Rejection-free Monte Carlo scheme for anisotropic particles

Daniel W. Sinkovits,[1,2] Stephen A. Barr,[3] and Erik Luijten[1,4,a)]
[1]*Department of Materials Science and Engineering, Northwestern University, Evanston, Illinois 60208, USA*
[2]*Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[3]*Department of Materials Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[4]*Department of Engineering Sciences and Applied Mathematics, Northwestern University, Evanston, Illinois 60208, USA*

(Received 23 January 2012; accepted 27 February 2012; published online 12 April 2012)

We extend the geometric cluster algorithm [J. Liu and E. Luijten, Phys. Rev. Lett. **92**, 035504 (2004)], a highly efficient, rejection-free Monte Carlo scheme for fluids and colloidal suspensions, to the case of anisotropic particles. This is made possible by adopting hyperspherical boundary conditions. A detailed derivation of the algorithm is presented, along with extensive implementation details as well as benchmark results. We describe how the quaternion notation is particularly suitable for the four-dimensional geometric operations employed in the algorithm. We present results for asymmetric Lennard-Jones dimers and for the Yukawa one-component plasma in hyperspherical geometry. The efficiency gain that can be achieved compared to conventional, Metropolis-type Monte Carlo simulations is investigated for rod–sphere mixtures as a function of rod aspect ratio, rod–sphere diameter ratio, and rod concentration. The effect of curved geometry on physical properties is addressed. © *2012 American Institute of Physics*. [http://dx.doi.org/10.1063/1.3694271]

I. INTRODUCTION

An important advantage of Monte Carlo simulations over molecular dynamics simulations is the possibility to perform "unphysical" moves that allow a more rapid relaxation of a system without altering its thermodynamic equilibrium properties.[1] Examples include the configurational-bias Monte Carlo method for polymeric systems[2,3] as well as the grand-canonical[4] and Gibbs ensemble[5] Monte Carlo methods that permit the simulation of two-phase coexistence without the need for an interface. Another example is the geometric cluster algorithm (GCA),[6–10] which is designed to overcome the difficulties presented by large size asymmetry, that is, a large ratio in size between the largest and the smallest particles in a system. This situation frequently occurs in colloidal and biological systems, and leads to a dynamic slowdown in which the motion of the larger species takes place on very long time scales compared to the smaller species, so that neither molecular dynamics simulations nor Monte Carlo simulations employing local moves are able to properly sample phase space for the larger species. An important physical phenomenon that can be probed by the GCA is the depletion interaction.[11] The GCA was inspired by a Monte Carlo scheme for producing configurations of non-overlapping hard spheres,[12] but generalizes this to multicomponent systems with arbitrary pairwise potentials, generating configurations of particles according to the Boltzmann distribution. Thus, it provides an alternative to the conventional Metropolis Monte Carlo scheme[13] by building clusters of particles that are subsequently moved collectively in a *rejection-free* manner, i.e., every cluster that is initiated will be moved once its construction is complete, irrespective of the nature of the interactions between the different particles. The resulting non-local algorithm greatly accelerates the numerical simulation of multicomponent systems[7] and has been employed to study a variety of phenomena, including the nanoparticle haloing mechanism for colloidal stabilization[14–17] and the effect of trace contaminants on the location of the coexistence curve of a Lennard-Jones fluid.[18,19] In addition, modifications of the GCA have been proposed for the simulation of collective motion in self-assembly processes,[20–23] the simulation of solvated systems,[24] and the simulation of polymer-induced phase separation in bacterial suspensions.[25] The geometric operations involved in the GCA have also inspired (Metropolis-type) simulation schemes,[26] and a biasing scheme has been proposed to employ the GCA to sample the radial distribution function in liquid mixtures even more efficiently.[27] Krauth[28] pointed out that the GCA can be interpreted as a mapping of a continuous fluid of interacting particles onto an Ising spin glass. Lastly, we note that for hard-sphere systems, cluster moves can also be employed to perform event-chain Monte Carlo simulations.[29]

In spite of the success of the GCA, it is subject to an important limitation. Namely, it is not ergodic when applied to anisotropic particles—in the course of a simulation, such objects will only explore a discrete set of orientations. Resolution of this drawback is particularly pressing since various systems that are currently at the forefront of soft condensed-matter research employ particles with a varying degree of anisotropy.[30] Notable examples are suspensions of rods and tetrapods,[31,32] Janus particles,[33–36] and colloids with rod-like depletants.[37] We have developed an extension to the GCA that lifts this limitation, permitting the simulation of anisotropic particles while retaining the rejection-free nature of the original GCA. A first account of this *hyperspherical geometric*

a)Electronic mail: luijten@northwestern.edu.

cluster algorithm, including an application to the aggregation of tetrapodal nanoparticles, has been published in Ref. 38. The present paper provides a detailed description and justification of the new algorithm as well as an examination of its efficiency under various circumstances. Since this algorithm requires the use of hyperspherical geometry, we include several appendices with implementation details and a discussion of testing procedures. Lastly, we also describe an alternative variant of the GCA, which is applicable to the simulation of anisotropic particles under conventional periodic (rather than hyperspherical) boundary conditions, albeit at the expense of rejecting a subset of all proposed cluster moves.

II. GEOMETRIC ALGORITHMS FOR ANISOTROPIC PARTICLES

A. Geometric transformations and anisotropic particles

The GCA constructs and moves clusters of particles according to a geometric transformation.[7,8] Generalizing the rotation over π radians about a pivot point proposed for two-dimensional (2D) hard-sphere fluids,[12] most implementations choose as this geometric transformation a point reflection with respect to a pivot point that for each cluster move is randomly chosen within the simulation cell. Alternative transformations are possible, as discussed below, but they all share the common requirement that periodic boundary conditions must be employed to ensure that, upon transformation, a particle is still located within the simulation cell.

Although a point reflection with respect to an arbitrary pivot point can translate a particle to any spatial position, it cannot generate arbitrary orientations. Indeed, successive point reflections merely toggle a particle's orientation between two states. Whereas this is not a matter of concern for spherically symmetric particles, the ergodic treatment of particles that are anisotropic (be it in their shape or in their interactions with other constituents of the system) requires the use of additional moves. The easiest and perhaps most obvious solution is to relax orientational degrees of freedom by supplementing the cluster moves with local rotation moves. However, an algorithm involving such moves will no longer be rejection-free (as it incorporates a Metropolis-type acceptance criterion) and the simulations (e.g., of long rods in an environment of small spherical particles) typically will suffer from a slow relaxation of the orientation of the large particles. Therefore, we instead consider the use of other geometric transformations besides point reflection, which will permit the ergodic simulation of anisotropic particles while retaining the rejection-free nature of the algorithm. In the GCA, acceptable transformations must satisfy the following two criteria:[39–41]

1. The transformation must be *self-inverse*, i.e., when applied twice it yields the identity operation.
2. The transformation must keep the action invariant when applied to all particles in the system. For the geometric transformations considered here, this is equivalent to requiring that the transformation is *isometric*, i.e., it preserves all interparticle distances (for a pair of extended objects, this implies preservation of the distance between each set of points chosen on the two objects, so the relative orientation of the two objects is preserved as well).

In three dimensions, these criteria allow for two more classes of transformation, plane reflection and line reflection (π rotation about an axis). A single plane reflection or line reflection is unable to transform a particle to every other orientation, but combinations of plane reflections or line reflections can produce arbitrarily small rotations, thus proving that a sequence of such transformations can yield any orientational change. In fact, two successive transformations suffice to achieve an arbitrary change in orientation.

Whereas these transformations thus may appear to resolve the ergodicity problem observed for anisotropic particles, a serious problem arises when they are applied in the presence of periodic boundary conditions (PBC). In the following discussion, it is important to recall (cf. Ref. 7) that the proof of detailed balance in the GCA hinges on the fact that the interaction energy of all particles included in the cluster remains invariant under the transformation, and so does the interaction energy of all particles that are not included in the cluster. The only energy change occurs for pairs of particles in which one particle is a member of the cluster and the other particle is not (we recall that the GCA is restricted to systems with a pairwise additive potential energy). Now, we note that, under PBC, reflections in arbitrarily oriented planes or lines are *not* isometric transformations, since generally they do not belong to the symmetry group of the Bravais lattice that defines the periodic replication of the primary simulation cell. We elucidate this by considering the treatment, within the GCA, of periodic images of particles that are included in a cluster (i.e., that are subjected to a transformation). The first option is to transform all periodic images of all particles that are explicitly added to the cluster. This ensures that the distances between all particles included in the cluster as well as all their periodic images, remain invariant upon transformation, so that their interaction energy is indeed unchanged. However, all transformed particles will be aligned with a *transformed* Bravais lattice, while the particles that were not part of the cluster and thus were not transformed will still align with the *original* Bravais lattice. This is illustrated by means of a 2D example in Fig. 1. As a consequence, the two sets (transformed and not transformed) of particles cannot be reconciled into a single periodic system.

To obtain a final configuration that obeys PBC, one can opt *not* to transform the periodic images of a particle that is included in the cluster, but instead (after the transformation) reconstruct its periodic images according to the PBC of the system. However, this changes the distances between particles in the cluster and periodic images of these particles. If any of these distances lies within the range of interaction of the pair potential, this will result in a change in the potential energy of particles in the cluster. More precisely, for a pair of particles (i, j) that are both included in the cluster, with respective constructed images (i', j') (where the prime can refer to any periodic copy), all separations $|\mathbf{r}'_j - \mathbf{r}_i|$ change for $i \neq j$. Moreover, the orientations of particle i and particle j' with respect to their center-of-mass separation vector $\mathbf{r}'_j - \mathbf{r}_i$ change for all pairs (i, j'), including (i, i'). The latter case is

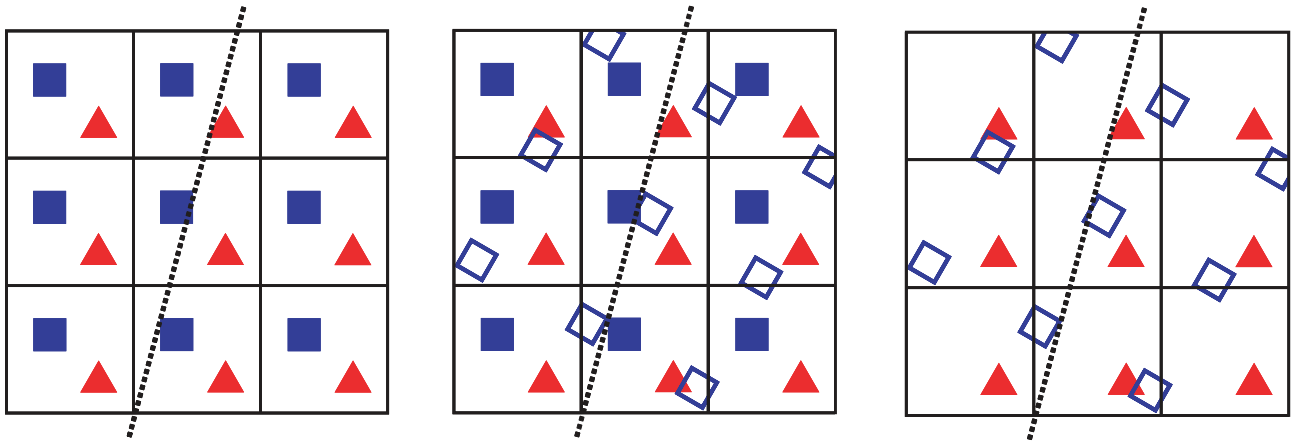

FIG. 1. Illustration of the conflict that arises in the geometric cluster algorithm upon application of a transformation that does not belong to the symmetry group of the Bravais lattice defining the periodic boundary conditions. The simulation cell contains two particles (solid square and solid triangle) and is periodically replicated (left panel). If the cluster contains the solid square as well as all its periodic images, the entire array of solid squares is reflected in the dotted line. The lattice implied by the reflected squares (open symbols) is tilted with respect to the original lattice (central panel). The red triangles are not part of the cluster and hence are not transformed. The resulting composite system of these triangles and the reflected squares is not periodic, as illustrated in the right panel; e.g., the shortest distance between a triangle and a square depends on which triangle is considered.

important for, e.g., dipolar interactions. Figure 2 provides a 2D example.

We note that some self-inverse transformations are symmetries of the Bravais lattice describing the PBC, so that transformed particles will align with the original Bravais lattice. For those transformations, the problem described above does not occur, and the two approaches (transforming all images along with a particle or reconstructing all periodic images upon transformation of a particle) produce an identical outcome. In certain cases, namely when anisotropic particles only take a discrete number of orientations, use of this restricted set of transformations is sufficient to generate an ergodic algorithm. This is exploited in the case of dimers on a square lattice.[42,43] For the case of anisotropic particles with continuous orientational degrees of freedom, one could consider to adopt a non-cubic periodic geometry (e.g., truncated octahedra), which allows a larger set of such transformations. Whereas this would make it possible for the particles to reach more orientations (from where they could be relaxed via local moves), the set of permitted transformations will always be finite and hence this does not yield an ergodic, rejection-free Monte Carlo scheme for anisotropic particles in a periodic geometry.

B. Geometric cluster algorithm with rejections

Before proceeding to the central objective of this paper, namely the formulation of a rejection-free GCA that overcomes the problems discussed in Sec. II A, we discuss a pragmatic solution for systems composed of particles that have a sufficiently short-ranged interaction potential. As noted, the GCA properly accounts for the change in potential energy between a particle that is transformed (i.e., included in the cluster) and a particle that is not transformed (i.e., not included in the cluster). The conflict between PBC and reflections in an arbitrarily oriented plane or line arises because a change in interaction energy occurs for pairs of particles that are both transformed. One might consider to account for this additional energy change via a Metropolis-style acceptance criterion. However, detailed balance would be very difficult to prove for such a scheme, because the particles that constitute a cluster will have different interaction strengths before and after the transformation, causing the cluster construction process to proceed with different probabilities in the reverse move. On the other hand, a proper algorithm can be devised by proceeding as discussed in conjunction with Fig. 2, i.e., transformation of each particle that is added to the cluster and subsequent reconstruction of the periodic images according to the PBC, provided that one *rejects* every cluster move in which any particle in the cluster interacts with a periodic image of another particle in the cluster in either the old or new positions of either particle. This criterion allows us to formulate an ergodic algorithm that simulates anisotropic particles based on the GCA, by performing cluster moves with arbitrary reflections and line reflections and then accepting the move only if there is no change in the interaction energy of all particles in the cluster (including their periodic images). If the interaction is sufficiently short ranged, a significant fraction of all proposed cluster moves will be accepted. We have implemented and tested this GCA with rejections for a system of Lennard-Jones dimers and confirmed that its results agree with a local-move Metropolis Monte Carlo simulation and with molecular dynamics simulations of the same system. Conversely, this algorithm breaks down completely when the interaction is not short ranged.

