

# A simulation algorithm for Brownian dynamics on complex curved surfaces

Cite as: J. Chem. Phys. 151, 164901 (2019); doi: 10.1063/1.5126201

Submitted: 30 August 2019 • Accepted: 7 October 2019 •

Published Online: 25 October 2019



View Online



Export Citation



CrossMark

Yuguang Yang<sup>1,2</sup> and Bo Li<sup>1,a)</sup>

## AFFILIATIONS

<sup>1</sup> Institute of Biomechanics and Medical Engineering, Applied Mechanics Laboratory, Department of Engineering Mechanics, Tsinghua University, Beijing 100084, China

<sup>2</sup> Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, Maryland 21218, USA

<sup>a)</sup> Author to whom correspondence should be addressed: libome@tsinghua.edu.cn

## ABSTRACT

Brownian dynamics of colloidal particles on complex curved surfaces has found important applications in diverse physical, chemical, and biological processes. However, most Brownian dynamics simulation algorithms focus on relatively simple curved surfaces that can be analytically parameterized. In this work, we develop an algorithm to enable Brownian dynamics simulation on extremely complex curved surfaces. We approximate complex curved surfaces with triangle mesh surfaces and employ a novel scheme to perform particle simulation on these triangle mesh surfaces. Our algorithm computes forces and velocities of particles in global coordinates but updates their positions in local coordinates, which combines the strengths from both global and local simulation schemes. We benchmark the proposed algorithm with theory and then simulate Brownian dynamics of both single and multiple particles on torus and knot surfaces. The results show that our method captures well diffusion, transport, and crystallization of colloidal particles on complex surfaces with nontrivial topology. This study offers an efficient strategy for elucidating the impact of curvature, geometry, and topology on particle dynamics and microstructure formation in complex environments.

Published under license by AIP Publishing. <https://doi.org/10.1063/1.5126201>

## INTRODUCTION

Dynamics of microscale and nanoscale particles, cells, and proteins on curved surfaces plays a critical role in a broad range of physical, chemical, and biological processes. Common examples include colloidal particle assembly, packing, defect formation, and crystallization on a nonflat interface;<sup>1–7</sup> protein diffusion on curved membranes;<sup>8,9</sup> cell dynamics (e.g., differentiation and migration) on curved substrates;<sup>10–12</sup> and collective locomotion of active particles system on curved surfaces.<sup>13–16</sup> Theoretical approaches including particle simulations<sup>17–22</sup> and continuum descriptions<sup>23,24</sup> are constantly employed to understand these dynamical processes. Despite extensive experimental evidence of emerging novel complexity arising from curvature and topology of surfaces,<sup>5,7,14,15,25</sup> algorithms of particle simulation are largely limited to simple surfaces that are analytically tractable (e.g., spherical, cylindrical, and ellipsoidal surfaces, and other simple geometric primitives).<sup>1,5,17,26</sup> Establishing a particle-scale simulation algorithm able to tackle surfaces with

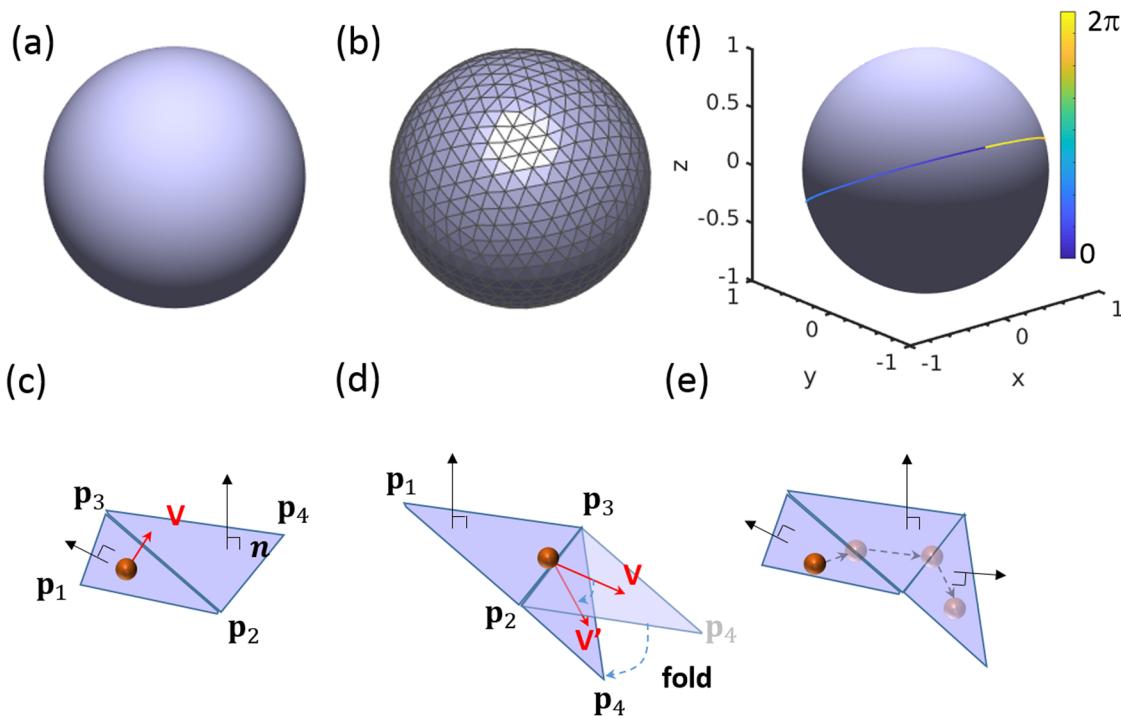
arbitrary complex geometry can significantly facilitate the understanding of how curvature and topology regulate individual or collective dynamics.

Brownian dynamics (BD) simulation, developed by Ermak and McCammon,<sup>27</sup> has been widely used for micro-sized particle simulation in the Cartesian space. BD simulation captures the hydrodynamics, Brownian forces, and particle interactions, and has become an indispensable tool to study the dynamical aspects of colloidal suspensions. Through decades of development, BD simulation has been harnessed to simulate dynamical processes beyond colloids, including cell migration,<sup>28</sup> tumor growth,<sup>29</sup> protein-protein interaction,<sup>18</sup> protein folding,<sup>30</sup> dynamics of polymer<sup>31</sup> and non-specific particles,<sup>32,33</sup> and colloidal hydrodynamics under confinement.<sup>34,35</sup> Considerable effort has been directed toward extending Brownian simulation and other molecular simulation methods to curved surfaces. Existing approaches can be classified into two categories: a global coordinate scheme and a local coordinate scheme.

As particles moving on a surface can be viewed as imposing a constraint on the positions, the global scheme originates from Lagrange classic mechanics<sup>36–38</sup> formulation under constraints. In the global scheme, calculation of forces and positions of the particles are carried out in the global coordinates, and the net effect of the constraints is equivalent to adding constraint forces perpendicular to the constraint surface to offset other forces driving particles out of the surfaces. The constraint force can be either exactly calculated from Lagrange multipliers or approximated by applying a restoring harmonic potential to confine the simulated particles on the surface. The resulting algorithms, commonly referred to as constrained BD, allow convenient calculations and analysis via global coordinates and reuse various computational techniques (e.g., parallel computation via domain decomposition) developed for simulations in Euclidean space. However, there are several limitations in the global scheme. First, the calculation of constraint forces requires analytically parameterizable surfaces, which are generally unavailable for many complex surfaces. Second, for highly curved surfaces, the update time step should be sufficiently small to ensure that constraint forces can stably maintain particles on

the surface and produce zero tangent velocity that would otherwise cause incorrect dynamics. As a comparison, the local scheme approach<sup>17,26</sup> directly evolves the particle dynamics in the local coordinates systems defined by the constraints. Using local coordinates system can guarantee that the constraints are always satisfied without imposing the constraints forces, thus reducing the complications in the computation of constrained forces and the choice of update time step size. The local scheme is generally more accurate and theoretically sound and is appropriate for the study of single particle dynamics on a simple, analytical, curved surface (e.g., spherical and cylindrical surfaces);<sup>26</sup> however, the local scheme will have significant difficulties for complex surfaces as their local coordinate system parametrizations are usually not available in these situations.

