Random sequential adsorption of unoriented rectangles at saturation

Wojciech Kasperek, Piotr Kubala,[*] and Michał Cieśla[†]
M. Smoluchowski Institute of Physics, Department of Statistical Physics, Jagiellonian University, Łojasiewicza 11, 30-348 Kraków, Poland

(Received 22 September 2018; published 17 December 2018)

This study presents an algorithm to generate a saturated random packing built of identical, unoriented rectangles. The algorithm is based on tracing regions that are unavailable for placing subsequent shapes. If these regions cover the whole packing the algorithm stops because no more objects can be added to the packing; thus it is saturated. The algorithm is used to study packings built of rectangles of side-to-side length ratio $\epsilon \in [1.0, 2.5]$. The densest packings are obtained for $\epsilon = 1.49 \pm 0.02$, and the packing fraction, in this case, reached 0.549641 ± 0.000017. The microstructural properties of the obtained packings are studied in terms of density autocorrelation function and propagation of orientational ordering.

DOI: 10.1103/PhysRevE.98.063310

I. INTRODUCTION

Random sequential adsorption (RSA) is one of numerous protocols used for random packing generation. A packing is built of objects that are added sequentially according to the following rules:

(1) A virtual object position and orientation is selected randomly inside a packing.

(2) If the virtual object does not intersect with any other previously placed shapes, it is added to the packing, and its position and orientation remain unchanged.

(3) Otherwise, if there is an overlap with at least one of the shapes, the virtual object is removed and abandoned.

For finite packing, after a large enough number of iterations, the packing becomes saturated, i.e., there is no possibility of adding another object that does not overlap with shapes added earlier.

Although packing problems originate from ancient times, the aforementioned protocol was first used by Flory in 1939 to study statistics of pairs of consecutive substituents of a long vinyl polymer [1]. In 1958 Renyi provided an analytical solution for the mean packing fraction of unit intervals on a one-dimensional line by solving the car parking problem [2]. The popularity of RSA owes to the fact that the random packings it generates resemble monolayers obtained in irreversible adsorption experiments [3,4]. Therefore a number of both utilitarian and fundamental studies on random packing created under the RSA protocol have appeared [5–8]. Apart from adsorption modeling, RSA packings are also useful for mesh generation, e.g., Voronoi meshes, for other stochastic methods [9]. They can serve as a starting configuration for molecular dynamics simulations of nonoverlapping objects, as well as for protocols used in dense packings generations [10–12]. In computer graphics, they help to render high-quality images [13].

Interestingly, most of the properties of RSA packings, e.g., mean packing fractions, are based on numerical simulations. However, RSA is very ineffective for almost saturated packings because the probability to find a large enough place for the particle decreases according to a power law with a number of iterations [14,15]. Therefore, a huge number of trials are required to add a subsequent object. Additionally, the RSA protocol in its above-described form cannot detect if packing is already saturated. This problem has been solved for spherically symmetric particles like disks or (hyper)spheres by tracing the space available for placing subsequent objects [16–19]. Although RSA packings of anisotropic objects are widely studied [20–24], there is no general method to generate saturated packings built of such objects. The main problem is that the shape of available space depends on virtual particle orientation. Thus the development of an efficient algorithm is much harder than for spherically symmetric particles.

The primary goal of this study is to present algorithm that effectively generates saturated RSA packings of rectangles. Recently, Zhang showed an algorithm that can be used for a wide range of polygons [25]; however, as we will show further, in the case of squares and rectangles our approach is much faster. The second goal is to find the rectangle for which the mean packing fraction reaches its maximum [23,24]. This problem has been studied already by Vigil and Ziff [20], but now, using strictly saturated packing, we can provide much more accurate results.

II. ALGORITHM

The algorithm uses the concept of exclusion zones. An exclusion zone is an area around a given shape in a packing where no other particle can be placed because it will overlap with this given shape. However, for a given anisotropic object, the shape of its exclusion zone depends on the orientation of another trial particle that can potentially be placed nearby (see Fig. 1). Thus, the subsequent particle can be added only outside exclusion zones defined by all shapes already added to the packing. The algorithm presented here is based on tracing available areas [16,18]. Because the shape of an available area

