^6 \pm 8.9 \cdot 10^4$	0.0816 ± 0.0084	$0.0820 \pm 1.1 \cdot 10^{-5}$
9 Loop	$4.94 \cdot 10^5 \pm 1.7 \cdot 10^4$	$2.87 \cdot 10^4 \pm 9.2 \cdot 10^2$	0.0582 ± 0.0038	$0.0612 \pm 1.2 \cdot 10^{-5}$
10 Chain	$2.07 \cdot 10^8 \pm 1.2 \cdot 10^7$	$1.43 \cdot 10^7 \pm 8.1 \cdot 10^5$	0.0689 ± 0.0080	$0.0604 \pm 3.5 \cdot 10^{-4}$
10 Loop	$3.81 \cdot 10^6 \pm 2.2 \cdot 10^5$	$1.71 \cdot 10^5 \pm 9.7 \cdot 10^3$	0.0448 ± 0.0051	$0.0453 \pm 4.7 \cdot 10^{-5}$

TABLE 4.4. Volume V , geometrical partition function z , and mean vibrational contribution $\bar{h} = z/V$ for the manifolds $M_{i,N}$ corresponding to the loop/chains of N particles. The computations were done using $n_t = 10^8$ points. Error bars are calculated using (3.16). The results are compared with the average vibrational contribution \tilde{h} estimated from our sampling algorithm. Specifically, we generate n_p points on $M_{i,N}$ (and repeat this $n_m = 10$ times) and estimate the average as $\tilde{h} = \frac{1}{n_m} \sum_{j=1}^{n_m} \tilde{h}_j$ where $\tilde{h}_j = \frac{1}{n_p} \sum_{i=1}^{n_p} f(X_i)$. The error bars of the latter are computed using the sample standard deviation using a sample of size $n_m = 10$.

5 Minimizing the Error Bars: Some Toy Models to Understand Parameter Dependence

Equation (3.16) estimates the relative standard deviation of the estimator \hat{Z} for Z . We would like to choose parameters in the algorithm to minimize this relative error. Previous groups studying similar algorithms in euclidean spaces have investigated how the overall number of operations scales with the dimension [26], but to our knowledge there is little analysis of how such an algorithm behaves for fixed dimension, and how to optimize its performance with fixed computing power.

To this aim we construct a hierarchy of toy models that suggest how the error formula may depend on parameters. We focus exclusively on the sum

$$\sum_i (1 - p_i) \tau_i / (n_i p_i)$$

at the right of (3.16), which estimates the relative variance of a product of ratio estimates. This is because the estimator \hat{Z}_k for ρ_k usually has a much smaller error, since it is estimated by straightforward Monte Carlo integration with independently chosen points. Our models may apply to other multiphase MCMC algorithms.

We consider an algorithm that chooses the same number of points n at each stage and has a fixed number of total points $n_t = kn$, where k is the number of stages.

Additionally, we fix the largest and smallest radii r_0 and r_k , respectively, and take the d -dimensional volume ratios between balls to be the same constant $v_i = v$ at each stage. The ratio v and number of stages k are related by

$$(5.1) \quad v^k = \left(\frac{r_0}{r_k} \right)^d = C,$$

where we will write C to denote a constant (not always the same) that does not depend on v or k .

We now make the approximation that $R_i \propto v$. This would be exactly true if M were a hyperplane of dimension d . The relative variance we wish to minimize is

$$(5.2) \quad g(v) = C \frac{v-1}{\log(v)} \sum_{i=0}^{k-1} \tau_i.$$

Recall that τ_i is the correlation time for the indicator function $\mathbf{1}_{B_{i+1}}(X_j)$, assuming points are generated in $M \cap B_i$ (see (3.11) and (3.12)).

We now consider several models for the correlation times τ_i that will allow us to investigate the behavior of $g(v)$. Ultimately, we would like to find the value v^* that minimizes this function.

5.1 Constant τ

Suppose the $\tau_i = \tau$ are equal and independent of k . This would happen, for example, if we were able to make independent samples and achieve $\tau_i = 1$. It also could happen if there were a scale-invariant MCMC algorithm such as the hit-and-run method or affine invariant samplers. It seems unlikely that such samplers exist for sampling embedded manifolds. In that case, we have

$$(5.3) \quad g(v) = C \frac{v-1}{\log(v)^2}.$$

Optimizing this over v gives $v^* \approx 4.9$ independently of dimension. This result is interesting in that it suggests a more aggressive strategy than the previously suggested choice $v = 2$ [26]; i.e., one should shrink the balls by a larger amount and hence have a smaller number of stages.