In this paper, we aim to develop a BD simulation algorithm that enables particle-scale simulation on complex curved surfaces beyond the analytical ones. We approximate a curved surface by triangle mesh surfaces and formulate the simulation algorithm with respect to them. Triangle mesh surfaces [Figs. 1(a) and 1(b)] have been widely used to approximate complex surfaces



**FIG. 1.** (a) A smooth spherical surface. (b) Approximation of a spherical surface by a triangle mesh surface. (c) The local coordinate system determined by the vertices of the triangle  $\mathbf{p}_1$ ,  $\mathbf{p}_2$ , and  $\mathbf{p}_3$ . A particle lying on the triangle surface with tangent velocity vector  $\mathbf{v}$ . (d) Illustration of the velocity folding. Consider a particle lying on the edge  $\mathbf{p}_2\mathbf{p}_3$  with tangent velocity  $\mathbf{v}$  and simultaneously lying on the surface  $(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4)$  with tangent velocity  $\mathbf{v}'$ . The tangent velocity  $\mathbf{v}'$  will be used to evolve particle position on the surface  $(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4)$ . The tangent speed  $\mathbf{v}'$  can be obtained by fixing the starting point of  $\mathbf{v}$  and then folding the face  $(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4')$  to the face  $(\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4)$ . (e) An example trajectory of a particle moving on a simple triangle mesh surface with its velocity folded twice. (f) A simple simulation experiment to demonstrate the validity of velocity folding. Consider a particle on a unit triangle mesh sphere (the mesh surface is sufficiently refined and its triangle faces have a length scale of  $\sim 0.018a$ ). We initialize a particle with a unit tangent velocity and use a large update time of  $2\pi$  to update its position for one step. We expect that after enough velocity foldings, the particle will move back to its original position. The simulated trajectory of the particle (colored by the passage of time) is shown on the sphere. The particle has an initial position of  $(-0.107\ 38, -0.875\ 985, 0.470\ 119)$  and a final position of  $(-0.103\ 186, -0.876\ 095, 0.470\ 885)$ , with the position difference within 0.01. There are in total 768 velocity foldings in this one update step.

(e.g., organs, bone, and other structures) in computer graphics, mechanics, medicine, and numerical methods like finite element method.<sup>23,39</sup> Therefore, a principled way to perform particle simulation on such triangle mesh surfaces can be used as an approximation to particle simulation on a wealth of smooth surfaces, ranging from flat surfaces to spherical surfaces, and to complex surfaces with nontrivial topology. Furthermore, the methodology to create and process triangle mesh surfaces are well established, including reconstructing complex surfaces from real-world 3D experiments<sup>40</sup> and improving the approximation accuracy by subdividing triangles.<sup>41</sup>

We develop a hybrid simulation scheme to combine strengths from both global and local schemes. Our algorithm first calculates forces and associated velocities using global coordinates, which is, in general, more convenient than using local coordinates; our algorithm then updates particles' positions in the local coordinates, which is more convenient since it avoids computing constraint forces and related complications mentioned earlier (i.e., small time step and nonzero tangent component). As we will show, the use of a triangle mesh surface enables straightforward conversion between global and local coordinates and thus the efficient implementation of the hybrid scheme.

This paper is organized as follows. We first review the basic geometry and coordinates transformations in triangle mesh surfaces. We then adapt the constrained BD theory to triangle mesh surfaces to establish the relationship between local velocities and global forces. We then introduce a “velocity folding” to update positions in the local coordinates in a principled manner. To verify our algorithm, we benchmark the algorithm with theory by simulating Brownian motion on a flat surface and a spherical surface. Subsequently, we demonstrate our algorithm by simulating single and multiple particle dynamics on complex surfaces with nontrivial topology. The effects of curvature and topology on BD are illustrated. Finally, we conclude the present paper and provide a discussion on the possible extensions based on our algorithm.

## THEORY AND ALGORITHM

### Coordinate systems on a triangle mesh

To facilitate our subsequent discussion, we start with a brief introduction to the coordinate systems on a triangle mesh surface. A triangle mesh surface [Fig. 1(b)] can be described by a vertex set consisting of vertices  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M \in \mathbb{R}^3$ , and a face set with each triangle face determined by three vertices ( $\mathbf{p}_i, \mathbf{p}_j$ , and  $\mathbf{p}_k$ ). The position of a particle lying on the face of the triangle surface can be described by both a global/lab coordinate  $\mathbf{r} = (r_1, r_2, r_3)^T$  and a local coordinate  $\mathbf{q} = (q_1, q_2)^T$  determined by the three vertices ( $\mathbf{p}_1, \mathbf{p}_2$ , and  $\mathbf{p}_3$ ) of the triangle face [Fig. 1(c)]. The local coordinate of every point in the interior of a triangle ( $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ ) is the barycentric coordinate given by

$$\mathbf{r} = [\mathbf{p}_2 - \mathbf{p}_1, \mathbf{p}_3 - \mathbf{p}_1] \cdot \mathbf{q} + \mathbf{p}_1, \quad (1)$$

with the constraints  $q_1 \geq 0$ ,  $q_2 \geq 0$ , and  $q_1 + q_2 \leq 1$ . From Eq. (1), the straightforward coordinate transformation from the local  $\mathbf{q}$  to the global  $\mathbf{r}$  is described by

$$\mathbf{r} = \mathbf{J} \cdot \mathbf{q} + \mathbf{p}_1, \mathbf{J} = [\mathbf{p}_2 - \mathbf{p}_1, \mathbf{p}_3 - \mathbf{p}_1], \quad (2)$$

where  $\mathbf{J} \in \mathbb{R}^{3 \times 2}$  is usually referred to as Jacobian matrix. Similarly, the transformation from the global  $\mathbf{r}$  to the local  $\mathbf{q}$  is given as

$$\mathbf{q} = \mathbf{J}^* \cdot (\mathbf{r} - \mathbf{p}_1), \mathbf{J}^* = (\mathbf{J}^T \cdot \mathbf{J})^{-1} \cdot \mathbf{J}^T, \quad (3)$$

where  $\mathbf{J}^* \in \mathbb{R}^{3 \times 2}$  is the pseudoinverse of the original Jacobian matrix  $\mathbf{J}$ .

The tangent velocity associated with the particle on a surface can also have a local description  $\mathbf{v}_q$  and a global description  $\mathbf{v}_r$ , with the transformation rule given by

$$\mathbf{v}_q = \mathbf{J}^* \cdot \mathbf{v}_r. \quad (4)$$

Equation (4) can be derived by taking time derivative on the two sides of Eq. (3) (note that  $\mathbf{p}_1$  is a constant vector).