[*] pkua.log@gmail.com
[†] michal.ciesla@uj.edu.pl

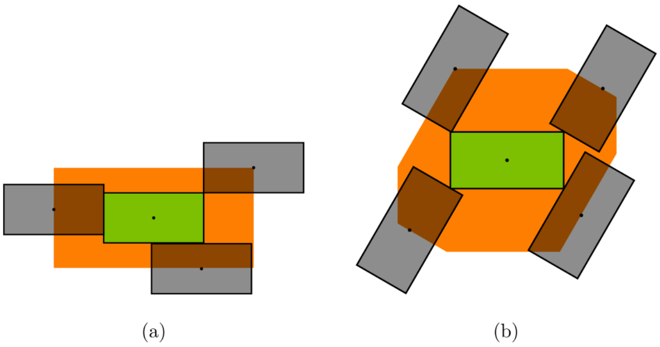

FIG. 1. Examples of exclusion zones of a green (center) rectangle. The exclusion zone is defined as the area where the center of another rectangle cannot be placed due to overlapping. The shape of the exclusion zone depends on the orientation of another rectangle, here the gray one. If the gray rectangle is parallel to the green one, the exclusion zone is rectangular; see the orange region in panel (a). If the gray rectangle is rotated by $\pi/3$, the exclusion zone will be described by the orange region in panel (b).

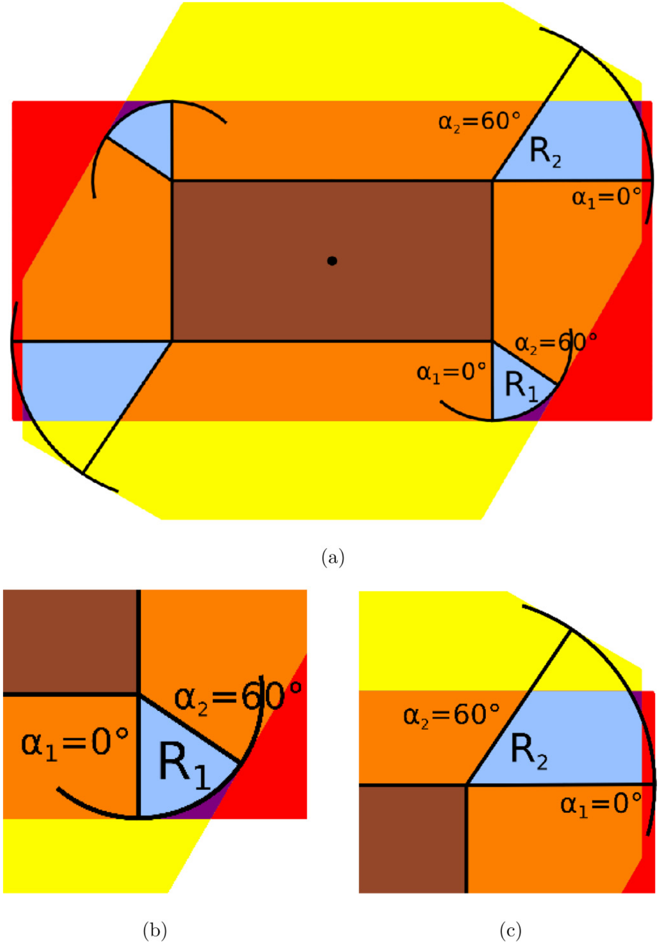

FIG. 2. (a) Example of the exclusion zone of a brown (dark gray) rectangle for a relative orientation $\alpha \in [0, \pi/3]$ interval. The red (gray) area corresponds to the exclusion zone for $\alpha = 0$, and the yellow (light gray) one is for $\alpha = \pi/3$. The resulting exclusion zone is their intersection less the violet (the darkest gray) areas shown enlarged in panels (b) and (c).

depends on the orientation of a trial particle, it is, in general, a three-dimensional object, where the third dimension is an angle. In our algorithm, available areas are approximated by three-dimensional voxels: two-dimensional squares with an additional coordinate denoting angle range. A point inside a voxel corresponds to a position and orientation of a trial particle [25].

At the beginning the packing is empty; thus it is fully covered by voxels with a full range of orientations. During packing generation, a voxel can be removed if and only if it is inside an exclusion zone of a single object added to the packing. It means that the square defined by voxel spatial coordinates has to be inside all the exclusion zones defined by its angle range. The voxel is removed only when there is no possibility of placing a trial particle in it. Thus, sampling in the RSA protocol is limited only to remaining voxels. When the sampling becomes ineffective, e.g., the number of consecutive unsuccessful tries of placing the next rectangle in a packing exceeds a given threshold value, each of the remaining voxels is split into eight identical smaller voxels, which allows us to approximate the available space more accurately. The RSA sampling and voxel splitting are iterated until all voxels are removed. Then we are sure that no shape can be placed anywhere in the packing.

