Journal of Statistical Physics, Vol. 60, Nos. 5/6, 1990

Geometric Properties of Random Disk Packings

Boris D. Lubachevsky[1] **and Frank H. Stillinger**[1]

Received January 9, 1990; final March 27, 1990

Random packings of $N \leqslant 2000$ rigid disks in the plane, subject to periodic boundary conditions on a square primitive cell, have been generated by a concurrent construction which treats all disks on an equal footing, as opposed to previously investigated sequential constructions. The particles start with random positions and velocities, and as they move about they grow uniformly in size, from points to jammed disks. The collection of packings displays several striking geometric features. These include (for large N) typically polycrystalline textures with irregular grain boundaries and linear shear fractures. The packings occasionally contain monovacancies and trapped but unjammed "rattler" disks. The latter appear to be confined to the grain boundaries. The linear shear fractures preserve bond orientational order, but disrupt translational order, within the crystalline grains. A new efficient event-driven simulation algorithm is employed to generate the histories of colliding and jamming disks. On a computer which can process one million floating-point instructions per second the algorithm processes more than one million pairwise collisions per hour.

KEY WORDS: Rigid disks; rigid spheres; random packings; rattlers; grain boundaries; vacancies; hexatic phases.

1. INTRODUCTION

Rigid disks and spheres have long enjoyed popularity among many-body physicists, materials scientists, and physical chemists, who have used them to explain structural and kinetic properties of matter.[7,11,13,16] In particular, random packings of spheres in three dimensions have been proposed as a representation for the short-range atomic order present in various amorphous solids,[12] and they are also relevant to the arrangement of colloidal particles in dense, glassy deposits.[8] Analogously, disk configurations in the plane have been advanced as a simple model for the

[1] AT&T Bell Laboratories, Murray Hill, New Jersey 07974.

0022-4715/90/0900-0561$06.00/0

arrangements of particles adsorbed on smooth surfaces,[6,26] and they may provide a helpful way to interpret experimental observations of monolayer colloidal suspensions confined between narrowly spaced glass plates.[20] Under rapid lateral compression or irreversible adsorption these two-dimensional systems can achieve states approximated by random disk packings. But in spite of the long scientific history of such connections and applications, many fundamental questions remain about the geometric nature of rigid sphere and disk packings.[25]

In the present paper, we study some novel aspects of random disk packings. This project has been facilitated by recent advances in programming techniques for event-driven simulations.[18] The approach used is readily applicable to random sphere packings in three dimensions as well, but that is reserved for later study.

The earliest examinations of random disk and sphere packings involved often clever, but imprecise and unsystematic, mechanical analog simulations.[5,19,24,27] More recently, construction algorithms for random packings utilizing digital computers have been implemented.[4,10] While the latter are precise and systematic, they use an intrinsically *sequential* model in which spheres are added one by one to an initial seed structure and the final aggregate, though possibly large, possesses a free surface. The packing model investigated herein is intrinsically *concurrent* and involves no free surface; it is consequently closer in spirit to the quenching or compressional procedures that are normally used experimentally to create amorphous solids. At least for the rigid disk case considered at length below, several unusual properties of the random packings generated have been identified, which we suspect would not readily appear with a sequential construction model. One of these properties is the presence of "rattlers," disks only loosely imprisoned by a ring of seven or more tightly jammed neighbors. This possibility had been previously postulated,[29] but, to the best of our knowledge, had not been observed directly prior to this study.

Section 2 provides some basic background information about the accessible configuration space for rigid disks and spheres. This is followed by a description in Section 3 of our dense packing model. Section 4 presents the results of experimenting with our model. Section 5 provides some discussion. Two appendices provide an outline of the algorithm[18] used to realize our model computationally.

2. SOME BASIC CONCEPTS

In its most general version, our inquiry would concern nonoverlapping arrangements of D-dimensional spheres, $D = 1, 2, 3, \ldots$, confined to a

"rectangular" region Ω_D of size $L_1 \times L_2 \times \cdots \times L_D$. We shall suppose that periodic boundary conditions apply; that is, Ω_D and its contents are periodically replicated in all directions to fill Euclidean D-space.

If its diameter is a, then the content of a D-dimensional sphere is[14]

$$s_D(a) = \pi^{D/2} a^D / 2^D \Gamma(1 + D/2)$$

Consequently, if N nonoverlapping such spheres inhabit the interior of Ω_D, the fraction ξ of that region covered by those spheres is

$$\xi = N s_D(a) / \Omega_D$$

In the limit of large Ω_D, explicit tight upper bounds are known for 1 and 2 dimensions and strongly conjectured in 3 dimensions[23,25]

$$\begin{aligned} \xi &\leqslant 1 && (D = 1) \\ &\leqslant \frac{\pi}{2\sqrt{3}} && (D = 2) \\ &\leqslant \frac{\pi}{3\sqrt{2}} && (D = 3) \end{aligned}$$

The bounds are attained, respectively, with the gapless linear array ($D = 1$), the triangular lattice ($D = 2$), and the face-centered cubic array ($D = 3$).

Let the center positions of the N spheres in Ω_D be denoted by the set of vectors $\mathbf{r}_1 \cdots \mathbf{r}_N \equiv \mathbf{R}$. The nonoverlap condition on all pairs of D-spheres and their periodic images, when the diameters are a, requires the DN-dimensional configuration vector $\mathbf{R}$ to belong to some subset $\mathfrak{S}(a)$ of the set of all possible center positions, $\Omega_D \times \Omega_D \times \cdots \times \Omega_D \equiv \mathfrak{S}(0)$. We shall refer to $\mathfrak{S}(a)$ as the "available configuration space" for the N nonoverlapping D-spheres in region Ω_D.

It is obvious that increasing a decreases the positional "freedom" of the N spheres, i.e., the content $C[\mathfrak{S}(a)]$ of the available configuration space is a continuous nonincreasing function of a. Indeed, beyond some $a_{\max}$ (generally dependent on D, N, and the shape of Ω_D) $C[\mathfrak{S}(a)]$ will vanish:

$$\begin{aligned} C[\mathfrak{S}(a)] &> 0 && (0 \leqslant a < a_{\max}) \\ &= 0 && (a_{\max} \leqslant a) \end{aligned}$$

That is, $a_{\max}$ is the largest diameter of the N spheres for which nonoverlapping positions in Ω_D still exist.

$\mathfrak{S}(a)$ undergoes a dimensional reduction as a increases to $a_{\max}$. For all $a < a_{\max}$ the dimension is DN; otherwise $C[\mathfrak{S}(a)]$ could not be positive. The dimension of $\mathfrak{S}(a_{\max})$ is necessarily less than DN; this is a mathematical consequence of close packing. The specific value of the reduced dimension depends on D, N, and the shape of Ω_D, but the presence of periodic boundary conditions and the implied free translation of any sphere packing assures that the reduced dimension must be at least D.