C. Transformations in isotropic space: The hyperspherical geometric cluster algorithm

The discussion in Sec. II A shows that the symmetries of the space in which a system is defined determine the allowed transformations and therefore the set of accessible

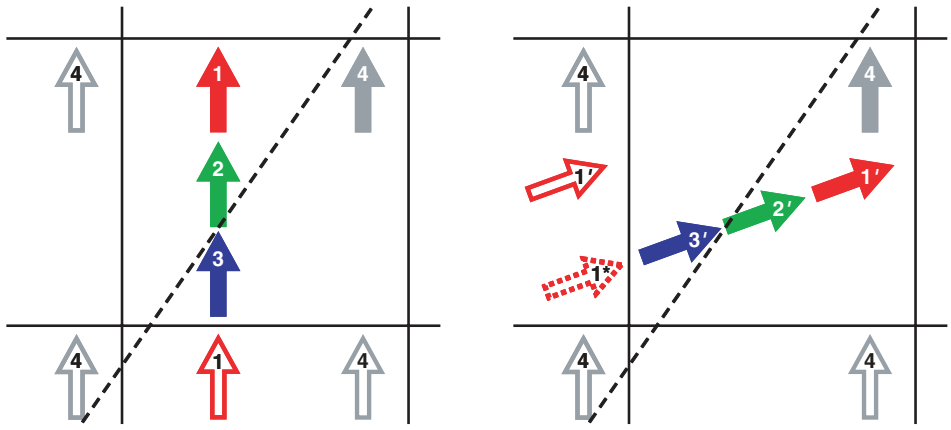

FIG. 2. Illustration of the conflict that arises in the geometric cluster algorithm if, upon incorporation of a particle in the cluster, its periodic images are not subjected to the same transformation. The simulation cell (left panel) contains four parallel arrow-shaped particles (solid objects), three of which are collinear. Some of their periodic images are shown as well (open objects). Particles 1–3 belong to the cluster, which is reflected in the dashed line. This transformation yields particles 1′–3′ (solid objects in right panel), which are periodically replicated. As illustrated by open arrow labeled 1′ in the right panel, these periodic images are no longer collinear with the particles that were part of the cluster, resulting in a change in interaction energy. This cannot be resolved through an alteration of the PBC such that the periodic image of 1′ is replaced with 1* (which corresponds to the approach taken in Fig. 1), as this would violate the periodicity of the particles that are not included in the cluster (represented by particle 4 and its images).

orientations. Thus, an ergodic algorithm for anisotropic particles requires the use of an isotropic space, which has continuous orientational symmetry in all directions. The only finite, isotropic three-dimensional (3D) space is the surface of a four-dimensional (4D) hypersphere (also referred to as a 3-sphere). Indeed, this geometry has been used before in Monte Carlo and molecular dynamics simulations. Specifically, the 2D counterpart of this geometry (i.e., the surface of a sphere) was first employed to avoid the use of Ewald summation for electrostatic interactions.[44] The case of general dimensionality was discussed shortly afterward.[45] Since then, it has been used in 3D systems, e.g., to deal efficiently with long-range electrostatic interactions[46] and to suppress finite-size effects.[47] Its use has also been extended to polymeric particles.[48] Being a three-dimensional manifold, this space is locally analogous to a 3D Euclidean space, so we can construct transformations analogous to every transformation in Euclidean space. An arbitrary plane reflection is a reflection through a hyperplane that contains the origin. Arbitrary line and point reflections can be constructed as the composition of two or three mutually orthogonal hyperplane reflections, respectively, that leave the origin fixed. In Appendix A we provide a detailed description on the construction of transformations using quaternions. Using hyperspherical geometry and an expanded set of transformations that includes analogues of arbitrary plane reflection and line reflection, we thus construct a *hyperspherical GCA* (h-GCA), which allows for the ergodic simulation of anisotropic particles and retains the property that it is a rejection-free Monte Carlo scheme.

The h-GCA follows the same procedure as the original GCA.[7,8] We include a step-by-step description of the cluster construction procedure for completeness.

1. A seed particle i is chosen, which forms the first particle in the cluster.
2. A self-inverse, isometric transformation is chosen, which transforms this particle from its old position and orientation i to a new position and orientation i'.
3. Each particle j that interacts with i or i' is considered for inclusion in the cluster. Particle j joins the cluster with probability

$$P(j|i) = \max[1 - \exp(-\beta \Delta U_{ij}), 0], \qquad (1)$$

where $\beta = 1/(k_B T)$, with k_B Boltzmann's constant and T the absolute temperature. ΔU_{ij} is the change in *pair* energy U between particles i and j resulting from the transformation of particle i, i.e., $\Delta U_{ij} = U(j, i') - U(j, i)$. In view of the superficial resemblance of Eq. (1) to a Metropolis acceptance criterion, we emphasize that ΔU_{ij} is *not* the total energy change that results from the transformation of i. If particle j joins the cluster, it is transformed to its new position and orientation. Furthermore, it is placed on a stack and removed from consideration for further transformations.
4. After all particles interacting with i or i' have been considered, a particle is retrieved from the stack, and the algorithm returns to step 3 with this particle as the new particle i.
5. When the stack is exhausted, the algorithm is finished, and all particles in the cluster will be in their new positions and orientations.

The proof of detailed balance for this algorithm is identical to the proof presented in Refs. 7 and 8.

To our knowledge, the fundamental connection between the permissible transformations in cluster moves and the symmetry of the space was not pointed out earlier, and it is not made explicit in the criteria stated in Sec. II A. In the original description of the GCA (Refs. 7 and 8) and in its subsequent applications,[14,16,17,49] the systems simulated contained only isotropic particles, and only point reflections were used for cluster moves, since these are sufficient to relax the translational degrees of freedom. Indeed, every point reflection is a central inversion, which is a symmetry of every Bravais lattice, illustrating why these transformations could be used with the GCA in PBC without problems. Cluster

algorithms have been proposed that are inspired by the GCA and that have been used to simulate systems of anisotropic particles in Euclidean space by employing other transformations than point reflections,[24,25] but both of these algorithms include a Metropolis-type acceptance criterion (i.e., they are not rejection-free) and rely on additional moves within the same simulation to achieve ergodicity. In contrast, the h-GCA described here is rejection-free even for anisotropic particles and autonomously generates an ergodic sequence of configurations that obey the Boltzmann distribution.

Whereas the h-GCA is of general applicability, we note that, apart from the ability to simulate anisotropic particles, it is subject to the same limitations as the original GCA.[7] Namely, the algorithm is only ergodic as long as the clusters do not comprise the entire system. This requirement limits the volume fractions and coupling strengths that can be examined. The maximum values attainable will depend on composition and degree of anisotropy of the various species; for spherical particles the highest volume fraction was found to be around 0.34. In addition, we point out a subtlety in the choice of geometric operations. For chiral particles, plane reflections, and point reflections are not permissible, as they lead to a chirality reversal. However, line reflections are still permissible and sufficient to relax the orientational degrees of freedom of such particles.

We conclude this algorithm description by noting that it is possible to use information about the seed particle in the choice of transformation, since the transformation is chosen after the seed particle has been selected. As first proposed in Ref. 8 (and later used in, e.g., Ref. 50), this can be exploited to control the average cluster size by constructing a geometric transformation that limits the maximum displacement of the seed particle. This is of particular importance for anisotropic particles, since it permits generation of transformations that limit the maximum change in orientation and thus offers control over the cluster size. We refer to such cluster moves which base their transformation on the position and orientation of the seed particle as *biased* cluster moves. A subtlety regarding the use of such moves for anisotropic particles is that a self-inverse transformation necessarily inverts one or more spatial dimensions, producing a discontinuous change in the orientation of the seed particle. However, transformations which invert dimensions that coincide with reflection symmetries of the seed particle are effectively identity operations on its orientation, from which one can construct transformations that effect arbitrarily small orientational changes. Evidently, particles that possess fewer or no reflection symmetries benefit less from biased cluster moves. The construction of the transformations for biased moves using quaternions is described in Appendix A 4.

III. DEMONSTRATION OF APPLICABILITY

A. Comparison with Metropolis Monte Carlo

We have implemented the algorithm described in Sec. II C. The coordinates and orientation of each particle are represented by a pair of quaternions, both of which are transformed when a particle is added to a cluster, as described in Appendix A. To efficiently deal with pair interactions, we employ the cell index method described in Appendix B 1. For compound particles ("molecules") we apply the mapping described in Appendix B 2.

As a first demonstration, we examine a system of 500 asymmetric dimers, each of which is composed of two spherical monomers labeled A and B, respectively, separated by a (geodesic) center-to-center distance σ. We place the dimers on the 3D surface of a hypersphere of radius R, which creates a space with volume

$$\Lambda = 2\pi^2 R^3. \tag{2}$$

Setting $R = 8.0932\sigma$, we have $\Lambda \approx 10464\sigma^3$ and a number density of $\rho = 4.7784 \times 10^{-2}$ dimers/σ^3. All monomers interact via a Lennard-Jones (LJ) potential,

$$U(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] + \varepsilon_c. \tag{3}$$

The potential is truncated at the cutoff $r_c = 2.5\sigma$ and the shift ε_c is chosen such that $U(r_c) = 0$. The distance r is the arc length distance along a geodesic on the hypersphere,

$$r = R\arccos(\hat{\mathbf{r}}_1 \cdot \hat{\mathbf{r}}_2), \tag{4}$$

where $\hat{\mathbf{r}}_1$ and $\hat{\mathbf{r}}_2$ are unit vectors denoting the position of the two interacting particles on the surface of the hypersphere. The dimers are asymmetric in the coupling strength between A and B monomers, with $\varepsilon_{AA} = 0.625k_BT$, $\varepsilon_{BB} = \varepsilon_{AA}/2 = 0.3125k_BT$, and $\varepsilon_{AB} = \sqrt{\varepsilon_{AA}\varepsilon_{BB}} \approx 0.44194k_BT$. The coupling strength and concentration are deliberately chosen to produce a system with moderately strong interactions, so that the average cluster size is a significant fraction of the total number of dimers.

To demonstrate the correctness of the h-GCA algorithm described in Sec. II C and of our implementation, we simulate this system both by means of the conventional Metropolis Monte Carlo algorithm and via the h-GCA. In the former method, a move consists of a combined rotation and translation, in which dimers are translated on the surface of the hypersphere over a maximum distance of 6σ and freely rotated to a new, uniformly chosen orientation. Defining a set of 1000 such moves as one sweep, we equilibrate the system for 10^3 sweeps and then perform 10^7 production sweeps, resulting in an average acceptance ratio of 0.356. The potential energy is sampled after every sweep. In the simulations employing the h-GCA, we pursue four different variants, each using a different type of transformation to perform the cluster moves: plane reflections, line reflections, biased plane reflections, and biased line reflections. The average cluster size depends on the transformation used. For each run, the system is equilibrated for 10^4 cluster moves, followed by 10^8 cluster moves during which the potential energy is sampled after every 10 moves. Table I lists the average potential energy for the simulation using conventional Monte Carlo moves as well as for those using different types of cluster moves. All results have a high precision (relative error $\mathcal{O}(10^{-5})$) and agree within the statistical uncertainty. The table also lists the relative cluster size. The distribution of both size and structure of the

TABLE I. Comparison of the average potential energy obtained in different simulations of the same system of asymmetric Lennard-Jones dimers in a hyperspherical geometry. The different simulation types listed refer to a conventional Metropolis-type ("local-move") simulation and to the hyperspherical geometric cluster algorithm with various choices for the geometric transformation used to construct the clusters. All runs agree within statistical uncertainty.

Simulation type	Average potential energy	Average cluster size	Relative cluster size
Local move	-246.1945 ± 0.0043	n/a	n/a
Plane reflection	-246.2042 ± 0.0035	141.37	28%
Line reflection	-246.2003 ± 0.0033	182.60	36%
Biased plane reflection	-246.2049 ± 0.0035	129.15	26%
Biased line reflection	-246.2025 ± 0.0033	182.68	37%

clusters may affect the rate of decorrelation, and hence statistical quality, of thermodynamic averages. However, despite the dependence of average cluster size on the choice of transformation, all h-GCA runs yield the same statistical accuracy.

B. Comparison with molecular dynamics simulations

It is desirable to compare our implementation of the h-GCA algorithm against a completely independent simulation code. Since, to the best of our knowledge, no Monte Carlo or molecular dynamics simulation packages are available that employ hyperspherical boundary conditions, we resort to a special case, namely the application of the h-GCA to a 2D system defined on the surface of a 3D sphere.[38] It is then possible to make a direct comparison to results obtained from a molecular dynamics package designed for 3D simulations, as described below.

As in the 3D case of Sec. III A, we define a system of 500 asymmetric dimers, composed of two monomers (labeled A and B) that have a (geodesic) center-to-center distance σ. Since the curved space is now two-dimensional, the monomers are spherical caps (sometimes referred to as spherical calottes). They interact again via a LJ potential, with asymmetric coupling parameters $\varepsilon_{AA} = 1.0 k_B T$, $\varepsilon_{BB} = \varepsilon_{AA}/2 = 0.5 k_B T$, and $\varepsilon_{AB} = \sqrt{\varepsilon_{AA}\varepsilon_{BB}} \approx 0.7071 k_B T$. To obtain different concentrations (four different values ranging from 0.05 dimers/σ^2 to 0.20 dimers/σ^2) we vary the sphere radius R. For each concentration, we perform two different types Monte Carlo simulations using the h-GCA algorithm, employing either plane reflections or line reflections (both transformations are used in unbiased form). With increasing concentration, samples are taken every 250, 80, 20, or 10 cluster moves, respectively. After equilibration for 1000 samples, $(1.4–7.0) \times 10^6$ samples are obtained. In a third series of simulations, we use the h-GCA code but actually perform conventional single-particle moves consisting of a translation and a rotation (cf. Sec. III A). The maximum displacement is σ and the maximum rotation of the rod director is 30°. Samples are taken every 30 000 steps. The systems are again equilibrated for 1000 samples, followed by $0.36–0.98 \times 10^6$ samples.

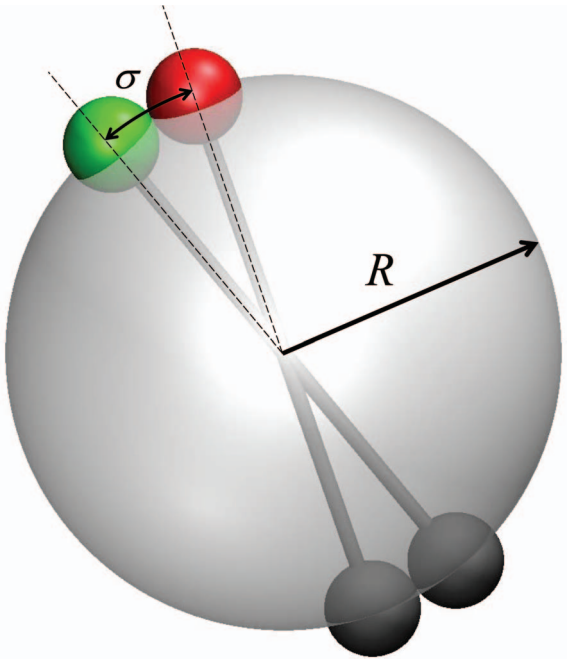

FIG. 3. Schematic of the method used to perform simulations on a spherical surface by means of a conventional molecular dynamics package. A dimer (comprised of a red and a green monomer, near the top of the image) on the surface of a sphere is rigidly bonded with dummy particles (dark spheres near the bottom of the image) at points antipodal to the monomers. As a result the center of mass of this "molecule" coincides with the center of the sphere. Since the dummy particles do not interact with any other particles, they do not interfere with the motion of the dimers on the spherical surface. Note that in the actual model definition (see main text) the 2D monomers are represented merely by the intersection of the monomers shown in this figure and the spherical surface; however, this is immaterial for the actual simulations.

