

GPU accelerated Discrete Element Method (DEM) molecular dynamics for conservative, faceted particle simulations

Matthew Spellings^{a,c}, Ryan L. Marson^{b,c}, Joshua A. Anderson^{a,c}, Sharon C. Glotzer^{a,b,c,*}

^a Chemical Engineering, University of Michigan, 2800 Plymouth Rd., Ann Arbor, MI 48109, USA

^b Materials Science & Engineering, University of Michigan, 2300 Hayward St., Ann Arbor, MI 48109, USA

^c Biointerfacing Institute, University of Michigan, 2800 Plymouth Rd., Ann Arbor, MI 48109, USA

ARTICLE INFO

Article history:

Received 3 June 2016

Received in revised form 6 January 2017

Accepted 10 January 2017

Available online 12 January 2017

Keywords:

GPU

Molecular dynamics

Discrete Element Method

Anisotropy

ABSTRACT

Faceted shapes, such as polyhedra, are commonly found in systems of nanoscale, colloidal, and granular particles. Many interesting physical phenomena, like crystal nucleation and growth, vacancy motion, and glassy dynamics are challenging to model in these systems because they require detailed dynamical information at the individual particle level. Within the granular materials community the Discrete Element Method has been used extensively to model systems of anisotropic particles under gravity, with friction. We provide an implementation of this method intended for simulation of hard, faceted nanoparticles, with a conservative Weeks–Chandler–Andersen (WCA) interparticle potential, coupled to a thermodynamic ensemble. This method is a natural extension of classical molecular dynamics and enables rigorous thermodynamic calculations for faceted particles.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

The impact of particle shape on the self-assembly of systems of colloidal- and nanoscale particles is receiving ever-increasing attention [1]. Hard particle simulations are the most straightforward way to determine the impact of particle shape on assembly and have been highly successful in elucidating the phase behavior of anisotropic particles [2,3]. Monte Carlo (MC) methods are ideal for probing the equilibrium behavior of such systems and can be implemented efficiently on modern highly parallel architectures [4,5]. However, studying nonequilibrium behavior often requires dynamical, rather than stochastic, simulation methods. While hard-particle Monte Carlo methods like those used in the previous studies can accurately predict the equilibrium behavior of shapes, it is more difficult to directly study nonequilibrium behaviors like crystal nucleation [6] and active matter [7,8]. Event-driven molecular dynamics [9–12] (EDMD) is one such dynamical method, but it can be difficult to parallelize or to extend for arbitrary shapes and can slow down at the moderate to high densities of interest in many self-assembly studies. As such, there are currently no EDMD codes publicly available for performing dynamical simulations of particles with arbitrary shape.

Treating the issue of shape in MD is hardly a new concept; beyond the previously mentioned techniques, it is also common to see coarse-grained models of shapes constructed from stacked spheres. Such stacked sphere models consist

* Corresponding author at: Biointerfacing Institute, University of Michigan, 2800 Plymouth Rd., Ann Arbor, MI 48109, USA.

E-mail address: sglotzer@umich.edu (S.C. Glotzer).

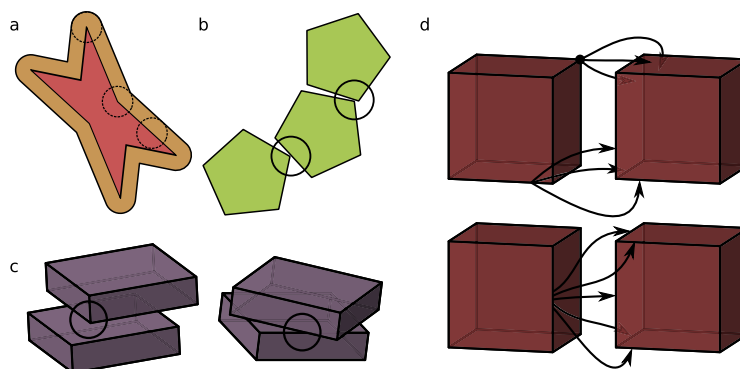


Fig. 1. The DEM model. (a) Shapes are represented by a polygon or polyhedron (inner shape), rounded out by a disk or sphere (dotted circles) to give the outer shape. The inner shape can be concave or convex. (b) In two dimensions, a repulsive contact force is evaluated between the nearest point on all pairs of vertices and edges. (c) In three dimensions, the contact force is evaluated between all pairs of vertices and faces and all pairs of edges and edges. (d) Because interactions are always evaluated between features, some interactions can be overcounted: when two aligned cubes come together, eight vertices will interact with three faces each, while four edges will interact with five neighboring edges and four other edges will interact with three neighboring edges each.

of rigid bodies of spheres in a more- or less-principled way to create particles with shape [13,14]. However, this leaves particles with a rough surface which could introduce artifacts in assembly behavior when compared to the faceted particles often created in experiments. Particles simulated via the method described in this manuscript have smooth faces, yielding a higher-fidelity model of particle shape.