The connectivity of $\mathfrak{S}(a)$ and its variation with $a < a_{\max}$ is a fundamental issue for the present study. This property is simple only for $D = 1$, where for all $0 < a < a_{\max}$ the region $\mathfrak{S}(a)$ consists of those $(N-1)!$ equivalent disconnected portions corresponding to particle permutations on a line that are not interconvertible by overall translation.[28] When $D > 1$, $\mathfrak{S}(a)$ will be connected, provided a is sufficiently small, for then any nonoverlap configuration could be continuously deformed into any other one while easily avoiding overlaps. However, as a increases for $D > 1$ one expects $\mathfrak{S}(a)$ to undergo a sequence of disconnections as the nonoverlap conditions impose more and more severe constraints on available sphere rearrangements. The presumption is that $\mathfrak{S}(a)$ should shed disconnecting portions that correspond to various nearly jammed and configurationally trapped sphere packings, the majority of which for large N have irregular structures. Each such disconnected portion would undergo its own dimensional reduction as increasing a caused it to reach its jamming limit, say at a_p for the packing denoted by p. Obviously

$$a_{\max} = \max_p (a_p)$$

Any jammed packing p belongs to a set of $(N-1)!$ equivalent packings that differ only by permutation of spheres (recall that free translations permitted by periodic boundary conditions automatically permit groups of N permutations to be freely accessed from one another, even at jamming). Our primary interest, however, concerns geometrically *distinguishable* packings that are *not* permutation-equivalent. When $D > 1$, the number M of such distinguishable packings can be expected to rise exponentially with N,[29]

$$\ln M(D, N, \Omega_D) = \kappa N + o(N), \qquad \kappa > 0$$

The objective of most random-sphere-packing inquiries concerns average values of selected properties for the collection of packings, subject to some appropriate weighting. Let q be an index for the M geometrically distinguishable packings, and let $w(q)$ be a set of normalized weights:

$$\sum_q w(q) = 1$$

Then, if $F(q)$ stands for the value of some property defined for the packings, its mean value is

$$\langle F \rangle = \sum_q w(q)\, F(q)$$

In particular, mean values of powers of the jamming diameters $\langle (a_q)^n \rangle$ are defined this way, and the mean covering fraction for the random packings becomes

$$\langle \xi \rangle = [N\pi^{D/2}/2^D \Gamma(1 + D/2)\, \Omega_D] \langle (a_q)^D \rangle \tag{2.1}$$

The set of weights $w(q)$ will depend on the method of preparation of the D-sphere packings. The next section provides a specific family of such methods.

3. MODEL

Our generation procedure begins by placing the desired number N of points randomly and with a uniform distribution within the periodic unit cell Ω_D. The N points are assigned initial velocities whose components are independently distributed at random between -1 and $+1$. In the absence of subsequent collisions each of these N points would continue to move at its initial velocity along a straight line that threads through an infinite sequence of image cells.

At the outset, time $t = 0$, the points are infinitesimal. However, they begin to grow at a common rate into elastic D-spheres with diameters given for $t \geqslant 0$ by some function $a(t)$. We require that $a(0) = 0$ and that $a(t)$ be continuous nondecreasing function with $a(t) \to +\infty$ for $t \to +\infty$. As a result of the particle "growth," collisions become possible for positive times, and will increase in frequency as $a(t)$ increases.

The intention is to sample initial configurations and velocities statistically by generating many starting points and using a common $a(t)$. We permit time to progress in any realization until the system jams up, at which point the collision rate in principle must diverge. The final packing achieved obviously depends on the specific combination of initial configuration of the N points, their initial velocities, and the nature of the time-dependent collision diameter $a(t)$. After averaging over the initial conditions, the weights $w(q)$ with which the distinguishable packings q are sampled still depend on $a(t)$.

The conventional collision dynamics of elastic D-spheres with constant diameters conserves kinetic energy. However, that is not the case when diameters change with time. The collision dynamics now must be altered in

a way that is indicated for each pair encounter by Fig. 1. Suppose particles numbered 1 and 2 for convenience have respective velocities $\mathbf{v}_1$ and $\mathbf{v}_2$ just before collision. As shown in Fig. 1, these velocities can be resolved into components parallel (p) and transverse (t) to the line of centers:

$$\mathbf{v}_1 = \mathbf{v}_1^{(p)} + \mathbf{v}_1^{(t)}$$
$$\mathbf{v}_2 = \mathbf{v}_2^{(p)} + \mathbf{v}_2^{(t)}$$

where $\mathbf{v}_i^{(p)} \cdot \mathbf{v}_i^{(t)} = 0$ and $\mathbf{v}_i^{(t)} \cdot (\mathbf{r}_2 - \mathbf{r}_1) = 0$, $i = 1, 2$. The transverse velocity components are unchanged by collision, whereas the parallel components are exchanged and modified in magnitude by an additive h. If $\mathbf{v}_1^*$ and $\mathbf{v}_2^*$ are the velocities just after impact (occurring at t_c), then

$$\begin{aligned} \mathbf{v}_1^* &= [\mathbf{v}_2^{(p)} + h\mathbf{u}_{12}] + \mathbf{v}_1^{(t)} \\ \mathbf{v}_2^* &= [\mathbf{v}_1^{(p)} + h\mathbf{u}_{21}] + \mathbf{v}_2^{(t)} \end{aligned} \tag{3.1}$$

where $\mathbf{u}_{12}$ is the unit vector:

$$\mathbf{u}_{12} = (\mathbf{r}_1 - \mathbf{r}_2)/|\mathbf{r}_1 - \mathbf{r}_2| = -\mathbf{u}_{21} \tag{3.2}$$

If $2h$ exceeds the diameter growth rate $a'(t_c)$, then collisions occur at discrete isolated times. For results reported in Section 4 we take

$$h = a'(t_c)$$

The difference in kinetic energies for the pair after and before the collision is proportional to

$$\begin{aligned} &\tfrac{1}{2}(|\mathbf{v}_1^*|^2 + |\mathbf{v}_2^*|^2 - |\mathbf{v}_1|^2 - |\mathbf{v}_2|^2) \\ &\quad = h(\mathbf{v}_1^{(p)} - \mathbf{v}_2^{(p)}) \cdot \mathbf{u}_{21} + h^2 \end{aligned} \tag{3.3}$$

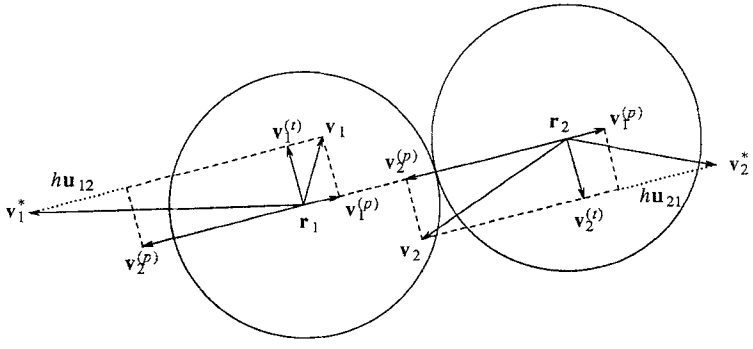

Fig. 1. Pair collision dynamics for growing D-spheres. Velocities are resolved into components parallel and transverse to the line of centers. Equations (3.1) and (3.2) specify how these components change upon impact.