The crucial point of this algorithm is to determine if a voxel is inside an exclusion zone of a single shape. Below we discuss this point in details in the case of rectangles.

A. Voxel inside the exclusion zone test

A three-dimensional interval from (x, y, α_1) to (x, y, α_2) is inside the exclusion zone of a given rectangle if and only if the point (x, y) is inside the intersection of all exclusion zones defined by virtual rectangles oriented from $[\alpha_1, \alpha_2]$. Because each exclusion zone is convex, their intersection is also convex. Thus, a voxel is inside an exclusion zone when all its spatial vertices are inside the intersection of exclusion zones for angular range determined by the angular size of the voxel. Thus, it is enough to provide the test whether a given point is inside intersection of exclusion zones. An example of such an intersection for an angular range $[0, \pi/3]$ is shown in Fig. 2. A point is inside the exclusion zone if it is inside the exclusion zones for relative orientation 0 and $\pi/3$, and it is not inside the violet areas shown in Figs. 2(b) and 2(c).

The algorithm determining whether a point is inside an exclusion zone of a given rectangle is as follows:

(1) Rotate and translate the coordination system to align it with a given rectangle. The rectangle sides should be parallel with the system's axes, and its center should be in the origin. The tested point in this coordinate system has the position (x, y), and the angle range is $[\alpha_1, \alpha_2]$.

(2) If the angle range does not fit in the interval $[\frac{k\pi}{2}, \frac{(k+1)\pi}{2}]$ for $k = 0, 1, 2$ or 3, divide it into subintervals, and each of them should be tested separately.

(3) Check whether the point is inside exclusion zones for α_1 and α_2 (whether rectangles with the center in the point and of orientations α_1 and α_2 intersect with the rectangle in the origin). If it is not inside both these zones, it is outside the exclusion zone for angles from the $[\alpha_1, \alpha_2]$ interval. Otherwise, continue.

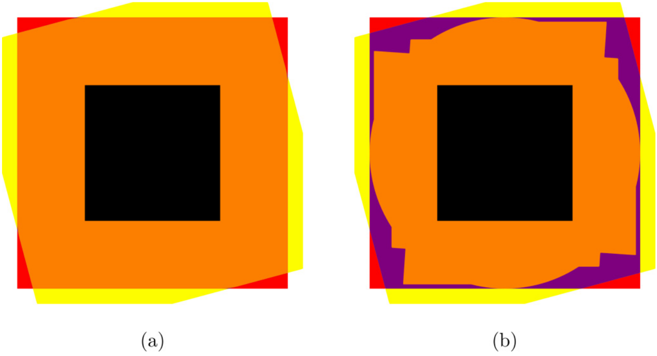

FIG. 3. Comparison of the regions built of spatially infinitely small voxels around a black square that are to be removed for our algorithm (a) and Zhang's algorithm (b). In both panels, these regions are marked orange (gray). Note that for our algorithm this region is equal to the exclusion zone, while for Zhang's algorithm it is smaller. The difference between these regions is colored purple (dark gray) in panel (b). These regions were calculated for the black square and angle range $[0, \pi/12]$.

(4) Check whether the point is inside a cone corresponding to circular sectors (see blue regions in Fig. 2). If not, the point is inside the exclusion zone. Otherwise, continue.

(5) Check whether the point is inside the circular sector (if its distance from the rectangle's corner is smaller than half the length of the rectangle side—shorter or longer depending on the corner). If not, the point is outside the exclusion zone. Otherwise, the point is inside the exclusion zone.