The Discrete Element Method (DEM) has been used extensively by the granular materials community to study dynamics of anisotropic, frictional particle systems [15–21]. This method models interactions between particles as interactions between the minimal set of lower-dimensional geometric features needed to capture the effects of particles' shapes. DEM is also a natural method to implement as a force field in a classical molecular dynamics (MD) framework, which lends itself to hardware acceleration. Programs using graphics processing units (GPUs) can achieve order-of-magnitude speedups over single-CPU programs, but only if they are made to take full advantage of the parallel nature of the GPU. Here we present an adaptation of DEM to run on GPUs within the HOOMD-Blue [22] MD framework. This sets our DEM implementation in a different class of algorithms than existing codes such as LIGGGHTS, where particle shape is constructed by stacking spheres. Furthermore, our implementation lacks the frictional contact force present in traditional DEM implementations by design – we have redesigned the algorithm to be appropriate for the kinds of forces present in colloidal and nanoscale materials, rather than those typical in granular matter. To our knowledge, our implementation is the first to bring DEM into the domain of classical MD. The method is intended purely for capturing steric repulsive forces between particles – such as those that have been ligated, charge-screened, or otherwise functionalized to only interact via short-ranged forces. Our method can be supplemented with additional colloidal or nanoscale forces typically used in coarse-grained MD particle simulations to account for intermediate or long-range forces like dipoles or charged surfaces [23]. This method is most directly applicable for people who would like to study the behavior of coarse-grained faceted colloidal- and nanoscale particles with higher shape fidelity than can be achieved with rigid bodies of spheres.

Finally, while it is tempting to draw comparisons between this and existing methodologies such as MC and EDMD, we have made no attempts at rigorous thermodynamic comparisons between various methods due to the inherent complexity of such comparisons. Here we have simply set forth the methodology, and leave it to the user to decide if DEM is the appropriate choice for their coarse-grained models. The implementation has been freely-available beginning with HOOMD-Blue v2.0, allowing users to easily and efficiently study the equilibrium and nonequilibrium behavior of faceted particles.

2. Model

Some assumptions must be made about the simulated particles for the method described here to be useful in classical molecular dynamics. First, we assume that the polytopal shapes have been rounded by a sphere or disk of a given radius, as shown in Fig. 1(a). Rounding arises from the isotropic, conservative interaction we compute between interacting points and prevents discontinuities in the force arising from corners of shapes. The optimal rounding radius to use depends strongly on the geometry of the particles – like the presence of a shell of surface ligands around otherwise sharply faceted shapes – and the quantities that are being studied. For studies of nanoparticle and colloidal assembly, we later show that the small amount of rounding required by our DEM algorithm has no discernible effect on self-assembly behavior for rounding radii of more than 10% of the edge length, depending on the shapes involved and the phenomena under investigation. Second, we assume that particles interact only through short-range, repulsive forces. These assumptions allow us to approximate interactions between the volumes of particles i and j as interactions between the nearest points of lower-dimensional features (faces, edges, or vertices) of the particles using simple point–point interactions, without integrating over the surface or volume of the particles. In general, we consider a minimal set of interactions between features that will prevent two

polyhedral shapes from overlapping as they collide. To prevent particle overlap in 2D, it is sufficient to consider interactions between all pairs of vertices and edges between the two particles, as shown in Fig. 1(b). In 3D, simulating polyhedral volumes requires calculating interactions between vertex–face pairs, and all edge–edge pairs as in Fig. 1(c).

Often in MD and MC simulations of anisotropic particles, a single “particle” is built from several spheres, joined together rigidly [13,24]; interactions are calculated between all pairs of spheres, but translations and rotations are applied to the body as a whole [25]. Ideally, to create an anisotropic potential from a shape, we would integrate an isotropic potential over the surface or volume of a pair of shapes. Within DEM, particles are instead built up out of the geometric features of a two- or three-dimensional polytope: vertices (V), edges (E), and faces (F). The functions $V(\cdot)$, $F(\cdot)$, and $E(\cdot)$ yield the coordinates of the vertices, faces, and edges of their argument, respectively, and r_{ab}^* returns the distance between the nearest points of two features with types a and b . We then define the potential energy U_{ij} between particles i and j in three dimensions using a point particle potential $\mathcal{U}$ as

$$U_{ij}^{3D} = \sum_{\substack{E_i \in E(i) \\ E_j \in E(j)}} \mathcal{U}(r_{EE}^*(E_i, E_j)) + \sum_{\substack{V_i \in V(i) \\ F_j \in F(j)}} \mathcal{U}(r_{VF}^*(V_i, F_j)) + \sum_{\substack{V_j \in V(j) \\ F_i \in F(i)}} \mathcal{U}(r_{VF}^*(V_j, F_i)) \quad (1)$$

By considering vertex–face and edge–edge interactions, one or more potential interactions U_{ij}^{3D} will become nonzero as two polyhedra come into contact, as illustrated in Fig. 1. In two dimensions these features are reduced to checks between vertices and edges only:

$$U_{ij}^{2D} = \sum_{\substack{V_i \in V(i) \\ E_j \in E(j)}} \mathcal{U}(r_{VE}^*(V_i, E_j)) + \sum_{\substack{V_j \in V(j) \\ E_i \in E(i)}} \mathcal{U}(r_{VE}^*(V_j, E_i)) \quad (2)$$

Similarly to three dimensions, in two dimensions vertex–edge computations are sufficient to make one or more terms in U_{ij}^{2D} become positive as two polygons contact each other. The nearest points given by r_{ab}^* can be found using standard point–line, line–line, and point–plane formulae. Forces are computed using the derivative of this potential and torques are based on the interaction point on each particle.

