

Speeding up Monte Carlo simulation of patchy hard cylinders^{*}

Alberto Giacomo Orellana, Emanuele Romani, and Cristiano De Michele^a

Dipartimento di Fisica, “Sapienza” Università di Roma, P.le A. Moro 2, 00185 Roma, Italy

Received 9 February 2018 and Received in final form 26 March 2018

Published online: 16 April 2018 – © EDP Sciences / Società Italiana di Fisica / Springer-Verlag 2018

Abstract. The hard cylinder model decorated with attractive patches proved to be very useful recently in studying physical properties of several colloidal systems. Phase diagram, elastic constants and cholesteric properties obtained from computer simulations based on a simple hard cylinder model have been all successfully and quantitatively compared to experimental results. Key to these simulations is an efficient algorithm to check the overlap between hard cylinders. Here, we propose two algorithms to check the hard cylinder overlap and we assess their efficiency through a comparison with an existing method available in the literature and with the well-established algorithm for simulating hard spherocylinders. In addition, we discuss a couple of optimizations for performing computer simulations of patchy anisotropic particles and we estimate the speed-up which they can provide in the case of patchy hard cylinders.

1 Introduction

Coarse-grained models which comprise hard rigid bodies decorated with attractive sites (also called *sticky spots*) have been widely used over the last years to study colloidal systems [1–5]. While hard rigid bodies without attraction were crucial to assess the theoretical possibility of entropy-driven phase transitions such as the liquid-to-solid of hard spheres [6] and isotropic-to-nematic [7, 8] ones, aggregation introduces a further level of complexity. Indeed, attractive interactions between hard particles allow their self-assembly, *i.e.* the spontaneous formation through free energy minimization of reversible aggregates.

A prominent self-assembly process is the formation of linear aggregates induced by the presence of two attractive sites on colloidal particles of anisotropic shape. Examples are provided by micellar systems [9–11], formation of fibers and fibrils [12–15], solution of short (with 6 to 20 base pairs) [16–18], long (with 10^2 to 10^6 base pairs) [19–22] and gapped [23] B-form DNA duplexes, filamentous viruses [24–27], chromonic liquid crystals [28] and inorganic nanoparticles [29]. In all these systems linear aggregates are sufficiently stiff to give rise to liquid crystalline (LC) phases, such as nematic, smectic and columnar phases, thus making their phase diagram rather rich.

Many theoretical approaches have been developed to study the isotropic-nematic transition as a result of reversible physical polymerization and collective ordering into the nematic LC phase in these systems [30–34]. The

hard cylinder (HC) model complemented with two attractive sites proved to be crucial to assess the validity of these theories [32, 33] through Metropolis Monte Carlo simulations. The bottleneck of these simulations is the test of overlap of two hard cylinders, therefore an efficient algorithm to perform this overlap check is key for the efficiency of these simulations.

After a brief description, in sect. 2.1, of HC models which we use in the present study, we describe in sect. 2.2 two novel algorithms for testing HCs overlap. In sect. 3.1 we assess their consistency and in sect. 3.2 we compare their performance i) with the algorithm proposed in ref. [35] (which is based on the same ideas of the one used in ref. [36]) and ii) with a standard algorithm to simulate hard spherocylinders (HSCs). We also illustrate in sects. 2.3.1 and 2.3.2 two useful tricks, namely bounding boxes (BBs) and multiple linked cell lists (MLLs), for speeding up simulations of patchy anisotropic particles, whose performance will be discussed in sect. 3.3. In sect. 4 the conclusions will be drawn.

2 Methods

2.1 Models

In all the simulations which we carried out for testing BBs and MLLs we used the patchy HC model shown in fig. 1(a). Cylinders have length L and diameter D and their aspect ratio is defined as $X_0 = L/D$. Each HC is also decorated with two attractive sites on its bases. These two attractive sites are located along the symmetry axis of the HC at a distance $X_0D/2 + 0.0091D$ from its center of mass.

^{*} Contribution to the Topical Issue “Advances in Computational Methods for Soft Matter Systems” edited by Lorenzo Rovigatti, Flavio Romano, John Russo.

^a e-mail: cristiano.demichela@uniroma1.it

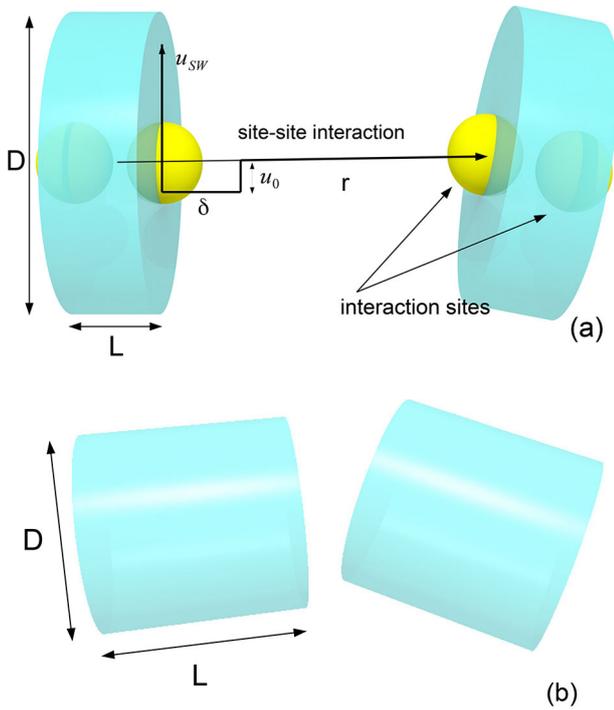


Fig. 1. Models used in our MC simulations. The patchy HC model (a) has been used to check the performance of MLLs and BBs methods, while HC without patches (b) have been used to test the efficiency of algorithms A1, A2 and A3.

Sites belonging to distinct particles interact via a square-well (SW) potential βu_{SW} , such that $\beta u_{SW} = -\beta u_0$, if $r < \delta$ and $\beta u_{SW} = 0$, if $r > \delta$, where r is the distance between interacting sites, $\delta = 0.273D$ is the interaction range (which corresponds to the diameter of the yellow spheres in fig. 1(a)) and βu_0 is the ratio between the binding energy and the thermal energy $k_B T$ with k_B the Boltzmann constant. We will make use of the adimensional temperature $T^* = k_B T / u_0$. Note that the patch position and size ensure that no branching occurs in the system. In sect. 3.2, where we test the efficiency of the three algorithms discussed in the present study, we will employ HC without patches (see fig. 1(b)) to carry out MC simulations.

2.2 Overlap of hard cylinders

In computer simulations of HCs the most time-consuming part is the test of their overlap. In the next three sections we describe three algorithms: a first algorithm (A1) which is exactly the one proposed in ref. [35] except for a small fix, a second algorithm (A2) which is a simplified and faster variant of A1 and, finally, a third algorithm (A3) where the most time-consuming part of A1 and A2 is completely new.

2.2.1 Algorithm A1

In this section, for reader's convenience, we outline the algorithm which has been proposed in ref. [35] and which we

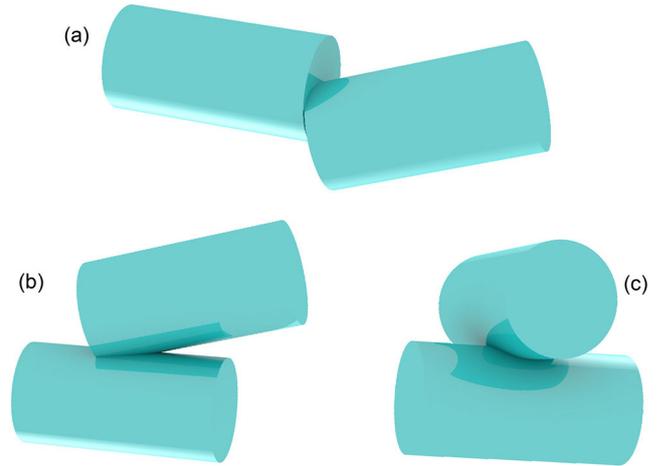


Fig. 2. Each HC is made of a rim and two disks, hence three different possible overlap configurations between two HCs can be identified: disk-disk (a), rim-disk (b) and rim-rim (c).

implemented as a reference for the two novel algorithms which we propose in this paper. We also suggest a small change to the algorithm in ref. [35] to fix convergence problems which emerged using the original implementation.