Recently Zhang proposed a different method to generate the saturated random packing of regular polygons [25]. In brief, to determine if a voxel can be removed, the author used a function depending on position and orientation of the new rectangle, which tells if it will intersect with any of the previously placed objects. In the case of an intersection, the function is negative, and positive otherwise. Then he estimated the maximum of this function over all possible positions and orientations of a particle placed inside a given voxel. If this estimation is negative, the voxel is removed. Note that the estimation has to be no less than the actual maximum to prevent us from removing voxels, in which the next particle may be successfully placed. Thus, for some voxels, the estimation may be positive even if the maximum is negative. Therefore, such voxels will not be removed. The method presented here is more efficient, because we do not use any estimations, but remove all the voxels that are inside the exclusion zone of any previously placed rectangle in a packing. To illustrate this difference, we draw regions built of spatially infinitely small voxels of a given orientation range, which are eliminated by both of the algorithms. The comparison is presented in Fig. 3. Although at first the difference between these regions seems not to be large, it significantly influences the speed of voxels' removal. When the voxel orientation range tends to zero, the shape of the region given by the Zhang algorithm approaches the exclusion zone. Thus, his algorithm will finally eliminate all unnecessary voxels but later than our method. It appears that the algorithm presented here generates saturated packing of squares even 100 times faster. This allows us to study bigger packings and thus improve statistics as well as decrease finite-size effects of obtained results.

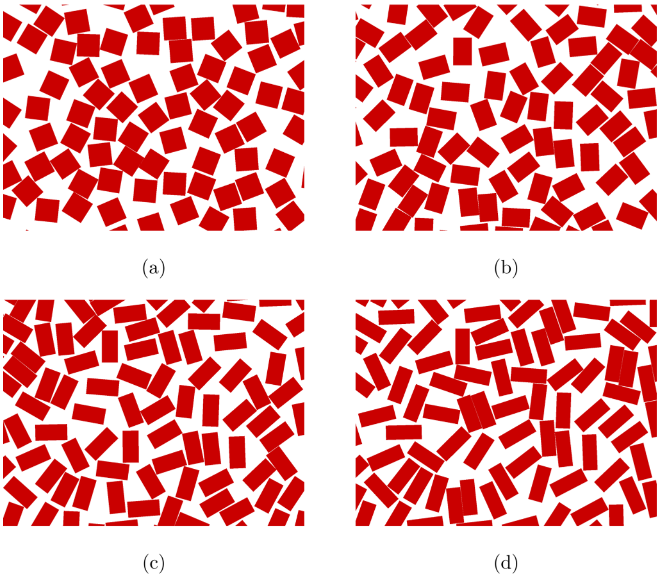

FIG. 4. Example fragments of saturated random packing built of (a) squares ($x = 1.0$) and rectangles of different width-to-height ratio: (b) $x = 1.5$; (c) $x = 2.0$; (d) $x = 2.5$.

III. SATURATED RANDOM PACKINGS OF RECTANGLES

To test the algorithm, we generate and analyze saturated random packings built of rectangles of different width-to-height ratio. Packings were square, and the surface area of each packing was 10^6. The unit of length was equal to $1/1000$ of packing side size. To lower finite-size effects periodic boundary conditions were used [26]. The lengths of the rectangle sides were $\sqrt{x}$ and $1/\sqrt{x}$. Thus, their surface was 1, and the width-to-height ratio was x. For each value of x, 100 independent saturated random packings were generated. Voxels were split when the number of consecutive failed attempts of adding a rectangle to packing exceeds 2×10^5. To prevent the overflow of a computer's memory, the total number of voxels was limited by 4×10^7. If splitting could violate this limitation, existing voxels were analyzed and removed only if they are not entirely inside an exclusion zone of particles inside a packing. Example fragments of obtained packings are shown in Fig. 4.

A. Saturated packing fraction

A packing fraction θ is the ratio of occupied surface to the whole surface of a packing. Because here a single shape has a unit surface, the packing fraction is given by

$$\theta = \frac{n}{10^6}, \tag{1}$$

where n is the number of particles in a packing of a surface equal to 10^6. The dependence of a saturated packing fraction on rectangle anisotropy x is shown in Fig. 5. For squares, the most symmetric shape, a local minimum is observed. It agrees with theoretical predictions [27]. It is interesting that the dependence of $\theta(x)$ for the relatively wide range