Because we are simulating nanoscale and colloidal systems, we choose a conservative pair potential $\mathcal{U}$ that is representative of the interactions of such materials and well vetted within the community. A truncated and shifted version of the Lennard–Jones (LJ) potential, the Weeks–Chandler–Andersen (WCA) potential, [26] creates a steep, purely repulsive force from the particle surface with a rounding radius of $\frac{1}{2}\sigma_{ij}$:

$$U_{ij}^{LJ}(r) = 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r} \right)^{12} - \left(\frac{\sigma_{ij}}{r} \right)^6 \right] \quad (3)$$

$$U_{ij}^{WCA}(r) = \begin{cases} U_{ij}^{LJ}(r) - U_{ij}^{LJ}(r_{\text{cut}}^{WCA}) & r < r_{\text{cut}}^{WCA} \\ 0 & r \geq r_{\text{cut}}^{WCA} \end{cases} \quad (4)$$

where $r_{\text{cut}}^{WCA} = 2^{\frac{1}{6}}\sigma_{ij}$.

When initializing particles on a lattice, e.g. prior to thermalization, one may encounter collisions of perfectly parallel edges when the lattice spacing is small. These collisions introduce a numerical instability for the molecular dynamics integrator: the points of interaction fluctuate at every timestep between the endpoints of each edge, yielding an unstable torque that changes sign at every timestep. To alleviate this issue while still only using point interactions, when two edges are sufficiently close to parallel the interaction point is taken to be the midpoint of overlap between the two edges.

Another common occurrence during simulation is for features to be “overcounted”; features that are shared among two separate faces (edges and/or vertices) are counted individually for polygonal face. This results from the fact that each polyhedron is decomposed into a set of polygonal faces (containing vertices and edges), and that the components of each face are subsequently checked against all other faces. A concrete example of this overcounting is shown in Fig. 1(d). If two cubes are touching perfectly face to face, they will have an interaction strength 66 times as large as a single vertex–face interaction: eight vertices are interacting with three faces each, four edges are interacting with five edges each, and four edges are interacting with three edges each. If the single vertex–vertex interaction had a strength of $1k_B T$, then the $1k_B T$ isosurface for the now $66k_B T$ interaction would have moved out by 10% of the rounding radius of the shape. This could lead to energetic “bumps” in the interaction: while the cutoff radius is not affected, the interaction is increased by a multiplicative factor according to the geometry of the two interacting sites, causing equipotential lines to expand slightly around vertices and edges. This effect should not matter to the extent that the potential used is a good approximation of a “hard” force field. Regardless of the geometry, we note that due to the cutoff in the WCA potential, it is impossible for the rounding radius to be increased by more than a factor of $2^{1/6} \approx 1.12$ with this overcounting effect.

3. Algorithm

The total force, torque, and potential energy for a given particle is the sum of the force, torque, and potential energy contributions between it and its neighbors. We evaluate these contributions from each particle’s features independently

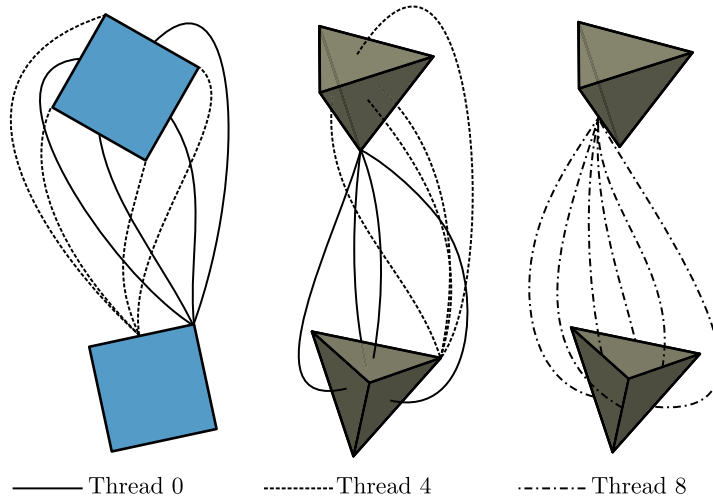


Fig. 2. Thread decomposition for the GPU. In 2D (left), each vertex is assigned a thread and each edge is assigned a thread; in 3D (center, right), each vertex is assigned two threads and each edge is assigned one thread.

by splitting the features among different GPU threads, then summing them efficiently in shared memory. In simulations constrained to only move in two dimensions (2D), two CUDA threads are assigned to each vertex of particle i , as shown in Fig. 2. The first thread assigned to a given vertex calculates and sums the force, torque, and potential energy contributions between that vertex and the nearest point to that vertex on each edge of each neighboring particle j . The second thread assigned to a given vertex calculates and sums the force, torque, and potential energy contributions between the nearest point on the edge beginning at that vertex (traveling counterclockwise) in particle i to each vertex in each neighboring particle j .