The existence of the collision requires

$$(\mathbf{v}_1^{(p)} - \mathbf{v}_2^{(p)}) \cdot \mathbf{u}_{21} > 0$$

Consequently, the difference in Eq. (3.3) is strictly positive, since $h > 0$. Hence, total kinetic energy in the system increases with each collision.

The work described below is restricted to the case $D = 2$, i.e., rigid disks in the plane. Furthermore, the diameter growth rate is taken to be constant:

$$a(t) = a_0 t \qquad (a_0 > 0) \tag{3.4}$$

so that jamming always occurs at a finite time. The sampling weights $w(q)$ for the jammed disk packings depend on the a_0 choice. As a general rule (at least for large N), jamming occurs in an irregular structure if a_0 is large compared to the mean initial particle speed. For very small a_0, however, the more extended collision dynamics in principle permits the system to rearrange into a more nearly regular crystalline packing. Indeed, the limit $a_0 \to 0+$ should result in achieving the packing with $a_{\max}$ with high probability. The sampling weights $w(q)$ thus will depend on a_0 in such a

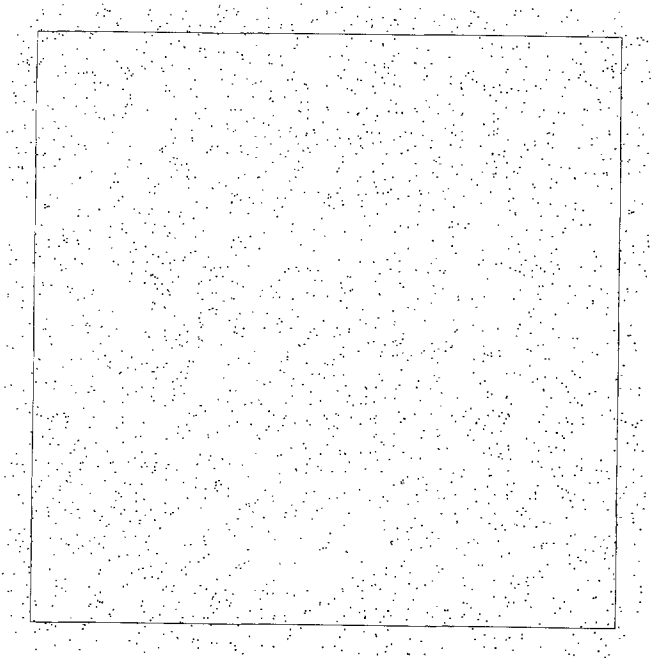

Fig. 2. An initial random configuration of 2000 points.

way that the mean covered area fraction $\langle \xi \rangle$, Eq. (2.1), will be a monotonically increasing function of a_0.

For the results reported in the next section,

$$a_0 = 3.2$$

This value of a_0 has been chosen by the following procedure. For 2000 disks, we doubled a_0 five times starting with $a_0 = 0.1$: with $a_0 = 3.2$, irregular packings resulted from any initial random configuration we tried, while with $a_0 = 0.1$, 0.2, 0.4, 0.8, and 1.6, irregular packings were not generated or were generated seldom.

The series of Figs. 2–5 represents four snapshots of 2000 disks expanding with speed $a_0 = 3.2$; these are qualitatively typical of the system evolution under these growth conditions. In all of our rigid-disk packing calculations we have taken the primitive cell Ω_2 to be a square ($L_x = L_y$). This choice excludes the occurrence of a perfect triangular lattice, the maximum density packing arrangement for disks. However for some choices of the integer N, specifically those of the form

$$N = n_1 n_2$$

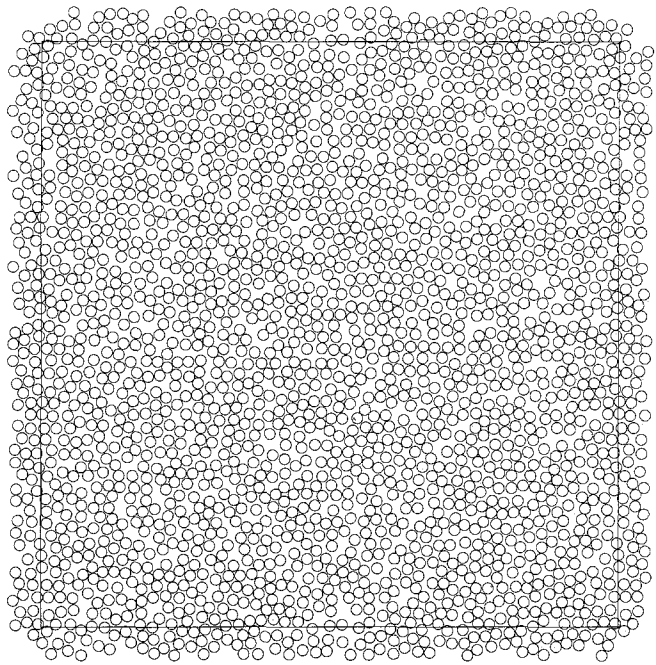

Fig. 3. Depiction of 2000 disks at $t = 2.9576$ after 2×10^4 pairwise collisions (20 impacts per disk). The covering fraction is $\xi = 0.5628$. The starting configuration is shown in Figure 2.