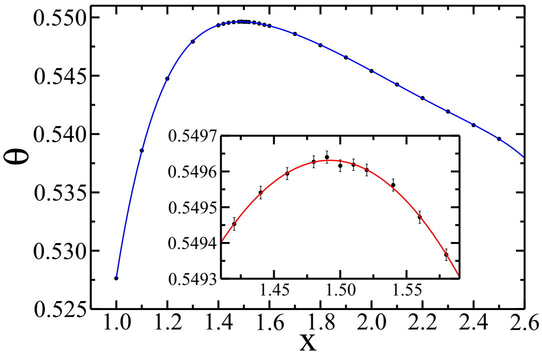

FIG. 5. Mean saturated packing fraction θ dependence on a rectangle width-to-height ratio x. The inset enlarges the region around the maximum. Dots correspond to mean values obtained from numerical simulations. Error bars correspond to the standard deviation of the mean saturated packing fraction. Solid lines are least-square fits of sixth- and second-order polynomials for the data in the main panel and the inset, respectively. These polynomials are $\theta(x) = -0.94305 + 4.5047\,x - 5.68\,x^2 + 3.837\,x^3 - 1.464\,x^4 + 0.29833\,x^5 - 0.025314\,x^6$, and $\theta(x) = 0.47377 + 0.10168\,x - 0.034071\,x^2$.

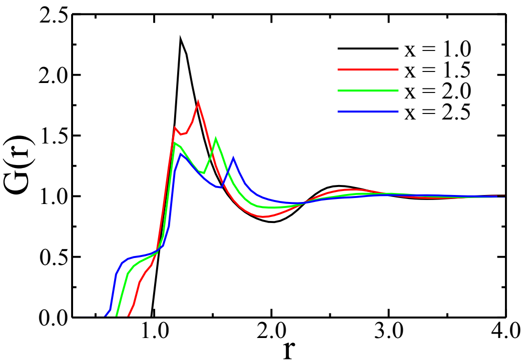

FIG. 6. Density correlation function for squares ($x = 1.0$) and rectangles ($x \in 1.5, 2.0, 2.5$) inside saturated random packings.

of anisotropies is approximated very well by a sixth-order polynomial. However, to determine the maximum of $\theta(x)$ precisely, we used quadratic fit only for the data near the maximum. The maximum of the mean saturated packing fraction is $\theta_{\max} = 0.549632 \pm 0.000017$, and it is obtained for anisotropy $x_{\max} = 1.492 \pm 0.022$. The error of $\theta_{\max}$ is the mean standard deviation of the mean saturated packing fractions obtained for x near the maximum: $\sigma(\theta) = \sqrt{\sum_{i=1}^{100}(\theta_i - \bar{\theta})^2}/100$. The error of $x_{\max}$ was estimated as a half width of the fitted parabola at the height of 0.000017 below its maximum.

Another parameter, which can be used for describing particle anisotropy instead of the width-to-height ratio, is the shape factor. It can be defined as

$$\zeta = \frac{C^2}{4\pi S}, \tag{2}$$

where C and S are the perimeter and surface area of a shape, respectively [28,29]. The shape factor is not smaller than 1, and the shape corresponding to its minimum value is a disk. Recently, it has been observed that the densest packing fraction for several different types of shapes, e.g., ellipses and two-dimensional spherocylinders, is reached for $\zeta_{\max} \approx 1.13$ [24]. In the case of rectangles, the shape factor is given by

$$\zeta_{\rm rec}(x) = \frac{\left(\sqrt{x} + \frac{1}{\sqrt{x}}\right)^2}{\pi}. \tag{3}$$

The densest packing fraction is observed for rectangles of $\zeta_{\rm rec}(1.492) = 1.325$, which is significantly larger than $\zeta_{\max}$. Therefore, the conjecture that for a given type of object the highest packing fraction is reached when shape factor is close to $\zeta_{\max}$ is false. On the other hand, the minimal possible value of $\zeta_{\rm rec}$ is 1.273. Thus, in this case, there is no possibility to approach $\zeta_{\max}$.

Two more things are worth mentioning when discussing obtained packing fractions. First, our results are approximately 100 times more precise than the previous one obtained by Vigil and Ziff [20]. Those authors estimated that the maximal packing fraction is obtained for $x \approx 2$ and its value is $\theta_{\max} = 0.545 \pm 0.002$. For $x \approx 1.5$ the obtained packing fraction was slightly smaller: $\theta = 0.543 \pm 0.003$. The saturated packing fraction obtained in this paper agrees with earlier results for $x = 2$ but is a little bit higher for $x = 1.5$. The maximum, however, according to our results is near $x = 1.5$ and not for $x = 2$. Second, the mean saturated packing fraction of squares is $\theta(x = 1) = 0.527640 \pm 0.000018$ and agrees with the value recently reported by Zhang [25], but again, due to the more efficient method of packing generation, we were able to decrease the statistical error approximately four times.