In fully three-dimensional (3D) simulations, two CUDA threads are assigned to each vertex of particle i and one thread is assigned to each edge of particle i , as shown in Fig. 2. The first vertex thread calculates the interaction between that vertex and the nearest point to each face in each neighboring particle j . The second vertex thread calculates the interaction between that vertex in each neighboring particle j and the nearest point of each face of particle i . The edge thread calculates the interaction between the nearest point on its edge of particle i to each edge of each neighboring particle j .

4. Results

4.1. Energy conservation

To perform rigorous thermodynamic calculations using this method, we must first ensure that the NVE integrator conserves energy when combined with our force algorithm. We analyze both the short-term and long-term energy conservation of our model [27]. We use reduced units ε_0 for energy, σ_0 for length, m_0 for mass, and $\tau_0 = \sqrt{\frac{m_0 \sigma_0^2}{\varepsilon_0}}$ for time. We run NVE simulations of a fluid of squares (with edge length $4.24\sigma_0$) in 2D and tetrahedra (with edge length $8.49\sigma_0$) in 3D which have been thermalized at temperature $T^* = 1$ in reduced units.

To measure the short-term energy conservation, we compute the standard deviation of the total energy per particle $\sigma(E/N)$, recorded at a high frequency over a short NVE simulation of duration $10\tau_0$. We measure $\sigma(E/N)$ for systems using both single and double precision floating point arithmetic with a varying integration timestep size δt and present the results in Fig. 3.

For small δt , rounding errors saturate the accuracy of the method, causing the energy conservation to plateau. These rounding errors appear at a much smaller value of δt when using double- rather than single-precision floating point arithmetic. At large δt , error is introduced through the coarse time step as $\frac{\Delta E}{\langle E \rangle} \sim \Delta t^2$ for the second-order integrator we use (a standard Velocity-Verlet algorithm with extensions for anisotropic particles [28], implemented as `md.integrate.nve` in HOOMD-Blue), increasing the energy deviation. For the remaining tests, which are performed in single precision only, we choose $\delta t = 0.01\tau_0$ as the timestep size to balance energy conservation and simulation speed.

To study the long-term energy conservation, we calculate the drift of the total energy per particle, $\frac{1}{N}(E(t) - E(0))$, over long simulations. For squares, we achieve an energy drift of $1.36(6) \cdot 10^{-5} \Delta E/(N/\varepsilon_0/\tau_0)$ and for tetrahedra we obtain $6.33(2) \cdot 10^{-4} \Delta E/(N/\varepsilon_0/\tau_0)$. The energy drifts are significantly higher than those reported for isotropic particles [29], $1.3(1) \cdot 10^{-7} \Delta E/(N/\varepsilon_0/\tau_0)$ over the same period of $10\tau_0$, likely due to the new rotational degrees of freedom and the approximation of expressing the energy between features as the potential evaluated between their closest points. In practice, we find these energy drifts to be acceptable for the coarse-grained simulations at which this method is targeted.

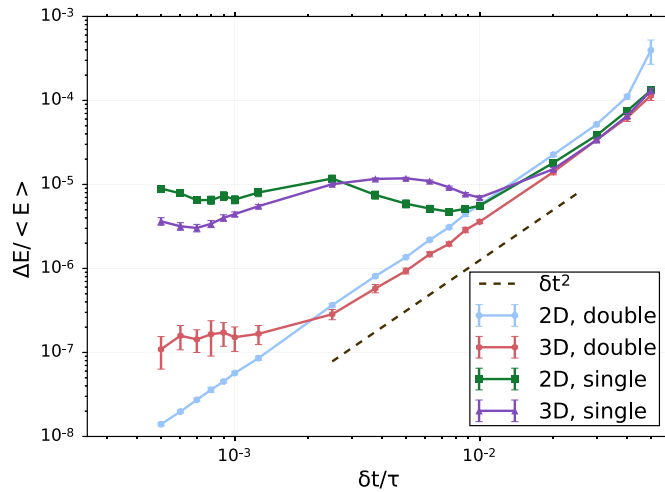


Fig. 3. Magnitude of total energy fluctuations for squares (2D) and tetrahedra (3D) in constant energy simulations for short times as a function of timestep size δt . Error bars indicate two standard errors of the mean for the energy fluctuations.

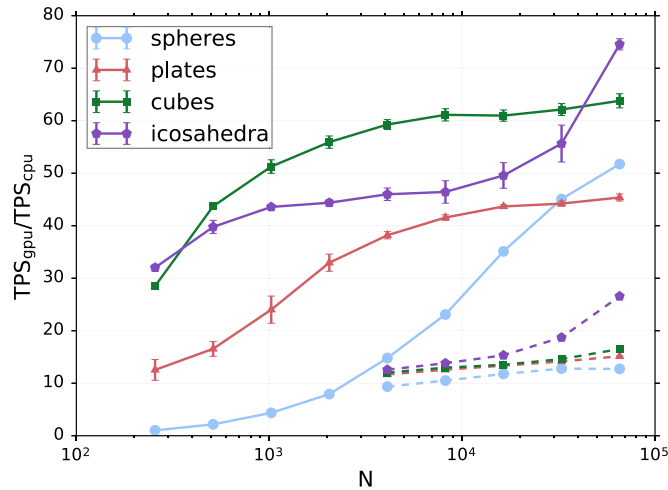


Fig. 4. Simulation speedup for different three-dimensional shapes by using a GPU (solid lines) and 18 CPU cores in parallel (dashed lines) relative to a single CPU core. Error bars indicate two standard errors of the mean.