### Constrained BD on a triangle mesh

Consider  $N$  Brownian particles that are constrained to move on a curved surface. We denote particle positions by vectors  $\mathbf{r}_i \in \mathbb{R}^3$  ( $i = 1, 2, \dots, N$ ) and capture the constraint on the particle positions by  $C(\mathbf{r}_i) = 0$ . The overdamped motion of particles under constraints is governed by<sup>31,33,37</sup>

$$\frac{d\mathbf{r}_i}{dt} = \beta \mathbf{D}_i \cdot (\mathbf{F}_i^P + \mathbf{F}_i^B + \mathbf{F}_i^C), \quad (5)$$

where  $\mathbf{F}_i^P \in \mathbb{R}^3$  denotes unconstrained deterministic forces (e.g., forces arising from particle-particle interaction, particle-substrate interaction, particle-field interaction, or active forces),  $\mathbf{F}_i^B \in \mathbb{R}^3$  denotes Brownian forces, and  $\mathbf{F}_i^C \in \mathbb{R}^3$  denotes constraint forces. The constraint forces are normal to the local tangent plane of the surface at  $\mathbf{r}_i$ ,<sup>37</sup> and they are introduced to ensure the resulting dynamics governed by Eq. (5) is satisfying the constraints  $C(\mathbf{r}_i) = 0$ .  $\beta = (kT)^{-1}$  is the inverse thermal energy, and  $\mathbf{D}_i$  is a  $3 \times 3$  diffusivity tensor, which can be used to capture additional physics, such as frictions due to particle-surface interaction and hydrodynamical coupling among particles.<sup>33,35,42,43</sup>

On the triangle mesh surface,  $\mathbf{F}_i^C$  is in the direction of  $\mathbf{n}_i$ , where  $\mathbf{n}_i$  is the normal of the triangle face containing the particle  $i$ . Requiring the constraint  $C(\mathbf{r}_i) = 0$  to hold all the time gives rise to

$$\frac{\partial C}{\partial t} = [\nabla_{\mathbf{r}_i} C]^T \cdot \frac{d\mathbf{r}_i}{dt} \propto \mathbf{n}_i^T \cdot \frac{d\mathbf{r}_i}{dt} = 0, \quad (6)$$

where we have used the fact that  $\mathbf{n}_i$  has the same direction as  $\nabla_{\mathbf{r}_i} C(\mathbf{r}_i)$  on the position  $\mathbf{r}_i$ . Plugging Eq. (5) into Eq. (6) and requiring  $\mathbf{F}_i^C$  to be collinear with  $\mathbf{n}_i$ , we can derive  $\mathbf{F}_i^C$  as

$$\mathbf{F}_i^C = -\mathbf{n}_i \cdot (\mathbf{n}_i^T \cdot \mathbf{D}_i \cdot \mathbf{n}_i)^{-1} \cdot \mathbf{n}_i^T \mathbf{D}_i \cdot (\mathbf{F}_i^P + \mathbf{F}_i^B). \quad (7)$$

The equation of motion given by Eq. (5) then reduces to

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{P}_i \cdot [\beta \mathbf{D}_i \cdot (\mathbf{F}_i^P + \mathbf{F}_i^B)], \quad (8)$$

where

$$\mathbf{P}_i = \mathbf{I} - \mathbf{D}_i \cdot \mathbf{n}_i (\mathbf{n}_i^T \cdot \mathbf{D}_i \cdot \mathbf{n}_i)^{-1} \mathbf{n}_i^T. \quad (9)$$

When the tensor  $\mathbf{D}_i$  is diagonal,  $\mathbf{P}_i$  reduces to  $\mathbf{P}_i = \mathbf{I} - \mathbf{n}_i \mathbf{n}_i^T$ , which is the familiar orthogonal projector onto the local tangent plane with the normal  $\mathbf{n}_i$ . Since our triangle mesh surface has zero curvature

everywhere except at edges (however, edges have zero measure), we do not need to consider the additional correction for the curvature effects of constraints.<sup>44</sup>

In constrained BD, the Brownian forces  $\mathbf{F}_i^B$  need to have zero components along the direction  $\mathbf{n}_i$ ,<sup>37</sup> which can be achieved by projecting the unconstrained Brownian force  $\mathbf{F}_i^{B'}$  onto the local tangent plane via

$$\mathbf{F}_i^B = (\mathbf{I} - \mathbf{n}_i \mathbf{n}_i^T) \cdot \mathbf{F}_i^{B'}, \quad (10)$$

where  $\mathbf{F}_i^{B'}$  has zero mean and variance given by

$$\left\langle \mathbf{F}_i^{B'}(t) \mathbf{F}_i^{B'}(t') \right\rangle = 2\beta^{-2} [\mathbf{D}_i]^{-1} \delta(t - t'), \quad (11)$$

with  $\delta$  being the Kronecker delta. It turns out our algorithm (Algorithm 1) has the most intuitive interpretation in terms of velocity. Rewrite the constrained dynamics of Eq. (8) as

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{P}_i \cdot \mathbf{v}_i^{uc}, \quad (12)$$

where  $\mathbf{v}_i^{uc} = \beta \mathbf{D}_i \cdot (\mathbf{F}_i^P + \mathbf{F}_i^B)$  is the velocity when there are no constraints on particles. Therefore, the effect of constraints is simply the projection of the original velocity onto the local tangent plane via projection operator  $\mathbf{P}_i$ .

The following algorithm (Algorithm 1) summarizes the key procedures to update particle positions on a triangle mesh at each time step. One critical procedure in updating local coordinates in step (3) is to handle the case that a particle is moving from one triangle to another triangle, as showed in Figs. 1(c) and 1(d). When the particle is moving to the edge between two triangles at  $\bar{t}$ , we fold the original tangent velocity  $\mathbf{v}$  to a new tangent velocity  $\mathbf{v}'$  [Fig. 1(d)], which has the same magnitude but different direction and continues the evolution of  $\mathbf{q}$  on the new triangle for the remaining time. We refer to such a method as “velocity folding” hereafter. The critical rationale behind this velocity folding is the fact that a particle on the surface under no force moves along geodesic according to Lagrange classic mechanics.<sup>36</sup> A geodesic is the shortest path between two points on the surface, which is the generalization of straight lines on a plane to shortest paths on surfaces. As we update  $\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \mathbf{v}_q \Delta t$ , we can view that this particle experiences zero force and moves at a constant velocity. When we fold its velocity as a particle transverses across different faces, we can ensure that the particle moves along a geodesic with a constant

speed on the triangle mesh surface during the whole interval  $\Delta t$ . We have verified this velocity folding in a simple numerical experiment [Fig. 1(f)]: we initialize a particle with unit tangent velocity on a triangle mesh surface approximating a spherical surface of unit radius; we use a large integration time step of  $2\pi$ , and after enough folding, the particle moves back to its original position with a negligible error.

The velocity folding requires determination of which edge has been hit by the moving particle and which face to the velocity is folded to. During one update step in  $\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \mathbf{v}_q \Delta t$ , the time and the edge that will be hit during the interval  $\Delta t$  can be determined by solving

$$t_1 = -\frac{q_2}{v_{q,2}}, \quad t_2 = \frac{(1 - q_1 - q_2)}{v_{q,1} + v_{q,2}}, \quad t_3 = -\frac{q_1}{v_{q,1}}. \quad (13)$$

Among  $t_1$ ,  $t_2$ , and  $t_3$ , if  $t_1$  is the minimum positive one, then the hitting edge is  $\mathbf{p}_1 \mathbf{p}_2$ ; if  $t_2$  is the minimum positive one, then the hitting edge is  $\mathbf{p}_2 \mathbf{p}_3$ ; and if  $t_3$  is the minimum positive one, then the hitting edge is  $\mathbf{p}_3 \mathbf{p}_1$ . Note that if all  $t_1$ ,  $t_2$ , and  $t_3$  are negative or if the minimal positive one among them is greater than the remaining time, then the particle will not move to a new triangle and no folding is needed.

After determining which face will be folded to, we can calculate the after-folding velocity by multiplying the before-folding velocity by a rotation matrix, since the velocity folding from one triangle to another one is equivalent to applying a rotation. The rotation matrix is defined by the rotation angle between the two surfaces and the rotational axis (which is the edge joining the two surfaces).

#### Remarks:

1. All matrices used for coordinate transformations and velocity folding can be precomputed and reused to accelerate the simulation.
2. There are a number of well-developed software to process mesh surfaces [e.g., MeshLab (<http://www.meshlab.net/>) and C++ library libigl (<https://libigl.github.io/>)] and to analyze dynamics and path on mesh surfaces (MATLAB). Particularly, MeshLab can be used to conveniently scale, deform, rotate, refine, and coarsen the mesh surface.
3. Our C++ implementation of this algorithm is also available (<https://github.com/yangyutu/BrownianMotionManifold.git>).

---

#### ALGORITHM 1. Key steps for updating particle positions (below, we leave out the subscript $i$ during each time step).