Let us consider two hard cylinders of length L and diameter D , which will be labelled 1 and 2. \mathbf{C}_1 and \mathbf{C}_2 will be their centers of mass and $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{u}}_2$ their orientations. If HCs are parallel, *i.e.* if $\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2 = \pm 1$, then they overlap if the following conditions are true:

$$\begin{cases} |\Delta \mathbf{C}_{\parallel}| < L, \\ |\Delta \mathbf{C}_{\perp}| < D, \end{cases} \quad (1)$$

where

$$\begin{cases} \Delta \mathbf{C}_{\parallel} = (\mathbf{C}_2 - \mathbf{C}_1) \cdot \hat{\mathbf{u}}_1, \\ \Delta \mathbf{C}_{\perp} = (\mathbf{C}_2 - \mathbf{C}_1) - |(\mathbf{C}_2 - \mathbf{C}_1) \cdot \hat{\mathbf{u}}_1| \hat{\mathbf{u}}_1. \end{cases} \quad (2)$$

Otherwise, if $\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2 \neq \pm 1$, the overlap test will be articulated in three main steps, namely:

- 1) The disk-disk overlap is checked by considering all the possible pairs of bases of the two cylinders (see fig. 2(a)).
- 2) If in the previous step no overlap is detected, one checks all possible rim-disk overlaps (see fig. 2(b)).
- 3) If in the previous two steps, no overlap is found, one has to finally check for rim-rim overlaps (see fig. 2(c)).

Now we describe in details the three steps just enumerated.

2.2.2 Disk-disk overlap

Let us consider two disks 1 and 2 of diameter D , whose centers of mass are \mathbf{D}_1 and \mathbf{D}_2 . Disk 1 can be any of the two disks belonging to cylinder 1 and disk 2 can be any of the two disks of cylinder 2. Each disk lies on a given

plane and the straight line Γ is the intersection of such two planes. Be \mathbf{P}_1 and \mathbf{P}_2 the two points on Γ which are closest to \mathbf{D}_1 and \mathbf{D}_2 , *i.e.* the two points which can be obtained by minimizing the quantity $|\mathbf{P}_1 - \mathbf{D}_1|$ with the constraints $(\mathbf{P}_1 - \mathbf{D}_1) \cdot \hat{\mathbf{u}}_1 = 0$ and $(\mathbf{P}_1 - \mathbf{D}_2) \cdot \hat{\mathbf{u}}_2 = 0$, which yields:

$$\mathbf{P}_1 = \frac{1}{|\mathbf{N}|^2} [(\mathbf{D}_1 \cdot \mathbf{N}) \mathbf{N} + (\mathbf{D}_1 \cdot \hat{\mathbf{u}}_1) (\hat{\mathbf{u}}_2 \wedge \mathbf{N}) - (\mathbf{D}_2 \cdot \hat{\mathbf{u}}_2) (\hat{\mathbf{u}}_1 \wedge \mathbf{N})], \quad (3)$$

$$\mathbf{P}_2 = \frac{1}{|\mathbf{N}|^2} [(\mathbf{D}_2 \cdot \mathbf{N}) \mathbf{N} + (\mathbf{D}_1 \cdot \hat{\mathbf{u}}_1) (\hat{\mathbf{u}}_2 \wedge \mathbf{N}) - (\mathbf{D}_2 \cdot \hat{\mathbf{u}}_2) (\hat{\mathbf{u}}_1 \wedge \mathbf{N})]. \quad (4)$$

A necessary condition for the two disks to overlap is the following one:

$$\begin{aligned} |\mathbf{P}_1 - \mathbf{D}_1| &< D/2, \\ |\mathbf{P}_2 - \mathbf{D}_2| &< D/2. \end{aligned} \quad (5)$$

Note that both conditions in eqs. (5) must be true. If both conditions in eqs. (5) occur, then the straight line Γ intersects the circumference of disk 1 in two points symmetrically with respect to segment $\overline{\mathbf{P}_1 \mathbf{D}_1}$, choose one of these two points and call it \mathbf{S}_1 . Similarly, the straight line Γ intersects the circumference of disk 2 in other two points symmetrically with respect to segment $(\mathbf{P}_2 - \mathbf{D}_2)$. Let us pick one of this two points and call it \mathbf{S}_2 . The two disks 1 and 2 overlap if and only if the following condition occurs:

$$|\mathbf{P}_1 - \mathbf{P}_2| \leq |\mathbf{S}_1 - \mathbf{P}_1| + |\mathbf{S}_2 - \mathbf{P}_2|, \quad (6)$$

where

$$\begin{aligned} |\mathbf{S}_1 - \mathbf{P}_1| &= \sqrt{\frac{D^2}{4} - |\mathbf{P}_1 - \mathbf{D}_1|^2}, \\ |\mathbf{S}_2 - \mathbf{P}_2| &= \sqrt{\frac{D^2}{4} - |\mathbf{P}_2 - \mathbf{D}_2|^2}. \end{aligned} \quad (7)$$

2.2.3 Disk-rim overlap

If no disk-disk overlap is found, one has to check for rim-disk overlaps. Let us consider the disk 1 whose center is \mathbf{D}_1 and the rim of cylinder 2. Be \mathbf{U} the point on the axis of cylinder 2 which is at the closest distance from \mathbf{D}_1 , *i.e.*

$$\mathbf{U} = \mathbf{C}_2 + [(\mathbf{D}_1 - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2] \hat{\mathbf{u}}_2. \quad (8)$$

There is no overlap if either the sphere enclosing disk 1 does not intersect the rim of cylinder 2, *i.e.* $|\mathbf{D}_1 - \mathbf{U}| > D$ or the center of the disk is located inside the region which can be obtained by prolonging cylinder 2 along its axis, *i.e.* if the following conditions are fulfilled:

$$\begin{cases} |\mathbf{D}_1 - \mathbf{U}| \leq \frac{D}{2}, \\ |(\mathbf{D}_1 - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2| > \frac{L}{2}. \end{cases} \quad (9)$$

On the contrary, one has an overlap if the center of disk 1, \mathbf{D}_1 , is inside cylinder 2, *i.e.*

$$\begin{cases} |\mathbf{D}_1 - \mathbf{U}| \leq \frac{D}{2}, \\ |(\mathbf{D}_1 - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2| \leq \frac{L}{2}. \end{cases} \quad (10)$$

If checks done so far do not establish whether the two cylinders overlap or not, an iterative scheme can be used. Be \mathbf{A} a point on the axis of cylinder 2 and \mathbf{T} a point onto the circumference of disk 2, such that the distance between \mathbf{A} and \mathbf{T} is at a minimum. \mathbf{T} can be determined by minimizing the quantity $|\mathbf{T} - \mathbf{A}|$ with the constraints $|\mathbf{T} - \mathbf{D}_1|^2 = \frac{D^2}{4}$ and $(\mathbf{T} - \mathbf{D}_1) \cdot \hat{\mathbf{u}}_1 = 0$, thus obtaining:

$$\mathbf{T} = \mathbf{D}_1 \pm \frac{D}{2} \frac{\mathbf{A} - \mathbf{D}_1 - [(\mathbf{A} - \mathbf{D}_1) \cdot \hat{\mathbf{u}}_1] \hat{\mathbf{u}}_1}{\sqrt{|(\mathbf{A} - \mathbf{D}_1) \wedge \hat{\mathbf{u}}_1|^2}}. \quad (11)$$

Hence, there is overlap if:

$$\begin{cases} |T_{\parallel}| \leq \frac{L}{2}, \\ |T_{\perp}| \leq \frac{D}{2}, \end{cases} \quad (12)$$

where

$$T_{\parallel} = (\mathbf{T} - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2$$

and

$$T_{\perp} = (\mathbf{T} - \mathbf{C}_2) - [(\mathbf{T} - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2] \hat{\mathbf{u}}_2.$$