An identical system can be defined in the LAMMPS molecular dynamics package[51] via a trick. Namely, we constrain the dimers to the surface of a sphere of radius R by rigidly bonding each monomer to a "dummy" particle of the same mass at the antipodal point, as illustrated in Fig. 3. The dummy particles do not interact with any particle, but they ensure that the center of mass of each molecule, composed of a dimer and two dummy particles making up the dummy dimer, lies at the center of the sphere. We restrict the motion of the molecules to rotations only, thus constraining the dimers to move on the surface of the sphere, but otherwise allowing their unrestricted translation and rotation on this surface. Furthermore, we modify the pairwise interactions between monomers such that the distance between particles is measured as the arc length and that the resulting forces act tangential to the sphere. We perform simulations with a time step of 0.001τ, where τ is the reduced time unit. After equilibration for 10^7 steps, the systems are simulated for $(2.2–10) \times 10^8$ time steps and samples are taken every 1000 steps.

In all simulations, we sample the potential energy, as summarized in Table II. For each dimer concentration, the results of the various Monte Carlo simulations and the molecular dynamics simulations have an uncertainty of $\mathcal{O}(10^{-4})$ and agree within statistical error.

TABLE II. Comparison of the average potential energy obtained in different simulations of the same system of 2D asymmetric Lennard-Jones dimers confined to a spherical surface. The simulation methods include conventional local-move Monte Carlo, the hyperspherical geometric cluster algorithm with two different geometric transformations, and molecular dynamics simulations. For each dimer concentration, all four methods agree within statistical uncertainty.

Concentration (dimers/σ^2)	0.05	0.10	0.15	0.20
Sphere radius R	28.2094792	19.947114	16.2867504	14.1047396
Molecular dynamics	-137.527 ± 0.020	-262.781 ± 0.022	-383.376 ± 0.023	-505.702 ± 0.023
Local move	-137.518 ± 0.012	-262.797 ± 0.026	-383.404 ± 0.024	-505.758 ± 0.029
Plane reflection	-137.501 ± 0.008	-262.789 ± 0.016	-383.413 ± 0.027	-505.724 ± 0.037
Line reflection	-137.486 ± 0.009	-262.802 ± 0.016	-383.411 ± 0.021	-505.815 ± 0.032

C. Comparison with known results for the Lennard-Jones fluid on a hypersphere

Since the implementation of a hyperspherical algorithm can be relatively involved, especially when use is made of quaternions (Appendix A) and a cell index method (Appendix B 1), it is advantageous to test the code by comparison to accurate tabulated data for a well-defined model. Comparison to data for the same model, e.g., a Lennard-Jones fluid, in Euclidean geometry is insufficient, as the intrinsic curvature of the hyperspherical space will lead to systematic differences in thermodynamic properties. Likewise, results that rely on extrapolation (e.g., finite-size scaling near a critical point) are less desirable as well, as the extrapolation techniques used may obfuscate numerical errors in the data. Relatively few suitable results are available in the literature. Schreiner and Kratky[52] have presented numerical results for the Lennard-Jones fluid on a hypersphere, which we will compare to here. Since these results are relatively old and hence of moderate numerical accuracy, we also compare to more recent results for the Yukawa one-component plasma in Sec. III D.

The results of Ref. 52 pertain to a system of N particles on a hypersphere, interacting through a Lennard-Jones potential along their short geodesic separation (this is equivalent to imposing a cutoff equal to half the circumference of the hypersphere). Two different combinations of number density and coupling strength are considered, namely $\rho = 0.8016\sigma^{-3}$ with coupling $\varepsilon/k_BT = 1/0.8347 \approx 1.1980$ (results in Table III) and $\rho = 0.7757\sigma^{-3}$ with coupling $\varepsilon/k_BT = 1/1.1686 \approx 0.8557$ (results in Table IV). We simulate this system for the same parameter combinations, using three different methods: single-particle displacements randomly and uniformly chosen within a sphere of radius 6σ around the current particle position, cluster moves with unbiased plane reflections, and cluster moves with unbiased line reflections. The runs employing local moves consist of an equilibration period of 1.5×10^7 single-particle steps and a production period of 3×10^9 steps, with samples taken every 3000 steps. The runs with cluster moves are equilibrated for 5×10^5 clusters and run for 2×10^6 cluster moves, with a sampling interval of 100 moves. Tables III and IV show the reduced average internal energy per particle, $U^* = U/(N\varepsilon)$, for 9 different system sizes for these three different algorithms, along with the data of Ref. 52. All results agree within the statistical uncertainty.

D. Comparison with known results for the Yukawa one-component plasma

Caillol and Gilles have carried out a highly detailed study of the Yukawa one-component plasma[53] and performed extensive numerical simulations[54] for this model in a hyperspherical geometry. The system consists of N identical point particles in a volume Λ that each carry a charge q and interact (in a Euclidean geometry) via a Yukawa potential $v_\alpha(r) = \exp(-\alpha r)/r$. This model can be shown to depend on only two reduced parameters, namely the coupling parameter $\Gamma = \beta q^2/a_{\text{ion}}$ and the reduced screening parameter

TABLE III. Comparison of the reduced internal energy per particle, $-U^* = -U/(N\varepsilon)$, for different types of simulations of a Lennard-Jones fluid on a hypersphere, with number density $\rho = 0.8016\sigma^{-3}$ and coupling strength $\varepsilon/k_BT = 1/0.8347 \approx 1.1980$.

N	Cluster reflection	Cluster line reflection	Local moves	Ref. 52
54	4.036 ± 0.002	4.034 ± 0.005	4.0343 ± 0.0006	4.008 ± 0.012
81	4.346 ± 0.001	4.338 ± 0.004	4.3450 ± 0.0007	4.349 ± 0.008
108	4.539 ± 0.002	4.541 ± 0.007	4.5348 ± 0.0009	4.541 ± 0.008
162	4.767 ± 0.002	4.766 ± 0.004	4.7657 ± 0.0008	4.776 ± 0.004
216	4.905 ± 0.002	4.900 ± 0.007	4.9055 ± 0.0009	4.901 ± 0.003
326	5.071 ± 0.001	5.067 ± 0.004	5.0701 ± 0.0009	5.072 ± 0.005
432	5.165 ± 0.001	5.158 ± 0.004	5.1668 ± 0.0010	5.160 ± 0.003
648	5.279 ± 0.001	5.279 ± 0.004	5.2775 ± 0.0010	5.287 ± 0.003
864	5.344 ± 0.001	5.342 ± 0.003	5.3459 ± 0.0009	5.344 ± 0.004

TABLE IV. Comparison of the reduced internal energy per particle, $-U^* = -U/(N\varepsilon)$, for different types of simulations of a Lennard-Jones fluid on a hypersphere, with number density $\rho = 0.7757\sigma^{-3}$ and coupling strength $\varepsilon/k_BT = 1/1.1686 \approx 0.8557$.

N	Cluster reflection	Cluster line reflection	Local moves	Ref. 52
54	3.705 ± 0.001	3.707 ± 0.003	3.7023 ± 0.0005	3.694 ± 0.009
81	3.993 ± 0.001	4.001 ± 0.005	3.9917 ± 0.0006	3.986 ± 0.009
108	4.167 ± 0.001	4.162 ± 0.004	4.1684 ± 0.0006	4.171 ± 0.009
162	4.383 ± 0.001	4.378 ± 0.005	4.3839 ± 0.0007	4.369 ± 0.006
216	4.512 ± 0.001	4.517 ± 0.004	4.5119 ± 0.0007	4.509 ± 0.004
326	4.666 ± 0.001	4.667 ± 0.005	4.6659 ± 0.0008	4.665 ± 0.004
432	4.755 ± 0.001	4.754 ± 0.003	4.7547 ± 0.0007	4.755 ± 0.003
648	4.859 ± 0.001	4.862 ± 0.003	4.8603 ± 0.0008	4.863 ± 0.003
864	4.922 ± 0.001	4.925 ± 0.003	4.9221 ± 0.0007	4.917 ± 0.005

$\alpha^* = \alpha a_{\rm ion}$. Here $a_{\rm ion}$ is the "ionic radius," to be defined below. A precise definition of the model employed, along with accurate numerical data for different derivatives of the free energy as a function of Γ and α^*, is provided in Ref. 54, making this an ideal starting point for testing purposes. Moreover, the tabulated data refer to a finite number of particles, $N = 600$, simulated in the canonical ensemble, so any numerical comparison does not involve extrapolations to the thermodynamic limit.

Unfortunately, we found that the numerical data presented in Ref. 54 contain systematic deviations, for all properties listed and all parameter values. To remedy this situation, and to provide a data set that can serve as a reference for future simulations in hyperspherical geometry, we have implemented an independent Metropolis-type Monte Carlo simulation of this model and used it to perform accurate simulations for a representative subset of the data listed in Ref. 54. In addition, we have confirmed that our implementation of the hyperspherical GCA yields results that are in accurate agreement with our results obtained from the conventional Monte Carlo simulations.

For completeness, we start with a precise definition of the Yukawa one-component plasma on a hypersphere $\mathcal{S}^3$. For a system with radius R, volume Λ given by Eq. (2), and number density $\rho = N/\Lambda$, the ionic radius $a_{\rm ion}^{\mathcal{S}^3} \equiv \psi_0 R$ is obtained from the space-filling condition $N v^{\mathcal{S}^3} = \Lambda$, where $v^{\mathcal{S}^3}$ is the single-particle volume in $\mathcal{S}^3$,

$$v^{\mathcal{S}^3} = 2\pi R^3 d(\psi_0), \tag{5}$$

with

$$d(\psi_0) = \psi_0 - \sin(\psi_0)\cos(\psi_0). \tag{6}$$

Thus, in terms of the number density $\rho = N/\Lambda$, ψ_0 is found from

$$2\pi\rho R^3 d(\psi_0) = 1. \tag{7}$$

The separation between a pair of particles located at $\mathbf{r}_i = (x_i, y_i, z_i, t_i)$ and $\mathbf{r}_j = (x_j, y_j, z_j, t_j)$, respectively, is their geodesic distance $R\psi \equiv R\arccos((\mathbf{r}_i \cdot \mathbf{r}_j)/R^2)$ (cf. Eq. (4)) and the exponentially decaying potential v_α in Euclidean space is replaced by the pair potential $v_\alpha^{\mathcal{S}^3}$,

$$v_\alpha^{\mathcal{S}^3}(\psi) = \begin{cases} \dfrac{1}{R}\dfrac{\sinh(\omega(\pi-\psi))}{\sin(\psi)\sinh(\omega\pi)} - \dfrac{4\pi}{\alpha^2\Lambda} & \text{for } \alpha R \geq 1, \\ \dfrac{1}{R}\dfrac{\sin(\omega(\pi-\psi))}{\sin(\psi)\sin(\omega\pi)} - \dfrac{4\pi}{\alpha^2\Lambda} & \text{for } \alpha R < 1, \end{cases} \tag{8}$$

with $\omega = \sqrt{|\alpha^2 R^2 - 1|}$. Note that the argument $\omega\pi$ in the denominator is misprinted as $\omega\psi$ for both cases in Ref. 54 (but stated correctly when originally derived as Eqs. (4.8) and (4.9) in Ref. 53). The potential energy then follows from a pairwise sum,

$$\beta V^{\mathcal{S}^3} = \frac{\beta q^2}{2}\sum_{i=1}^{N}\sum_{j\neq i} v_\alpha^{\mathcal{S}^3}(\psi_{ij}) + N\beta\mathcal{A}. \tag{9}$$

The additive constant comprises two contributions,

$$\beta\mathcal{A} = \beta\mathcal{A}_{\rm OCP} + \beta\delta\mathcal{A}, \tag{10}$$

where $\mathcal{A}_{\rm OCP}$ is the one-component plasma (or $\alpha = 0$) contribution (first derived as Eq. (4.38) in Ref. 53),

$$\beta\mathcal{A}_{\rm OCP} = -\frac{9}{10}\Gamma - \frac{3\Gamma}{4\pi R^*} + \frac{\Gamma}{2R^* d(\psi_0)} \times \left[\frac{3}{2} + \sin^2(\psi_0) - \frac{\psi_0\sin^2(\psi_0)}{d(\psi_0)}\right]. \tag{11}$$

Here $R^* = R/a_{\rm ion}^{\mathcal{S}^3} = 1/\psi_0$ is the reduced radius and the coupling constant is now defined in terms of the hyperspherical ionic radius, $\Gamma = \beta q^2/a_{\rm ion}^{\mathcal{S}^3}$. The second contribution in Eq. (10) depends on the reduced screening parameter $\alpha^* = \alpha a_{\rm ion}^{\mathcal{S}^3}$,

$$\beta\delta\mathcal{A} = \begin{cases} \dfrac{\Gamma}{2}\left(\dfrac{3}{2\pi R^*} - \dfrac{\omega\coth(\omega\pi)}{R^*} - \dfrac{4\pi}{\alpha^{*2}\Lambda^*}\right) & \text{for } \alpha R \geq 1, \\ \dfrac{\Gamma}{2}\left(\dfrac{3}{2\pi R^*} - \dfrac{\omega\cot(\omega\pi)}{R^*} - \dfrac{4\pi}{\alpha^{*2}\Lambda^*}\right) & \text{for } \alpha R < 1, \end{cases} \tag{12}$$

where $\Lambda^* = \Lambda/(a_{\rm ion}^{\mathcal{S}^3})^3$ is the reduced volume. This is Eq. (3.6) in Ref. 54, but note that the $\alpha R < 1$ term contains a misprint (the correct form appears as Eq. (4.42) in Ref. 53).