---

- (1) Calculate all unconstrained forces and the resulting velocity  $\mathbf{v}^{uc}$  [Eq. (12)] in global coordinate for each particle [note that Brownian forces need to be generated by Eq. (10)].
  - (2) Transform  $\mathbf{v}^{uc}$  to tangent velocity  $\mathbf{v}_r$  via projection  $\mathbf{v}_r = \mathbf{P} \cdot \mathbf{v}^{uc}$  and then get the local description  $\mathbf{v}_q$  via transformation  $\mathbf{v}_q = \mathbf{J}^* \cdot \mathbf{v}_r$ .
  - (3) Update local coordinates using local tangent velocity  $\mathbf{v}_q$  and Euler-Maruyama scheme<sup>45</sup> via  $\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \mathbf{v}_q \Delta t$ .
  - (4) Transform updated local coordinates  $\mathbf{q}(t + \Delta t)$  back to global coordinates  $\mathbf{r}(t + \Delta t)$  via Eq. (2).
-

## METHODS

In all numerical experiments, we use particle radius  $a = 1000$  nm, absolute temperature  $T = 293$  K, and integration time step  $\Delta t = 0.0001$  s. For simplicity, we ignore hydrodynamics and other particle–surface interactions in our simulation and set diffusivity  $D = 2.145 a^2/\text{s}$ . The characteristic time scale associated with diffusion is  $\tau = a^2/D \approx 0.47$  s. All length in this work is measured in the scale of radius  $a$ . In simulation of multiple particles, we introduce electrostatic repulsive interactions and depletion attraction between particles, which are given by<sup>46</sup>

$$\mathbf{F}_i = -\sum_{j \neq i} \nabla u_{ij}(r_{ij}), \quad (14)$$

$$u_{ij}(r_{ij}) = B \exp[-\kappa(r_{ij} - 2a)] + \Delta\Pi V_{\text{ex}}(r_{ij}),$$

where we use electrostatic prefactor  $B = 2290 kT$  and Debye length  $\kappa^{-1} = 20$  nm,  $r_{ij}$  is particle pair separation, and  $V_{\text{ex}}$  is the excluded volume between spheres,<sup>47</sup> which is given by

$$V_{\text{ex}}(r_{ij}) = \frac{4\pi}{3} (a+L)^3 \left[ 1 - \frac{3}{4} \left( \frac{r_{ij}}{a+L} \right) + \frac{1}{16} \left( \frac{r_{ij}}{a+L} \right)^3 \right]. \quad (15)$$

$\Delta\Pi$  is the osmotic pressure difference between the bulk and excluded volume region. We use  $\Delta\Pi = 5.8 \times 10^{-6} kT/\text{nm}^3$  and  $L = 0.2a$  such that there exist  $\sim 5kT$  pair attractions between particles.

In the simulation, we store each particle's global coordinate, local coordinate, and the triangle face index. Given simulated trajectories, we construct the position distribution on the surface by counting and normalizing the frequency of the particle appearing inside each face of the mesh surface. To construct displacement distribution on the flat and the spherical surfaces, we initiated 400 000 trajectories at the same initial position and run the simulation up to the observation time.

In principle, our algorithm can simulate arbitrary surfaces that can be represented by triangle meshes. In the following simulation, we demonstrate four different mesh surfaces, ranging from relatively simple flat and spherical surfaces to complex torus and knot surfaces. The basic geometry information for the four triangle mesh surfaces used in the following sections is given in Table I.

Our algorithm can be run on using a single desktop CPU core for medium system size ( $< 1000$  particles) within reasonable time. For example, the time for simulating 480 particles on the knot surface for  $1 \times 10^6$  steps is about 20 min.

TABLE I. Geometry information for the triangle mesh surfaces.

	Flat surface	Spherical surface	Torus surface	Knot surface
Number of faces	2288	81 920	56 064	449 832
Number of vertices	6864	245 970	168 192	224 916
Average face area ( $a^2$ )	~0.0014	~0.00015	~0.018	~0.0084
Face length scale ( $a$ )	~0.05	~0.018	~0.19	~0.13
Total face area ( $a^2$ )	~6.24	~25	~1971	~7580

## RESULTS AND DISCUSSION

## Brownian simulation in a flat plane

We first benchmark our simulation algorithm on a two-dimensional (2D) flat plane represented by a triangle mesh surface, as shown in Figs. 2(a) and 2(b). The mesh has an average face area of  $\sim 0.0014a^2$ , corresponding to a length scale of  $\sim 0.05a$ . We simulate a single particle that undergoes Brownian motion and experiences no other external force. Figures 2(a) and 2(b) show a representative trajectory of a particle starting at origin.

To obtain more quantitative verification on the simulation algorithm, we studied the position distribution of a diffusing particle. For a freely diffusing Brownian particle, its probability distribution at time  $t$  is given by the following diffusion equation:

$$\frac{\partial p(\mathbf{r}, t)}{\partial t} = D \nabla^2 p(\mathbf{r}, t), \quad (16)$$

where  $p(\mathbf{r}, t)$  is the probability density at time  $t$  and position  $\mathbf{r}$ ,  $D$  is the diffusivity parameter that takes the same value of diffusivity used in Brownian motion simulation, and  $\nabla^2$  is the Laplacian operator. When the diffusion is confined to a plane and the initial starting position is at origin, we have theoretical solution given by

$$p(x, y, t) = \frac{1}{4\pi Dt} \exp\left(-\frac{x^2 + y^2}{4Dt}\right), \quad (17)$$

which is the 2D Gaussian distribution with variance  $4Dt$ . The solution Eq. (17) can also be parametrized by distance  $r$  defined as  $r = \sqrt{x^2 + y^2}$ ,

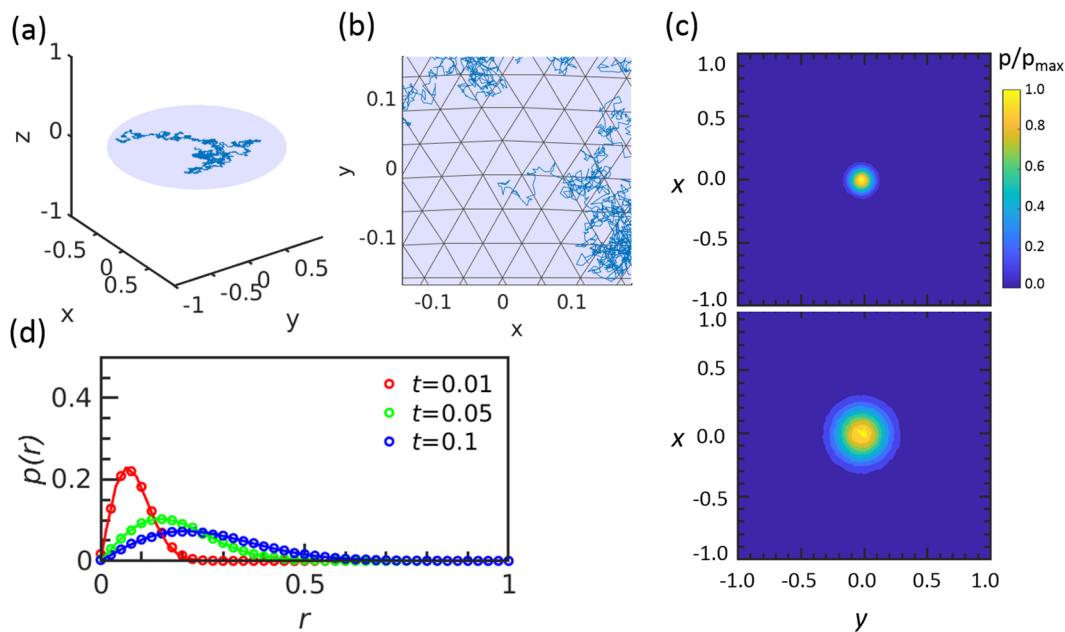
$$p(r, t) = \frac{r}{2Dt} \exp\left(-\frac{r^2}{4Dt}\right). \quad (18)$$

Figures 2(c) and 2(d) show the position distribution constructed from 2D histogram of multiple realizations of BD simulations at different observation time  $t$ . The symmetric shape of the distribution agrees with the expectation. We compare the 1D distribution constructed from simulation trajectory with theoretical prediction in Eq. (18), which shows excellent agreement between simulation and theory at different observation time.