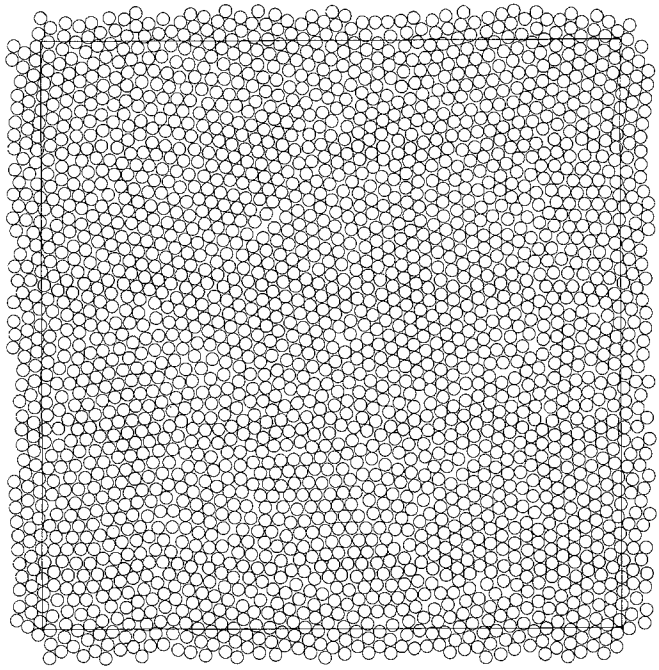

Fig. 4. Depiction of 2000 disks at $t = 3.3901$ after 2×10^5 pairwise collisions (200 impacts per disk). The covering fraction is $\xi = 0.7394$. This shows a continuation initiated in Figs. 2 and 3.

where the ratio of integer factors n_1 and n_2 closely approximates the irrational value

$$n_1/n_2 \simeq 3^{1/2}/2$$

a slightly strained version of the triangular lattice can be fitted into Ω_2 with its principal directions aligned with the sides of Ω_2. Since our primary interest concerns irregular disk packings and our choice of a_0 would discriminate against the regular lattice anyway, the square shape chosen for Ω_2 is reasonable.

Collision rates increase without bound as the jammed packing limit is approached. This results, first, from the diminishing mean distance that particles must travel between successive collisions. Second, it has been pointed out that kinetic energy (hence the particle mean speed) increases for elastic collisions with growing disks. For practical reasons we set all particle velocities to zero repeatedly during late stages of the dynamics to alleviate (but not eliminate) the collision rate divergence problem. This

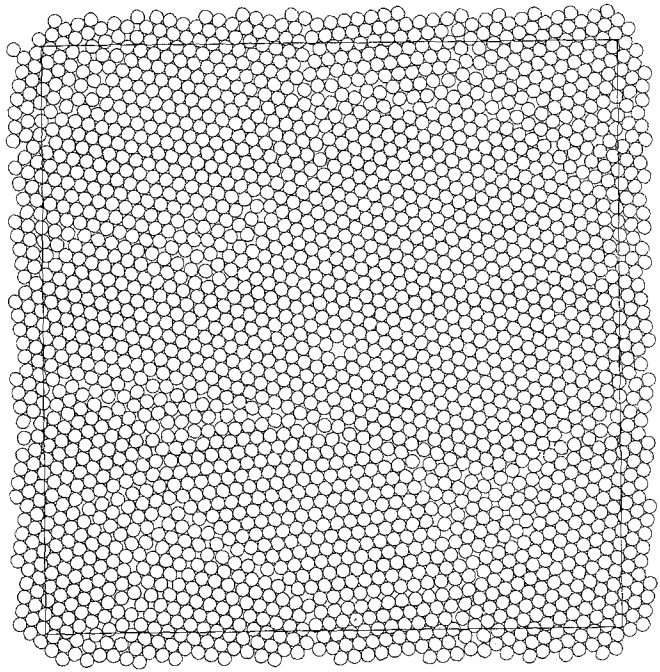

Fig. 5. Depiction of 2000 disks at $t = 3.6733$ after 2×10^6 pairwise collisions (2000 *impacts* per disk). The covering fraction is $\xi = 0.8681$. This shows a continuation initiated in Figs. 2–4.

action is utilized only after the disks have become so nearly jammed that virtually no influence on the final packing distribution is expected to occur.

Numerically, we identify the jammed state as having occurred when the seven significant digits of the disk diameter stabilize despite continuing collisions. Note that the computations are carried out with double precision, i.e., with precision at least 10^{-14}. While further lengthy computation with even higher precision could reduce this remaining looseness somewhat, we believe that nothing new would be learned thereby.

As an alternative to our method of producing jammed packings, one might have utilized particles of fixed size and reduced the system area. Two variants would be possible: (a) impenetrable boundaries, and (b) periodic boundary conditions. The former has the undesirable property of producing an anomalous boundary region with properties that presumably differ from those of bulk packings. The latter requires decision about momentum discontinuity to be required when a periodic cell boundary is crossed. Both variants might be implemented with particle accelerations between collisions in the spirit of "constant pressure molecular dynamics" as introduced by Andersen.[2] Our simple particle growth method avoids these issues.

4. DISK PACKING RESULTS

As a simple introduction to the complex behavior with larger numbers of disks, we first consider the specific case $N = 27$. With a square primitive cell and periodic boundary conditions, only two fundamentally distinct families of packings occur. These are illustrated in Figs. 6 and 7. Aside from the possibility of overall free translation and rotations by integer multiples of 90°, these types of patterns have been repeatedly generated from the random initial conditions.

The configuration presented in Fig. 6 is completely jammed. Every disk is tightly constrained by at least three neighbors with which it is in contact. No continuous rearrangement of the pattern is possible without violating the disk nonoverlap condition. Consequently, the limiting reduced dimension of the disconnected manifold for this completely jammed packing is 2.

The disk packing exhibited in Fig. 7 possesses a distinctive characteristic, namely the presence of a "rattler" disk. Twenty-six of the disks are tightly jammed against one another, and can only translate as a rigid whole, thanks to the periodic boundary conditions. The 27th disk, as the figure clearly shows, is free to move within a rigid cage of eight jammed neighbors. As a result, the limiting reduced dimension of this packing's disconnected manifold is 4.

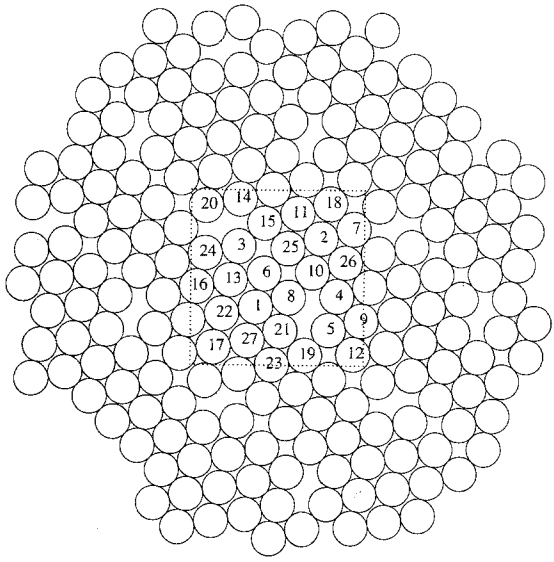

Fig. 6. Completely jammed random packing formed from 27 disks in a square domain, with periodic boundary conditions. The covering fraction is $\xi = 0.83007$.