5.2 Diffusive Scaling

As a second scenario, we model τ_i as

$$(5.4) \quad \tau_i \propto r_i^2.$$

This is a “diffusive scaling” because it represents the time scale of a simple diffusion or random walk in a ball of radius r_i (see (5.18) in Section 5.3 below.) From (3.19), we have $\tau_i \propto v^{-2i/d}$. Hence (5.2) becomes, after some algebra and using (5.1),

$$(5.5) \quad g(v) \propto g_d(v) \equiv \frac{v-1}{\log(v)(1-v^{-2/d})}.$$

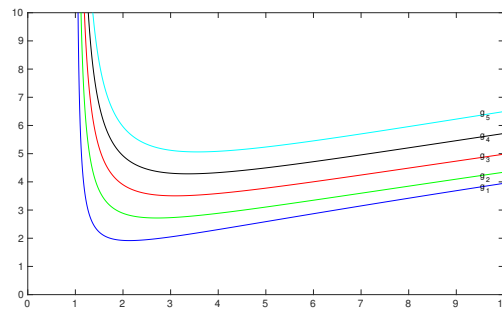


FIGURE 5.1. Plots of the functions $g_d(v)$ as in (5.5) for $d = 1, 2, 3, 4, 5$.

Plots of $g_d(v)$ are given in Figure 5.1. Examples of minimizers v_d^* are $\{2.6, 2.7, 3.1, 3.4, 3.6, 3.7, 4.1, 4.5, 4.7\}$ for $d = \{1, 2, 3, 4, 5, 6, 10, 20, 50\}$, respectively. Some analysis shows that $v_d^* \rightarrow v^*$ as $d \rightarrow \infty$, where v^* is the optimal ratio assuming constant τ above. The same analysis shows that $g_d(v_d^*) \approx \frac{d(v^*-1)}{2 \log^2(v^*)}$ for large d , so the relative variance in this model is proportional to the manifold's intrinsic dimension d . The figure shows that $g_d(v)$ increases more slowly for $v > v^*$ than for $v < v^*$. This also suggests taking larger values of v in computations.

We found it surprising that the model with diffusive scaling predicted smaller ratios than the one with constant scaling, since it has a correlation time that decreases as the balls shrink so one might prefer to shrink them rapidly. Some insight into why comes from thinking about the *marginal* cost of adding an additional ball, given we already have k' balls. For the first model, this marginal cost is constant no matter what k' is, and equal to the cost of just the first ball. For the second model, the marginal cost can decrease as k' increases, since the contribution to the cost from the j^{th} ball is a decreasing function of j . So, with diffusive scaling, each additional ball does not increase the error by as much. Another way of seeing this is to plot the correlation times as a function of ball number, and note that the first and last balls have fixed correlation times. The total error is roughly the area under this plot. But the area under an exponential function connecting $(0, Cr_0^d)$ to (k', Cr_k^d) increases less when $k' \rightarrow k' + 1$ than the area under a constant function connecting $(0, C')$ to (k', C') (here C, C' are constants.)

5.3 A Brownian Motion Model in Balls

As a third model, we approximate the MCMC sequence X_0, X_1, \dots with a continuous-time Brownian motion and simplify the geometry to two concentric balls in a d -dimensional plane. This allows us to find an analytic formula for τ_i . While this is a natural model to consider, we will describe later why it fails to capture the behavior of almost any system of interest. Nevertheless, the reasons

behind its failure help build intuition into high-dimensional geometry, so we feel it useful to describe this mathematical exercise.

What follows is an exact derivation of the correlation times in this model. Readers interested in only the result may skip to equation (5.21) and the subsequent discussion.

Let X_t be a d -dimensional Brownian motion on a d -dimensional ball B_0 of radius R_0 centered at the origin, with reflecting boundary conditions at $|x| = R_0$. We consider the stochastic process

$$(5.6) \quad F(X_t) = \mathbf{1}_{B_1}(X_t),$$

where B_1 is a ball with radius $R_1 < R_0$, also centered at the origin. Our goal is to derive an analytic formula for the correlation time of the process, defined for a continuous-time process to be (compare with (3.11)) [33]

$$(5.7) \quad \tau = 2 \int_0^{+\infty} \frac{C(t)}{C(0)} dt.$$

This is expressed in terms of the stationary covariance function of $F(X_t)$:

$$(5.8) \quad C(t) = \mathbb{E}[(\mathbf{1}_{B_1}(X_0) - \bar{F})(\mathbf{1}_{B_1}(X_t) - \bar{F})], \quad \bar{F} = \mathbb{E}[\mathbf{1}_{B_1}(X)].$$

The expectation is calculated assuming that X_0 is drawn from the stationary distribution, which for a Brownian motion is the uniform distribution.