## Brownian simulation in a spherical surface

We next verify our algorithm by simulating a freely diffusing particle on a spherical surface of radius  $a$ . We use a sufficiently refined mesh surface whose triangle faces have a length scale of  $\sim 0.018a$ . Figure 3(a) shows a typical realization of the Brownian simulation of 10 s, with its starting position at the north pole. We then constructed the displacement distribution on the surface from multiple trajectories using histogram method on the mesh (see Methods). The distribution exhibits symmetric shape around the starting position, which agrees with our expectation [Fig. 3(b)].

To quantify the accuracy of our algorithm, we benchmark the displacement distribution with theory. For a particle undergoing Brownian motion on a spherical surface, the probability density of its position is captured by the same diffusion equation in Eq. (16). If the particle starts at the north pole, due to its uniform sampling in

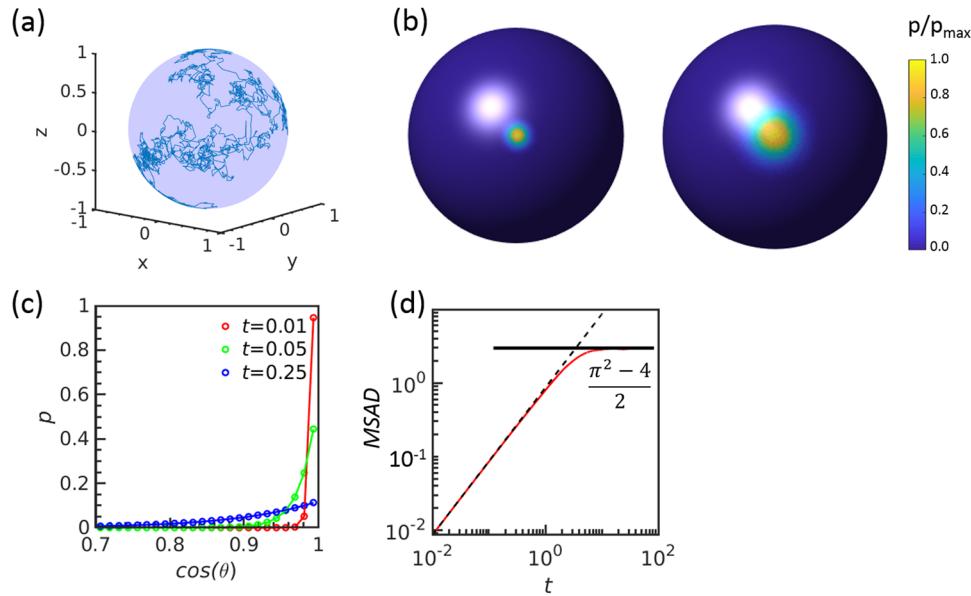


**FIG. 2.** [(a) and (b)] A representative 1 s simulated trajectory of a freely diffusing colloidal particle on a flat 2D triangle mesh surface embedded in 3D space. The magnified top view is showed in (b). (c) Position distribution on the 2D flat surface at different observation time,  $t = 0.01$  s and  $0.05$  s. The distribution is constructed from simulation using trajectories starting at the origin. (d) Position distribution parametrized by distance  $r$  constructed from simulation (symbols) and predicted by theory [Eq. (18); solid lines] at different observation time,  $t = 0.01$  s,  $0.05$  s, and  $0.1$  s.

the azimuth angles  $\phi$  coordinate, the evolution of its position density in polar angle  $\theta$  coordinate is given by<sup>17</sup>

$$p(\theta, t) = \sum_{l \geq 0}^{\infty} \frac{2l+1}{4\pi R^2} P_l(\cos \theta) \exp\left[-\frac{l(l+1)Dt}{R^2}\right], \quad (19)$$

where  $P_l$  are the Legendre polynomials with degree  $l$  and  $R$  is the radius of the sphere. Figure 3(c) shows the position distribution parameterized by  $\cos \theta$ , showing excellent agreement between simulation and theory [Eq. (18)] at different observation time.



**FIG. 3.** (a) A representative 10 s trajectory of a diffusing particle on a spherical surface of radius  $a$ . (b) Position distribution of a diffusing particle at different observation time,  $t = 0.01$  s (left) and  $0.05$  s (right) on the spherical surface. (c) Position distribution parameterized by  $\cos \theta$ , where  $\theta$  is the polar angle, at different observation time. Symbols denote simulation results, and solid lines denote theoretical prediction by Eq. (18). (d) MSAD analysis of a long simulation trajectory (5000 s) on the spherical surface. The dashed line is the theoretical prediction  $MSAD = 4Dt$  that applies to short-time diffusion. The solid horizontal line is the long-time theoretical limit of  $(\pi^2 - 4)/2$  [Eq. (21)].

We further examine the mean-square angular displacement (MSAD)  $\langle \theta(t)^2 \rangle$  of a freely diffusing particle on the sphere, which is defined by

$$\langle \theta(t)^2 \rangle = \left\langle \left[ \cos^{-1} \left( \frac{\mathbf{r}(\tau+t) \cdot \mathbf{r}(\tau)}{\|\mathbf{r}(\tau+t)\| \|\mathbf{r}(\tau)\|} \right) \right]^2 \right\rangle_{\tau}, \quad (20)$$

where the expectation is taken by averaging over a long trajectory  $\mathbf{r}(t)$ .

In the short-time limit where the particle is mainly diffusing on its local neighborhood (e.g., within its own hemisphere) without experiencing the global geometry confinement, the MSAD is characterized by  $MSAD = 4Dt/R^2$ . In the long-time limit where the particle position  $\mathbf{r}(t)$  will be uniformly distributed on the spherical surface, the MSAD has an asymptote given by<sup>17</sup>

$$\langle [\Delta\theta(t \rightarrow \infty)]^2 \rangle = \frac{1}{4\pi} \int_0^{2\pi} \int_0^\pi \theta^2 \sin(\theta) d\theta d\phi = \frac{\pi^2 - 4}{2}. \quad (21)$$

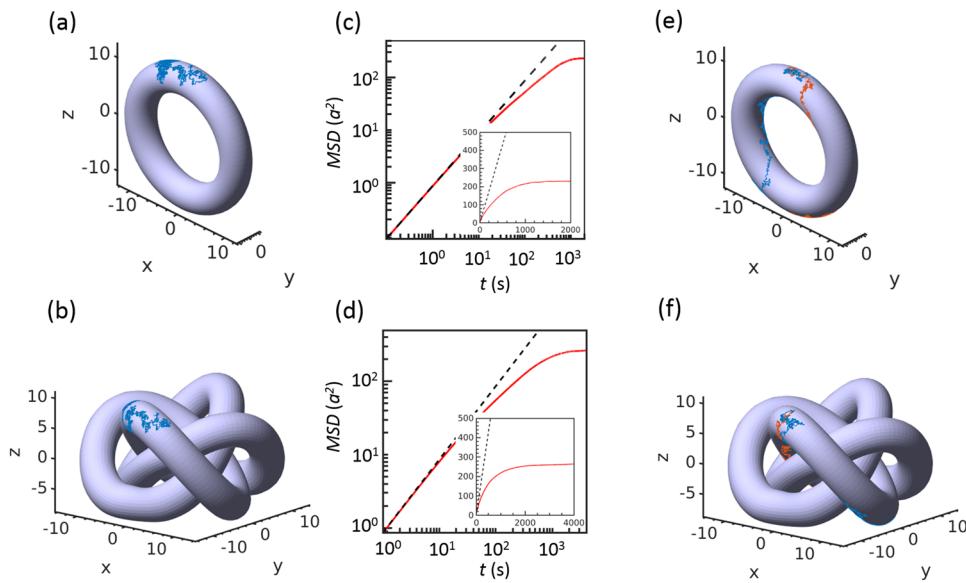
Note that the long-time asymptote is independent of the diffusivity  $D$  but depends on the surface geometry. Figure 3(d) shows the MSAD constructed from a long simulation trajectory (5000 s), which agrees well with theory in both short-time limit ( $\ll 1$  s) and long-time limit ( $> 10$  s). The transition from short-time limit to long-time limit occurs at  $\sim 1$  s, where the MSAD starts to fall below the short-time MSAD curve. This transition time approximately corresponds to a diffusion distance  $\sqrt{4Dt} \sim 3a$ , where the particle diffuses across the equator toward another pole and starts to experience confinement of surface geometry.