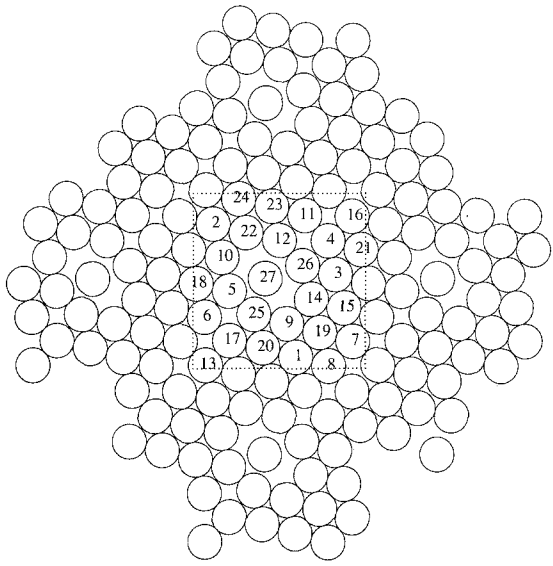

Fig. 7. Random packing for 27 disks showing the occurrence of an unjammed rattler. Periodic boundary conditions apply to the square domain used. The covering fraction is $\xi = 0.82974$.

Notice that the jammed subset of 26 disks in Fig. 7 appears to possess a reflection symmetry parallel to the diagonals of the square primitive cell Ω_2. No such symmetry applies to Fig. 6.

While numerical experiments with relatively small systems suffice to show the existence of rattlers, other attributes of random disk packings become clear only for large N. Some of these attributes are illustrated in Fig. 8, the final configuration of the 2000-disk experiment whose early stages were shown in Figs. 2–5. To aid visualization, all disks have been shaded except for four rattlers.

The predominant texture of the configuration shown in Fig. 8 can be described as polycrystalline. Individual crystalline domains are separated by grain boundaries across which crystal orientation changes. These grain boundaries exhibit a high incidence of pentagonal holes, i.e., gaps surrounded by five disks. Note that the four rattlers all reside within grain boundaries.

Figure 8 also reveals the presence of linear shear fractures that run across crystal grains. These run parallel to the primary crystallographic directions of the affected grains. They seem to appear only in the final stages of the numerical experiment, producing sequences of virtually identical quadrilateral holes. It is important to realize that intragrain transla-

tional long-range order is disrupted by these shear fracture lines, whereas orientational long-range order (directions of nearest-neighbor pairs) is preserved. This observation relates to the existence of "hexatic" phases of two-dimensional systems with just this property: long-range order in the orientational, but not translational, degrees of freedom.[22] However, the accepted description of hexatic phases involves unbonded dislocations, not linear shear fractures.[21] Whether or not linear shear fractures could be a significant structural feature in real hexatic systems must remain an open question for now.

A monovacancy is obvious near the center of Fig. 8. That only one arises in a 2000-disk packing suggests that they have low occurrence probability under the packing formation conditions used.

In order to analyze more deeply the local geometric properties of the random packings, we have found it instructive to classify disks according to their values of a dimensionless "jamming" parameter γ:

$$\gamma = \Sigma/3a \tag{4.1}$$

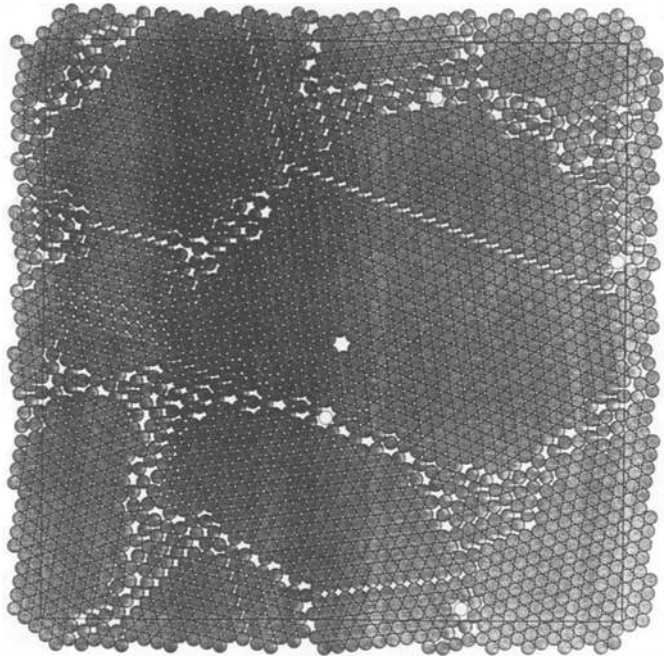

Fig. 8. Depiction of 2000 disks at $t = 3.6840$ after 21×10^6 pairwise collisions (21,000 impacts per disk). The covering fraction is $\xi = 0.8732$. Significant rattlers are not shaded. This is the final configuration in the series whose beginning is in Fig. 2. Notice the monovacancy near the center of the pattern, whose precursor was clear in Fig. 5.

Here Σ is the sum of the distances to the three closest neighbors of the disk under consideration, and a is the final common value of the disks' diameters. Figure 9 reproduces the configuration shown in Figure 8, but with four γ ranges visually distinguished. The four rattlers continue to stand out with large values of $\gamma \geqslant 10^{-1}$. The surprising result is that a substantial fraction of the disks seem to enjoy a small, but distinguishably positive, motional freedom, with jamming parameter values in the range $10^{-7} \leqslant \gamma \leqslant 10^{-1}$. These slightly loose particles are not randomly arranged over the pattern, but, like the obvious rattlers, tend to accumulate at grain boundaries. This observation suggests that the limiting dimension of $\mathfrak{S}(a)$ for most random packings is of order N.

Figure 10 presents a magnified view of the rattler and its seven neighbors that has been highlighted with a small square in Fig. 9.