We know that

$$(5.9) \quad C(0) = \text{Var}(F(X_t)) = \bar{F}(1 - \bar{F})$$

(as for a Bernoulli random variable), so we focus on computing the integral in (5.7). Notice that $\bar{F} = (R_1/R_0)^d$.

We start by writing (5.8) in terms of the probability density $p(t, x, y)$ for $X_t = x$ given $X_0 = y$. If X_0 is in the stationary distribution, then the joint density of $X_0 = y$ and $X_t = x$ is $Z_0^{-1} p(t, x, y)$, where $Z_0 = \text{vol}(B_0)$. Using this expression to write (5.8) as an integral over $dx dy$, integrating over time, and rearranging the integrals gives

$$(5.10) \quad \int_0^\infty C(t) dt = Z_0^{-1} \int_{B_0} \int_{B_0} (\mathbf{1}_{B_1}(x) - \bar{F})(\mathbf{1}_{B_1}(y) - \bar{F}) \cdot \int_0^\infty p(t, x, y) dt dy dx.$$

To calculate the time integral we must evaluate $\bar{p}(x, y) = \int_0^{+\infty} p(t, x, y) dt$. Notice that $p(t, x, y)$ satisfies the forward Kolmogorov equation with boundary and initial conditions (see, e.g., [13]):

$$(5.11) \quad \begin{aligned} p_t &= \frac{1}{2} \Delta_x p \quad \text{for } x \in B_0, & \nabla p \cdot \hat{n} &= 0 \quad \text{for } x \in \partial B_0, \\ p(0, x, y) &= \delta(x - y), \end{aligned}$$

where \hat{n} is the vector normal to the boundary. Therefore $\bar{p}(x, y)$ satisfies the equation

$$\Delta_x \bar{p}(x, y) = \int_0^{+\infty} \Delta_x p(t, x, y) dt = 2 \int_0^{+\infty} p_t(t, x, y) dt = -2\delta(x - y).$$

Therefore $\bar{p}(x, y)$ is proportional to the Green's function for the Laplace equation in a ball.

We now wish to evaluate the integral over the y -coordinate,

$$u(x) = \int_{B_0} f(y) \bar{p}(x, y) dy,$$

where we have defined $f(x) = \mathbf{1}_{B_1}(x) - \bar{F}$ to be the centered function whose covariance we wish to evaluate. One can check that $u(x)$ solves the Poisson equation

$$(5.12) \quad \Delta_x u = -2f \quad \text{for } x \in B_0, \quad \nabla u \cdot \hat{n} = 0 \quad \text{for } x \in \partial B_0.$$

We now solve this equation analytically for $u(x)$. For $d = 1$, (5.12) reduces to an ODE, whose solution is

$$(5.13) \quad u(x) = \begin{cases} \left(\frac{R_1}{R_0} - 1\right)x^2, & |x| < R_1, \\ \frac{R_1}{R_0}x^2 - 2R_1|x| + R_1^2, & R_1 < |x| < R_0. \end{cases}$$

For $d \geq 2$, we solve (5.12) by looking at radial solutions $u = u(|x|) = u(r)$. In polar coordinates (5.12) is

$$(5.14) \quad u_{rr} + \frac{d-1}{r}u_r = -2f \quad \text{for } r \in (0, R_0), \quad u_r(R_0) = 0.$$

We solve this equation using the elementary radial solutions $\Phi(r)$ of the Laplace equation ($\Delta u = 0$) in the disk,⁶ and requiring the solution to satisfy $\lim_{r \rightarrow 0} u(r) = 0$ and to be continuous at $r = R_1$. For $d = 2$ we have

$$(5.15) \quad u(r) = \begin{cases} \frac{R_1^2 - R_0^2}{R_0^2} r^2, & 0 \leq r \leq R_1, \\ R_1^2(\log R_1 - \log r) + \frac{R_1^2}{2} \left(\frac{r^2}{R_0^2} - 1 \right), & R_1 < r \leq R_0, \end{cases}$$

⁶These are given by

$$\Phi(r) = \begin{cases} C \log(r) + K, & d = 2, \\ \frac{C}{2-d} r^{2-d} + K, & d \geq 3, \end{cases}$$

where C and K are constants [9].