The latter condition means that there is overlap if and only if point \mathbf{T} is inside cylinder 2. An iterative scheme to find \mathbf{A} and \mathbf{T} is the following one:

- 1) An initial guess \mathbf{A}_i for \mathbf{A} is calculated as follows: $\mathbf{A}_i = \mathbf{C}_2 - [(\mathbf{C}_2 - \mathbf{D}_1) \cdot \hat{\mathbf{u}}_2] \hat{\mathbf{u}}_2$. In passing we note that in ref. [35] a different and simpler guess for \mathbf{A} is used, which anyway does not ensure a proper convergence of this iterative scheme.
- 2) From eq. (11) one defines the points \mathbf{T}_+ and \mathbf{T}_- as follows:

$$\begin{aligned} \mathbf{T}_+ &= \mathbf{D}_1 + \frac{D}{2} \frac{[\mathbf{A} - \mathbf{D}_1 - ((\mathbf{A} - \mathbf{D}_1) \cdot \hat{\mathbf{u}}_1) \hat{\mathbf{u}}_1]}{\sqrt{|(\mathbf{A} - \mathbf{D}_1) \wedge \hat{\mathbf{u}}_1|^2}}, \\ \mathbf{T}_- &= \mathbf{D}_1 - \frac{D}{2} \frac{[\mathbf{A} - \mathbf{D}_1 - ((\mathbf{A} - \mathbf{D}_1) \cdot \hat{\mathbf{u}}_1) \hat{\mathbf{u}}_1]}{\sqrt{|(\mathbf{A} - \mathbf{D}_1) \wedge \hat{\mathbf{u}}_1|^2}}. \end{aligned}$$

- 3) If $T_{+\perp} < T_{-\perp}$, then one sets $\mathbf{T}_{new} = \mathbf{T}_+$, otherwise if $T_{+\perp} > T_{-\perp}$, then $\mathbf{T}_{new} = \mathbf{T}_-$;
- 4) A new value of \mathbf{A} is set, *i.e.* $\mathbf{A} = T_{new\parallel} \hat{\mathbf{u}}_2 + \mathbf{C}_2$, where $T_{new\parallel} = (\mathbf{T}_{new} - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2$;
- 5) Continue starting from step 2) until the variation of $|\mathbf{T} - \mathbf{A}|$ is less than a given threshold.

2.2.4 Rim-rim overlap

If previous checks do not establish or exclude the overlap of the two cylinders, rim-rim overlaps must be checked. Be

\mathbf{V}_1 and \mathbf{V}_2 two points lying on the axis of cylinder 1 and on that of cylinder 2, respectively, such that $|\mathbf{V}_1 - \mathbf{V}_2|$ is at a minimum. These two points can be calculated by solving analytically the following system of equations:

$$\begin{cases} (\mathbf{V}_1 - \mathbf{V}_2) \cdot \hat{\mathbf{u}}_1 = 0, \\ (\mathbf{V}_1 - \mathbf{V}_2) \cdot \hat{\mathbf{u}}_2 = 0, \end{cases}$$

from which one obtains that

$$\begin{aligned} \mathbf{V}_1(\lambda_1) &= \mathbf{C}_1 + \lambda_1 \hat{\mathbf{u}}_1, \\ \mathbf{V}_2(\lambda_2) &= \mathbf{C}_2 + \lambda_2 \hat{\mathbf{u}}_2, \end{aligned}$$

where, if $\mathbf{C}_{12} \equiv \mathbf{C}_1 - \mathbf{C}_2$,

$$\begin{aligned} \lambda_1 &= \frac{1}{(\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2)^2 - 1} [\mathbf{C}_{12} \cdot \hat{\mathbf{u}}_1 + (\mathbf{C}_{12} \cdot \hat{\mathbf{u}}_2) (\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2)], \\ \lambda_2 &= \frac{1}{(\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2)^2 - 1} [-\mathbf{C}_{12} \cdot \hat{\mathbf{u}}_2 + (\mathbf{C}_{12} \cdot \hat{\mathbf{u}}_2) (\hat{\mathbf{u}}_1 \cdot \hat{\mathbf{u}}_2)]. \end{aligned}$$

It can be seen that rim-rim overlap occurs if and only if:

$$\begin{aligned} |\mathbf{V}_1 - \mathbf{V}_2| &\leq D, \\ |\lambda_1| &\leq \frac{L}{2}, \\ |\lambda_2| &\leq \frac{L}{2}. \end{aligned}$$

2.2.5 Algorithm A2

Rim-rim and disk-disk overlap checks are carried out as in algorithm A1, while we suggest some modifications of the disk-rim overlap check, which is the most time-consuming part of algorithm A1.

First we note that in the rim-disk overlap check of A1 both disks of each cylinder are checked for a possible overlap with the rim of the other cylinder. Anyway, one can check only the disk which is closest to the symmetry axis of the other cylinder, thus halving the number of checks to carry out. We also suggest to change the order of checks as follows: first disk-disk and rim-rim checks are performed and afterwards the disk-rim one. The latter order of overlap checks could be particularly beneficial in denser phases, such as nematic, smectic or columnar ones, where rim-rim and disk-disk overlaps are more likely. Changing the order of disk-disk and rim-rim overlap checks does not provide any benefit though. Finally, we propose a simplification of the iterative procedure which is used in the disk-rim overlap check of A1.

As in the disk-rim overlap check of algorithm A1 we first verify whether conditions in eqs. (9) and (10) are fulfilled, if not we proceed as described in the following. Let us consider one of the two disks of cylinder 1 and the rim of cylinder 2, we propose the following simpler approach to check their overlap (see fig. 3):

1) Calculate $\mathbf{A}_0 = \mathbf{C}_2 + [(\mathbf{D}_1 - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2] \hat{\mathbf{u}}_2$. Set $A_i = A_0$.

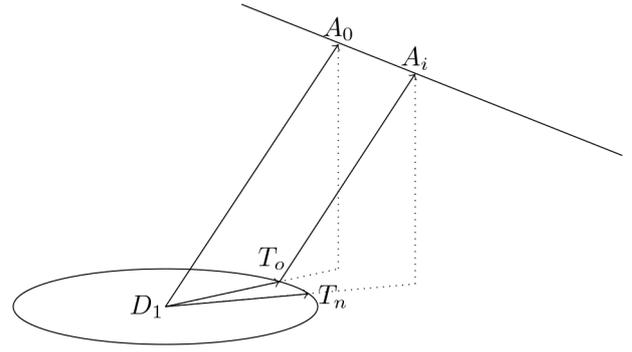


Fig. 3. Graphical representation of the first two steps of the iterative procedure for checking the disk-rim overlap in algorithm A2. An initial guess A_0 is calculated by which point T_o on the disk is built. From T_o one calculates A_i on the cylinder axis and then by using A_i one builds a new point T_n on the disk.

2) Calculate the component \mathbf{J}_{\parallel} of vector $\mathbf{J} = \mathbf{A}_i - \mathbf{D}_1$ onto the plane of disk 1, *i.e.*

$$\mathbf{J}_{\parallel} = \mathbf{J} - (\mathbf{J} \cdot \hat{\mathbf{u}}_1) \hat{\mathbf{u}}_1.$$

3) If we have already calculated \mathbf{T}_n once, then set $\mathbf{T}_o = \mathbf{T}_n$.

4) Calculate the point \mathbf{T}_n as the intersection of vector \mathbf{J}_{\parallel} applied to \mathbf{C}_1 with the circumference of disk 1, *i.e.*

$$\mathbf{T}_n = \mathbf{D}_1 + \frac{1}{2} D \mathbf{J}_{\parallel} / |\mathbf{J}_{\parallel}|. \quad (13)$$

5) If point \mathbf{T}_n does not vary appreciably from \mathbf{T}_o , *i.e.*

$$\max_{\alpha} \left\{ \frac{|T_{n,\alpha} - T_{o,\alpha}|}{\max\{1, |T_{n,\alpha}|\}} \right\} \leq \epsilon,$$

where $T_{n,\alpha}$ ($T_{o,\alpha}$) is the component α (with $\alpha \in \{x, y, z\}$) of vector \mathbf{T}_n (\mathbf{T}_o), then terminate, since convergence has been reached (in our simulations we used $\epsilon = 10^{-14}$).