In the simulations, we focus on derivatives of the excess free energy per particle f with respect to Γ and α^*, respectively, which are given by

$$\Gamma\frac{\partial f}{\partial\Gamma} = \frac{1}{N}\langle\beta V^{\mathcal{S}^3}\rangle, \tag{13}$$

$$\frac{\partial f}{\partial\alpha^*} = \frac{1}{N}\langle\beta W^{\mathcal{S}^3}\rangle, \tag{14}$$

where the angular brackets indicate thermodynamic averages. The functional $W^{\mathcal{S}^3}$ is given by

$$\beta W^{\mathcal{S}^3} = N(\partial\beta\mathcal{A}/\partial\alpha^*) - \frac{\Gamma}{2}\sum_{i=1}^{N}\sum_{j\neq i} w_\alpha^{\mathcal{S}^3}(\psi_{ij}). \tag{15}$$

Since the first term in Eq. (10) is independent of α, the derivative of $\beta\mathcal{A}$ follows from Eq. (12),

$$\frac{\partial\beta\mathcal{A}}{\partial\alpha^*} = \begin{cases} \dfrac{\Gamma}{2}\left[\dfrac{4}{\pi\alpha^3R^3} + \dfrac{\alpha R}{\omega}\left(\dfrac{\omega\pi}{\sinh^2(\omega\pi)} - \coth(\omega\pi)\right)\right] & \text{for } \alpha R \geq 1. \\ \dfrac{\Gamma}{2}\left[\dfrac{4}{\pi\alpha^3R^3} - \dfrac{\alpha R}{\omega}\left(\dfrac{\omega\pi}{\sin^2(\omega\pi)} - \cot(\omega\pi)\right)\right] & \text{for } \alpha R < 1. \end{cases} \tag{16}$$

This is Eq. (3.10) in Ref. 54, but note that there is a sign error in the expression for $\alpha R < 1$. Lastly, the pair function $w_\alpha^{\mathcal{S}^3}(\psi_{ij})$ appearing in Eq. (15) is given by

$$w_\alpha^{\mathcal{S}^3}(\psi_{ij}) = \begin{cases} \dfrac{\pi\alpha R}{\omega}\dfrac{\cosh(\omega\pi)\sinh(\omega(\pi-\psi))}{\sin(\psi)\sinh^2(\omega\pi)} - \dfrac{\alpha R}{\omega}\dfrac{(\pi-\psi)\cosh(\omega(\pi-\psi))}{\sin(\psi)\sinh(\omega\pi)} - \dfrac{4}{\pi\alpha^3R^3} & \text{for } \alpha R \geq 1, \\ -\dfrac{\pi\alpha R}{\omega}\dfrac{\cos(\omega\pi)\sin(\omega(\pi-\psi))}{\sin(\psi)\sin^2(\omega\pi)} + \dfrac{\alpha R}{\omega}\dfrac{(\pi-\psi)\cos(\omega(\pi-\psi))}{\sin(\psi)\sin(\omega\pi)} - \dfrac{4}{\pi\alpha^3R^3} & \text{for } \alpha R < 1. \end{cases} \tag{17}$$

This is Eq. (3.9) in Ref. 54, but note that the sign is reversed for the first two terms in the expression for $\alpha R < 1$ and that the argument $\omega\pi$ is misprinted as $\omega\psi$ in the denominator of the second term in both cases.

Caillol and Gilles[54] have systematically studied this model for a large range of the coupling parameter $0.10 \leq \Gamma \leq 100$ and for a reduced screening parameter $0.10 \leq \alpha^* \leq 6.0$. Here, we employ the same range for α^* and concentrate on $\Gamma = 0.10$ and $\Gamma = 10$. Following Ref. 54, our system contains $N = 600$ particles in all simulations. For each combination of Γ and α^*, we equilibrate the system for 10^4 sweeps, followed by 10^6 production sweeps. A sweep consists of N Monte Carlo steps and in each step a displacement attempt is made for one randomly selected particle. Thus, each production run corresponds to 6×10^8 Monte Carlo steps. For $\Gamma = 0.10$ the acceptance rate depends only very weakly on the maximum displacement and was always larger than 90% in our simulations. For $\Gamma = 10$, the maximum displacement was adjusted to yield acceptance rates ~30% for the smallest values of the screening parameter and ~80% for the largest values of α^*. In each simulation, we sample the free-energy derivatives Eqs. (13) and (14). To provide an impression of the functional dependence of these derivatives on the screening parameter and to make a comparison to the results of Ref. 54, we plot $\partial f/\partial\alpha^*$ for $\Gamma = 0.10$ in Fig. 4. As shown, there is a systematic difference between the results of Ref. 54 and our data. We emphasize that results obtained by means of the h-GCA agree within statistical error with our data obtained via Metropolis-type simulations for all α^*. Moreover, the discrepancy cannot be attributed to the misprints identified in some of the equations in Ref. 54 for $\alpha R = \alpha^* R^* < 1$, since $R^* = 1/\psi_0 \approx 1/0.1993028778$, so all data points except $\alpha^* = 0.10$ and $\alpha^* = 0.15$ pertain to $\alpha R \geq 1$.

To extend this comparison to a significantly stronger coupling and to other thermodynamic properties, we display $\Gamma(\partial f/\partial\Gamma)$ as a function of α^* for $\Gamma = 10$ in Fig. 5. As shown, the relative discrepancy between the results of Ref. 54 and those of our Metropolis-type simulations is far smaller than for $\partial f/\partial\alpha^*$. However, in absolute magnitude the discrepancy is similar. The maximal difference between both data sets shifts to higher values of α^* than in Fig. 4. This is due to the increase in Γ, not to the difference in physical property; indeed, the maximum discrepancy in $\partial f/\partial\alpha^*$ (not shown) exhibits a similar shift to higher values of α^* when Γ is increased. The results obtained by means of simulations employing the h-GCA agree again with those from our Metropolis-type simulations within statistical uncertainty. For comparison purposes we provide the numerical data for both free-energy derivatives in tabulated form in Appendix C.

IV. EFFICIENCY OF THE h-GCA

A. Rod–sphere mixtures

The GCA defined for periodic systems in Euclidean geometry stands out because of its ability to accelerate simulations of size-asymmetric mixtures compared to conventional Monte Carlo simulations.[7] In this section, we demonstrate that the h-GCA is capable of similar efficiency gains. The standard test case for the standard GCA is a binary mixture of interacting spherical particles, with the two components denoted as "large" and "small." A large particle typically finds itself surrounded by many small particles, hindering its diffusion. Rapid positional decorrelation can be achieved in a Monte Carlo simulation through non-local moves, provided that these do not have a vanishingly small acceptance rate—precisely what the GCA guarantees. For the h-GCA, we

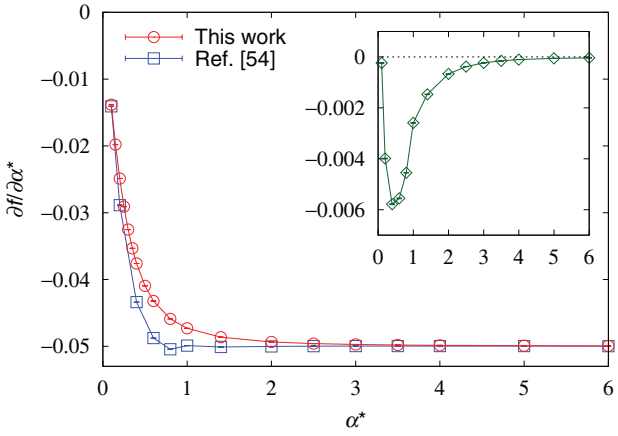

FIG. 4. Derivative of the free energy per particle with respect to the reduced screening parameter α^* for the Yukawa one-component plasma at coupling constant $\Gamma = 0.10$, as a function of α^*. The open circles are results obtained in this work via a Metropolis-type simulation for $N = 600$ particles. The open squares are data published in Ref. 54 for the identical system. Although the data agree for the unscreened case ($\alpha^* \to 0$) and for the strongly screened Yukawa one-component plasma, there is a systematic deviation at intermediate values of α^*, as highlighted in the inset (which displays $\partial f/\partial\alpha^*$ [Ref. 54] $-\partial f/\partial\alpha^*$[this work]). Independent simulations employing the hyperspherical geometric cluster algorithm agree with the Metropolis-type simulations of this paper (open circles) within statistical error. All error bars are significantly smaller than the symbol size.

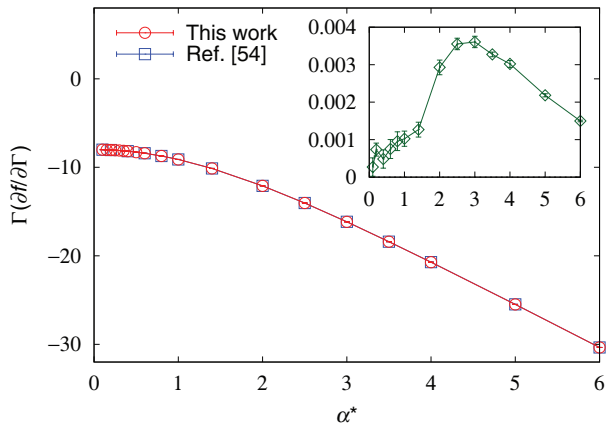

FIG. 5. Derivative of the free energy per particle with respect to the coupling constant Γ for the Yukawa one-component plasma at $\Gamma = 10$, as a function of the reduced screening parameter α^*. As in Fig. 4, the open circles are results obtained in this work via a Metropolis-type simulation for $N = 600$ particles and the open squares are data published in Ref. 54 for the identical system. Whereas the two data sets seemingly are in better agreement than for $\partial f/\partial\alpha^*$ at $\Gamma = 0.10$ (Fig. 4), the inset (displaying $\Gamma(\partial f/\partial\Gamma)$ [Ref. 54] $-\Gamma(\partial f/\partial\Gamma)$[this work]) shows that the discrepancy is of similar magnitude and is largest for intermediate values of α^*, although the position of the maximum discrepancy has shifted to a larger α^* with increasing Γ. Like for $\Gamma = 0.10$, independent simulations employing the hyperspherical geometric cluster algorithm agree with the Metropolis-type simulations of this paper within statistical error. All error bars in the main figure are significantly smaller than the symbol size. The error bars in the inset reflect the combined statistical uncertainty.

must demonstrate that rapid orientational decorrelation can be achieved as well. Therefore, we consider a mixture of small spherical particles and large rod-like particles. The rigid rods are composed of l spherical monomers of diameter σ stacked linearly with a center-to-center spacing σ between neighboring monomers (Fig. 6). The small spheres have diameter σ'. The small spheres and the monomers forming the rods all have hard-sphere interactions. Both the aspect ratio l of the rod and the diameter ratio $\alpha \equiv \sigma/\sigma'$ of the monomers and the small spheres affect the decorrelation time, so we will study the role of both parameters in our comparison of h-GCA and Metropolis-type simulations.

In the cluster algorithm, we employ reflections in randomly oriented planes. Each cluster move starts with a rod—this is possible without loss of ergodicity since the small spheres will relax due to overlaps with rods. In the local-move algorithm (conventional Monte Carlo), either a rod or a small sphere is selected for a trial move with 50% probability. With increasing density the maximum translations and rod rotations are tuned such that the order-parameter autocorrelation time (defined below) is minimized.

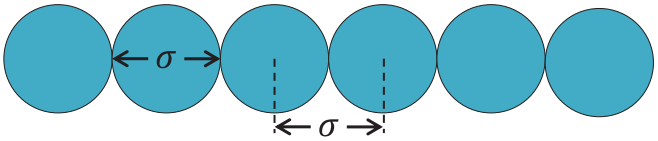

FIG. 6. Schematic illustration of the arrangement of spherical monomers into a rigid rod, as employed in the efficiency tests of the hyperspherical geometric cluster algorithm.

To quantify the computational effort required to simulate each system, we characterize the arrangement of the rods through an order parameter based on the second Legendre polynomial,

$$\frac{1}{N(N-1)}\sum_i\sum_{j\neq i}\left(\frac{3}{2}\left[\mathbf{s}_i\cdot\mathbf{s}_j-\frac{(\mathbf{s}_i\cdot\mathbf{r}_j)(\mathbf{s}_j\cdot\mathbf{r}_i)}{1+\mathbf{r}_i\cdot\mathbf{r}_j}\right]^2-\frac{1}{2}\right), \tag{18}$$

where $\mathbf{r}_i$ is the four-dimensional unit vector on the surface of the hypersphere denoting the position of particle i and $\mathbf{s}_i$ is the four-dimensional vector tangential to the surface of the hypersphere serving as the director of rod i; the sums run over all rods. This is very similar to the nematic order parameter,[55] except that the relative orientations of the rods are evaluated pairwise, rather than being compared to a global director. Furthermore, the second term within the square brackets is a geometric correction that accounts for the fact that the directors $\mathbf{s}_i$ and $\mathbf{s}_j$ belong to different tangent spaces for $\mathbf{r}_i \neq \mathbf{r}_j$, due to the curvature of the hyperspherical surface. Inclusion of this term, which is derived in Appendix A 5, is equivalent to evaluating the dot product of the two directors after they are brought together without rotation along their common geodesic. The autocorrelation time of this parameter, expressed in CPU time, is a measure for the effort required to generate independent configurations.

First, we vary the diameter ratio α by systematically reducing the size of the small spheres while keeping their volume fraction constant. The system contains 100 rods, each 10 monomers long, at a volume fraction of 5%. Computing the volume of a monomer via Eqs. (5) and (6) with $\psi_0 = \sigma/(2R)$ and the volume of the hypersphere surface via Eq. (2), we fix the radius of the hypersphere at 8.0932σ. In successive simulations, the small-sphere diameter σ' is decreased such that the diameter ratio α varies from 1 to 9. At the same time, the number of small spheres is increased as α^3 from 1000 to 729 000 to maintain their volume fraction at 5% as well.

Figure 7 displays the autocorrelation time[8,56] of the order parameter Eq. (18) as a function of α. Clearly, the CPU time per sample increases with α owing to the increasing number of particles. However, for simulations that employ the Metropolis Monte Carlo algorithm the autocorrelation time increases even faster, as it becomes more difficult to move the rods embedded in a finely dispersed environment of small particles. Indeed, the simulations become prohibitively slow even for moderate α. This contrasts starkly with the efficiency of the h-GCA, for which the order-parameter autocorrelation time increases only slowly with α. As for the GCA,[7] this efficiency difference results from the ability of the algorithm to realize non-local displacements without rejection, along with the preferential updating of those small particles that influence the large species. As shown in Fig. 7, at $\alpha = 1$, the h-GCA is $\mathcal{O}(100)$ faster than the conventional simulations; for $\alpha = 2$ this speed-up has already increased to nearly a factor $\mathcal{O}(10^4)$.

Next, we examine the effect of rod aspect ratio on the efficiency of both conventional Monte Carlo and h-GCA simulations. In these tests, we vary the rod length from two to ten monomers while keeping the size of the monomers and the

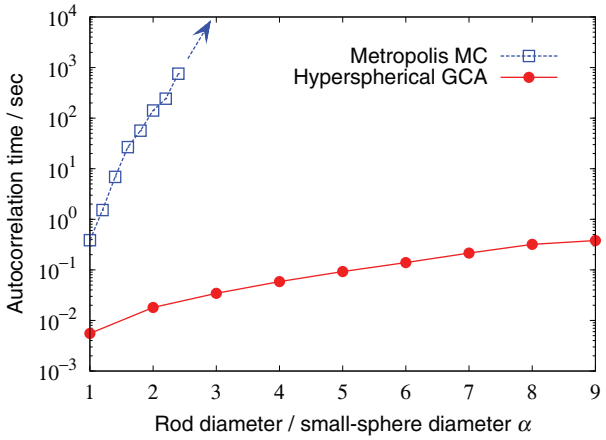

FIG. 7. Efficiency comparison between conventional Metropolis Monte Carlo simulations and the hyperspherical GCA, for a mixture of rods and spherical particles, as a function of rod–sphere size asymmetry. The rods have length 10σ and diameter σ. The spherical particles have diameter σ/α. Both species are present at a volume fraction of 5%. The figure displays the autocorrelation time of the order parameter Eq. (18), describing the arrangement of the rods, as a function of the diameter ratio α. As the small particles are decreased in size (i.e., α is increased), their number is increased to keep their volume fraction constant. For conventional simulations, the autocorrelation time increases more rapidly than the number of particles in the system as the rods become trapped in an environment of finely dispersed small spheres. The h-GCA does not suffer from this slowdown and is capable of decorrelating systems with diameter ratios as large as $\alpha = 9$, i.e., with small-sphere diameter 90 times smaller than the length of the rods. Even at a modest diameter ratio of $\alpha = 2$ the h-GCA produces decorrelated samples nearly 8000 times more efficiently than conventional Monte Carlo simulations.