### Single particle dynamics on complex surfaces

After benchmarking, we simulate Brownian particles on complex surfaces with nontrivial topology, including a torus and a knot.

(Fig. 4), to demonstrate the capability of our algorithm. These two surfaces are challenging for previous methods<sup>17,21</sup> since they might not have compact parametric forms that allow the computation of constraining forces or direct evolution of dynamics in the local coordinates (the standard torus has an analytical form, but its nonstandard variants will not). Figs. 4(a) and 4(b) show the example trajectories of a simulated particle freely diffusing on a torus and a knot, respectively. Using the global coordinates of the particle, the mean-squared displacement (MSD) analysis of a long simulation trajectory lasting for 10 000 s is provided in Figs. 4(c) and 4(d). On the short time scale  $t < \sim 3$  s (corresponding to diffusion distance  $\sqrt{4Dt} \sim 5a$  and the tube on both surfaces have a diameter of  $5a$ ), the MSD curves on both surfaces can be well approximated by  $MSD \approx 4Dt$  since the particle is approximately diffusing on its local flat tangent plane.

On a slightly longer time scale (e.g.,  $t$  within a range between 10 s and 100 s), which corresponds to diffusion distance of  $\sim 10a$ – $30a$ , the particle trajectory starts to experience confinement from surface geometry and the MSD curve falls under the short time limit of  $4Dt$ . The impact of global geometry and topology fully emerges on a large time scale  $t > 1000$  s, where the MSD curve plateaus around  $\sim 230a^2$  on the torus and after 2000 s it plateaus around  $\sim 270a^2$  on the knot (corresponding to a length scale of  $\sim 15a$  and  $\sim 16a$ , respectively). The MSD curve takes longer time to plateau at the knot surface ( $\sim 2000$  s) than at the torus surface ( $\sim 1000$  s) because the knot surface has a larger area ( $\sim 7579a^2$ ) than the torus surface ( $\sim 1970a^2$ ). The square root of the plateau MSD value can be interpreted as either confinement length scale or the average distance between two positions randomly sampled on the surfaces. The topology thus has a nontrivial impact on the confinement length scale. Although the knot has a much larger dimension ( $\sim 29a \times 29a \times 20a$ ) than the torus ( $25a \times 5a \times 25a$ ), the confinement length scale on the knot is actually comparable with that on the torus. This is because multiple interconnected tube surfaces in



**FIG. 4.** Trajectories of 100 s Brownian simulation of single particle diffusing on surfaces with nontrivial topology, including a torus (a) and a knot (b). The torus has a dimension of  $\sim 25a \times 5a \times 25a$ , and the knot has a dimension of  $29a \times 29a \times 20a$ . [(c) and (d)] MSD analysis of a long simulation trajectory (10 000 s) on the torus (c) and the knot (d). We calculate the MSD in terms of Euclidean distance (instead of geodesic distance) based on particle lab coordinates. The dashed lines are the theory  $MSD = 4Dt$  that applies to diffusion on flat 2D surfaces. Insets are MSD plot in the linear scale. [(e) and (f)] Two example trajectories of 100 s Brownian simulation of a single diffusive particle on the torus (e) and the knot (f), where the particle is subject to external field force along negative  $z$  direction characterized by  $F_z = -3 kT/a$ .

the knot enable diffusing particles to have higher chances to get closer to their previously sampled position in terms of Euclidean distance.

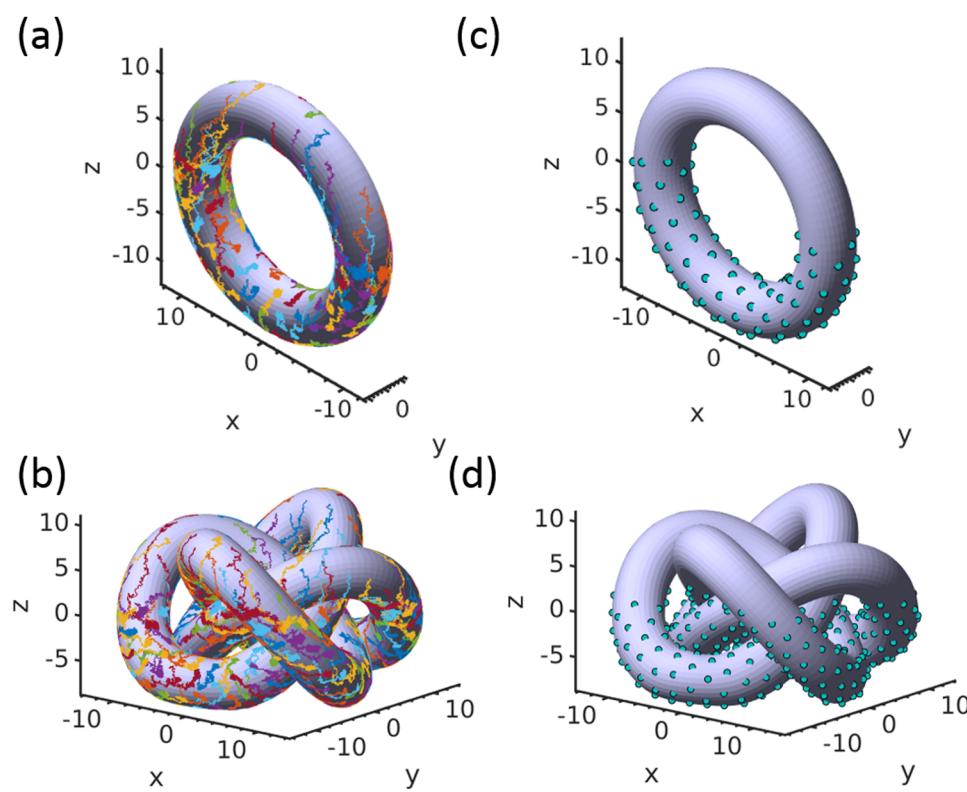
By applying external field force along the negative  $z$  direction (e.g., gravity, electric field<sup>48,49</sup>), our algorithm can be used to study the single particle diffusive transport driven by the interplay of external force, Brownian motion, and geometry constraint [Figs. 4(e) and 4(f)]. Compared with the free diffusion case, particles under external forces quickly move down the surface and equilibrate at the surface bottom. As particles quickly move down the surface, tiny Brownian force perturbation can dramatically impact the trajectory path down to the bottom. For example, in the torus, the particles might fall down either in the front surface or in the back surface even they start with the same initial position [Fig. 4(e)]. Likewise, particles on the top of a ring of a knot can move down along different tube surfaces and reach different parts at the bottom [Fig. 4(f)]. Compared with particle transport in Euclidean space, the higher sensitivity to Brownian motion can be possibly employed to engineering applications like switches and particle sorting devices.