6) Set \mathbf{A}_i as the intersection of the line which passes through \mathbf{T}_n and which is perpendicular to $\hat{\mathbf{u}}_2$, *i.e.*

$$\mathbf{A}_i = \mathbf{C}_2 + \hat{\mathbf{u}}_2 (\mathbf{T}_n - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2. \quad (14)$$

7) Go to step 2).

After convergence is reached one has to check whether the point \mathbf{T}_n is inside cylinder 2, *i.e.*

$$\begin{cases} |\mathbf{T}_n - \hat{\mathbf{u}}_2 (\mathbf{T}_n - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2| \leq D/2, \\ |(\mathbf{T}_n - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2| \leq L/2. \end{cases} \quad (15)$$

If conditions in eqs. (15) are fulfilled then the two cylinders overlap. We note that this iterative procedure is equivalent to the one in algorithm A1, if one considers only \mathbf{T}_+ in steps 2) and 3) of the disk-rim overlap check of A1. Indeed, solution \mathbf{T}_- corresponds to a maximum of the distance and it can be safely discarded in each iteration as we do.

2.2.6 Algorithm A3

The last algorithm is identical to algorithm A2 except that we propose a different and faster method to check for the disk-rim overlap which is based on the explicit calculation of the intersection points of a disk and a rim.

Given two cylinders 1 and 2 of the same diameter D whose centers are \mathbf{C}_1 and \mathbf{C}_2 and whose orientations are $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{u}}_2$, consider one of the two disks belonging to cylinder 1, whose center is \mathbf{D}_1 , and the rim of cylinder 2. An intersection point $\mathbf{r} = (x, y, z)$ of disk and rim must fulfill the following conditions:

$$\begin{aligned} |\mathbf{r} - \mathbf{D}_1| &= D/2, \\ (\mathbf{r} - \mathbf{D}_1) \cdot \hat{\mathbf{u}}_1 &= 0, \\ |\mathbf{r} - \mathbf{C}_2 - \hat{\mathbf{u}}_2 (\mathbf{r} - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2| &= D/2. \end{aligned} \quad (16)$$

To simplify these equations we consider a new reference system with the origin in the center of the disk whose axes $\hat{\mathbf{x}}'$, $\hat{\mathbf{y}}'$ and $\hat{\mathbf{z}}'$ are built as follows:

- 1) Axis $\hat{\mathbf{x}}'$ is perpendicular to the disk, *i.e.* $\hat{\mathbf{x}}' = \hat{\mathbf{u}}_1$.
- 2) Axis $\hat{\mathbf{y}}'$ is obtained with the following procedure: if \mathbf{A}_i is the point calculated in step 1) of the iterative procedure of algorithm A2, *i.e.*

$$\mathbf{A}_i = \mathbf{C}_2 + [(\mathbf{D}_1 - \mathbf{C}_2) \cdot \hat{\mathbf{u}}_2] \hat{\mathbf{u}}_2,$$

then if $\mathbf{J} = \mathbf{A}_i - \mathbf{D}_1 - \hat{\mathbf{u}}_1 (\mathbf{A}_i - \mathbf{D}_1) \cdot \hat{\mathbf{u}}_1$, one sets

$$\hat{\mathbf{y}}' = \frac{\mathbf{J}}{|\mathbf{J}|}.$$

- 3) Finally, axis $\hat{\mathbf{z}}'$ is the cross product of $\hat{\mathbf{x}}'$ and $\hat{\mathbf{y}}'$, *i.e.*

$$\hat{\mathbf{z}}' = \hat{\mathbf{x}}' \wedge \hat{\mathbf{y}}'.$$

In the new reference system one has $\mathbf{C}_2 = (C_{2,x'}, C_{2,y'}, C_{2,z'})$, $\hat{\mathbf{u}}_2 = (u_{2,x'}, u_{2,y'}, u_{2,z'})$ and $\mathbf{r} = (x', y', z')$. Equations (16) reduce to

$$y'^2 + z'^2 = \frac{D^2}{4}, \quad (17)$$

$$c_0 y'^2 + c_1 z'^2 + c_2 y' z' + c_3 + c_4 y' + c_5 z' = 0, \quad (18)$$

where

$$\begin{aligned} c_0 &= 1 - u_{2,y'}^2, \\ c_1 &= 1 - u_{2,z'}^2, \\ c_2 &= -2u_{2,y'} u_{2,z'}, \\ c_3 &= C_{2,x'}^2 (1 - u_{2,x'}^2) + C_{2,y'}^2 c_0 + C_{2,z'}^2 c_1 \\ &\quad - 2(C_{2,x'} C_{2,y'} u_{2,x'} u_{2,y'} + C_{2,x'} C_{2,z'} u_{2,x'} u_{2,z'} \\ &\quad + C_{2,y'} C_{2,z'} u_{2,y'} u_{2,z'}) - \frac{D^2}{4}, \\ c_4 &= 2(C_{2,z'} u_{2,y'} u_{2,z'} + C_{2,x'} u_{2,x'} u_{2,y'} - C_{2,y'} c_0), \\ c_5 &= 2(C_{2,x'} u_{2,x'} u_{2,z'} + C_{2,y'} u_{2,y'} u_{2,z'} - C_{2,z'} c_1). \end{aligned}$$

Note that eq. (17) is that of a circumference of radius $D/2$ lying on the $y'z'$ -plane and centered in the origin of the

reference system, while eq. (18) is that of a generic ellipse which lies on the $y'z'$ -plane too. If we scale the $\hat{\mathbf{y}}'$ and $\hat{\mathbf{z}}'$ axis by a factor $D/2$, *i.e.* we apply the transformation

$$\begin{aligned} y'' &= \frac{y'}{D/2}, \\ z'' &= \frac{z'}{D/2}, \end{aligned}$$

then eqs. (17) and (18) become:

$$y''^2 + z''^2 = 1, \quad (19)$$

$$d_0 y''^2 + d_1 z''^2 + d_2 y'' z'' + d_3 + d_4 y'' + d_5 z'' = 0, \quad (20)$$

where

$$\begin{aligned} d_0 &= c_0 (D/2)^2, \\ d_1 &= c_1 (D/2)^2, \\ d_2 &= c_2 (D/2)^2, \\ d_3 &= c_3, \\ d_4 &= c_4 (D/2), \\ d_5 &= c_5 (D/2). \end{aligned}$$

Now, if we solve for y'' in eq. (19) and we plug the resulting expression of y'' in eq. (20), one obtains

$$y'' = \frac{(d_0 - d_1) z''^2 - d_5 z'' - d_0 - d_3}{d_2 z'' + d_4}, \quad (21)$$

$$q_4 z''^4 + q_3 z''^3 + q_2 z''^2 + q_1 z'' + q_0 = 0, \quad (22)$$

where

$$\begin{aligned} q_0 &= d_0^2 + 2d_0 d_3 + d_3^2 - d_4^2, \\ q_1 &= -2d_2 d_4 + 2d_0 d_5 + 2d_3 d_5, \\ q_2 &= -2d_0^2 + 2d_0 d_1 - d_2^2 - 2d_0 d_3 + 2d_1 d_3 + d_4^2 + d_5^2, \\ q_3 &= 2d_2 d_4 - 2d_0 d_5 + 2d_1 d_5, \\ q_4 &= d_0^2 - d_0 d_1 + d_1^2 + d_2^2. \end{aligned}$$

It is well known that the analytic solution of eq. (22) is numerically inaccurate due to roundoff errors [37, 38] and as such it cannot be used in practice. At present, several alternative quartic solvers exist, which are quite fast and numerically robust [39–42]. In the present work, building on the quartic solver proposed in ref. [41], we developed our own quartic solver which proved to be very fast and resilient to numerical errors. It will be thoroughly described in a forthcoming publication where we will compare it against other solvers. The efficiency of the quartic solver is rather important for achieving a good performance. The use of the algorithm described in ref. [40] or the one based on the calculation of eigenvalues of a companion matrix [39] results in a slowing-down of around 30% and 60%, respectively.