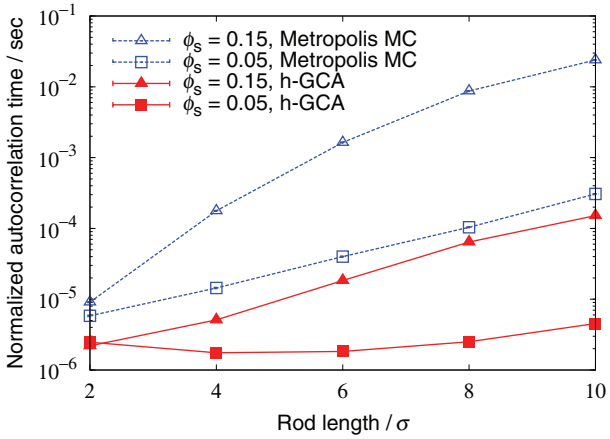

FIG. 8. Efficiency comparison between conventional Metropolis Monte Carlo simulations and the hyperspherical GCA, for a mixture of rods and spherical particles, as a function of rod aspect ratio. The spherical particles as well as the rods have diameter σ. Both species are present at a constant volume fraction of 5%. The autocorrelation time of the order parameter Eq. (18) is displayed as a function of rod length, for two different volume fractions ϕ_s of spherical particles. To facilitate comparison (at fixed rod length, and for a given algorithm) of the two volume fractions, the autocorrelation time is normalized by the number of spherical particles (960 at $\phi_s = 0.05$ and 2 880 at $\phi_s = 0.15$). At all rod lengths and for both volume fractions the h-GCA outperforms conventional Monte Carlo simulations by up to two orders of magnitude. For additional discussion see the text.

spherical particles constant. To maintain a constant rod volume faction, their number is adjusted such that all systems contain a total of 960 rod monomers. The hypersphere radius is set to 7.9838σ to achieve a volume fraction of 5%. The small spheres have a diameter equal to that of the monomers, $\sigma' = \sigma$, and we consider two different small-sphere volume fractions $\phi_s = 0.05$ and 0.15, corresponding to systems with 960 and 2880 spherical particles, respectively. In the conventional simulations we choose the maximum translation and rotation to minimize the autocorrelation time; this ranges from a maximum translation of σ and a uniformly chosen random orientation for the shortest rods at the lower volume fraction to a maximum translation of 0.2σ and a maximum rotation of 2° for the longest rods at the higher volume fraction.

We find that the h-GCA performs more efficiently than conventional Monte Carlo for both volume fractions and for all rod lengths within the range considered. At the lower volume fraction of small spheres, $\phi_s = 0.05$, the autocorrelation time of the order parameter increases approximately exponentially with rod length for conventional Monte Carlo simulations (Fig. 8). This mirrors the corresponding decrease in the observed acceptance rate. On the other hand, the h-GCA simulations exhibit almost no decrease in efficiency with increasing rod aspect ratio. The autocorrelation time even decreases as the length of the rods is increased from 2σ to 6σ; this results from variation in average cluster composition. At the higher sphere volume fraction, $\phi_s = 0.15$, the simulation time per sample is longer for both algorithms due in part to the larger number of small particles affected by each rod displacement. To deemphasize this effect, Fig. 8 displays the autocorrelation time normalized by the number of small spheres. Thus, for each rod length the difference between the results for $\phi_s = 0.05$ and $\phi_s = 0.15$ (comparing results obtained with the same algorithm) is due to the increased obstruction of rods as they find themselves in an environment containing more spheres. Although the h-GCA simulations remain considerably more efficient than Metropolis Monte Carlo even for the higher small-sphere volume fraction, they do exhibit an exponential increase of the autocorrelation time with increasing rod length. This results from the fact that this system approaches the percolation threshold, where most clusters contain nearly all the rods in the system; for rod length 10σ and $\phi_s = 0.15$, on average 97% of the rods (and 56% of the small spheres) belong to the cluster.

B. One-component systems

Besides size-asymmetric mixtures, the h-GCA also can be more efficient for certain one-component systems. To illustrate this, we simulate monodisperse hard rods on a spherical surface using the 2D h-GCA with unbiased reflections in lines passing through the center of the sphere (which are sufficient to relax the orientational degrees of freedom, owing to the Gaussian curvature of the surface). Each rod is composed of circular monomers, assembled as in Fig. 6. Their total area fraction is varied from 0.05 to 0.30 and we consider rod lengths ranging from 2σ to 21σ. The number of rods is varied to keep the total number of monomers 2520. To monitor the simulation efficiency, we measure the autocorrelation time of the nematic order parameter Eq. (18). For comparison purposes, we simulate the same systems using local moves, employing 13 different combinations of maximum displacement and maximum rotation. For each rod length and volume

fraction, we choose the shortest autocorrelation time of the 13 runs.

In the simulations with local moves, the autocorrelation time increases monotonically and near-exponentially with both rod length and area fraction (leveling off for systems with both long rods and high area fraction). When cluster moves are used, the autocorrelation time is approximately constant as a function of concentration for systems of short rods and as a function of rod length for low concentrations. However, when a certain threshold area fraction is reached (near 0.30 for rod length 4σ, decreasing to area fraction 0.10 for rod length 21σ) the autocorrelation time starts to increase steeply. This threshold correlates with an average cluster size of $\sim$10% of all rods. Since the cluster size distribution is bimodal, the threshold likely signals the appearance of very large clusters that contain almost all rods. Comparison of local moves and cluster moves shows that their efficiency is comparable below this threshold, as the typical cluster size is small. However, near the threshold, where clusters take an intermediate size, the h-GCA is considerably more efficient, with autocorrelation times that are at least 20 times shorter than in the conventional Monte Carlo algorithm. At higher area fractions and rod lengths, this advantage disappears until local moves have a twice shorter autocorrelation time for rods $\geq 10\sigma$ at area fraction 0.30.

Similarly, the h-GCA can also increase the efficiency of simulations of aggregating particles. Whereas conventional Monte Carlo algorithms move one particle at a time, the h-GCA proposes non-local moves that enable large-scale restructuring of aggregates.[57] For completeness, we note that since the introduction of the GCA,[7] related algorithms have been proposed for systems of attractive particles undergoing self-assembly. In these, the notion of constructing correlated collective moves put forward in the GCA is combined with a Metropolis-style acceptance criterion.[20–23] Specifically, clusters of particles are built according to a weaker interaction than the real interaction between the particles. The proposed cluster move is then subject to an acceptance criterion to account for the energy change that is ignored during the cluster construction. This approach accelerates the rearrangement of aggregates of strongly interacting particles, despite the inclusion of rejection. Furthermore, once the rejection-free notion is dismissed, it is no longer necessary to employ transformations that are self-inverse. This has the advantage that internal degrees of freedom of an aggregate can be relaxed by, e.g., shifting a subset of the aggregate over a small distance.

V. EFFECTS OF HYPERSPHERICAL GEOMETRY

Having established in Sec. IV that the h-GCA is more efficient than conventional methods for a variety of systems, we address its requirement that the simulations be conducted in hyperspherical geometry. The intrinsic curvature of this geometry introduces finite-size effects that must be compared to those encountered in simulations carried out in Euclidean geometry.

Simulations performed using either periodic boundary conditions or hyperspherical boundary conditions (HBC) display ensemble-dependent finite-size effects. In addition, the use of PBC leads to finite-size effects arising from the anisotropy introduced by the periodic lattice of cells, whereas the use of HBC leads to finite-size effects owing to the intrinsic curvature of the space. To illustrate the latter, we consider the second virial coefficient. Calculation of this coefficient involves a volume integration. The difference between Euclidean geometry and hyperspherical geometry for this integration is borne out by the difference in the surface area of a spherical shell of radius r. On a hypersphere of radius R, this shell has a surface area

$$4\pi R^2 \sin^2(r/R) \approx 4\pi r^2 \left(1 - \frac{r^2}{3R^2}\right). \tag{19}$$

Thus, the first-order correction to Euclidean geometry scales as $\mathcal{O}(R^{-2})$. This causes the second virial coefficient to exhibit a corresponding system-size dependent deviation from its value in Euclidean geometry. Indeed, it was derived for a Lennard-Jones fluid of N particles that, at constant density, this correction scales with system size as $R^{-2} \propto N^{-2/3}$, with similar corrections for other thermodynamic properties, including the internal energy per particle, the pressure, and the constant-volume heat capacity.[52]

To illustrate these finite-size effects, we simulate a Lennard-Jones fluid (particle diameter σ) with a potential cutoff at 2.5σ, at a coupling strength $\varepsilon = (2/3)k_B T$ and a number density $\rho = 0.2\sigma^{-3}$. We vary the number of particles from $N = 50$ to $N = 102\,400$ and perform the simulations under PBC as well as under HBC. For PBC, the average internal energy per particle $U(N)$ for $N > 100$ is well described by $U(N) = U_0 + aN^x$ with scaling exponent $x = -1.002 \pm 0.006$ and $U_0 = -0.79849$ (1). On the other hand, for HBC we find $U_0 = -0.79847$ (2) and $x = -0.686 \pm 0.001$. While observing the agreement in the thermodynamic limit for both geometries, we also note that the scaling exponent for HBC is likely an effective exponent caused by the relatively close vicinity of several finite-size corrections. Indeed, it is expected that[52]

$$U(N) = U_0 + U_1 N^x + U_2 N^{-1} + U_3 N^{2x} + \cdots, \tag{20}$$

with $x = -2/3$. In a nonlinear least-squares fit of all HBC data in which the exponent of the regular scaling correction is kept fixed at -1 (to avoid interference with the other corrections) we find $U_0 = -0.79846$ (2), consistent with the fits above containing only a single correction term, and $x = -0.676 \pm 0.006$, within two standard deviations of the predicted power-law. The approach to the thermodynamic limit, for both PBC and HBC, is illustrated in Fig. 9. Whereas the internal energy approaches the thermodynamic limit more slowly for HBC than for PBC, the differences are small for structural properties such as the radial distribution function, even for the smallest system, see Fig. 10. This makes simulations employing HBC equivalent to those employing PBC when the local structure of a fluid state must be determined.[38]

Finite-size effects in hyperspherical geometry have also been reported for other systems. In Ref. 58, the internal energy of the Yukawa one-component plasma of Sec. III D was found to exhibit corrections proportional to $N^{-2/3}$ for coupling strengths $\Gamma \leq 3$, whereas for stronger couplings the leading correction was found to scale as N^{-1}. On the other hand, for the restricted primitive model in hyperspherical geometry the

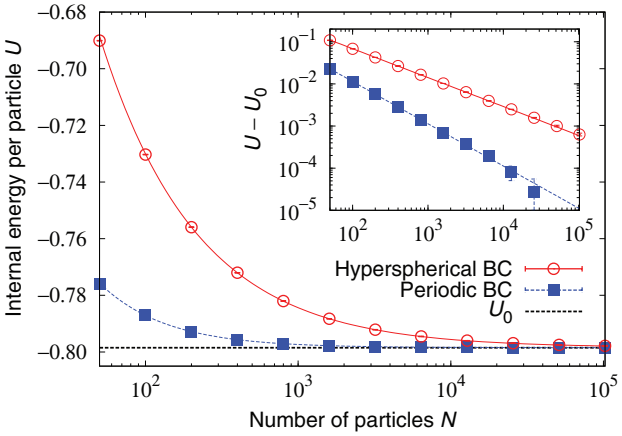

FIG. 9. Finite-size behavior of the internal energy per particle for a Lennard-Jones fluid in Euclidean geometry with periodic boundary conditions and in hyperspherical geometry. Both curves approach the same thermodynamic limit U_0, with power-law finite-size corrections (cf. inset) that agree with theoretical predictions (see text for further discussion).

internal energy per particle was reported to exhibit no significant finite-size effects, even for moderate system sizes.[47,59]

One concern that arises in the use of hyperspherical geometry is the formation of defects. Indeed, for spherical particles with a soft, purely repulsive potential, a comparison of PBC and HBC confirmed that HBC artificially inhibit crystallization, whereas PBC artificially promote crystallization due to the periodicity of the space.[60] Likewise, spherical boundary conditions can affect liquid-crystalline order by inducing topological defects, as was observed for 2D simulations on a sphere.[61] However, whereas such defects are guaranteed to exist in S_2 by the Poincaré–Hopf theorem, the hypersphere surface S_3 is parallelizable,[62] which implies that global defect-free nematic order can be achieved. Nevertheless, it remains an open question whether the strain generated by the intrinsic curvature of this geometry affects the formation of nematic order and to what extent this is influenced by the choice of projection, as discussed in Appendix B 2.

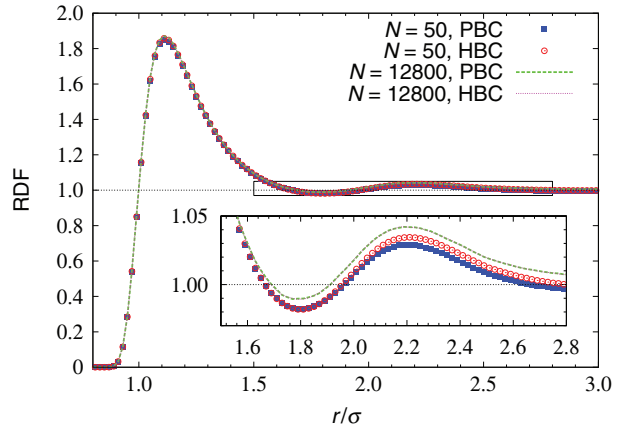

FIG. 10. Comparison of the radial distribution function for a Lennard-Jones fluid in periodic Euclidean geometry (PBC) and in hyperspherical geometry (HBC). Even for the smallest system size ($N = 50$), the structural differences are very small, demonstrating that the local structure is not strongly affected by curvature effects. For the larger system ($N = 12\,800$) the differences are undetectable. Interestingly, at larger separations the system in Euclidean space exhibits stronger finite-size deviations (see inset), which we ascribe to the anisotropic nature of the boundary conditions.

VI. CONCLUSIONS

In summary, we have presented an extension of the geometric cluster algorithm[7] to hyperspherical geometry. In contrast with simulations performed under periodic boundary conditions, this isotropic space permits the use of geometric transformations in the cluster construction process that relax all degrees of freedom of anisotropic particles. Thus, an ergodic, rejection-free Monte Carlo scheme of general applicability is obtained. As first illustrated for tetrapodal particles,[38] this hyperspherical geometric cluster algorithm accelerates the simulation of size-asymmetric mixtures of anisotropic particles by several orders of magnitude compared to conventional Metropolis Monte Carlo or molecular dynamics simulations. We have demonstrated the validity of this scheme for a broad range of models, including dimers in two- and three-dimensional space, the Lennard-Jones fluid, and the one-component Yukawa plasma. For the latter two systems, we have provided benchmark results against which future implementations can be compared. Furthermore, we have discussed a variety of implementation details, ranging from the use of the cell index method in hyperspherical geometry to the use of the quaternion notation to realize geometric transformations. The efficiency of this algorithm has been examined for rod–sphere mixtures, both as a function of rod aspect ratio and as a function of rod–sphere diameter ratio. In addition, we have demonstrated that even for one-component systems, the new algorithm can yield significant efficiency improvements. Lastly, for those situations where use of periodic boundary conditions in Euclidean geometry is preferred, we have derived a variant of the original geometric cluster algorithm that is no longer rejection-free but is capable of relaxing orientational degrees of freedom.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. DMR-1006430 and DGE-0948017.