4.2. Performance

We evaluate the speed benefit of our GPU parallelization scheme using an NVIDIA Quadro M6000 relative to both 18 cores and a single core of an Intel E5-2680V2 CPU for dense fluids of several systems with shapes of varying complexity in three dimensions: spheres, triangular plates, cubes, and icosahedra. The anisotropic shapes are modeled with the DEM potential, and spheres are modeled with a central WCA interaction, as in Equation (4). As shown in Fig. 4, we achieve speedups of 15–75 times on the GPU and 10–27 times using domain decomposition, depending on the particle shape and system size. For many shapes the relative speed saturates at system sizes of a few thousand particles. Importantly, Fig. 4 also shows little to no speed up at small particle numbers on the GPU over the CPU for isotropic particles. This is due to the fact that threads are being under-utilized on the GPU for the isotropic particles; however, as can be seen in Fig. 4, the DEM algorithm saturates the GPU even for relatively modest numbers of particles. We note that the apparent jump in icosahedron performance for both GPU and parallel CPU implementations is due to a decrease of speed on the single-core CPU, likely due to memory locality effects, rather than an increase in GPU speed. Typical absolute performance numbers, in intensive units of particle-timesteps per second (PTPS) – that is, the number of MD timesteps completed per second multiplied by the number of particles in the system – are reported in Table 1. In contrast, spheres are only just beginning to saturate the GPU at 65,000 particles. This finding demonstrates that the feature-based DEM parallelization scheme allows users to take advantage of GPU performance even for relatively small systems. Put another way, this method allows one to simulate the dynamics of a few thousand micron-sized particles for seconds of simulated time in a day on the GPU, rather than a week or more on a single CPU core.

Table 1

Absolute performance of CPU, GPU, and domain-decomposition parallel CPU MPI implementations, in particle-timesteps per second (PTPS). Numbers in parentheses show two standard errors of the mean in the least significant digit. Parallel CPU results are not available for small systems due to the domain decomposition scheme used [30].

Shape	N	$PTPS_{CPU}$	$PTPS_{GPU}$	$PTPS_{CPU,parallel}$
Sphere	256	$3.25(2) \cdot 10^6$	$3.3(3) \cdot 10^6$	N/A
Sphere	4096	$3.137(1) \cdot 10^6$	$4.64(7) \cdot 10^7$	$2.932(1) \cdot 10^7$
Sphere	65 536	$3.07(2) \cdot 10^6$	$1.586(8) \cdot 10^8$	$3.901(5) \cdot 10^7$
Cube	256	$6.50(5) \cdot 10^3$	$1.85(4) \cdot 10^5$	N/A
Cube	4096	$6.49(7) \cdot 10^3$	$3.85(5) \cdot 10^5$	$7.81(8) \cdot 10^4$
Cube	65 536	$5.48(9) \cdot 10^3$	$3.49(5) \cdot 10^5$	$9.04(1) \cdot 10^4$
Icosahedron	256	$1.83(2) \cdot 10^3$	$5.9(1) \cdot 10^4$	N/A
Icosahedron	4096	$1.76(1) \cdot 10^3$	$8.1(2) \cdot 10^4$	$2.21(3) \cdot 10^4$
Icosahedron	65 536	$9.23(6) \cdot 10^2$	$6.88(9) \cdot 10^4$	$2.452(9) \cdot 10^4$

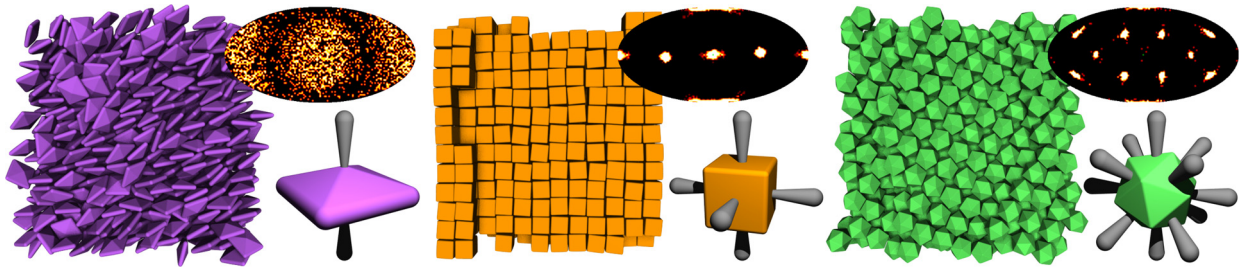


Fig. 5. Self-assembled crystal structures in systems of square bipyramids, cubes, and icosahedra. Bond orientational order diagrams depicting a global histogram of neighboring particle positions are in the upper-right of each snapshot. Particles with cones indicating preferred nearest-neighbor positions are located to the lower-right of each snapshot.