Before proceeding to the calculation of the roots of eq. (22) one has to check the value of q_4 . If $q_4 = 0$ the disk axis is parallel to the rim axis, *i.e.* cylinder 1 and 2 are parallel. In this case cylinder 1 and 2 overlap if and only

if the conditions in eq. (1) are fulfilled. If $q_4 \neq 0$, we can proceed with calculating the roots of eq. (22) through the quartic solver. If there are no real roots, disk and rim do not intersect, otherwise one needs to check whether any intersection point belongs to the cylinder 2. If this is the case, the two cylinders 1 and 2 overlap.

2.3 Some tricks of the trade

Here we will describe two tricks which significantly speed-up simulations of patchy HCs: MLLs and BBs. Before entering into the details of these two methods, it is useful to define the speed-up $S_{\mathcal{O}}$ of some optimization \mathcal{O} (where \mathcal{O} can be either *MLL* or *BB*) as the ratio of the running time τ_{\emptyset} with optimization \mathcal{O} disabled over the running time $\tau_{\mathcal{O}}$ with such optimization enabled, *i.e.*

$$S_{\mathcal{O}} \equiv \frac{\tau_{\emptyset}}{\tau_{\mathcal{O}}}. \quad (23)$$

In the following two sections we illustrate MLLs and BBs methods and we also discuss the X_0 - and ϕ -dependence of S_{MLL} and S_{BB} .

2.3.1 Multiple linked cell lists

Linked cell lists (LL) as described in textbooks such as refs. [43, 44] are used to avoid unnecessary distance calculations by partitioning the simulation box into a set of cubic cells \mathcal{C}_{HC} , where each cell encloses completely a particle. In the present case a cubic cell has to contain entirely each patchy HC. For each sticky spot of a given HC, one has to calculate the interaction potential βu_{SW} with every sticky spot belonging to all other HCs within the same cell or within the 26 adjacent cells. Since the range of attraction of interacting sites of patchy colloids (*e.g.*, see refs. [23, 30, 33]) is typically much smaller than HC dimensions, the simulation box can be partitioned into cubic cells \mathcal{C}_{HC} and \mathcal{C}_{SS} of two different sizes: one for HCs and the other one much smaller for sticky spots. Each cubic cell of \mathcal{C}_{HC} contains completely an HC, while each cubic cell of \mathcal{C}_{SS} has a size just greater than the range of attraction of attractive sites. By using these two different partitionings \mathcal{C}_{HC} and \mathcal{C}_{SS} of the simulation box, two separate linked cell lists are built for HCs and sticky spots. As in ref. [45] we will refer to this approach as the MLLs method. To understand how much MLLs can be beneficial to the simulation performance consider the model shown in fig. 1(a), where an HC is decorated with two small sticky spots on its two bases. If only one set of cubic cells is used, the minimum size L_{HC} of each cubic cell into which the simulation box is partitioned for building the linked cell lists of HCs has to be:

$$L_{HC} = \sqrt{4D^2 + (X_0D + \Delta L)^2}, \quad (24)$$

where ΔL ensures that both HCs and yellow spheres (see fig. 1(a)) can be contained entirely inside a cubic cell. The

number of attractive sites n_{SS} within a cubic cell is

$$n_{SS} \approx 2L_{HC}^3 \frac{\phi}{\pi X_0 D^3/4} \quad (25)$$

and in the limit of large or small aspect ratio, one has

$$n_{SS} \approx \begin{cases} \frac{8\phi}{\pi} X_0^2, & \text{if } X_0 \gg 1, \\ \frac{\phi}{\pi} \frac{1}{X_0}, & \text{if } X_0 \ll 1. \end{cases} \quad (26)$$

For a sticky spot belonging to a given HC the number of sticky spots for which βu_{SW} has to be calculated is $27n_{SS}$ without using MLL. On the contrary, if MLLs are employed this number of calculations reduces to $n_{SS}^0 \approx 1$, which does not depend on X_0 , *i.e.* $n_{SS}^0 = n_{SS}^0(\phi)$. Since the speed-up S_{MLL} is expected to be roughly proportional to n_{SS}/n_{SS}^0 , *i.e.*

$$S_{MLL} \propto \frac{n_{SS}(\phi, X_0)}{n_{SS}^0(\phi)} \quad (27)$$

by plugging eq. (26) into eq. (27) one obtains that

$$S_{MLL} \propto \begin{cases} \frac{\phi}{n_{SS}^0(\phi)} X_0^2, & \text{if } X_0 \gg 1, \\ \frac{\phi}{n_{SS}^0(\phi)} \frac{1}{X_0}, & \text{if } X_0 \ll 1. \end{cases} \quad (28)$$

2.3.2 Bounding boxes

The check of HCs overlap is rather time consuming, hence it would be advisable to reduce as much as possible the number of these checks in a simulation. A possible strategy is to surround each HC with a bounding neighborhood having a similar shape as the HC and to check HCs overlap only between HCs having overlapping bounding neighborhoods.

In ref. [46] it has been suggested to build an oriented BB around each HC and to test HCs overlap only if their BBs overlap as shown in fig. 4. This method is rather effective because the overlap between two parallelepipeds can be checked very efficiently by using the Separating Axis Theorem [47].

To estimate the speed-up S_{BB} consider an HC labelled with \mathcal{A} within a cubic cell $c \in \mathcal{C}_{HC}$. The number of HCs within c and within all cubic cells adjacent to c is equal to $27n_{SS}/2$. Without BBs one has to check the overlap between \mathcal{A} and all the $27n_{SS}/2$ neighbor HCs. If BBs are used, the number of overlap checks reduces to n_{BB}^0 which can be assumed to depend only on ϕ . Hence, S_{BB} can be estimated as follows:

$$S_{BB} \propto \frac{n_{SS}(\phi, X_0)}{n_{BB}^0(\phi)}. \quad (29)$$

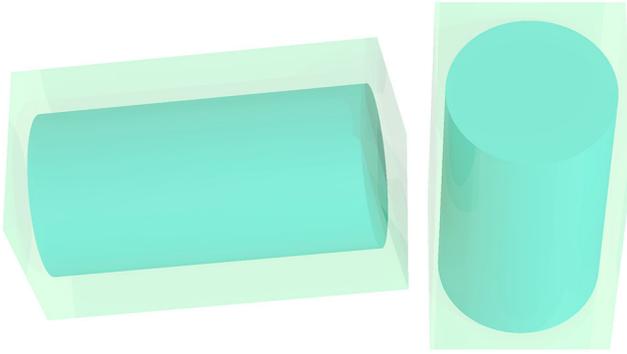


Fig. 4. Bounding boxes surrounding two hard cylinders. Since the two BBs do not overlap, the overlap check between HCs has not to be performed in this case.

By plugging eq. (26) into eq. (29), as with MLLs for large and small aspect ratio, we can expect the following limiting behavior of S_{BB} :

$$S_{BB} \propto \begin{cases} \frac{\phi}{n_{BB}^0(\phi)} X_0^2, & \text{if } X_0 \gg 1, \\ \frac{\phi}{n_{BB}^0(\phi)} \frac{1}{X_0}, & \text{if } X_0 \ll 1. \end{cases} \quad (30)$$

Boxes which are enclosed in the cylinders can be used to perform a fast overlap pre-check, where two HCs overlap if the enclosed boxes overlap. We make also use of this optimization in the simulations performed in this work.

3 Results

In this section we show the results for all the tests which we carried out to evaluate the consistency and efficiency of algorithms A1, A2 and A3 discussed above and of MLLs and BBs methods. All simulations have been carried out on a Linux server based on Intel Xeon E5-4620 v2 2.60GHz octa-core processor. The code has been compiled with gcc version 4.8.4 using -O3 and -ffast-math optimization flags.