Construction and comparison of many jammed structures, sampling initial conditions uniformly, is necessary to assign significance to the properties discussed thus far. With this in mind, we have generated several other random packings of 2000 disks. Figure 11 shows the final configura-

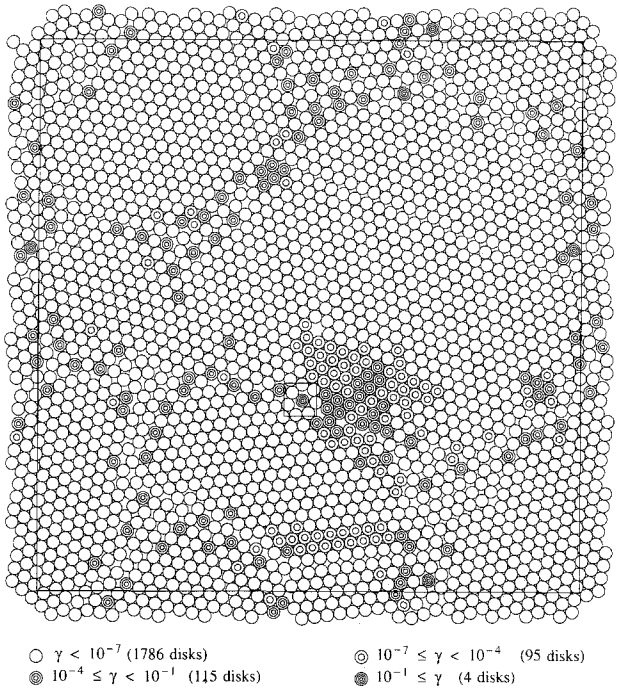

Fig. 9. The configuration shown in Fig. 8 with disks classified according to the "jamming" γ defined in Eq. (4.1).

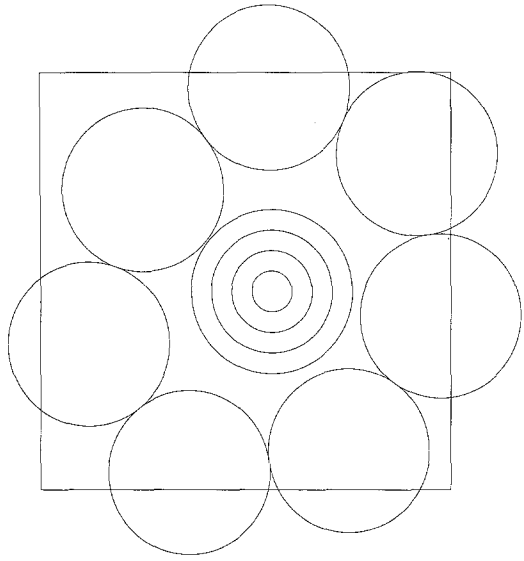

Fig. 10. A magnified fragment near the center of Fig. 9 (it is marked with a square in Fig. 9). The magnification clearly reveals a rattler.

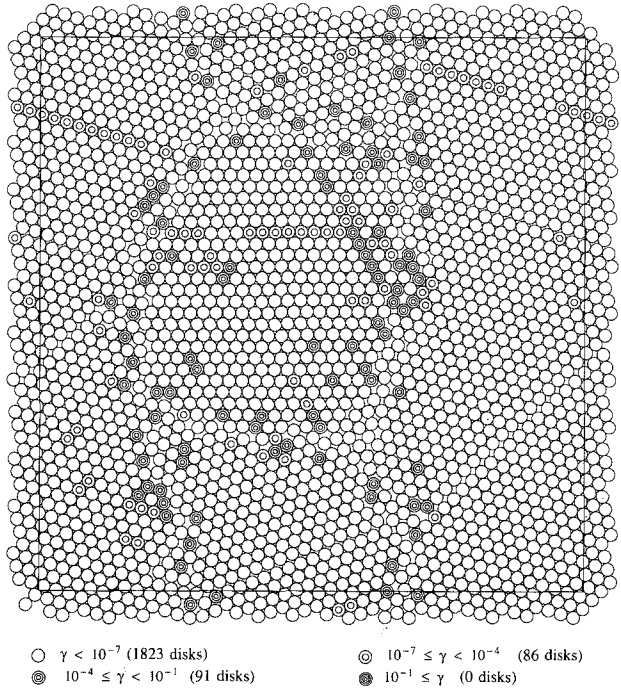

Fig. 11. Another jammed configuration of 2000 disks started with a different random configuration. It jammed at $t = 3.7013$ after 23×10^6 pairwise collisions (23,000 impacts per disk). The covering fraction is $\xi = 0.8814$.

tion of one of these other numerical experiments, with γ classification of disks. Our results have consistently shown polycrystalline textures, with frequent linear shear fractures. The relatively scarce monovacancies and loose ($\gamma \geqslant 10^{-1}$) rattlers may or may not occur in any particular instance, as would be expected statistically, but in any case their concentrations remain quite low.

5. DISCUSSION

The principal conclusion to be drawn from the present work is that our concurrent construction procedure produces random disk packings with several surprising characteristics that are unlikely to emerge from standard sequential construction procedures. These characteristics include the presence of rattlers, vacancies, and linear shear fractures. In particular, we have found that obvious rattler disks tend to concentrate at grain boundaries. Although it is possible for a sequential method to produce packings with grain boundaries (specifically by starting with an irregular seed cluster of a few disks), it seems unlikely that an isotropic distribution of such boundaries would arise in an extended packing created this way.

Several nontrivial extensions of our study deserve future consideration. One would involve mistures of two or more species of disks with an arbitrary symmetric matrix of collision distances $a_{ij}(t)$ for species i and j. This would be particularly fascinating in light of recent advocacy of two-disk-size models for quasicrystalline order in two dimensions.[17,31] Another direction for generalization would be to consider more elaborate particle shapes, such as hard ellipses,[30] for which random packings will doubtless display interesting patterns of local particle orientational order.

We have carried out a few exploratory calculations designed to observe the effect of changing a_0 in Eq. (3.4) on packing properties. As mentioned earlier, we expect that the slower the growth rate, the less defective should the packings tend to be. Indeed, this is the trend observed. The most irregular packings were produced with the largest growth rate, $a_0 = 1000$, and have covering fraction ξ approximately 0.85; the most regular packings were achieved with the slowest growth rate, $a_0 = 0.001$, and have covering ξ just over 0.90 (maximum = 0.9069). Further computation would be required to yield precise results for the way that a_0 affects the mean and variance of ξ.

The most obvious direction for extension of the present study is toward packing of rigid spheres in three dimensions. A natural question is whether "rattlers" would continue to be found inhabiting cages of neighbors comprising no fewer than 13 spheres. Some preliminary calculations have been performed, which seem to answer that question in the

affirmative: not only do the rattlers appear in random sphere packings, but they occur apparently with higher probability than in the corresponding disk case.

Finally, we express the hope that the results of our numerical experiments will stimulate rigorous mathematical analysis of the properties of random disk and sphere packings.