B. Microstructural properties

Microstructural properties of obtained random packings were studied in terms of density and order correlation functions. The density correlation function describes density fluctuations as a function of distance from a reference particle:

$$G(r) = \lim_{dr \to 0} \frac{N(r, r + dr)}{2\pi r\, dr\, \theta}, \tag{4}$$

where $N(r, r + dr)$ is the mean number of rectangles with centers at a distance between r and $r + dr$ from the center of a reference particle. The normalization factor in denominator causes $G(r) \to 1$ for $r \to \infty$. In addition, in general, for anisotropic objects, the autocorrelation function depends on the direction of r; here we averaged it over angular variables. To calculate $G(r)$, $dr = 0.02$ was used. The density correlation functions for some example packings are shown in Fig. 6. As we noted before, the distance unit is equal to 1/1000 of the side size of the packing, and all studied shapes have a unit surface area. It means that, for example, a square shape ($x = 1$) has sides equal to 1, and a rectangle of $x = 2$ has side sizes $\sqrt{2} \approx 1.4142$ and $1/\sqrt{2} \approx 0.7071$. The density fluctuations are damped fast, which agrees with expectations that the density autocorrelation function decays superexponentially in RSA packings [30]. For $r = 4$ there are no visible fluctuations. According to the results reported in

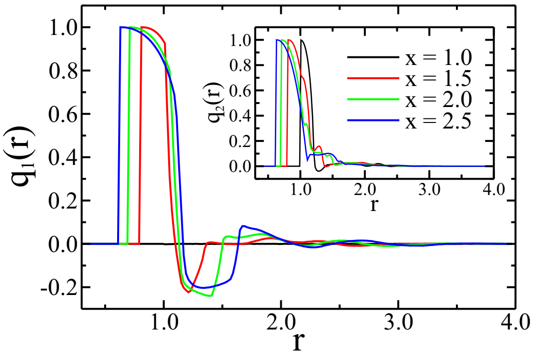

FIG. 7. Orientational order propagation for squares ($x = 1.0$) and rectangles ($x \in 1.5, 2.0, 2.5$) inside saturated random packings. The main panel shows the dependence of $q_1(r)$ and the inset shows $q_2(r)$.

Ref. [26] it means that finite-size effects are far below the statistical error. At short distances, interestingly, for rectangles, two maxima are observed. This is probably related to parallel and perpendicular orientations of neighboring particles, which suggests that the local orientational ordering should be present. To study it, two different order parameters were used:

$$q_1(r) = \langle 2\cos^2\varphi(r) - 1\rangle, \tag{5}$$

$$q_2(r) = \langle 4.0[\cos^4\varphi(r) + \sin^4\varphi(r)] - 3.0\rangle, \tag{6}$$

where $\varphi(r)$ is a relative orientation of particles at a distance r, and $\langle\cdot\rangle$ denotes averaging over all pairs of particles at this distance. The first parameter works well for anisotropic particles like rectangles and has been used previously in similar studies, e.g., Refs. [24,31]. It, however, does not work for squares because there two perpendicular axes are equivalent. Therefore, even for a fully orientationally oriented set of squares it can give a value close to zero, because statistically, for one-half of the squares we get $\cos\varphi = 1$, and for the other half $\cos\varphi = 0$. For this case, parameter q_2 was introduced because it takes these axes into account equivalently. Dependence of both order parameters on the distance is shown in Fig. 7. As expected, orientational ordering is present only at very short distances. Besides parallel alignment for small r, for slightly larger distances for rectangles q_1 indicates a dominance of perpendicular alignment. Any orientational order vanishes for $r > 2.0$. Note that, as predicted above, for squares the order parameter $q_1 \approx 0$, for all r, even where the order parameter q_2 indicates orientational ordering.

IV. SUMMARY

We have proposed what we view as the most effective method to generate saturated random sequential adsorption packings of rectangles. Then our algorithm was used to find the width-to-height ratio of the rectangle that follows to the densest RSA packing. The optimal ratio is approximately 1.49 ± 0.02, and the densest packing fraction is 0.549641 ± 0.000017. The density autocorrelation function for the obtained packings decays very fast with the distance. There is no global orientational ordering, but it can be observed locally. The presented algorithm also can be used in further studies to precisely study the packing growth kinetics for anisotropic shapes [14,15,32,33] and if it is governed by the same exponent as the dependence of iterations needed to generate saturated random packings on packing size [34].