3.1 Consistency of algorithms A1, A2 and A3

To assess the consistency of the three algorithms discussed in sect. 2.2, we performed three NTV MC simulations in the isotropic phase of $N = 980$ patchy HCs (as in fig. 1(a)) with $X_0 = 0.31$, at $\phi = 0.273$ and $T^* = 0.5$ using the three algorithms A1, A2 and A3. These simulations, which used the same pseudo-random sequence, lasted 5×10^7 MC steps starting from the same initial configuration. If algorithms A1, A2 and A3 consistently give the same result for every overlap check of HCs which occurs during the simulation, one must expect the same energy values throughout the whole simulation.

Figure 5 shows the potential energy V as a function of MC steps for algorithms A1, A2 and A3. It can be seen that the energy obtained by using the three different algorithms is the same over the whole simulation.

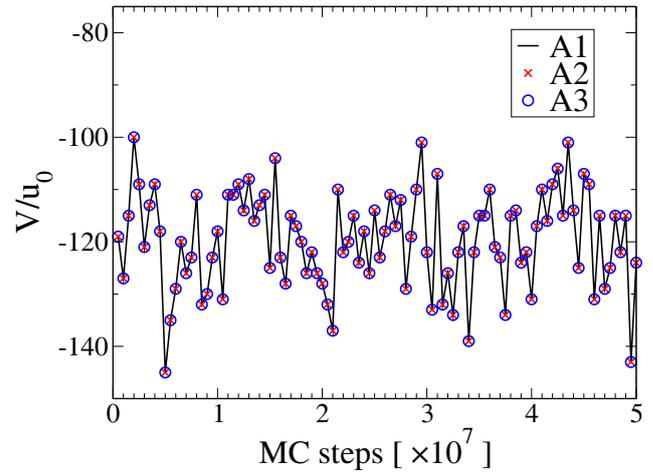


Fig. 5. Energy during three MC simulations of 5×10^7 MC steps carried out with the patchy HC model for the three algorithms A1 (black straight line), A2 (red crosses) and A3 (blue circles).

3.2 Performance of algorithms A1, A2 and A3

Having assessed the consistency of the three algorithms we proceed with evaluating their performance. We carried out NVT MC simulations with $N = 1000$ HCs without patches (as in fig. 1(b)) which lasted 2×10^5 MC steps and we measured their running time $\tau_{A\alpha}$ with $\alpha = 1, 2, 3$. The relative efficiency of algorithm A2 against A1 and of algorithm A3 against A1 will be quantified by τ_{A1}/τ_{A2} and τ_{A1}/τ_{A3} , respectively.

Figure 6(a) shows τ_{A1}/τ_{A2} while fig. 6(b) shows τ_{A1}/τ_{A3} and it can be seen that both algorithms A2 and A3 provide a significant speed-up compared to A1 at small elongations. At large elongation the most likely overlap is of rim-rim type (see fig. 2(c)) and, since the same method is used in all these algorithms to detect such overlaps, the same performance for all of them is expected. Remarkably, algorithm A3 is more than 100% faster than A1 at small elongations. In fig. 6 τ_{A2} and τ_{A3} are also shown as insets of panel (a) and (b), respectively, to provide information about real computational costs of simulations. In many past numerical studies HSCs [48–51] have been preferred to HCs because of an easier implementation of the code and of better computational performance. Here, we compare simulations of HCs (using algorithms A2 and A3) against simulations of HSCs. The overlap algorithm for HSCs is the one proposed in ref. [52] and the aspect ratio of an HSC of diameter D and length L of its cylindrical part is defined as $X_0 = (L + D)/D$. We simulate $N = 1000$ HCs and HSCs in the canonical ensemble for 2×10^5 MC steps at concentrations $\phi = 0.10, 0.20, 0.30$ and for several aspect ratios ranging from 1.4 to 5. Figure 7 shows τ_{A2}/τ_{HSC} (a) and τ_{A3}/τ_{HSC} (b), where τ_{HSC} is the running time for HSCs, as a function of X_0 for the three different volume fractions studied. We note that the efficiency of both algorithms A2 and A3 compared to HSCs (*i.e.* τ_{A2}/τ_{HSC} and τ_{A3}/τ_{HSC}) weakly depends on

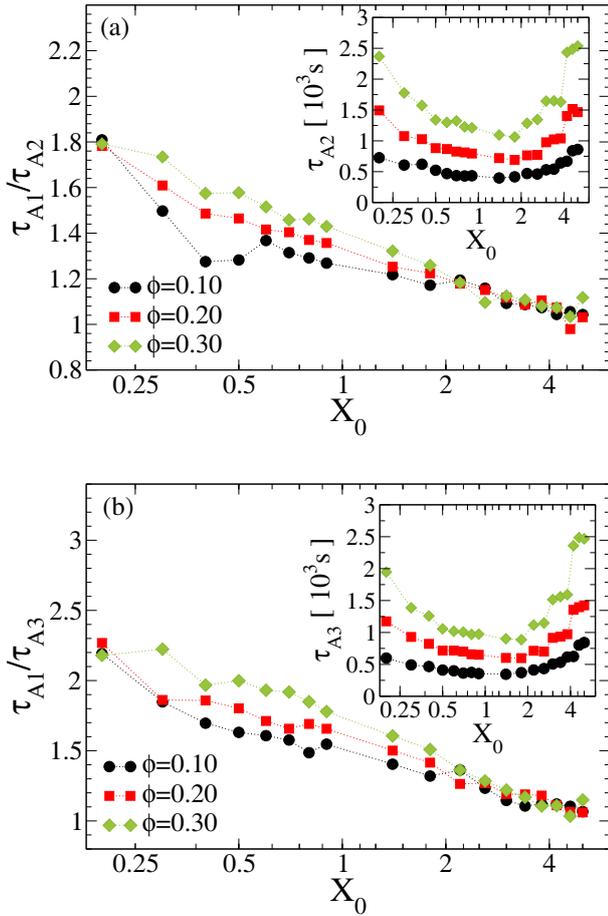


Fig. 6. Performance comparison of algorithm A2 (a) and A3 (b) with A1 for $\phi = 0.10, 0.20, 0.30$ as a function of the aspect ratio X_0 . The insets show τ_{A2} and τ_{A3} , *i.e.* the computational cost in seconds for performing the simulations.

the volume fraction. The X_0 -dependence of τ_{A2}/τ_{HSC} and τ_{A3}/τ_{HSC} is rather different instead. In fact, τ_{A2}/τ_{HSC} depends significantly on the aspect ratio X_0 increasing up to 3 on decreasing X_0 , as can be seen from fig. 7(a). We conclude that the simulation of HCs with algorithm A2 becomes much less efficient for short elongations in comparison with HSCs. On the contrary, the efficiency of algorithm A3 compared to HSCs (*i.e.*, τ_{A3}/τ_{HSC}) exhibits no significant X_0 -dependence. As shown in fig. 7(b) algorithm A3 is just 1.5–2 times slower than HSCs for all investigated aspect ratios. In this figure, as an inset, we show τ_{HSC} so that the real computational cost in seconds of the HSCs simulation can be conveniently quantified.

3.3 Performance of MLLs and BBs methods

To test the performance of BBs and MLLs methods we simulated $N = 1000$ patchy hard cylinders (as in fig. 1(a)) in a cubic box with periodic boundary conditions using a standard canonical (NVT) Metropolis MC.