and for $d \geq 3$ we have

$$(5.16) \quad u(r) = \begin{cases} \frac{R_1^d - R_0^d}{dR_0^d} r^2, & 0 \leq r \leq R_1, \\ -\frac{2R_1^d}{d(2-d)} r^{2-d} + \frac{R_1^d}{dR_0^d} r^2 + \frac{R_1^2}{2-d}, & R_1 \leq r \leq R_0. \end{cases}$$

We can now evaluate the integral over the x -coordinate to obtain

$$(5.17) \quad \begin{aligned} \int_0^{+\infty} C(t) dt &= Z_0^{-1} \int_{B_0} f(x) u(x) dx \\ &= \text{vol}(S_{d-1}) Z_0^{-1} \int_0^{R_0} f(r) u(r) r^{d-1} dr, \end{aligned}$$

where in the last step we have written the integral using polar coordinates. Here $\text{vol}(S_{d-1})$ is the surface area of the boundary of the unit d -dimensional sphere. Substituting our analytic expression for $u(x)$, and using (5.9) in (5.7), we obtain

$$(5.18) \quad \tau = R_0^2 h_d(\nu).$$

Here $\nu = (R_0/R_1)^d$ is the volume ratio as in (3.19), and the functions $h_d(\nu)$ are given by

$$(5.19) \quad h_d(\nu) = \begin{cases} \frac{4}{3} \frac{\nu - 1}{\nu^2}, & d = 1, \\ \frac{\nu^{-1} - 1 + \log(\nu)}{\nu - 1}, & d = 2, \\ \frac{4}{d^2 - 4} \frac{(d-2)\nu^{-1} - d\nu^{2/d-1} + 2}{\nu^{2/d}(1 - \nu^{-1})}, & d \geq 3. \end{cases}$$

The functions $h_d(\nu)$ are proportional to the correlation time and are plotted in Figure 5.2 for $d = 1, 2, 3, 4, 5$. For fixed ν and $d \geq 3$ the correlation time decreases with the dimension, and for all d we have

$$\lim_{\nu \rightarrow 1} h_d(\nu) = \lim_{\nu \rightarrow +\infty} h_d(\nu) = 0.$$

We found all of these observations to be surprising and will discuss them further in Section 5.4.

We use (5.18) to approximate the correlation time τ_i in (5.2) as

$$(5.20) \quad \tau_i = r_i^2 h_d(\nu).$$

After some algebra, we obtain

$$(5.21) \quad g(\nu) \propto l_d(\nu) \equiv \frac{(\nu - 1)h_d(\nu)}{\log(\nu)(1 - \nu^{-2/d})}.$$

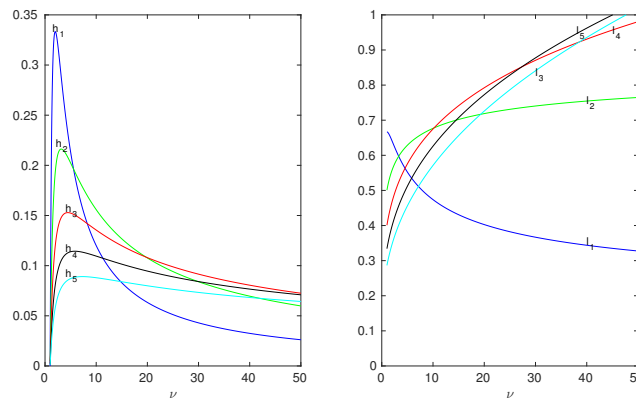


FIGURE 5.2. Left: plots of $h_d(\nu)$ from (5.19), which are proportional to the correlation times of the process $\mathbf{1}_{B_1}(X_t)$ in (5.6). Right: plots of $l_d(\nu)$ in (5.21), which are proportional to the relative variance of the product of ratios. Both functions are analytically computed for a Brownian motion X_t in two concentric spheres for dimensions $d = 1, 2, 3, 4, 5$.

One can verify from (5.5) and (5.21) that $l_d(\nu) = g_d(\nu)h_d(\nu)$, so this model's only difference from that of Section 5.2 is that it additionally models how τ_i depends on ν .

Plots of the function $l_d(\nu)$ for dimensions up to $d = 5$ are given in Figure 5.2. The function $l_1(\nu)$ is strictly decreasing, while $l_d(\nu)$ is strictly increasing for $d \geq 2$. It follows that the ratio ν^* that minimizes (5.21) is $\nu^* = 1$ for $d \geq 2$.