APPENDIX A: TRANSFORMATIONS USING QUATERNIONS

1. Coordinate representation

A complete specification of the position and orientation of a particle on a hypersphere requires six degrees of freedom. These can be represented using six Euler angles, three for position and three for orientation. However, the need to perform transformations in the GCA makes the use of Euler angles cumbersome, since arbitrary rotations are complex to compute in this representation. As an alternative, one can employ a four-dimensional Cartesian coordinate system to represent each position on the hypersphere by a unit four-vector. The orientation of a particle is represented by a set of three mutually orthogonal unit four-vectors, each tangent to the

hypersphere (and thus orthogonal to the position vector). This set of four mutually orthogonal vectors, taken as a set of column vectors, becomes a 4×4 orthogonal matrix, which is the rotation matrix to transform the particle from a standard position and orientation, described by the identity matrix, to the current position and orientation. Thus, the position and orientation of a particle are represented by a rotation matrix and transformation of a particle is accomplished simply through matrix multiplication.

However, it is even more advantageous to represent transformations using quaternions.[63–65] First, rounding errors in the simulations cause the rotation matrices to lose their orthogonality, which introduces dilation and shear into the transformation, whereas quaternion transformations can only produce dilation, and cannot produce shear. Thus, while rotation matrices must be orthonormalized to prevent distortion due to numerical drift, quaternions need only to be normalized, which is simpler and computationally less costly. Secondly, it is easier to construct arbitrary transformations in the quaternion representation. In Secs. A 3 and A 4 of this Appendix we describe the construction of several important classes of transformations. In our implementation of the h-GCA, we use transformations in the quaternion representation to describe the position and orientation of each particle. When the position vector or any directors are needed during the simulation, they are computed from the quaternions. We emphasize that the use of quaternions in this work is different from the conventional use of quaternions in molecular dynamics simulation,[66–68] where each particle is paired with a quaternion of unit length to represent its orientation. We associate *two* unit quaternions with each particle to represent the six degrees of freedom of position and orientation. A pair of quaternions provides a natural representation of a four-dimensional transformation that defines the position and orientation of a particle relative to a standard position and orientation. In this appendix, we explain the construction and use of quaternion transformations in the h-GCA.

2. Quaternion properties

To describe geometric transformations using quaternions in a hyperspherical geometry, we first recall some general properties of quaternions. Quaternions are four-component vectors composed of a scalar part and a three-component vector part,[65,69–71]

$$\mathsf{q} \equiv [q_0, \mathbf{q}], \tag{A1}$$

with the following properties:
Addition,

$$\mathsf{p} + \mathsf{q} \equiv [p_0 + q_0, \mathbf{p} + \mathbf{q}]; \tag{A2}$$

multiplication,

$$\mathsf{pq} \equiv [p_0 q_0 - \mathbf{p} \cdot \mathbf{q},\ p_0 \mathbf{q} + q_0 \mathbf{p} + \mathbf{p} \times \mathbf{q}]; \tag{A3}$$

scalar multiplication,

$$s\mathsf{q} \equiv [s q_0, s\mathbf{q}]; \tag{A4}$$

conjugation,

$$\mathsf{q}^* \equiv [q_0, -\mathbf{q}]; \tag{A5}$$

dot product,

$$\mathsf{p} \cdot \mathsf{q} = p_0 q_0 + \mathbf{p} \cdot \mathbf{q}; \tag{A6}$$

and norm,

$$|\mathsf{q}| = \sqrt{\mathsf{q} \cdot \mathsf{q}}. \tag{A7}$$

Multiplication is non-commutative due to the cross product in Eq. (A3). Consequently, conjugation distributes over multiplication while reversing the order,

$$(\mathsf{pq})^* = \mathsf{q}^* \mathsf{p}^*. \tag{A8}$$

In this work, all quaternions will be unit quaternions, so that quaternion conjugation is equivalent to quaternion inversion, $\mathsf{q}^* = \mathsf{q}^{-1}$, where the latter form is our preferred notation.

3. Quaternion transformations in 3D

First we review transformations on a 2-sphere, which are equivalent to rotations and reflections of a three-dimensional object. Thus, a single quaternion is used to describe the position and orientation of a particle on a sphere or, equivalently, to describe the three orientational degrees of freedom of a 3D object. In this section, we adopt the latter perspective. The well-known quaternion formula for a rotation of a three-dimensional vector $\mathbf{v}$ by ϕ radians about an axis $\hat{\mathbf{a}}$ is[65,69–71]

$$\mathrm{Rot}_{\mathsf{q}}(\mathbf{v}) = \mathsf{q}\,[0, \mathbf{v}]\,\mathsf{q}^{-1}, \quad \text{for } \mathsf{q} = \left[\cos\left(\frac{\phi}{2}\right), \sin\left(\frac{\phi}{2}\right)\hat{\mathbf{a}}\right]. \tag{A9}$$

This equation is the general form for proper rotations, which can be extended to improper rotations by including a minus sign.[70] A special case of improper rotations is a pure reflection. Reflection in a plane containing the origin and with normal $\hat{\mathbf{n}}$ is expressed via

$$\mathrm{Ref}_{\mathsf{q}}(\mathbf{v}) = -\mathsf{q}\,[0, \mathbf{v}]\,\mathsf{q}^{-1}, \quad \text{for } \mathsf{q} = [0, \hat{\mathbf{n}}]. \tag{A10}$$

Thus, we can represent a general transformation in three dimensions as

$$\mathrm{Trans}_{\mathsf{q}}(\mathbf{v}) = \hat{q}\,\mathsf{q}\,[0, \mathbf{v}]\,\mathsf{q}^{-1}, \tag{A11}$$

where $\hat{q}$ is a unit scalar that is $+1$ for a proper rotation and -1 for an improper rotation (including reflections). Any possible value for the quaternion q, provided $|\mathsf{q}| = 1$, represents a valid transformation. When transformations are composed, the quaternions of the transformations multiply and the unit scalars multiply as well,

$$\mathrm{Trans}_{\mathsf{g}} = \mathrm{Trans}_{\mathsf{p}}\left(\mathrm{Trans}_{\mathsf{q}}(\mathbf{v})\right) = \hat{p}\hat{q}\,\mathsf{pq}\,[0, \mathbf{v}]\,\mathsf{q}^{-1}\mathsf{p}^{-1}$$
$$\Longrightarrow \hat{g} = \hat{p}\hat{q},\ \mathsf{g} = \mathsf{pq}. \tag{A12}$$

To illustrate this formula, we consider two reflections in planes with normals $\hat{\mathbf{n}}_1$ and $\hat{\mathbf{n}}_2$, respectively. According to Eq. (A12) the composite transformation,

$$\mathrm{Trans}_{\mathsf{g}}(\mathbf{v}) = \mathrm{Ref}_{\mathsf{p}=[0,\hat{\mathbf{n}}_2]}(\mathrm{Ref}_{\mathsf{q}=[0,\hat{\mathbf{n}}_1]}(\mathbf{v})), \tag{A13}$$

yields

$$\hat{g} = \hat{p}\hat{q} = (-1)(-1) = +1, \tag{A14}$$

$$\begin{aligned} \mathsf{g} &= \mathsf{pq} = [0, \hat{\mathbf{n}}_2]\,[0, \hat{\mathbf{n}}_1] \\ &= [-\hat{\mathbf{n}}_2 \cdot \hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2 \times \hat{\mathbf{n}}_1] = [-\cos\theta, -(\hat{\mathbf{n}}_1 \times \hat{\mathbf{n}}_2)\sin\theta], \end{aligned} \tag{A15}$$

where θ is the angle between the two normals. Since quaternion transformation is invariant under the change $\mathsf{g} \mapsto -\mathsf{g}$, comparison of Eq. (A15) to Eq. (A9) shows that the composite operation is indeed a rotation about the line of intersection of the two reflection planes by an angle twice the angle between the planes.

In simulations, the quaternion q and the unit scalar $\hat{q}$ together encode the orientation of a 3D object relative to a standard orientation,[72] arbitrarily defined. The current value of any vector attribute of an object, such as the position of each component part relative to the center of mass, is computed using Eq. (A11) to transform the vector from its value in standard orientation to the current orientation.

4. Quaternion transformations in 4D

Transformations on a 4D hypersphere are equivalent to rotations and reflections of an object in four dimensions. There are six degrees of freedom, analogous to the three translational and three rotational degrees of freedom in three-dimensional Euclidean space. Each transformation requires two quaternions, q_L and q_R. These are used in a 4D rotation formula similar to Eq. (A9),

$$\mathrm{Rot}_\mathsf{q}(\mathsf{v}) = \mathsf{q}_\mathrm{L}\mathsf{v}\mathsf{q}_\mathrm{R}, \tag{A16}$$

with the difference that the quaternions multiplying on the left and right are independent. The quaternion v to be transformed is identified with a Euclidean unit four-vector $\hat{\mathbf{v}}$ of the same components, which can represent either the 4D spatial position normalized by the radius of the hypersphere, $\mathbf{r}/R$, or one of the directors of the particle. To extend Eq. (A16) to include improper rotations, an operation must be performed that inverts an odd number of the spatial degrees of freedom. This is accomplished using the conjugation operation on v, which for notational clarity we denote as inversion, since the two are equivalent for the unit quaternions considered here. The general formula for quaternion transformations including all rotations and reflections is[70]

$$\mathrm{Trans}_\mathsf{q}(\mathsf{v}) = \mathsf{q}_\mathrm{L}\mathsf{v}^{\hat{q}}\mathsf{q}_\mathrm{R}, \tag{A17}$$

where we use a unit scalar $\hat{q} = \pm 1$ to indicate the absence or presence of quaternion conjugation.

To represent the position and orientation of a particle in the simulation, each particle is associated with a quaternion transformation, which encodes the current position (and orientation) of the particle relative to a standard position (and orientation), analogous to the treatment of particle orientation in 3D. Any four-dimensional vector attribute of a particle, such as the position vector or a director, has a specified value in standard position. To calculate one of these vectors in the current position, we apply the transformation associated with the particle using Eq. (A17), where the vector to be transformed is identified with the quaternion v of the same components. The definition of the standard position is arbitrary, but it is most convenient to assign a reference point near the center of the particle with a position vector identified with the scalar quaternion 1, because it is easy to construct specific transformations around this point.

A second application of transformations is the construction of cluster moves. When a transformation $\mathrm{Trans}_\mathsf{p}$ (quaternions p_L and p_R) associated with a cluster move is applied to a particle whose position and orientation are defined by the transformation $\mathrm{Trans}_\mathsf{q}$, the new, composed transformation $\mathrm{Trans}_{\mathsf{q}'}$ (quaternions q'_L and q'_R) associated with the particle is

$$\mathrm{Trans}_{\mathsf{q}'}(\mathsf{v}) = \mathrm{Trans}_\mathsf{p}(\mathrm{Trans}_\mathsf{q}(\mathsf{v})) = \mathsf{p}_\mathrm{L}(\mathsf{q}_\mathrm{L}\mathsf{v}^{\hat{q}}\mathsf{q}_\mathrm{R})^{\hat{p}}\mathsf{p}_\mathrm{R}. \tag{A18}$$

Depending on the sign of $\hat{p}$, there are two different forms for $\mathrm{Trans}_{\mathsf{q}'}$,

$$\mathsf{q}'_\mathrm{L} = \mathsf{p}_\mathrm{L}\mathsf{q}_\mathrm{L}, \quad \mathsf{q}'_\mathrm{R} = \mathsf{q}_\mathrm{R}\mathsf{p}_\mathrm{R}, \quad \hat{q}' = \hat{q}, \quad \text{for } \hat{p} = +1, \tag{A19}$$

$$\mathsf{q}'_\mathrm{L} = \mathsf{p}_\mathrm{L}\mathsf{q}_\mathrm{R}^{-1}, \quad \mathsf{q}'_\mathrm{R} = \mathsf{q}_\mathrm{L}^{-1}\mathsf{p}_\mathrm{R}, \quad \hat{q}' = -\hat{q}, \quad \text{for } \hat{p} = -1. \tag{A20}$$

Table V lists the constraints needed to construct the self-inverse transformations on a hypersphere that are the analogues of the three classes of transformations in 3D Euclidean space.

Specific transformations are constructed according to this table as follows. For a plane reflection, the quaternion p is the normal of the hyperplane in 4D space. For a line reflection, the constraint derives from a more general formula, where the quaternions p_L and p_R are constructed as

$$\mathsf{p}_\mathrm{L} = \mathsf{b}\mathsf{a}^{-1}, \quad \mathsf{p}_\mathrm{R} = \mathsf{a}^{-1}\mathsf{b}. \tag{A21}$$

Here a and b span a 2D vector space that rotates while the components orthogonal to it remain invariant. This formula derives from two successive plane reflections, using quaternions a and b, respectively. The rotation takes place in the direction from a to b over twice the angle between these two quaternions. The analogue of a line reflection in 3D is generated by a rotation angle π, which requires $\mathsf{a} \cdot \mathsf{b} = 0$, constraining p_L and p_R in Table V to have zero scalar part. For a point reflection in a pivot p on the hypersphere, a reflection is performed in the line that passes through the center of the hypersphere and through p.

If the quaternions are chosen uniformly within the constraints, the resulting transformation will be uniformly

TABLE V. Construction of hyperspherical analogues of the 3D Euclidean self-inverse transformations employed in the GCA.