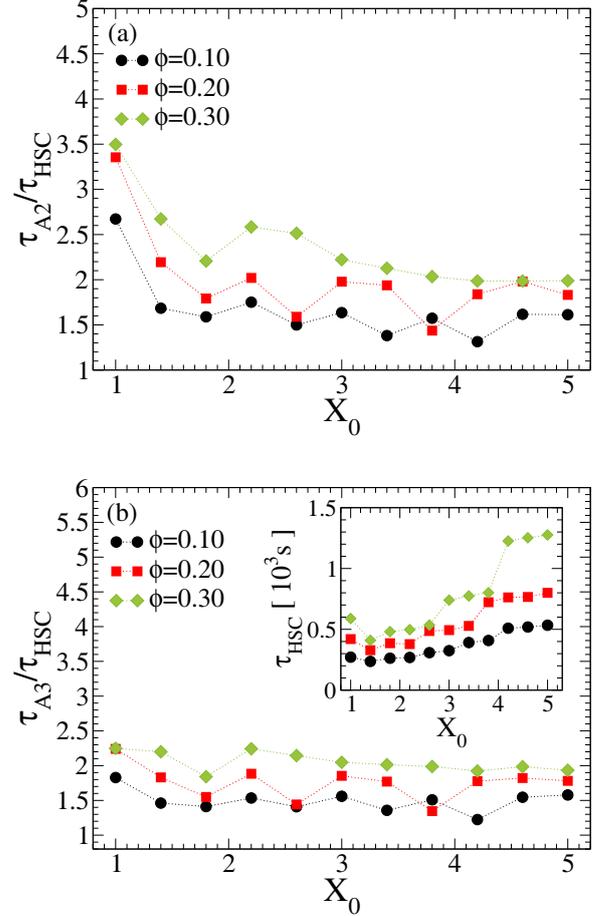


Fig. 7. Performance comparison of algorithm A2 (a) and A3 (b) with HSCs for $\phi = 0.10, 0.20, 0.30$ as a function of the aspect ratio X_0 . The inset in panel (b) shows τ_{HSC} , *i.e.* the computational cost for carrying out the simulation of HSCs in seconds.

Figure 8(a) shows the speed-up S_{MLL} which can be obtained by using MLLs in a simulation of patchy HCs for $\phi = 0.10, 0.20, 0.30$ as a function of X_0 at $T^* = 1$. Although the speed-up S_{MLL} is far from the limiting behavior in eq. (28), it can be seen that S_{MLL} increases on either increasing X_0 for $X_0 > 1$ or on decreasing X_0 for $X_0 < 1$ as expected. We note also that S_{MLL} is almost independent of ϕ . An explanation for this behavior is that by increasing the volume fraction from 0.10 to 0.30 $n_{SS}^0(\phi)$ in eq. (27) increases, thus making the speed-up almost independent of ϕ .

Figure 8(b) shows the speed-up S_{BB} provided by BBs as a function of X_0 for $\phi = 0.10, 0.20, 0.30$. As was with MLL, according to eq. (30) the speed-up S_{BB} increases on either increasing X_0 for $X_0 > 1$ or on decreasing X_0 for $X_0 < 1$. Moreover, we observe no dependence of S_{BB} on volume fraction. This behavior can be explained again by an increase of $n_{BB}^0(\phi)$ in eq. (29) on increasing ϕ , which counterbalances the speed-up increase provided by the use of BBs.

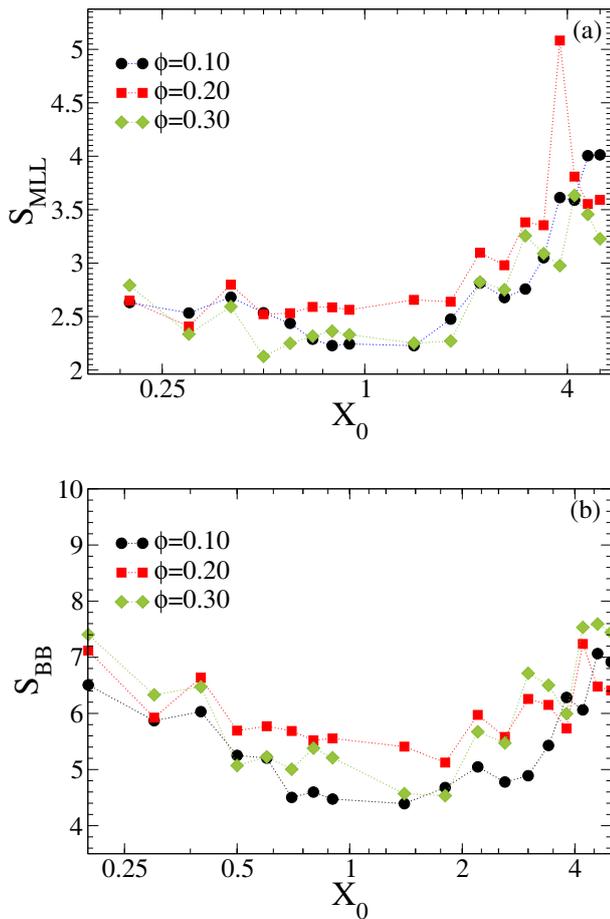


Fig. 8. Speed-up S_{MLL} obtained by using MLL (a) and speed-up S_{BB} obtained by using BB (b) for $\phi = 0.10, 0.20, 0.30$ as a function of the aspect ratio X_0 .

4 Conclusions

In this paper we propose two novel algorithms (A2 and A3) for testing the overlap of two HCs. Algorithm A2 is a more efficient variant of the algorithm proposed in ref. [35], while algorithm A3 features a completely different and faster rim-disk overlap check, which is based on the solution of a quartic equation. We assess the efficiency of these algorithms by comparing them with the algorithm A1 and with HSCs. With respect to algorithm A1, algorithm A3 turned out to be up to 2.5 faster, while algorithm A2, even if it is a bit less efficient than algorithm A3, has the advantage of being easier to implement. In comparison with HSCs algorithm A3 is less than twice as slow, thus making the simulation of HCs almost on a par with those with HSCs.

We also describe and test a couple of optimizations which can be used to speed-up the simulation of HCs, namely MLLs and BBs. Patchy HCs can take a significant advantage from MLLs where separate linked cell lists are built for the attractive sites which decorate the HC. On the contrary, BBs can be used both with and without patches and provide a great speed-up too.

Very recently HCs have been used as a very effective coarse-grained model of DNA duplexes [23, 30, 33]. Our algorithms and optimizations provide a valuable tool to ease the study of these systems via MC simulations. Another example for which our numerical techniques could be very beneficial is provided by chromonic liquid crystals. Chromonic liquid crystals, such as sunset yellow (SSY) [53] or disodium cromoglycate (DSCG) [54,55], are constituted of disk-like particles which form linear aggregates by reversible polymerization due to the attractive (stacking) interaction between their bases. In particular the aggregate structure and stacking energy of SSY have been rather well characterized [56–62]. A SSY molecule can be appropriately modeled as a thick disk of length L and diameter D decorated with two attractive sites, *i.e.* as the patchy disk shown in fig. 1(a). Experimental values of the elastic properties and phase diagram of SSY are available [53,60]. To grasp a deeper understanding of these experimental results, computer simulations could be carried out by using the patchy HC model and thus fully exploiting all the techniques which have been discussed in the present paper.

We note that algorithms A1, A2 and A3 can be straightforwardly generalized to simulate a mixture of hard cylinders of different aspect ratio and/or size, thus making their applicability rather wide in soft-matter physics. Furthermore, the use of BBs and MLLs is not restricted to HC model, since they can be also conveniently employed in simulations of patchy particles of arbitrary shape. Furthermore, in the case of more attractive or multiple bonding, which is a rather common situation in associating fluids [63,64], we expect that, even if absolute computational timing increases, the speed-up achieved by the techniques described in the present paper is not significantly affected.

Finally, it is worth observing that several methods for collision detection, which have been specifically developed for computer graphics or robotics [47, 65, 66], could be in principle adopted for testing HCs overlap. Anyway, it is not clear whether these approaches can be more efficient than the algorithms which have been discussed in this paper and it would be very interesting to carry out a careful comparison of all these methods.

Author contribution statement

All the authors were involved in the preparation of the manuscript. All the authors have read and approved the final manuscript.

References

1. P. Teixeira, J. Tavares, *Curr. Opin. Colloid Interface Sci.* **30**, 16 (2017).
2. F. Sciortino, E. Zaccarelli, *Curr. Opin. Colloid Interface Sci.* **30**, 90 (2017).
3. F. Sciortino, *Collect. Czech. Chem. Commun.* **75**, 349 (2010).