5.4 Comparing the Toy Models with Data

Our first two models for the correlation time predicted an error that was minimized at an intermediate value ν^* , equal to about $\nu^* \approx 5$ for the model with constant τ , and slightly smaller for the model with diffusive scaling though approaching it as $d \rightarrow \infty$. The third was totally different—it predicted taking as small a ν as possible. What does the empirical evidence suggest?

Figure 5.3 plots the relative standard deviation of \widehat{Z} , obtained empirically from averaging over several simulations, as a function of $\log(\nu)$ for several different manifolds of varying dimensions and codimensions. For this figure we calculated the volume using $k = 1, \dots, 40$ and computed ν using (5.1), assuming fixed r_0 and r_k . All curves have the same basic shape, which is qualitatively the same as that predicted by the first two models but not the third.

What went wrong with the third model? The only difference it has with the diffusive scaling model is an additional assumption for how the correlation time depends on ν . This correlation time was small for large ν , something to be expected since a Brownian motion should leave B_1 very quickly when B_1 is much

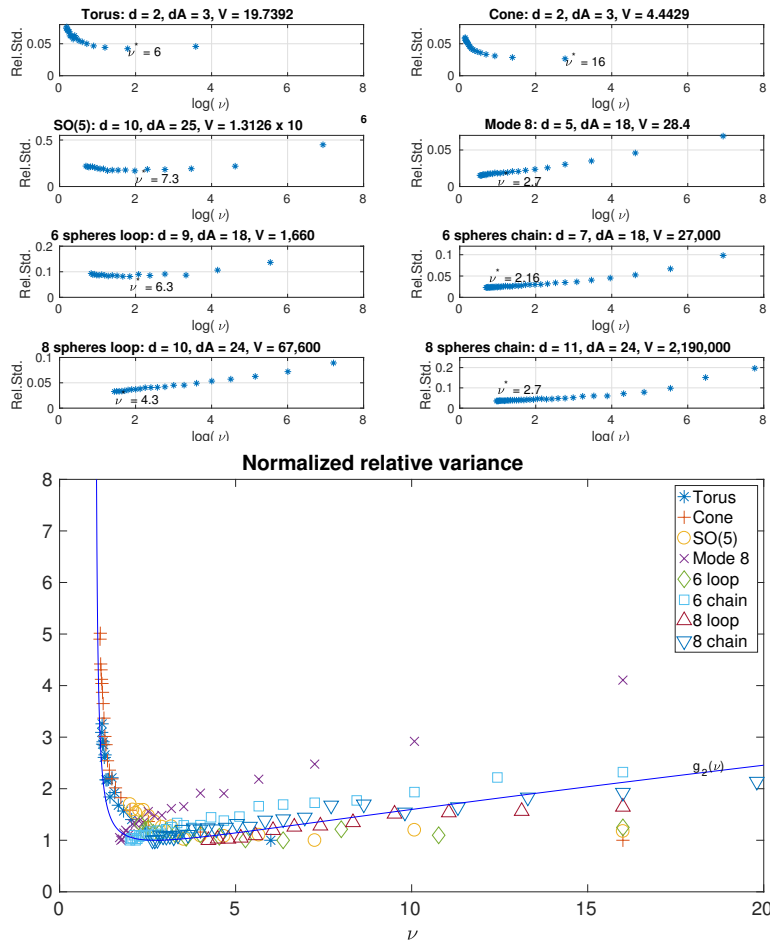


FIGURE 5.3. Top: plots of the relative standard deviation σ_r of \widehat{Z} as a function of the ball ratio ν for eight different manifolds. These are each obtained using formula (3.16) from a single run of $n_t = 10^6$ points. The minimizers ν^* are reported in the plots and plotted as smaller black dots. Bottom: plots of the relative variance σ_r^2 of \widehat{Z} of the same manifolds, combined together and normalized so that they all have the minimum in $\nu = 1$, superimposed on the theoretical model $g_2(\nu)$ given in (5.5).

smaller than B_0 . But, importantly, the correlation time was also small for small ν ; small enough to make the overall relative variance decrease monotonically as ν decreases. The fact that the correlation time approaches 0 as $\nu \rightarrow 1$ implies that a Brownian motion leaves B_1 very quickly even when B_1 is nearly as big as B_0 .