### Multiple particle dynamics on complex surfaces

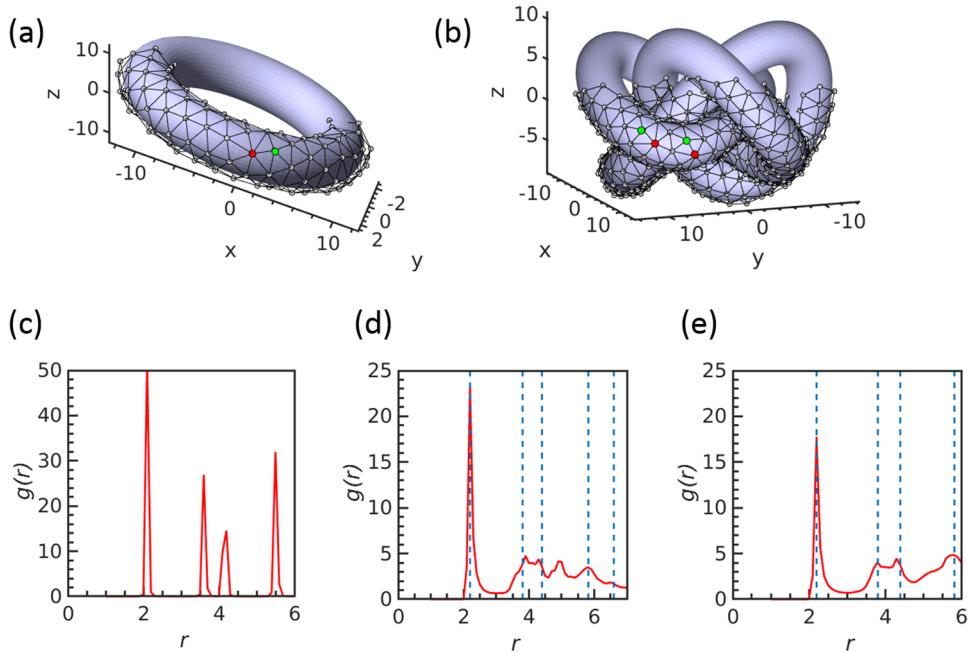
Dynamics of multiple particles on curved surfaces has generated significant excitement for both theoretical exploration<sup>16,26</sup> and its potential to aid engineering novel structures and devices.<sup>13–15</sup> As a final demonstration of our algorithm, we simulate colloidal particle assembly on curved surfaces under external force field [Fig. 5].

We simulate multiple colloidal particles with mild attraction levels ( $\sim 5kT$ ) on the surfaces and apply the external force field with  $F_z = -10kT/a$  (e.g., gravity). In a flat 2D plane, crystallization of such weakly attractive colloidal particles requires radial inward force to compress and concentrate the colloids, otherwise entropy will drive all particles to sample the whole space.<sup>22,50</sup> In our example, the external forces, although acting in the  $z$  direction, also concentrate particles toward the bottom and induce crystallization. Figures 5(a) and 5(b) show the trajectories of multiple particles, starting from a uniformly distributed initial configuration, moving toward the bottom and finally forming a solid crystalline structure [Figs. 5(c) and 5(d)].

At equilibrium, besides predominant hexagonal close-packed structures formed on the torus surface and the knot surface, we also identify topological defects including mainly heptagons (seven-fold coordinated particle) and pentagons (five-fold coordinated particle) [Figs. 6(a) and 6(b)]. Pentagons and heptagons facilitate the tiling of a curve surface (e.g., soccer balls), and they are commonly found in crystals on curved surfaces.<sup>5,25</sup> We proceed to quantify the impact of curvature and topology on the crystalline packing structure by analyzing the pair correlation distribution  $g(r)$  (using 3D Euclidean distance), as shown in Figs. 6(c)–6(e), whose value can be interpreted as the probability (up to a scaling factor) to find another particle at a given distance  $r$  from the center of one particle. For comparison, we also compute  $g(r)$  for an equilibrated finite-size circular crystalline lattice (300 particles) on a 2D plane obtained via electric field compression.<sup>3,22,51</sup>



**FIG. 5.** [(a) and (b)] Assembly trajectories and equilibrium configuration of 120 particles on a torus surface under external field in the negative  $z$  direction characterized by  $F_z = -10kT/a$ . [(c) and (d)] Assembly trajectories and equilibrium configuration of 480 particles on a knot surface under the same external field as in (a).



**FIG. 6.** [(a) and (b)] Topological defects in the equilibrium structure of particle assembly on the torus (a) and the knot (b). Particles colored red are heptagons (seven-fold coordinated particle) and particles colored green are pentagons (five-fold coordinated particle). The neighbors for each particle are identified by Voronoi diagram. (c) Pair distribution function  $g(r)$  for an equilibrium, finite-sized, circular-shaped, hexagon close-packed lattice on a flat plane. [(d) and (e)] Pair distribution function  $g(r)$  for the equilibrium structure in (a) and (b). Dashed lines are the positions of peaks in  $g(r)$  for a 2D hexagonally close-packed crystal in (c). The positions of the dashed lines are shifted to align with the first peaks in (d) and (e).

As particles predominantly form hexagonally close-packed (HCP) structure on surfaces, all  $g(r)$  functions exhibit similar peak locations corresponding to the first four coordinate shells. There are several key observations. First, for lattices on a plane,  $g(r)$  has sharper peaks and vanishing values between peaks, whereas on a curved surface, the curvature renders larger variations in particles' pair distance, and as a result,  $g(r)$  exhibits wider and shorter peaks. Moreover, the knot surface is more complex and has larger curvature variation than the torus surface (i.e., more twists and tubes), which not only causes more defects and imperfect HCP structures [Fig. 6(b)] but also leads to wider and shorter first peak in  $g(r)$  on the knot surface than on the torus surface. Besides the peaks in  $g(r)$  corresponding to first four coordinate shells, we also observe an additional small and wide peaks at the position of  $\sim 5a$  in Figs. 6(d) and 6(e). This can be attributed to the surface geometry where particles are packing on the tube with a diameter of  $5a$ .

## CONCLUSION AND OUTLOOK

In this work, we have developed a BD simulation algorithm for particles moving on general curved surfaces. By approximating surfaces via triangle mesh surfaces and performing constrained Brownian simulation on the meshes, we can simulate Brownian particles dynamics on surfaces with complex geometry and topology. We have verified the simulation algorithm by comparing simulation results and theoretical predictions on a 2D flat surface and a spherical surface. We demonstrate the application of our algorithms in simulating the single particle diffusion and transport and crystallization of collective particles on a torus surface and a knot surface. Our results illustrate the impact of curvature and topology on particle dynamics and equilibrium structures.

Since approximating complex curved surfaces via triangle mesh surfaces is well established, our algorithms will find application in a wide range of areas involving complex curved surfaces. For example, besides the study of point defects arises from curved geometry,<sup>4</sup> our tools can also be applied to study grain boundaries,<sup>3,22,52</sup> glasses,<sup>24,53–55</sup> and jamming<sup>56,57</sup> in colloidal systems on curved surfaces. It can also be used as a flexible tool to simulate proteins or macromolecules on a membrane with complex geometry<sup>8,9,58</sup> and viral assembly.<sup>59</sup> The algorithm can also be applied to active systems to study the impact of geometry and topology on the nonequilibrium dynamics, including lining, jamming, clustering, and oscillation.<sup>20,60–63</sup> Furthermore, by simultaneously simulating underlying surface dynamics, our algorithm can be extended to explore extremely complex phenomena, such as surface morphology dynamics driven by active particles.<sup>15</sup>

## ACKNOWLEDGMENTS

Support from the National Natural Science Foundation of China (Grant Nos. 11961131005 and 11672161) is acknowledged. We also want to thank Professor Misha Kazhdan and Professor Gregory Chirikjian at Johns Hopkins University for inspiring discussion.

The authors declare no competing financial interest.

## REFERENCES

- G. Meng, J. Paulose, D. R. Nelson, and V. N. Manoharan, *Science* **343**, 634 (2014).
- H. Serna, E. G. Noya, and W. Góźdź, *Langmuir* **35**, 702 (2019).
- T. D. Edwards, Y. Yang, D. J. Beltran-Villegas, and M. A. Bevan, *Sci. Rep.* **4**, 6132 (2014).
- V. N. Manoharan, *Science* **349**, 1253751 (2015).