3D transformation	Constraints			Degrees of freedom
Plane reflection	$\hat{p} = -1$	$\mathsf{p}_\mathrm{L} = \mathsf{p}$	$\mathsf{p}_\mathrm{R} = -\mathsf{p}$	3
Line reflection	$\hat{p} = +1$	$\mathsf{p}_\mathrm{L} = [0, \hat{\mathbf{p}}_\mathrm{L}]$	$\mathsf{p}_\mathrm{R} = [0, \hat{\mathbf{p}}_\mathrm{R}]$	4
Point reflection	$\hat{p} = -1$	$\mathsf{p}_\mathrm{L} = \mathsf{p}$	$\mathsf{p}_\mathrm{R} = \mathsf{p}$	3

random, generating an unbiased cluster move. To perform a biased cluster move, which aids in controlling cluster size,[8] a transformation is constructed that leads to a small displacement of the particle that acts as the cluster seed. For this, it is convenient to construct the biased transformation with reference to a particle in standard position, since simple formulas exist for pure rotation and pure translation of a particle from this position. A rotation of a particle in the standard position over ϕ radians about the axis $\hat{\mathbf{a}}$ is constructed via

$$\begin{aligned} &\hat{p} = +1, \quad \mathsf{p}_{\mathrm{L}} = \mathsf{p}, \quad \mathsf{p}_{\mathrm{R}} = \mathsf{p}^{-1}, \\ &\text{for } \mathsf{p} = \left[\cos\left(\tfrac{\phi}{2}\right), \sin\left(\tfrac{\phi}{2}\right)\hat{\mathbf{a}}\right], \end{aligned} \tag{A22}$$

identical to the formula for the rotation of a three-dimensional object, Eq. (A9). A pure translation from the standard position over θ radians in the direction of $\hat{\mathbf{b}}$ without rotation (parallel transport along a geodesic) is constructed via

$$\begin{aligned} &\hat{p} = +1, \quad \mathsf{p}_{\mathrm{L}} = \mathsf{p}, \quad \mathsf{p}_{\mathrm{R}} = \mathsf{p}, \\ &\text{for } \mathsf{p} = \left[\cos\left(\tfrac{\theta}{2}\right), \sin\left(\tfrac{\theta}{2}\right)\hat{\mathbf{b}}\right], \end{aligned} \tag{A23}$$

Employing these expressions along with those of Table V, we construct a self-inverse transformation $\mathrm{Trans}_{\mathsf{p}}$ that displaces a particle from the standard position by a controlled amount. This transformation must be centered around the actual position of the chosen particle. The resulting transformation $\mathrm{Trans}_{\mathsf{p}'}$ is obtained by first transforming the particle back to standard position using $\mathrm{Trans}_{\mathsf{q}}^{-1}$, followed by the cluster transformation $\mathrm{Trans}_{\mathsf{p}}$, and subsequent reversal of the first transformation via $\mathrm{Trans}_{\mathsf{q}}$,

$$\begin{aligned} \mathrm{Trans}_{\mathsf{p}'}(\mathsf{v}) &= \mathrm{Trans}_{\mathsf{q}}\left[\mathrm{Trans}_{\mathsf{p}}\left(\mathrm{Trans}_{\mathsf{q}}^{-1}(\mathsf{v})\right)\right] \\ &= \mathsf{q}_{\mathrm{L}}\left[\mathsf{p}_{\mathrm{L}}\left(\mathsf{q}_{\mathrm{L}}^{-1}\mathsf{v}^{\hat{q}}\mathsf{q}_{\mathrm{R}}^{-1}\right)^{\hat{p}}\mathsf{p}_{\mathrm{R}}\right]^{\hat{q}}\mathsf{q}_{\mathrm{R}}. \end{aligned} \tag{A24}$$

As this composite transformation will be applied to all particles in the cluster, it is computationally more efficient to precompute the associated unit scalar $\hat{p}'$ and the quaternions p'_{L} and p'_{R}. Whereas the scalar remains unaltered, $\hat{p}' = \hat{p}$, there are four separate cases for the quaternions,

$$\mathsf{p}'_{\mathrm{L}} = \mathsf{q}_{\mathrm{L}}\mathsf{p}_{\mathrm{L}}\mathsf{q}_{\mathrm{L}}^{-1}, \quad \mathsf{p}'_{\mathrm{R}} = \mathsf{q}_{\mathrm{R}}^{-1}\mathsf{p}_{\mathrm{R}}\mathsf{q}_{\mathrm{R}}, \quad \text{for } \hat{p} = +1, \quad \hat{q} = +1, \tag{A25}$$

$$\mathsf{p}'_{\mathrm{L}} = \mathsf{q}_{\mathrm{L}}\mathsf{p}_{\mathrm{R}}^{-1}\mathsf{q}_{\mathrm{L}}^{-1}, \quad \mathsf{p}'_{\mathrm{R}} = \mathsf{q}_{\mathrm{R}}^{-1}\mathsf{p}_{\mathrm{L}}^{-1}\mathsf{q}_{\mathrm{R}}, \quad \text{for } \hat{p} = -1, \quad \hat{q} = +1, \tag{A26}$$

$$\mathsf{p}'_{\mathrm{L}} = \mathsf{q}_{\mathrm{L}}\mathsf{p}_{\mathrm{L}}\mathsf{q}_{\mathrm{R}}, \quad \mathsf{p}'_{\mathrm{R}} = \mathsf{q}_{\mathrm{L}}\mathsf{p}_{\mathrm{R}}\mathsf{q}_{\mathrm{R}}, \quad \text{for } \hat{p} = +1, \quad \hat{q} = -1, \tag{A27}$$

$$\mathsf{p}'_{\mathrm{L}} = \mathsf{q}_{\mathrm{L}}\mathsf{p}_{\mathrm{R}}^{-1}\mathsf{q}_{\mathrm{R}}, \quad \mathsf{p}'_{\mathrm{R}} = \mathsf{q}_{\mathrm{L}}\mathsf{p}_{\mathrm{L}}^{-1}\mathsf{q}_{\mathrm{R}}, \quad \text{for } \hat{p} = -1, \quad \hat{q} = -1. \tag{A28}$$

5. Derivation of the order parameter curvature correction

The order parameter in Eq. (18) is similar to the nematic order parameter. The two differences are that the orientation of the director is evaluated pairwise, rather than compared with the global average director, and that the dot product of the directors is corrected by a term which takes into account the curvature of the hyperspherical space. Here, we derive this correction term.

For two particles i and j at positions r_i and r_j ($\mathsf{r}_i \neq \mathsf{r}_j$), it is generally incorrect to take the dot product of the directors s_i and s_j as a measure of their alignment, as these directors lie in different tangent spaces. This can easily be seen as follows. Note that each director s_i must be orthogonal to the position r_i. If s_i has a component in the direction of r_j, i.e., $\mathsf{s}_i \cdot \mathsf{r}_j \neq 0$, then $\mathsf{s}_i \cdot \mathsf{s}_j$ will always be less than one, irrespective of the orientation of particle j, because s_j cannot have a component in the direction of r_j. Thus, the dot product $\mathsf{s}_i \cdot \mathsf{s}_j$ will tend to underestimate their degree of alignment.

To recover the familiar behavior of the dot product in Euclidean space, we bring the two particles to the same location without rotation prior to taking the dot product. To do so, we construct a transformation that brings particle i to the location of particle j along their common geodesic by performing two plane reflections. This rotates (cf. Eq. (A21)) the position of the particle over twice the angle between the normals of the reflection planes without changing its orientation, i.e., so-called parallel transport. We first reflect through the normal r_i and then through the normal $\mathsf{r}_{\mathrm{m}} \equiv (\mathsf{r}_j + \mathsf{r}_i)/|\mathsf{r}_j + \mathsf{r}_i|$, which is the midpoint on the hypersphere between the two particles, so that r_i will be rotated to r_j. The transformation of the director s_i of particle i has the following form:

$$\mathsf{s}'_i = \mathsf{r}_{\mathrm{m}}\left(\mathsf{r}_i\mathsf{s}_i^{-1}\mathsf{r}_i\right)^{-1}\mathsf{r}_{\mathrm{m}} = \mathsf{r}_{\mathrm{m}}\mathsf{r}_i^{-1}\mathsf{s}_i\mathsf{r}_i^{-1}\mathsf{r}_{\mathrm{m}}. \tag{A29}$$

To simplify this, we use an alternative form of the dot product,

$$2(\mathsf{a}\cdot\mathsf{b}) = \mathsf{a}\mathsf{b}^{-1} + \mathsf{b}\mathsf{a}^{-1}, \tag{A30}$$

which can be rearranged as

$$\mathsf{a}\mathsf{b}^{-1} = -\mathsf{b}\mathsf{a}^{-1} + 2(\mathsf{a}\cdot\mathsf{b}). \tag{A31}$$

Applying Eq. (A31) twice and using the fact that the director of a particle is orthogonal to the position vector, $\mathsf{r}_i \cdot \mathsf{s}_i = 0$, we find

$$\mathsf{s}'_i = \mathsf{s}_i - 2(\mathsf{s}_i\cdot\mathsf{r}_{\mathrm{m}})\mathsf{r}_{\mathrm{m}}. \tag{A32}$$

Thus, the dot product of s'_i and s_j becomes

$$\mathsf{s}'_i\cdot\mathsf{s}_j = \mathsf{s}_i\cdot\mathsf{s}_j - 2(\mathsf{s}_i\cdot\mathsf{r}_{\mathrm{m}})(\mathsf{s}_j\cdot\mathsf{r}_{\mathrm{m}}). \tag{A33}$$

Substitution of $\mathsf{r}_{\mathrm{m}} = (\mathsf{r}_i + \mathsf{r}_j)/\sqrt{2(1+\mathsf{r}_i\cdot\mathsf{r}_j)}$ yields

$$\begin{aligned} \mathsf{s}'_i\cdot\mathsf{s}_j &= \mathsf{s}_i\cdot\mathsf{s}_j - \frac{[\mathsf{s}_i\cdot(\mathsf{r}_i+\mathsf{r}_j)][\mathsf{s}_j\cdot(\mathsf{r}_i+\mathsf{r}_j)]}{1+\mathsf{r}_i\cdot\mathsf{r}_j} \\ &= \mathsf{s}_i\cdot\mathsf{s}_j - \frac{(\mathsf{s}_i\cdot\mathsf{r}_j)(\mathsf{s}_j\cdot\mathsf{r}_i)}{1+\mathsf{r}_i\cdot\mathsf{r}_j}. \end{aligned} \tag{A34}$$

When the quaternions are identified with four-dimensional vectors this results in the term within square brackets in Eq. (18).

APPENDIX B: IMPLEMENTATION DETAILS

1. Cell index method

For each particle that is added to a cluster, a pairwise potential requires $\mathcal{O}(N)$ operations in a simple implementation that considers the interaction between this particle and all other particles in the system. For a finite-range potential, this can be improved to $\mathcal{O}(1)$ operations if one only considers the interactions with neighboring particles that lie within the cutoff of the pair interaction. A common method to accomplish this is the cell index method.[68,73] In this method, the simulation cell is divided into subcells with a side length larger than the cutoff radius, so that all particles that interact with a given particle will be found located in the subcell containing this particle or in the subcells immediately bordering this subcell. To efficiently scan over the particles that are located in these subcells, all particles in the system are indexed in memory according to the cell to which they belong.

Implementation of this method in hyperspherical space presents some challenges, since the intrinsic curvature permits no convenient tiling of space into cells. Indeed, any tiling of the hypersphere surface would be so complicated as to require significant computation to determine which cell a particle belongs to. On the other hand, this mapping is trivial for simulations taking place in 3D Euclidean space. If a cubic simulation cell of dimensions $L \times L \times L$ is divided into M^3 cubic subcells of linear size $\ell = L/M$, then the cell indices of a particles with coordinates (x, y, z) are $(n_x, n_y, n_z) = (\lfloor x/\ell \rfloor, \lfloor y/\ell \rfloor, \lfloor z/\ell \rfloor)$, where each index is an integer $0 \le n_i < M$. Therefore, we opt to define a system of hypercubical subcells by tiling the 4D Euclidean space containing the hyperspherical system. The subcell indices (n_x, n_y, n_z, n_t) for each particle are then found analogously from its 4D position vector. In terms of memory requirements, this approach becomes progressively less efficient at larger system sizes because only $\mathcal{O}(R^3)$ of all $\mathcal{O}(R^4)$ subcells will intersect with the surface of a hypersphere of radius R, whereas all other subcells will never contain any particles (cf. Fig. 11). To address this, we only store those subcells that intersect with the hypersphere surface, i.e., we use a linear array containing only the shaded cells in Fig. 11. This, in turn, requires a mapping of the cell indices onto the linear array of non-consecutive subcells. We devise this mapping by noting that each row of subcells only contains up to two contiguous blocks that intersect with the hypersphere surface, so that the elimination of non-intersecting cells amounts to the reduction of each row to these contiguous blocks. We precompute a look-up table containing the indices of the first and last subcells of these blocks, for all $\mathcal{O}(R^3)$ rows. During the simulation, this look-up table then directly permits determining the memory address of a given cell subcell. This approach is nearly as fast as a direct look-up employing a 4D array of subcells, but with a memory requirement that scales only as $\mathcal{O}(R^3)$.

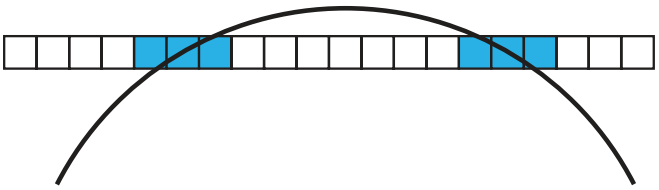

FIG. 11. Lower-dimensional representation of the intersection of the hypersphere surface (represented by arc) and the subcells defined in Euclidean space. The subcells that intersect the hypersphere are shaded.

2. Molecules

Rather than having explicitly anisotropic interactions or shape, anisotropic particles can be composed of rigidly bonded, isotropic monomers. Here, we discuss two topics arising in the implementation of such molecular particles, namely their construction in hyperspherical geometry and their treatment in the cluster construction process.

If a molecule conceived in Euclidean space is projected onto hyperspherical space, the intrinsic curvature of the latter leads to a distortion of the molecular shape. The degree of distortion is inversely related to the system size, which makes the shape of the molecule system-size-dependent. Various projection schemes exist, each of which preserve some molecular properties while abandoning others. This is analogous to the choices encountered in the creation of geographic maps.

As a default scheme, we implement a projection that preserves the distance and direction of each monomer from a reference point (e.g., its center of mass) on the molecule. To construct a molecule in hyperspherical space, each monomer is positioned using a pure translation from this reference point, cf. Eq. (A23). If we consider a molecule in 3D Euclidean space with its reference point at the origin and place the reference point at $(0, 0, 0, R)$ on a hypersphere of radius R, the position (x', y', z', t') of a monomer is obtained from its position (x, y, z) in 3D Euclidean space as

$$x' = R\frac{x}{r}\sin\left(\frac{r}{R}\right), \tag{B1}$$

$$y' = R\frac{y}{r}\sin\left(\frac{r}{R}\right), \tag{B2}$$

$$z' = R\frac{z}{r}\sin\left(\frac{r}{R}\right), \tag{B3}$$

$$t' = R\cos\left(\frac{r}{R}\right), \tag{B4}$$

where $r = \sqrt{x^2 + y^2 + z^2}$. This projection keeps radial arrangements of monomers straight. However, straight lines not passing through the reference point will experience a concave outward distortion. This will affect, for example, the sides of a cylindrical particle. If the center of such a particle is chosen as the reference point, the concave distortion will prevent two parallel cylinders from meeting flush. This will affect their interaction, e.g., in the case of a depletion attraction induced by small depletants. It is possible to resolve this problem by adopting a projection that makes the sides of the cylinder geodesic, but this would result in a nonuniform cylinder diameter. In most cases, the level of distortion is small enough to render differences between projections negligible already for moderately large systems, but it may be important in systems

that are particularly sensitive to particle alignment or when high accuracy is desired.