We believe this is an artifact of our highly restrictive geometry, and almost any problem of interest will not have this property. In high dimensions, virtually all of

the volume of an object will be concentrated near the extremal parts of its boundary, a fact that can be checked for example by calculating the volume of a spherical shell of width $\epsilon \ll 1$. A sphere's boundary therefore is concentrated near the outermost spherical shell, so when a Brownian motion enters B_1 it will with overwhelming probability stay near the boundary, where it is easy for it to leave B_1 . Therefore, we expect the Brownian motion to quickly enter and leave B_1 repeatedly, like a Brownian motion jumping back and forth between the boundaries of a thin strip. The correlation time therefore should be short.

When $B_1 \cap M$, $B_0 \cap M$ are not spherical, then some parts of their boundaries will lie further from the origin than others, and it is in these “corners” that we expect a Brownian motion to get stuck for longer times. In convex geometry such corners are known to control the computational complexity of Monte-Carlo algorithms [32], an effect that is simply not possible to capture with Brownian motion in spheres. We expect that inhomogeneous boundary densities are the norm rather than the exception in any problem of scientific interest, and therefore do not believe the predictions of this third model.

6 Conclusions

We introduced algorithms to sample and integrate over a connected component of a manifold M , defined by equality and inequality constraints in euclidean space. Our sampler is a random walk on M that does not require an explicit parametrization of M and uses only the first derivatives of the equality constraint functions. This is simpler than other samplers for manifolds that use second derivatives. Our integration algorithm was adapted from the multiphase Monte Carlo methods for computing high-dimensional volumes, based on intersecting the manifold with a decreasing sequence of balls.

We discussed several toy models aimed at understanding how the integration errors depend on the factor by which the balls are shrunk at each step. Our models and data suggested taking larger factors than those suggested in the literature. We also estimated these errors by computing results analytically for a Brownian motion in balls, but the results of these calculations disagree with computational experiments with the sampler. We discussed the geometric intuition behind this failure.

We tested the algorithms on several manifolds of different dimensions whose distributions or volumes are known analytically, and then we used our methods to compute the entropies of various clusters of hard sticky spheres. These calculations make specific predictions that could be tested experimentally, for example using colloids interacting via sticky DNA strands.

We expect our methods to apply to other systems of objects that can be effectively modeled as “sticky,” i.e., interacting with a short-range potential [28]. For example, one could calculate the entropies of self-assembling polyhedra, a system that has been realized experimentally in the hope that it will lead to efficient drug-delivery methods [31]; one could calculate the entire free energy landscape of

small clusters (building on results in [17]), or one could more accurately calculate the phase diagrams of colloidal crystals [27, 34] by computing entropic contributions to the free energy.

While our methods have worked well on the systems considered in this paper, we expect that applying them to certain larger systems will require additional techniques and ideas. For example, the manifold corresponding to the floppy degrees of freedom of a body-centered cubic crystal is thought to have a shape like a bicycle wheel, with long thin spokes leading out of a small round hub near the center of the crystal [19]. Sampling efficiently from a manifold with such different length scales will be a challenge, though one may draw on techniques developed to sample the basins of attraction of jammed sphere packings, for which the same issues arise [29]. A different, though possibly related problem, is to sample from manifolds whose constraint functions are not linearly independent. Such manifolds arise, for example, in clusters of identical spheres [21] or origami-based models [20] and could be important for understanding how crystals nucleate.

There are many possible improvements to the Monte Carlo methods described here. For the sampling algorithm, one improvement would be to use gradient information of the density function f to create a more effective proposal distribution. It may be possible to find a manifold version of exact discrete Langevin methods described, for example, in [25, 33]. An interesting extension to the sampler would be to allow it to jump between manifolds of different dimensions. With such a sampler, one could calculate the ratio of entropies in (4.7) directly without first calculating the volumes.

Our integration algorithm could also be enhanced in several ways. It could be extended to the case when the intermediate manifolds $M \cap B_i$ are not connected, for example, by using umbrella sampling techniques. It could be more adaptive: the radii R_i could be chosen more systematically using variance estimators, as is done in thermodynamic integration [18]. We could use additional ratios such as $N_{i,i+2}$ (see (3.2)) to improve the ratio estimates. We could also let n_i vary with ball radius so the errors at each ratio are more equally distributed; according to the diffusive scaling model of Section 5.2, we should use more points to estimate the larger ratios. Finally, it might be possible to change the probability distribution using smoother “steps” when breaking up the volume into a product of ratios, as in [18] and thermodynamic integration [12], though how to adapt this strategy to manifolds is not immediately clear.