APPENDIX A. ALGORITHM

Let state of a sphere i at time t be the vector $\mathbf{s}_i(t) = (\mathbf{r}_i(t), \mathbf{v}_i(t))$, where $\mathbf{r}_i(t)$ is the center position vector of sphere i, and $\mathbf{v}_i(t)$ is the velocity vector. The global state $\mathbf{S}(t)$ of the system at time t is the set $\mathbf{s}_1(t)$, $\mathbf{s}_2(t)$,..., $\mathbf{s}_N(t)$.

A "naive" serial algorithm advances the global state $\mathbf{S}(t)$ from event to event, where an event is either a collision of two spheres or a boundary crossing by a center of a sphere. All N states $\mathbf{s}_1(t)$, $\mathbf{s}_2(t)$,..., $\mathbf{s}_N(t)$ are examined and updated at times $t_0 \leqslant t_1 \leqslant t_2 \leqslant \ldots$, where t_0 is the initialization time and t_{i+1} is the nearest next event time seen at time t_i. The naive scheme is inefficient for large N because (a) the same event is repeatedly scheduled an order of N times until it occurs, and (b) at a typical cycle, most spheres are not participating in events; still, they are examined by the algorithm as potential participants.

Aside from problems (a) and (b), there exists the problem (c) of finding an inexpensive method of determining the nearest collision for a chosen sphere. A straightforward method is to compare the chosen sphere with all $N-1$ others. The standard improvement in this method is the division of the region Ω_D into an order of N rectangular sectors $\omega^{(k)}$,

$$\Omega_D = \bigcup_k \omega^{(k)}, \qquad \omega^{(k)} \cap \omega^{(l)} = \varnothing, \qquad \text{if} \quad k \neq l$$

Only spheres in the neighboring sectors have to be checked to determine the immediate next collision. For each sector $\omega^{(k)}$, a membership list of spheres whose centers belong to $\omega^{(k)}$ is maintained. The overhead of the method results from examining additional boundary crossing events when spheres change sector membership. Despite of the overhead, the method reduces the work from $O(N)$ to $O(1)$ per one collision scheduled.

A natural idea for improvement in (a) and (b) is to postpone examining and updating a sphere state until the event actually occurs. Implementing this idea does not appear as easy as it might seem. As the simulation progresses, a scheduled collision of a given sphere may require rescheduling. The need for such rescheduling and the desire not to lose information about already planned collisions led in ref. 1 to a complicated data structure and update scheme called "time-table" in ref. 9. Observe

that, with all its inefficiency, the naive scheme has a very attractive double-buffering data structure. The structure consists of only two copies of the global state vector S, the old and the new, so that the new vector is computed on the basis of the old one and, in turn, becomes the old one during the next cycle.

We use a different algorithm proposed in ref. 18. The attraction of this algorithm is that it utilizes a simple and easy to handle double-buffering data structure, while avoiding problems (a) and (b). Problem (c) is handled using the standard technique of sectoring. In most cases the algorithm examines and processes only the events whose processing is unavoidable, e.g., sphere collisions and boundary crossings. Sometimes, like the naive algorithm, it also processes events whose examining is not necessary. However, the fraction of such overhead events is small and does not grow with N, while the speedup due to simplicity of data handling is substantial.

The following is an outline of the algorithm. The basic data unit in the algorithm is called *event* and has the following format

$$event = (time,\ state,\ partner)$$

where *time* is the time to which *state* of a sphere corresponds. Note that *state* is the new state of the sphere *immediately after* the event, e.g., if the sphere has experienced a collision at *time*, the velocity-coordinate of the *state* is the new velocity vector after the collision; *partner* identifies the other sphere, if any, involved in the event. If there is no partner in the event, the program assigns a special "no-value" symbol Λ to the *partner* coordinate. If $time = +\infty$, then the other three coordinates in the *event* have no value, i.e., $state = type = partner = \Lambda$.

At any stage of simulation, the algorithm maintains two events for each sphere: an old, already processed in the past event and a new, next scheduled event. This information is stored in array $event[1{:}N, 1{:}2]$, where, as before, N is the number of spheres of the simulated system. Let us agree to understand a reference like $time[3, 1]$ as the *time* coordinate of element $event[3, 1]$ of this array.

Two arrays, $new[1{:}N]$ and $old[1{:}N]$, with elements equal to 1 or 2, are maintained. For sphere i, $new[i]$ is the pointer to the new event and $old[i]$ is the pointer to the old event. Thus, the new event for sphere i is stored at $event[i, new[i]]$ and the old event is stored at $event[i, old[i]]$. When $new[i]$ is updated, $old[i]$ is updated immediately afterward, so that the relation $new[i] + old[i] = 3$ remains invariant.

Figure 12 represents the algorithm pseudocode given in ref. 18. The algorithm is formulated in terms of basic functions *interaction_time*, *jump*, and *advance* (here "basic" means that the actual computations of these functions are not represented). The functions have the following semantics.

Given *state*1 of sphere 1 at *time*1 and *state*2 of sphere 2 at *time*2, function *interaction_time* computes the *time* of the next potential interaction of sphere 1 with sphere 2 while ignoring the presence of other sphere and boundaries:

$$time \leftarrow interaction\_time\ (state1,\ time1,\ state2,\ time2) \qquad (A.1)$$

where $time \geqslant \max(time1,\ time2)$. If *interaction_time* cannot find such finite *time*, e.g., when the spheres are moving away from each other, we assume that $+\infty$ is returned. In the actual program, *interaction_time* is represented as a subroutine; the computations of this subroutine are specified in Appendix B.

initially *current_time* ← 0 and for $i = 1,2,...N$:
new[*i*] ← 1, *old*[*i*] ← 2, *time*[*i*,1] ← 0, *partner*[*i*,1] ← Λ,
state[*i*,1] ← initial state of sphere *i*, *event*[*i*,2] ← *event*[*i*,1]