Another issue is how to handle molecules in cluster moves. One option is to consider each monomer separately. When a given monomer joins the cluster, all other monomers belonging to the same molecule immediately join the cluster as well. This approach is easy to implement, since each monomer is treated as an individual particle bonded infinitely strongly to the other monomers in the molecule, and it reduces the number of pair interactions that need to be considered. However, an alternative is to consider the change in *molecular* pairwise energy when deciding whether a molecule should join the cluster. This has the advantage that the energy change between one pair of monomers residing on molecules i and j, respectively, may be offset by the energy change between another pair of monomers residing on the same molecules. Thus, the total energy change in Eq. (1) is decreased and the probability that the second molecule is added to the cluster is decreased. This is often a desirable situation, as it lowers the average cluster size. The situation is illustrated in Fig. 12.

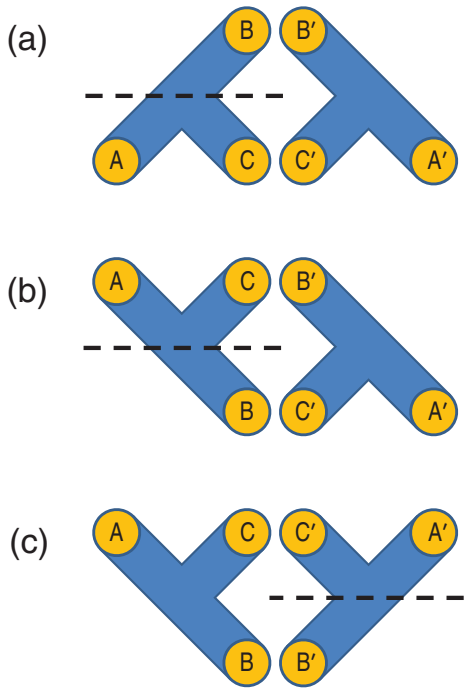

FIG. 12. Illustration of how, for compound particles, the use of molecular pairwise interactions in Eq. (1) can lead to a lower cluster addition probability than if each pair of interacting monomers is treated individually. Panel (a) shows two y-shaped molecules with a short-range attraction between the orange monomers. The cluster is initiated with the molecule on the left-hand side, which is reflected in the horizontal dashed line to produce the configuration in panel (b). Since the total interaction energy in panels (a) and (b) is identical, there would be no energy change if the molecule on the right-hand side were reflected as well. Thus, panel (b) represents the final situation, provided that the total molecular pair interaction is considered. However, the situation is different in a simpler implementation that considers each monomer separately. In that case, the monomer pair BB' is broken when going from (a) to (b) and this would lead to a finite probability that the particle on the right would be reflected as well, cf. panel (c). Both simulation schemes will produce configurations that follow the Boltzmann distribution, but the first scheme will lead to more rapid decorrelation.

TABLE VI. Numerical results for the free-energy derivatives $\Gamma(\partial f/\partial\Gamma)$ and $\partial f/\partial\alpha^*$ (Eqs. (13) and (14)) as a function of the screening parameter α^*, for coupling constant $\Gamma = 0.10$. The system size is $N = 600$.

α^*	$\Gamma(\partial f/\partial\Gamma)$	$(\partial f/\partial\alpha^*)$
0.10	-0.0254561 ± 0.0000050	-0.0138426 ± 0.0000033
0.15	-0.0260191 ± 0.0000049	-0.0198040 ± 0.0000044
0.20	-0.0268027 ± 0.0000049	-0.0248773 ± 0.0000052
0.25	-0.0277778 ± 0.0000048	-0.0290937 ± 0.0000057
0.30	-0.0289369 ± 0.0000047	-0.0325264 ± 0.0000059
0.35	-0.0302379 ± 0.0000046	-0.0353325 ± 0.0000059
0.40	-0.0316915 ± 0.0000045	-0.0376018 ± 0.0000057
0.50	-0.0349404 ± 0.0000043	-0.0409557 ± 0.0000052
0.60	-0.0385668 ± 0.0000040	-0.0432181 ± 0.0000046
0.80	-0.0466020 ± 0.0000036	-0.0458943 ± 0.0000035
1.00	-0.0553006 ± 0.0000032	-0.0473101 ± 0.0000026
1.40	-0.0737204 ± 0.0000027	-0.0486326 ± 0.0000017
2.00	-0.1025218 ± 0.0000021	-0.0493536 ± 0.0000010
2.50	-0.1269711 ± 0.0000018	-0.0495999 ± 0.0000007
3.00	-0.1516159 ± 0.0000016	-0.0497312 ± 0.0000005
3.50	-0.1763698 ± 0.0000015	-0.0498078 ± 0.0000004
4.00	-0.2011879 ± 0.0000013	-0.0498573 ± 0.0000003
5.00	-0.2509443 ± 0.0000011	-0.0499135 ± 0.0000002
6.00	-0.3007904 ± 0.0000010	-0.0499427 ± 0.0000002

APPENDIX C: TABULATED DATA FOR THE YUKAWA ONE-COMPONENT PLASMA

To provide a benchmark for future implementations of particle-based simulations employing hyperspherical boundary conditions, we provide here accurate results for the Yukawa one-component plasma described in Sec. III D. All data pertain to a system of $N = 600$ particles. The system is characterized by merely two independent parameters,[53,54] namely the coupling constant Γ and the reduced screening parameter α^*. Specifically, temperature and number density

TABLE VII. Numerical results for the free-energy derivatives $\Gamma(\partial f/\partial\Gamma)$ and $\partial f/\partial\alpha^*$ (Eqs. (13) and (14)) as a function of the screening parameter α^*, for coupling constant $\Gamma = 10$. The system size is $N = 600$.

α^*	$\Gamma(\partial f/\partial\Gamma)$	$(\partial f/\partial\alpha^*)$
0.10	-8.010389 ± 0.000087	-0.287230 ± 0.000010
0.15	-8.024719 ± 0.000086	-0.429104 ± 0.000015
0.20	-8.044996 ± 0.000085	-0.569195 ± 0.000019
0.30	-8.102299 ± 0.000085	-0.843174 ± 0.000026
0.35	-8.139407 ± 0.000084	-0.976795 ± 0.000029
0.40	-8.182353 ± 0.000082	-1.108102 ± 0.000032
0.50	-8.284705 ± 0.000081	-1.363449 ± 0.000037
0.60	-8.408731 ± 0.000079	-1.608943 ± 0.000041
0.80	-8.721279 ± 0.000073	-2.068120 ± 0.000045
1.00	-9.115195 ± 0.000068	-2.483051 ± 0.000048
1.40	-10.124408 ± 0.000057	-3.176550 ± 0.000046
2.00	-12.093377 ± 0.000044	-3.899275 ± 0.000038
2.50	-14.036783 ± 0.000030	-4.272682 ± 0.000024
3.00	-16.160547 ± 0.000022	-4.508193 ± 0.000017
3.50	-18.401794 ± 0.000018	-4.657289 ± 0.000013
4.00	-20.719180 ± 0.000016	-4.753759 ± 0.000010
5.00	-25.488143 ± 0.000012	-4.862051 ± 0.000007
6.00	-30.354569 ± 0.000009	-4.915599 ± 0.000005

are *not* additional control variables, but enter through Γ and α^*. The numerical data pertain to derivatives of the excess free energy per particle, Eqs. (13) and (14). The quantities are listed as a function of α^* for $\Gamma = 0.10$ (Table VI) and for $\Gamma = 10$ (Table VII).

[1]D. Frenkel and B. Smit, *Understanding Molecular Simulation*, 2nd ed. (Academic, San Diego, 2002).
[2]M. N. Rosenbluth and A. W. Rosenbluth, J. Chem. Phys. **23**, 356 (1955).
[3]J. I. Siepmann and D. Frenkel, Mol. Phys. **75**, 59 (1992).
[4]G. E. Norman and V. S. Filinov, High Temp. **7**, 216 (1969).
[5]A. Z. Panagiotopoulos, Mol. Phys. **61**, 813 (1987).
[6]E. Luijten and J. Liu, in *The Monte Carlo Method in the Physical Sciences*, AIP Conference Proceedings Vol. 690, edited by J. E. Gubernatis (American Institute of Physics, Melville, NY, 2003), pp. 225–231.
[7]J. Liu and E. Luijten, Phys. Rev. Lett. **92**, 035504 (2004).
[8]J. Liu and E. Luijten, Phys. Rev. E **71**, 066701 (2005).
[9]E. Luijten, Comput. Sci. Eng. **8**, 20 (2006).
[10]C. Day, Phys. Today **57**(3), 25 (2004).
[11]S. Asakura and F. Oosawa, J. Chem. Phys. **22**, 1255 (1954).
[12]C. Dress and W. Krauth, J. Phys. A **28**, L597 (1995).
[13]N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).
[14]J. Liu and E. Luijten, Phys. Rev. Lett. **93**, 247802 (2004).
[15]C. J. Martinez, J. Liu, S. K. Rhodes, E. Luijten, E. R. Weeks, and J. A. Lewis, Langmuir **21**, 9978 (2005).
[16]J. Liu and E. Luijten, Phys. Rev. E **72**, 061401 (2005).
[17]S. A. Barr and E. Luijten, Langmuir **22**, 7152 (2006).
[18]J. Liu, N. B. Wilding, and E. Luijten, Phys. Rev. Lett. **97**, 115705 (2006).
[19]D. J. Ashton, J. Liu, E. Luijten, and N. B. Wilding, J. Chem. Phys. **133**, 194102 (2010).
[20]C. H. Mak, J. Chem. Phys. **122**, 214110 (2005).
[21]A. Troisi, V. Wong, and M. A. Ratner, J. Chem. Phys. **122**, 024102 (2005).
[22]S. Whitelam and P. L. Geissler, J. Chem. Phys. **127**, 154101 (2007).
[23]A. Bhattacharyay and A. Troisi, Chem. Phys. Lett. **458**, 210 (2008).
[24]C. A. Hixson, J. P. Benigni, and D. J. Earl, Phys. Chem. Chem. Phys. **11**, 6335 (2009).
[25]J. Schwarz-Linek, A. Winkler, L. G. Wilson, N. T. Pham, T. Schilling, and W. C. K. Poon, Soft Matter **6**, 4540 (2010).
[26]A. Uhlherr and D. N. Theodorou, J. Chem. Phys. **125**, 084107 (2006).
[27]J. G. Malherbe and W. Krauth, Mol. Phys. **105**, 2393 (2007).
[28]W. Krauth, *Statistical Mechanics: Algorithms and Computations* (Oxford University Press, Oxford, 2006).
[29]E. P. Bernard, W. Krauth, and D. B. Wilson, Phys. Rev. E **80**, 056704 (2009).
[30]S. C. Glotzer and M. J. Solomon, Nature Mater. **6**, 557 (2007).
[31]L. Manna, E. C. Scher, and A. P. Alivisatos, J. Am. Chem. Soc. **122**, 12700 (2000).
[32]L. Manna, D. J. Milliron, A. Meisel, E. C. Scher, and A. P. Alivisatos, Nature Mater. **2**, 382 (2003).
[33]L. Hong, A. Cacciuto, E. Luijten, and S. Granick, Nano Lett. **6**, 2510 (2006).
[34]L. Hong, A. Cacciuto, E. Luijten, and S. Granick, Langmuir **24**, 621 (2008).
[35]S. Jiang, Q. Chen, M. Tripathy, E. Luijten, K. S. Schweizer, and S. Granick, Adv. Mater. **22**, 1060 (2010).
[36]Q. Chen, J. K. Whitmer, S. Jiang, S. C. Bae, E. Luijten, and S. Granick, Science **331**, 199 (2011).
[37]L. Helden, R. Roth, G. H. Koenderink, P. Leiderer, and C. Bechinger, Phys. Rev. Lett. **90**, 048301 (2003).
[38]D. W. Sinkovits and E. Luijten, Nano Lett. **12**(4) (2012).
[39]U. Wolff, Phys. Rev. Lett. **62**, 361 (1989).
[40]J. R. Heringa and H. W. J. Blöte, Physica A **251**, 224 (1998).
[41]J. R. Heringa and H. W. J. Blöte, Physica A **254**, 156 (1998).
[42]W. Krauth and R. Moessner, Phys. Rev. B **67**, 064503 (2003).
[43]S. Papanikolaou, E. Luijten, and E. Fradkin, Phys. Rev. B **76**, 134514 (2007).
[44]J. P. Hansen, D. Levesque, and J. J. Weis, Phys. Rev. Lett. **43**, 979 (1979).
[45]K. W. Kratky, J. Comp. Phys. **37**, 205 (1980).
[46]J. M. Caillol and D. Levesque, J. Chem. Phys. **94**, 597 (1991).
[47]S. Hanassab and T. J. VanderNoot, J. Electroanal. Chem. **528**, 135 (2002).
[48]P. J. Råsmark, T. Ekholm, and C. Elvingson, J. Chem. Phys. **122**, 184110 (2005).
[49]C. H. Mak, Phys. Rev. E **73**, 065104(R) (2006).
[50]S. Amokrane, A. Ayadim, and J. G. Malherbe, J. Chem. Phys. **123**, 174508 (2005).
[51]S. J. Plimpton, J. Comp. Phys. **117**, 1 (1995).
[52]W. Schreiner and K. W. Kratky, Mol. Phys. **50**, 435 (1983).
[53]J. M. Caillol and D. Gilles, J. Stat. Phys. **100**, 905 (2000).
[54]J. M. Caillol and D. Gilles, J. Stat. Phys. **100**, 933 (2000).
[55]P. J. Collings and M. Hird, *Introduction to Liquid Crystals: Chemistry and Physics* (Taylor & Francis, London, 1997).
[56]K. Binder and E. Luijten, Phys. Rep. **344**, 179 (2001).
[57]A. Winkler, D. W. Sinkovits, and E. Luijten, "Phase behavior of tetrapods" (unpublished).
[58]J. M. Caillol, J. Chem. Phys. **111**, 6538 (1999).
[59]S. Hanassab and T. J. VanderNoot, Mol. Simul. **29**, 527 (2003).
[60]P. Jund and R. Jullien, Europhys. Lett. **42**, 637 (1998).
[61]H. Shin, M. J. Bowick, and X. Xing, Phys. Rev. Lett. **101**, 037802 (2008).
[62]T. Frankel, *The Geometry of Physics* (Cambridge University Press, Cambridge, 1997).
[63]W. R. Hamilton, Proc. R. Ir. Acad. **2**, 424 (1844), see http://www.maths.tcd.ie/pub/HistMath/People/Hamilton/Quatern1.
[64]W. R. Hamilton, Proc. R. Ir. Acad. **3**, 1 (1847), see http://www.maths.tcd.ie/pub/HistMath/People/Hamilton/Quatern2.
[65]W. R. Hamilton, *Lectures on Quaternions* (Hodges and Smith, Dublin, 1853).
[66]D. J. Evans, Mol. Phys. **34**, 317 (1977).
[67]D. J. Evans and S. Murad, Mol. Phys. **34**, 327 (1977).
[68]M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon, Oxford, 1987).
[69]J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality* (Princeton University Press, Princeton, New Jersey, 1999).
[70]J. H. Conway and D. A. Smith, *On Quaternions and Octonions: Their Geometry, Arithmetic, and Symmetry* (A K Peters, Natlick, Massachusetts, 2003).
[71]A. J. Hanson, *Visualizing Quaternions* (Morgan Kaufman, San Francisco, 2006).
[72]C. F. F. Karney, J. Mol. Graphics Modell. **25**, 595 (2007).
[73]R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw-Hill, New York, 1981).