Acknowledgment. E. Z. and M. H.-C. acknowledge support from Department of Energy Grant DE-SC0012296. The authors would additionally like to thank Gabriel Stoltz and Tony Lelievre for interesting discussions that helped to improve this article.

Bibliography

- [1] Björck, Å.; Golub, G. H. Numerical methods for computing angles between linear subspaces. *Math. Comp.* **27** (1973), 579–594. doi:10.2307/2005662
- [2] Brubaker, M.; Salzmänn, M.; Urtasun, R. A family of MCMC methods on implicitly defined manifolds. *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (La Palma, Canary Island)*, 161–172.
- [3] Byrne, S.; Girolami, M. Geodesic Monte Carlo on embedded manifolds. *Scand. J. Stat.* **40** (2013), no. 4, 825–845. doi:10.1111/sjos.12036
- [4] Ciccotti, G.; Lelievre, T.; Vanden-Eijnden, E. Projection of diffusions on submanifolds: application to mean force computation. *Comm. Pure Appl. Math.* **61** (2008), no. 3, 371–408. doi:10.1002/cpa.20210
- [5] Ciccotti, G.; Vanden-Eijnden, E. Second-order integrators for Langevin equations with holonomic constraints. *Chemical Physics Letters* **429** (2006), no. 1-3, 310–316. doi:10.1016/j.cplett.2006.07.086
- [6] Connelly, R.; Whiteley, W. Second-order rigidity and prestress stability for tensegrity frameworks. *SIAM J. Discrete Math.* **9** (1996), no. 3, 453–491. doi:10.1137/S0895480192229236
- [7] Diaconis, P. Patterns in eigenvalues: the 70th Josiah Willard Gibbs lecture. *Bull. Amer. Math. Soc. (N.S.)* **40** (2003), no. 2, 155–178. doi:10.1090/S0273-0979-03-00975-3
- [8] Diaconis, P.; Holmes, S.; Shahshahani, M. Sampling from a manifold. *Advances in modern statistical theory and applications: a Festschrift in honor of Morris L. Eaton*, 102–125. Institute of Mathematical Statistics (IMS) Collections, 10. Institute of Mathematical Statistics, Beachwood, OH, 2013.
- [9] Evans, L. C. *Partial differential equations*. Second edition. Graduate Studies in Mathematics, 19. American Mathematical Society, Providence, R.I., 2010. doi:10.1090/gsm/019
- [10] Feng, L.; Pontani, L.-L.; Dreyfus, R.; Chaikin, P.; Brujic, J. Specificity, flexibility and valence of DNA bonds guide emulsion architecture. *Soft Matter* **9** (2013), 9816–9823. doi:10.1039/C3SM51586A
- [11] Fixman, M. Classical statistical mechanics of constraints: a theorem and application to polymers. *Proceedings of the National Academy of Sciences* **71** (1974), no. 8, 3050–3053.
- [12] Frenkel, D.; Ladd, A. New Monte Carlo method to compute the free energy of arbitrary solids. Applications to the fcc and hcp phases of hard spheres. *J. Chem. Phys.* **81** (1984), no. 7, 3188–3193. doi:10.1063/1.448024
- [13] Gardiner, C. *Stochastic methods: a handbook for the natural and social sciences*. Fourth edition. Springer, Berlin, Heidelberg, 2009.
- [14] Hairer, E. Symmetric projection methods for differential equations on manifolds. *BIT* **40** (2000), no. 4, 726–734. doi:10.1023/A:1022344502818
- [15] Hinch, E. J. Brownian motion with stiff bonds and rigid constraints. *J. Fluid Mech.* **271** (1994), 219–234. doi:10.1017/S0022112094001746
- [16] Holmes-Cerfon, M. Sticky-sphere clusters. *Annual Reviews of Condensed Matter Physics* **8** (2017), 77–98. doi:10.1146/annurev-conmatphys-031016-025357
- [17] Holmes-Cerfon, M.; Gortler, S. J.; Brenner, M. P. A geometrical approach to computing free-energy landscapes from short-ranged potentials. *Proc. Natl. Acad. Sci.* **110** (2012), no. 1, E5–E14. doi:10.1073/pnas.1211720110
- [18] Hou, F.; Goodman, J.; Hogg, D. The probabilities of orbital-companion models for stellar radial velocity data. Preprint, 2014. arXiv:1401.6128 [astro-ph.IM]
- [19] Jenkins, I. C.; Casey, M. T.; McGinley, J. T.; Crocker, J. C.; Sinno, T. Hydrodynamics selects the pathway for displacive transformations in DNA-linked colloidal crystallites. *Proc. Natl. Acad. Sci.* **111** (2014), no. 13, 4803–4808. doi:10.1073/pnas.1318012111