Table 2

Table of simulation parameters used to produce the hard particle assemblies shown in Fig. 5. Parameters include number of particles N , rounding radius compared to edge length r_{round}/r_{edge} , particle mass m , components of inertia tensor $I_{\alpha\beta}$, simulation pressure P , thermostat and barostat time constants τ_T and τ_P , and simulated time t in the NPT ensemble before creating the snapshots shown in the figure. The edge length used for the bipyramid is that of each of the four edges on its equator.

Shape	r_{round}/r_{edge}	m/m_0	$I_{xx}/m_0\sigma_0^2$	$I_{yy}/m_0\sigma_0^2$	$I_{zz}/m_0\sigma_0^2$	$P/k_B T/\sigma_0^2$	t/τ_{tau_0}
Bipyramid	$\frac{1}{6\sqrt{2}} \approx 0.118$	14.3	18.6	18.6	35.0	0.375	$3.6 \cdot 10^4$
Cube	$\frac{1}{6\sqrt{2}} \approx 0.118$	339	1296	1296	1296	0.1	$4.0 \cdot 10^4$
Icosahedron	$\frac{1}{12} \approx 0.0833$	644	4441	4441	4441	0.05	$6.3 \cdot 10^4$

4.3. Assembly

We simulate the self-assembly of shapes into crystals to compare our results to those of hard particle colloidal crystals obtained via MC simulations [2]. We simulate 1,728 square bipyramids, cubes, and icosahedra in the NPT ensemble after thermalization at a dilute gas over a range of pressures, as shown in Fig. 5. Mass and moments of inertia are calculated based on the volume of the Minkowski sum of the polyhedron and a sphere of rounding radius $0.5\sigma_0$ and the inertia tensor of the inner polyhedron without rounding, respectively, using a density of $1 \frac{m_0}{\sigma_0^3}$. We use a standard Martyna–Tobias–Klein NPT thermostat (available as *md.integrate.npt* in HOOMD-Blue) applied to both translational and rotational degrees of freedom [31,32,28] and the barostat computes the pressure from the virial tensor, just as in standard MD [33]. The time constants τ_T for the thermostat and τ_P for the barostat are each $1 \tau_0$ and the thermal energy for the thermostat is set to $1 \frac{k_B T}{\epsilon_0}$. We generate systems with a range of pressures and select the samples with the cleanest assembly behavior to show here. Numerical details of the simulations are presented in Table 2. Similarly to MC results from hard particles with perfectly sharp corners, we find that the wide, flat bipyramids with a height of $\frac{1}{\sqrt{2}}$ relative to their equatorial edge length form a nematic phase, cubes form a simple cubic crystal, and icosahedra form a face-centered cubic crystal. Additionally, Fig. 5 contains particle bond orientational order diagrams (BOODs) inset at right. The BOODs show a histogram of particle positions, at a fixed distance, averaged over every particle in the system and projected onto the surface of a sphere; the sphere is then “unrolled” into a map projection to show the entire diagram. The BOOD gives a qualitative image of the local orientation and environment of each particle relative to its neighbors. Additional information on BOODs, and their role in structure classification, can be found in the references [34,2]. At the lower right, a cone represents preferred nearest-neighbor positions for surrounding particles, which was extracted from the BOOD.

5. Summary

Here we have described a method to simulate conservative interactions of purely repulsive, rounded polytopes in molecular dynamics simulations. Because evaluation of this potential is more intensive than that of a point particle, we are able to more efficiently utilize graphics processing units for smaller system sizes with this potential than with isotropic potentials. Although the simulated shapes are still rounded, this method affords less opportunity for unphysical interdigitation than when rigid bodies of tangent or overlapping disks or spheres are used because the particle surface is smooth rather than rough.

In the future, there are many potentially useful optimizations that could be applied to this method. Currently, the contact point search and pair potential evaluation happen within the same GPU kernel; in three dimensions in particular, this leads to large register usage and low GPU occupancy. By splitting the contact search and force evaluation into two separate steps, some of this inefficiency could be avoided. In general, the search for contact points could also be improved through the use of shape-local cell lists [35], octrees [36], or other spatial data structures.

In summary, the DEM-MD method presented here enables dynamical simulations of faceted particles. With the particular integrators used, the method as implemented is most relevant to the simulation of nanoparticles and colloids in solvents where particle motion is dominated by inertia. Our implementation is available as of version 2.0 of HOOMD-Blue,¹ along with Brownian and Langevin thermostats for modeling non-inertial regimes. Because the interactions are conservative, this method is useful for computation of thermodynamic quantities. Having real dynamical information enables a more direct mapping to studies of nonequilibrium processes such as crystallization, vitrification, jamming, and self-assembly of active matter than Monte Carlo simulations.

Acknowledgements

We thank Richmond Newman for helpful discussion of optimization techniques. This work was supported as part of the Center for Bio-Inspired Energy Science, an Energy Frontier Research Center funded by the US Department of Energy, Office of Science, Basic Energy Sciences under Award DE-SC0000989 and by the National Science Foundation, Division of Materials Research Award # DMR 1409620. M.S. was also supported in part by a National Science Foundation, Integrative Graduate Education and Research Traineeship (IGERT) under Grant DGE 0903629. R.L.M. acknowledges support from the University of Michigan Rackham Merit Fellowship program. Computational resources and services were provided by Advanced Research Computing at the University of Michigan, Ann Arbor. The Glotzer Group at the University of Michigan is an NVIDIA GPU Research Center. Hardware support by NVIDIA Corp. is gratefully acknowledged.