4. G.R. Yi, D.J. Pine, S. Sacanna, J. Phys: Condens. Matter **25**, 193101 (2013).
5. J.P.K. Doye, A.A. Louis, I.C. Lin, L.R. Allen, E.G. Noya, A.W. Wilber, H.C. Kok, R. Lyus, Phys. Chem. Chem. Phys. **9**, 2197 (2007).
6. B.J. Alder, T.E. Wainwright, J. Chem. Phys. **27**, 1208 (1957).
7. L. Onsager, Ann. N.Y. Acad. Sci. **51**, 627 (1949).
8. D. Frenkel, B.M. Mulder, J.P. McTague, Phys. Rev. Lett. **52**, 287 (1984).
9. A. Khan, Curr. Opin. Colloid Interface Sci. **1**, 614 (1996).
10. P. van der Schoot, M. Cates, Langmuir **10**, 670 (1994).
11. D.M. Kuntz, L.M. Walker, Soft Matter **4**, 286 (2008).
12. J.M. Jung, R. Mezzenga, Langmuir **26**, 504 (2010).
13. C.F. Lee, Phys. Rev. E **80**, 031902 (2009).
14. A. Ciferri, Liq. Cryst. **34**, 693 (2007).
15. A. Aggeli, M. Bell, L.M. Carrick, C.W.G. Fishwick, R. Harding, P.J. Mawer, S.E. Radford, A.E. Strong, N. Boden, J. Am. Chem. Soc. **125**, 9619 (2003).
16. M. Nakata, G. Zanchetta, B.D. Chapman, C.D. Jones, J.O. Cross, R. Pindak, T. Bellini, N.A. Clark, Science **318**, 1276 (2007).
17. G. Zanchetta, M. Nakata, M. Buscaglia, N.A. Clark, T. Bellini, J. Phys.: Condens. Matter **20**, 494214 (2008).
18. G. Zanchetta, F. Giavazzi, M. Nakata, M. Buscaglia, R. Cerbino, N.A. Clark, T. Bellini, Proc. Natl. Acad. Sci. U.S.A. **107**, 17497 (2010).
19. C. Robinson, Tetrahedron **13**, 219 (1961).
20. F. Livolant, A.M. Levelut, J. Doucet, J.P. Benoit, Nature **339**, 724 (1989).
21. K. Merchant, R.L. Rill, Biophys. J. **73**, 3154 (1997).
22. F. Tombolato, A. Ferrarini, J. Chem. Phys. **122**, 054908 (2005).
23. M. Salamonczyk, J. Zhang, G. Portale, C. Zhu, E. Kentzinger, J.T. Gleeson, A. Jakli, C. De Michele, J.K.G. Dhont, S. Sprunt *et al.*, Nat. Commun. **7**, 13358 EP (2016).
24. F. Tombolato, A. Ferrarini, E. Grelet, Phys. Rev. Lett. **96**, 258302 (2006).
25. E. Barry, D. Beller, Z. Dogic, Soft Matter **5**, 2563 (2009).
26. E. Grelet, S. Fraden, Phys. Rev. Lett. **90**, 198302 (2003).
27. S. Tomar, M.M. Green, L.A. Day, J. Am. Chem. Soc. **129**, 3367 (2007).
28. J. Lydon, J. Mater. Chem. **20**, 10071 (2010).
29. K. Liu, Z. Nie, N. Zhao, W. Li, M. Rubinstein, E. Kumacheva, Science **329**, 197 (2010).
30. C. De Michele, T. Bellini, F. Sciortino, Macromolecules **45**, 1090 (2012).
31. C. De Michele, L. Rovigatti, T. Bellini, F. Sciortino, Soft Matter **8**, 8388 (2012).
32. K.T. Nguyen, F. Sciortino, C. De Michele, Langmuir **30**, 4814 (2014).
33. T. Kouriabova, M. Betterton, M. Glaser, J. Mater. Chem. **20**, 10366 (2010).
34. X. Lü, J. Kindt, J. Chem. Phys. **120**, 10328 (2004).
35. N. Ibarra-Avalos, A. Gil-Villegas, A. Martinez Richa, Mol. Simul. **33**, 505 (2007).
36. R. Blaak, D. Frenkel, B.M. Mulder, J. Chem. Phys. **110**, 11652 (1999).
37. H.E. Salzer, Math. Comput. **14**, 279 (1960).
38. M. Abramowitz, I.A. Stegun, *Handbook of Mathematical Functions*, 10th edition (National Bureau of Standards, 1964).
39. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes - The Art of Scientific Computing*, 3rd edition (Cambridge University Press, 2007).
40. N. Flocke, ACM Trans. Math. Softw. **41**, 30 (2015).
41. P. Strobach, internal technical report (AST-Consulting Inc., 2015) <https://doi.org/10.13140/2.1.3955.7440>.
42. P. Strobach, J. Comput. Appl. Math. **234**, 3007 (2010).
43. M.P. Allen, D.J. Tildesley, *Computer Simulation of Liquids*, 3rd edition (Clarendon Press, Oxford, 1991).
44. D. Frenkel, B. Smit, *Understanding Molecular Simulation*, 1st edition (Academic Press, 2002).
45. C. De Michele, Comput. Phys. Commun. **182**, 1846 (2011).
46. C. De Michele, J. Comput. Phys. **229**, 3276 (2010).
47. M.G. Coutinho, *Dynamic Simulations of Multibody Systems*, 1st edition (Springer-Verlag New York, 2001).
48. S.C. McGrother, D.C. Williamson, G. Jackson, J. Chem. Phys. **104**, 6755 (1996).
49. P. Bolhuis, D. Frenkel, J. Chem. Phys. **106**, 666 (1997).
50. J.A.C. Veerman, D. Frenkel, Phys. Rev. A **43**, 4334 (1991).
51. M.P. Allen, G.T. Evans, D. Frenkel, B.M. Mulder, *Hard Convex Body Fluids* (John Wiley & Sons, Inc., 1993) pp. 1–166.
52. C. Vega, S. Lago, Comput. Chem. **18**, 55 (1994).
53. S. Zhou, Y.A. Nastishin, M.M. Omelchenko, L. Tortora, V.G. Nazarenko, O.P. Boiko, T. Ostapenko, T. Hu, C.C. Almasan, S.N. Sprunt *et al.*, Phys. Rev. Lett. **109**, 037801 (2012).
54. S. Zhou, K. Neupane, Y.A. Nastishin, A.R. Baldwin, S.V. Shiyankovskii, O.D. Lavrentovich, S. Sprunt, Soft Matter **10**, 6571 (2014).
55. S. Zhou, A.J. Cervenka, O.D. Lavrentovich, Phys. Rev. E **90**, 042505 (2014).
56. V.R. Horowitz, L.A. Janowitz, A.L. Modic, P.A. Heiney, P.J. Collings, Phys. Rev. E **72**, 041710 (2005).
57. Y.A. Nastishin, H. Liu, T. Schneider, V. Nazarenko, R. Vasyuta, S.V. Shiyankovskii, O.D. Lavrentovich, Phys. Rev. E **72**, 041711 (2005).
58. D.J. Edwards, J.W. Jones, O. Lozman, A.P. Ormerod, M. Sinyureva, G.J.T. Tiddy, J. Phys. Chem. B **112**, 14628 (2008).
59. F. Chami, M.R. Wilson, J. Am. Chem. Soc. **132**, 7794 (2010).
60. H.S. Park, S.W. Kang, L. Tortora, Y. Nastishin, D. Finotello, S. Kumar, O.D. Lavrentovich, J. Phys. Chem. B **112**, 16307 (2008).
61. M.P. Renshaw, I.J. Day, J. Phys. Chem. B **114**, 10032 (2010).
62. L. Joshi, S.W. Kang, D.M. Agra-Kooijman, S. Kumar, Phys. Rev. E **80**, 041703 (2009).
63. H. Docherty, A. Galindo, Mol. Phys. **104**, 3551 (2006).
64. G. Jiménez, S. Santillán, C. Avendaño, M. Castro, A. Gil-Villegas, Oil Gas Sci. Technol. – Rev. IFP Energ. Nouv. **63**, 329 (2008).
65. E.G. Gilbert, D.W. Johnson, S.S. Keerthi, IEEE J. Robot. Autom. **4**, 193 (1988).
66. M. Montanari, N. Petrinic, E. Barbieri, ACM Trans. Graph. **36**, 30 (2017).