ACKNOWLEDGMENTS

This work was supported by Grant No. 2016/23/B/ST3/01145 of the National Science Center, Poland. Numerical simulations were carried out with the support of the Interdisciplinary Center for Mathematical and Computational Modeling (ICM) at the University of Warsaw under Grant No. G-27-8.

[1] P. J. Flory, J. Am. Chem. Soc. **61**, 1518 (1939).
[2] A. Rényi, Publ. Math. Inst. Hung. Acad. Sci. **3**, 109 (1958).
[3] J. Feder, J. Theor. Biol. **87**, 237 (1980).
[4] G. Y. Onoda and E. G. Liniger, Phys. Rev. A **33**, 715 (1986).
[5] J. W. Evans, Rev. Mod. Phys. **65**, 1281 (1993).
[6] S. Torquato and F. H. Stillinger, Rev. Mod. Phys. **82**, 2633 (2010).
[7] Z. Adamczyk, Curr. Opin. Coll. Interface Sci. **17**, 173 (2012).
[8] S. Torquato, J. Chem. Phys. **149**, 020901 (2018).
[9] M. S. Ebeida, S. A. Mitchell, A. A. Davidson, A. Patney, P. M. Knupp, and J. D. Owens, Computer-Aided Design **43**, 1506 (2011).
[10] B. D. Lubachevsky and F. H. Stillinger, J. Stat. Phys. **60**, 561 (1990).
[11] S. Torquato and Y. Jiao, Phys. Rev. E **82**, 061302 (2010).
[12] S. Torquato, Phys. Rep. **745**, 1 (2018).
[13] M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory to Implementation* (Morgan Kaufmann, Cambridge, MA, 2016).
[14] Y. Pomeau, J. Phys. A: Math. Gen. **13**, L193 (1980).
[15] R. H. Swendsen, Phys. Rev. A **24**, 504 (1981).
[16] J.-S. Wang, Int. J. Mod. Phys. C **5**, 707 (1994).
[17] M. S. Ebeida, S. A. Mitchell, A. Patney, A. A. Davidson, and J. D. Owens, *Computer Graphics Forum*, Vol. 31 (Wiley Online Library, 2012), pp. 785–794.
[18] G. Zhang and S. Torquato, Phys. Rev. E **88**, 053312 (2013).
[19] E. R. Chen and M. Holmes-Cerfon, J. Nonlinear Sci. **27**, 1743 (2017).
[20] R. D. Vigil and R. M. Ziff, J. Chem. Phys. **91**, 2599 (1989).
[21] P. Viot, G. Tarjus, S. Ricci, and J. Talbot, J. Chem. Phys. **97**, 5212 (1992).
[22] J. D. Sherwood, J. Phys. A: Math. Gen. **30**, L839 (1997).
[23] M. Cieśla, G. Pająk, and R. M. Ziff, Phys. Chem. Chem. Phys. **17**, 24376 (2015).
[24] M. Cieśla, G. Pająk, and R. M. Ziff, J. Chem. Phys. **145**, 044708 (2016).
[25] G. Zhang, Phys. Rev. E **97**, 043311 (2018).

[26] M. Cieśla and R. M. Ziff, J. Stat. Mech.: Theory Exp. (2018) 043302.

[27] A. Baule, R. Mari, L. Bo, L. Portal, and H. A. Makse, Nat. Commun. **4**, 2194 (2013).

[28] P. Richard, J.-P. Troadec, L. Oger, and A. Gervois, Phys. Rev. E **63**, 062401 (2001).

[29] F. Moučka and I. Nezbeda, Phys. Rev. Lett. **94**, 040601 (2005).

[30] B. Bonnier, D. Boyer, and P. Viot, J. Phys. A: Math. Gen. **27**, 3671 (1994).

[31] M. Cieśla and J. Barbasz, Colloid. Surf. B **110**, 178 (2013).

[32] E. L. Hinrichsen, J. Feder, and T. Jøssang, J. Stat. Phys. **44**, 793 (1986).

[33] A. Baule, Phys. Rev. Lett. **119**, 028003 (2017).

[34] M. Cieśla, J. Stat. Phys. **166**, 39 (2017).