References

- [1] J. Henzie, M. Grünwald, A. Widmer-Cooper, P.L. Geissler, P. Yang, Self-assembly of uniform polyhedral silver nanocrystals into densest packings and exotic superlattices, *Nat. Mater.* 11 (2) (2012) 131–137, <http://dx.doi.org/10.1038/nmat3178>.
- [2] P.F. Damasceno, M. Engel, S.C. Glotzer, Predictive self-assembly of polyhedra into complex structures, *Science* 337 (6093) (2012) 453–457, <http://dx.doi.org/10.1126/science.1220869>.
- [3] K.L. Young, M.L. Personick, M. Engel, P.F. Damasceno, S.N. Barnaby, R. Bleher, T. Li, S.C. Glotzer, B. Lee, C.A. Mirkin, A directional entropic force approach to assemble anisotropic nanoparticles into superlattices, *Angew. Chem., Int. Ed. Engl.* (2013) 13980–13984, <http://dx.doi.org/10.1002/anie.201306009>.
- [4] J.A. Anderson, E. Jankowski, T.L. Grubb, M. Engel, S.C. Glotzer, Massively parallel Monte Carlo for many-particle simulations on GPUs, *J. Comput. Phys.* 254 (2013) 27–38, <http://dx.doi.org/10.1016/j.jcp.2013.07.023>.
- [5] J.A. Anderson, M.E. Irrgang, S.C. Glotzer, Scalable Metropolis Monte Carlo for simulation of hard shapes, *Comput. Phys. Commun.* 204 (2016) 21–30, <http://dx.doi.org/10.1016/j.cpc.2016.02.024>.
- [6] R. Ni, S. Belli, R. van Roij, M. Dijkstra, Glassy dynamics, spinodal fluctuations, and the kinetic limit of nucleation in suspensions of colloidal hard rods, *Phys. Rev. Lett.* 105 (8) (2010) 088302, <http://dx.doi.org/10.1103/PhysRevLett.105.088302>.
- [7] H.H. Wensink, V. Kantsler, R.E. Goldstein, J. Dunkel, Controlling active self-assembly through broken particle-shape symmetry, *Phys. Rev. E* 89 (1) (2014) 10302, <http://dx.doi.org/10.1103/PhysRevE.89.010302>.
- [8] M. Spellings, M. Engel, D. Klotz, S. Sabrina, A.M. Drews, N.H.P. Nguyen, K.J.M. Bishop, S.C. Glotzer, Shape control and compartmentalization in active colloidal cells, *Proc. Natl. Acad. Sci. USA* 112 (34) (2015) E4642–E4650, <http://dx.doi.org/10.1073/pnas.1513361112>.
- [9] M.N. Bannerman, R. Sargant, L. Lue, Dynamo: a free $O(N)$ general event-driven molecular dynamics simulator, *J. Comput. Chem.* 32 (15) (2011) 3329–3338, <http://dx.doi.org/10.1002/jcc.21915>.
- [10] M. Marín, Event-driven hard-particle molecular dynamics using bulk-synchronous parallelism, *Comput. Phys. Commun.* 102 (1–3) (1997) 81–96, [http://dx.doi.org/10.1016/S0010-4655\(97\)00011-8](http://dx.doi.org/10.1016/S0010-4655(97)00011-8).
- [11] S. Miller, S. Luding, Event-driven molecular dynamics in parallel, *J. Comput. Phys.* 193 (1) (2004) 306–316, <http://dx.doi.org/10.1016/j.jcp.2003.08.009>.
- [12] F. Smallenburg, L. Filion, M. Marechal, M. Dijkstra, Vacancy-stabilized crystalline order in hard cubes, *Proc. Natl. Acad. Sci. USA* 109 (44) (2012) 17886–17890, <http://dx.doi.org/10.1073/pnas.1211784109>.
- [13] B.S. John, A. Stroock, F.A. Escobedo, Cubatic liquid–crystalline behavior in a system of hard cuboids, *J. Chem. Phys.* 120 (19) (2004) 9383–9389, <http://dx.doi.org/10.1063/1.1711594>.
- [14] C.L. Phillips, J.A. Anderson, G. Huber, S.C. Glotzer, Optimal filling of shapes, *Phys. Rev. Lett.* 108 (19) (2012) 198304, <http://dx.doi.org/10.1103/PhysRevLett.108.198304>.
- [15] J. Ghaboussi, R. Barbosa, Three-dimensional discrete element method for granular materials, *Int. J. Numer. Anal. Methods Geomech.* 14 (June 1987) (1990) 451–472, <http://dx.doi.org/10.1002/nag.1610140702>.

¹ HOOMD-blue web page: <http://codeblue.umich.edu/hoomd-blue>.