- <sup>5</sup>W. T. Irvine, V. Vitelli, and P. M. Chaikin, *Nature* **468**, 947 (2010).
- <sup>6</sup>O. Kruglova, P.-J. Demeyer, K. Zhong, Y. Zhou, and K. Clays, *Soft Matter* **9**, 9072 (2013).
- <sup>7</sup>A. B. Subramaniam, M. Abkarian, L. Mahadevan, and H. A. Stone, *Nature* **438**, 930 (2005).
- <sup>8</sup>F. Jaskolski and J. Henley, *Neuroscience* **158**, 19 (2009).
- <sup>9</sup>M. Weiss, H. Hashimoto, and T. Nilsson, *Biophys. J.* **84**, 4043 (2003).
- <sup>10</sup>D. Baptista, L. Teixeira, C. van Blitterswijk, S. Giselbrecht, and R. Truckenmüller, *Trends Biotech.* **37**, 838 (2019).
- <sup>11</sup>A. J. Ridley *et al.*, *Science* **302**, 1704 (2003).
- <sup>12</sup>W. Xi, S. Sonam, T. B. Saw, B. Ladoux, and C. T. Lim, *Nat. Commun.* **8**, 1517 (2017).
- <sup>13</sup>I. R. Bruss and S. C. Glotzer, *Soft Matter* **13**, 5117 (2017).
- <sup>14</sup>P. W. Ellis *et al.*, *Nat. Phys.* **14**, 85 (2018).
- <sup>15</sup>F. C. Keber *et al.*, *Science* **345**, 1135 (2014).
- <sup>16</sup>S. Shankar, M. J. Bowick, and M. C. Marchetti, *Phys. Rev. X* **7**, 031039 (2017).
- <sup>17</sup>P. Castro-Villarreal, A. Villada-Balbuena, J. M. Méndez-Alcaraz, R. Castañeda-Priego, and S. Estrada-Jiménez, *J. Chem. Phys.* **140**, 214115 (2014).
- <sup>18</sup>R. R. Gabdoulline and R. C. Wade, *Methods* **14**, 329 (1998).
- <sup>19</sup>Z. Gan, Z. Wang, S. Jiang, Z. Xu, and E. Luijten, *J. Chem. Phys.* **151**, 024112 (2019).
- <sup>20</sup>A. Martín-Gómez, D. Levis, A. Díaz-Guilera, and I. Pagonabarraga, *Soft Matter* **14**, 2610 (2018).
- <sup>21</sup>S. Paquay and R. Kusters, *Biophys. J.* **110**, 1226 (2016).
- <sup>22</sup>Y. Yang, R. Thyagarajan, D. M. Ford, and M. A. Bevan, *J. Chem. Phys.* **144**, 204904 (2016).
- <sup>23</sup>R. Backofen, A. Voigt, and T. Witkowski, *Phys. Rev. E* **81**, 025701 (2010).
- <sup>24</sup>G. L. Hunter and E. R. Weeks, *Rep. Prog. Phys.* **75**, 066501 (2012).
- <sup>25</sup>W. T. Irvine, M. J. Bowick, and P. M. Chaikin, *Nat. Mater.* **11**, 948 (2012).
- <sup>26</sup>L. Apaza and M. Sandova, *Soft Matter* **14**, 9928 (2018).
- <sup>27</sup>D. L. Ermak and J. A. McCammon, *J. Chem. Phys.* **69**, 1352 (1978).
- <sup>28</sup>E. J. Banigan, T. H. Harris, D. A. Christian, C. A. Hunter, and A. J. Liu, *PLoS Comput. Biol.* **11**, e1004058 (2015).
- <sup>29</sup>R. L. Klank, S. S. Rosenfeld, and D. J. Odde, *Convergent Sci. Phys. Oncol.* **4**, 015001 (2018).
- <sup>30</sup>A. Rojnuckarin, S. Kim, and S. Subramaniam, *Proc. Natl. Acad. Sci. U. S. A.* **95**, 4288 (1998).
- <sup>31</sup>A. Montesi, D. C. Morse, and M. Pasquali, *J. Chem. Phys.* **122**, 084903 (2005).
- <sup>32</sup>R. Kutteh, *J. Chem. Phys.* **132**, 174107 (2010).
- <sup>33</sup>Y. Yang and M. A. Bevan, *J. Chem. Phys.* **147**, 054902 (2017).
- <sup>34</sup>J. W. Swan and J. F. Brady, *Phys. Fluids* **22**, 103301 (2010).
- <sup>35</sup>J. W. Swan and J. F. Brady, *J. Fluid Mech.* **687**, 254 (2011).
- <sup>36</sup>V. I. Arnol'd, *Mathematical Methods of Classical Mechanics* (Springer Science & Business Media, 2013).
- <sup>37</sup>E. J. Hinch, *J. Fluid Mech.* **271**, 219 (1994).
- <sup>38</sup>D. C. Morse, *Adv. Chem. Phys.* **128**, 65 (2004).
- <sup>39</sup>D. S. Lo, *Finite Element Mesh Generation* (CRC Press, 2014).
- <sup>40</sup>Y. J. Zhang, *Geometric Modeling and Mesh Generation from Scanned Images* (Chapman and Hall/CRC, 2018).
- <sup>41</sup>M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon Mesh Processing* (CRC Press, 2010).
- <sup>42</sup>J. L. Bitter, Y. Yang, G. Duncan, H. Fairbrother, and M. A. Bevan, *Langmuir* **33**, 9034 (2017).
- <sup>43</sup>J. F. Brady and G. Bossis, *Annu. Rev. Fluid Mech.* **20**, 111 (1988).
- <sup>44</sup>T. D. Edwards, Y. Yang, W. N. Everett, and M. A. Bevan, *Sci. Rep.* **5**, 13612 (2015).
- <sup>45</sup>D. Higham, *SIAM Rev.* **43**, 525 (2001).
- <sup>46</sup>T. D. Edwards and M. A. Bevan, *Langmuir* **28**, 13816 (2012).
- <sup>47</sup>Y. Yang, T. D. Edwards, and M. A. Bevan, *J. Colloid Interface Sci.* **449**, 270 (2015).
- <sup>48</sup>T. D. Edwards and M. A. Bevan, *Langmuir* **30**, 10793 (2014).
- <sup>49</sup>I. Torres-Díaz, B. Rupp, Y. Yang, and M. A. Bevan, *Soft Matter* **14**, 934 (2018).
- <sup>50</sup>J. J. Juárez and M. A. Bevan, *Adv. Funct. Mater.* **22**, 3833 (2012).
- <sup>51</sup>X. Tang *et al.*, *ACS Nano* **10**, 6791 (2016).
- <sup>52</sup>A. Bausch *et al.*, *Science* **299**, 1716 (2003).
- <sup>53</sup>W. C. Poon, *MRS Bull.* **29**, 96 (2004).
- <sup>54</sup>W. Härtl, *Curr. Opin. Colloid Interface Sci.* **6**, 479 (2001).
- <sup>55</sup>C. P. Royall, A. Malins, A. J. Dunleavy, and R. Pinney, *J. Non-Cryst. Solids* **407**, 34 (2015).
- <sup>56</sup>K. Stratford, R. Adhikari, I. Pagonabarraga, J.-C. Desplat, and M. E. Cates, *Science* **309**, 2198 (2005).
- <sup>57</sup>V. Trappe, V. Prasad, L. Cipelletti, P. Segre, and D. A. Weitz, *Nature* **411**, 772 (2001).
- <sup>58</sup>W. Gózdz, *Langmuir* **24**, 12458 (2008).
- <sup>59</sup>R. D. Cadena-Nava *et al.*, *J. Virol.* **86**, 3318 (2012).
- <sup>60</sup>C. Dombrowski, L. Cisneros, S. Chatkaew, R. E. Goldstein, and J. O. Kessler, *Phys. Rev. Lett.* **93**, 098103 (2004).
- <sup>61</sup>S. Henkes, Y. Fily, and M. C. Marchetti, *Phys. Rev. E* **84**, 040301 (2011).
- <sup>62</sup>S. Z. Lin, S. L. Xue, B. Li, and X. Q. Feng, *J. Mech. Phys. Solids* **112**, 650 (2018).
- <sup>63</sup>Y. Liu, Y. Yang, B. Li, and X. Q. Feng, *Soft Matter* **15**, 2999 (2019).