- [20] Johnson-Chazhykov, D.; Menon, G. The building game: from enumerative combinatorics to conformational diffusion. *J. Nonlinear Sci.* **26** (2016), no. 4, 815–845. doi:10.1007/s00332-016-9291-z
- [21] Kallus, Y.; Holmes-Cerfon, M. Free energy of singular sticky-sphere clusters. *Physical Review E* **95** (2017), no. 2, 022130. doi:10.1103/PhysRevE.95.022130
- [22] Lee, J. M. *Introduction to smooth manifolds*. Graduate Texts in Mathematics, 218. Springer, New York, 2003. doi:10.1007/978-0-387-21752-9
- [23] Leimkuhler, B.; Matthews, C. Efficient molecular dynamics using geodesic integration and solvent-solute splitting. *Proc. A.* **472** (2016), no. 2189, 20160138, 22 pp. doi:10.1098/rspa.2016.0138
- [24] Lelièvre, T.; Rousset, M.; Stoltz, G. Langevin dynamics with constraints and computation of free energy differences. *Math. Comp.* **81** (2012), no. 280, 2071–2125. doi:10.1090/S0025-5718-2012-02594-4
- [25] Liu, J. S. *Monte Carlo strategies in scientific computing*. Springer Series in Statistics. Springer, New York, 2001.
- [26] Lovász, L.; Vempala, S. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *J. Comput. System Sci.* **72** (2006), no. 2, 392–417. doi:10.1016/j.jcss.2005.08.004
- [27] Macfarlane, R. J.; Lee, B.; Jones, M. R.; Harris, N.; Schatz, G. C.; Mirkin, C. A. Nanoparticle superlattice engineering with DNA. *Science* **334** (2011), no. 6053, 204–208. doi:10.1126/science.1210493
- [28] Manoharan, V. N. Colloidal matter: packing, geometry, and entropy. *Science* **349** (2015), no. 6251, 942. doi:10.1126/science.1253751
- [29] Martiniani, S.; Schrenk, K. J.; Stevenson, J. D.; Wales, D. J.; Frenkel, D. Structural analysis of high-dimensional basins of attraction. *Phys. Rev. E* **94** (2016), no. 3, 031301. doi:10.1103/PhysRevE.94.031301
- [30] Meng, G.; Arkus, N.; Brenner, M.; Manoharan, V. The free-energy landscape of clusters of attractive hard spheres. *Science* **327** (2010), no. 5965, 560–563. doi:10.1126/science.1181263
- [31] Pandey, S.; Ewing, M.; Kunas, A.; Nguyen, N.; Gracias, D. H.; Menon, G. Algorithmic design of self-folding polyhedra. *Proc. Natl. Acad. Sci.* **108** (2011), no. 50, 19885–19890. doi:10.1073/pnas.1110857108
- [32] Simonovits, M. How to compute the volume in high dimension? *Math. Program.* **97** (2003), no. 1, 337–374. doi:10.1007/s10107-003-0447-x
- [33] Sokal, A. Monte Carlo methods in statistical mechanics: Foundations and new algorithms. *Functional integration: basics and applications*, 131–192. Springer, Boston, 1997.
- [34] Tkachenko, A. V. Generic phase diagram of binary superlattices. *Proc. Natl. Acad. Sci.* **113** (2016), no. 37, 10269–10274. doi:10.1073/pnas.1525358113
- [35] Wales, D. *Energy landscapes*. Cambridge University Press, Cambridge, 2003.
- [36] Wampler, C. W.; Sommese, A. J. Numerical algebraic geometry and algebraic kinematics. *Acta Numer.* **20** (2011), 469–567. doi:10.1017/S0962492911000067
- [37] Zhou, J.; Shi, J. Characteristic classes on Grassmann manifolds. Preprint, 2008. arXiv:0809.0808 [math.FA]

JONATHAN GOODMAN

Courant Institute

251 Mercer St.

New York, NY 10012

USA

E-mail: goodman@cims.nyu.edu

MIRANDA HOLMES-CERFON

Courant Institute

251 Mercer St.

New York, NY 10012

USA

E-mail: holmes@cims.nyu.edu

EMILIO ZAPPA

Courant Institute

251 Mercer St.

New York, NY 10012

USA

E-mail: zappa@cims.nyu.edu

Received February 2017.

Revised September 2017.