- [16] S. Galindo-Torres, F. Alonso-Marroquín, Y. Wang, D. Pedroso, J.M. Castano, Molecular dynamics simulation of complex particles in three dimensions and the study of friction due to nonconvexity, *Phys. Rev. E* 79 (6) (2009) 060301, <http://dx.doi.org/10.1103/PhysRevE.79.060301>.
- [17] F. Alonso-Marroquín, Y. Wang, An efficient algorithm for granular dynamics simulations with complex-shaped objects, *Granul. Matter* 11 (5) (2009) 317–329, <http://dx.doi.org/10.1007/s10035-009-0139-1>.
- [18] J. Wang, H.S. Yu, P. Langston, F. Fraige, Particle shape effects in discrete element modelling of cohesive angular particles, *Granul. Matter* 13 (1) (2010) 1–12, <http://dx.doi.org/10.1007/s10035-010-0217-4>.
- [19] S. Mack, P. Langston, C. Webb, T. York, Experimental validation of polyhedral discrete element model, *Powder Technol.* 214 (3) (2011) 431–442, <http://dx.doi.org/10.1016/j.powtec.2011.08.043>.
- [20] P. Langston, J. Ai, H.-S. Yu, Simple shear in 3D DEM polyhedral particles and in a simplified 2D continuum model, *Granul. Matter* (2013) 13–15, <http://dx.doi.org/10.1007/s10035-013-0421-0>.
- [21] M. Botton, E. Azéma, N. Estrada, F. Radjaï, A. Lizcano, Quasistatic rheology and microstructural description of sheared granular materials composed of platy particles, *Phys. Rev. E* 87 (2013) 032206, <http://dx.doi.org/10.1103/PhysRevE.87.032206>.
- [22] J. Anderson, C. Lorenz, A. Travesset, General purpose molecular dynamics simulations fully implemented on graphics processing units, *J. Comput. Phys.* 227 (10) (2008) 5342–5359, <http://dx.doi.org/10.1016/j.jcp.2008.01.047>.
- [23] K.J.M. Bishop, C.E. Wilmer, S. Soh, B.A. Grzybowski, Nanoscale forces and their uses in self-assembly, *Small* 5 (14) (2009) 1600–1630, <http://dx.doi.org/10.1002/sml.200900358>.
- [24] T.D. Nguyen, E. Jankowski, S.C. Glotzer, Self-assembly and reconfigurability of shape-shifting particles, *ACS Nano* 5 (11) (2011) 8892–8903, <http://dx.doi.org/10.1021/nn203067y>.
- [25] A. Rahman, F.H. Stillinger, Molecular dynamics study of liquid water, *J. Chem. Phys.* 55 (7) (1971) 3336–3359, <http://dx.doi.org/10.1063/1.1676585>.
- [26] D. Chandler, J. Weeks, H. Andersen, Van der Waals picture of liquids, solids, and phase transformations, *Science* 220 (1983) 787–794.
- [27] M.P. Allen, D.J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, 1989.
- [28] H. Kamberaj, R.J. Low, M.P. Neal, Time reversible and symplectic integrators for molecular dynamics simulations of rigid molecules, *J. Chem. Phys.* 122 (2005) 224114, <http://dx.doi.org/10.1063/1.1906216>.
- [29] J.A. Anderson, S.C. Glotzer, The development and expansion of HOOMD-blue through six years of GPU, arXiv:1308.5587.
- [30] J. Glaser, T.D. Nguyen, J.A. Anderson, P. Lui, F. Spiga, J.A. Millan, D.C. Morse, S.C. Glotzer, Strong scaling of general-purpose molecular dynamics simulations on GPUs, *Comput. Phys. Commun.* 192 (2015) 97–107, <http://dx.doi.org/10.1016/j.cpc.2015.02.028>.
- [31] G.J. Martyna, D.J. Tobias, M.L. Klein, Constant pressure molecular dynamics algorithms, *J. Chem. Phys.* 101 (5) (1994) 4177, <http://dx.doi.org/10.1063/1.467468>.
- [32] J. Cao, G.J. Martyna, Adiabatic path integral molecular dynamics methods, II: algorithms, *J. Chem. Phys.* 104 (5) (1996) 2028, <http://dx.doi.org/10.1063/1.470959>.
- [33] M.E. Tuckerman, J. Alejandre, R. López-Rendón, A.L. Jochim, G.J. Martyna, A Liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal–isobaric ensemble, *J. Phys. A, Math. Gen.* 39 (2006) 5629–5651, <http://dx.doi.org/10.1088/0305-4470/39/19/S18>.
- [34] J. Roth, A.R. Denton, Solid-phase structures of the Dzugutov pair potential, *Phys. Rev. E* 61 (6) (2000) 6845–6857, <http://dx.doi.org/10.1103/PhysRevE.61.6845>.
- [35] L. Verlet, Computer “experiments” on classical fluids, I: thermodynamical properties of Lennard–Jones molecules, *Phys. Rev.* 159 (1967) 98–103, <http://dx.doi.org/10.1103/PhysRev.159.98>.
- [36] D. Meagher, Geometric modeling using octree encoding, *Comput. Graph. Image Process.* 19 (1982) 129–147, [http://dx.doi.org/10.1016/0146-664X\(82\)90104-6](http://dx.doi.org/10.1016/0146-664X(82)90104-6).