1. while *current_time* < *end_time* do {
2. *current_time* ← $\min_{1 \le i \le N}$ *time* [*i*, *new*[*i*]] ;
   i_* ← an index which supplies this minimum (i.e., *current_time*) ;
3. *new*[i_*] ← *old*[i_*] ; *old*[i_*] ← 3−*new*[i_*] ;
4. **P** ← $\min_{j \in A(i_*)} P_{i_*j}$, where $A(i_*) = \{j \mid 1 \le j \le N,\ j \ne i_*,\ time[j, new[j]] \ge P_{i_*j}\}$;
   if **P** < $+\infty$ then j_* ← an index which supplies this minimum (i.e., **P**) ;
5. **Q** ← $\min_{1 \le k \le K} Q_{i_*k}$;
   if **Q** < $+\infty$ then k_* ← an index which supplies this minimum (i.e., **Q**) ;
6. **R** ← min{**P**, **Q**} ; *time*[i_*, *new*[i_*]] ← **R** ;
7. if **R** < $+\infty$ then {
8. *state*1 ← *advance* (*state*[i_*, *old*[i_*]], *time*[i_*, *old*[i_*]], **R**) ;
9. if **Q** < **P** then {
10. *state* [i_*, *new*[i_*]] ← *jump* (*state*1, k_*) ;
11. *partner* [i_*, *new*[i_*]] ← Λ ;
    } /* end **Q** < **P** close */
12. else { /* case **Q** ≥ **P** */
13. *time*[j_*, *new*[j_*]] ← **R** ;
14. *state*2 ← *advance* (*state*[j_*, *old*[j_*]], *time*[j_*, *old*[j_*]], **R**) ;
15. (*state* [i_*, *new*[i_*]], *state* [j_*, *new*[j_*]]) ← *jump* (*state*1, *state*2) ;
16. m_* ← *partner* [j_*, *new*[j_*]] ;
17. *partner* [i_*, *new*[i_*]] ← j_* ; *partner* [j_*, *new*[j_*]] ← i_* ;
18. if $m_* \ne \Lambda$ and $m_* \ne i_*$ then { /* update third party m_* */
19. *state* [m_*, *new*[m_*]] ←
        advance (*state*[m_*, *old*[m_*]], *time*[m_*, *old*[m_*]], *time*[m_*, *new*[m_*]]) ;
20. *partner* [m_*, *new*[m_*]] ← Λ ;
    } /* end update third party */
  } /* end **Q** ≥ **P** close */
 } /* end **R** < $+\infty$ close */
} /* end while loop */

Fig. 12. The simulation algorithm.

The change of the sphere velocities at the moment of collision, discussed in Section 3, is represented in the algorithm by function *jump*, also implemented as a subroutine. Given *state*1 and *state*2 of colliding spheres 1 and 2, subroutine *jump* returns *new_state*1 and *new_state*2 of these spheres immediately after the interaction:

$$(new\_state1,\ new\_state2) \leftarrow jump\ (state1,\ state2)$$

Functions *interaction_time* and *jump*, as described, depend on two sphere arguments. The algorithm pseudocode in Fig. 12 also employs one-argument variants of *interaction_time* and *jump* to express boundary crossing. Thus, if k is an index for the set of K boundaries (which includes the exterior boundaries of Ω_D as well as interior boundaries which define the sectors), then sequences

$$time \leftarrow interaction\_time\ (state1,\ time1,\ k) \tag{A.2}$$

and

$$new\_state \leftarrow jump\ (state,\ k) \tag{A.3}$$

show the invocation formats of these one-sphere functions. In (A.2), *time* is the nearest time of crossing boundary k. The *jump* in (A.3) is the identity map when k is an internal boundary, because neither position nor velocity of a sphere experiences a jump at the instance of crossing internal boundary. When k is an external boundary, one of the coordinates of the *position* vector experiences a jump: the sphere disappears at a face of Ω_D and immediately reappears at the opposite face.

Given *state*0 of a sphere at *time*0 and a value *time*1 $\geqslant$ *time*0, function *advance* computes *state*1 this sphere would have at *time*1 ignoring possible collisions with the other spheres or boundary crossings on the interval [*time*0, *time*1]:

$$state1 \leftarrow advance\ (state0,\ time0,\ time1)$$

In our problem, *advance* has the obvious form of advancement along a straight line parallel to *velocity*0 starting from *position*0 at *time*0.

In Fig. 12, /" and "/ mark the beginning and the end of a comment, and the minimum over an empty set of values is assumed to be $+\infty$. The following short-hand notations are used:

$$P_{ij} \equiv interaction\_time\ (state[i,\ old[i]],\ time[i,\ old[i]],$$
$$state[j,\ old[j]],\ time[j,\ old[j]])$$

where $1 \leqslant i, j \leqslant N$ and

$$Q_{ik} \equiv interaction\_time(state[i, old[i]], time[i, old[i]], k)$$

where $1 \leqslant i \leqslant N$ and $1 \leqslant k \leqslant K$.

The main cycle in Fig. 12 consists essentially of two steps: (1) selecting the next sphere i_* to process its event (line 2), (2) processing the event (the rest of the cycle).

Processing the event means scheduling next events for the chosen sphere and the other involved spheres, if any. **P** and **Q** are the nearest next interaction times. There are two main cases in such scheduling, depending on the type of the future event:

(a) $\mathbf{Q} < \mathbf{P}$, when the scheduled interaction involves only one sphere i_* (lines 8, 10, and 11).

(b) $\mathbf{Q} \geqslant \mathbf{P}$, when the scheduled interaction involves i_*, a second party j_* (lines 8 and 13–17), and may involve a third party m_*, the previous partner, if any, of j_* (lines 19 and 20).

The pseudocode in Fig. 12 does not specify the way minimizations in lines 2, 4, and 5 are implemented. A straightforward method to find the minimum of $time[i, new[i]]$ for i ranging from 1 to N in line 2 requires $O(N)$ operations per event. To reduce the cost, the algorithm instead organizes values $time[i, new[i]]$ into an implicit *heap* structure.[15] Two pointer arrays $pht[1:N]$ and $pth[1:N]$ are maintained, so that $time[pht[m], new[pht[m]]]$ is the value which is implicitly located at the mth position of the imaginary heap array and *pth* is the inverse map for *pht*, i.e., $pth[pht[m]] = m$ for all m. In particular, $time[pht[1], new[pht[1]]]$ corresponds to the heap tree root, i.e., the minimum value, so that line 2 can be simply rewritten as

$$i_* \leftarrow pht[1], \qquad current\_time \leftarrow time[i_*, new[i_*]]$$

This method requires updating the heap structure (arrays *pht* and *pth*) each time a value of $time[i, new[i]]$ is changed in other sections of the algorithm. Including this updating, the total cost of finding the minimum next event time is $O(\log N)$ operations per one event.

The main difficulty in the direct method for minimization in lines 4 and 5 is the need to compute the $N-1$ values $P_{i_* j}$ in line 4 and the K values $Q_{i_* k}$ in line 5. The $O(N+K)$ complexity burden of these computations reduces to $O(1)$ if sectorization is used, as described above. For example, when $D = 2$ only those $P_{i_* j}$ are computed for which the center of disk j belongs to one of the nine sectors neighboring the one whose member is i_*. This small number of $P_{i_* j}$ is subject to minimization in line 4.

The last but not the least comment in this section concerns the actual computational speed achieved by this algorithm. On a VAX8550 computer whose accepted value of speed is about 1 MFLOP (million floating-point instructions per second; for comparison, the accepted value of speed of CRAY-1 is more than 20 MFLOPs), the FORTRAN-implemented algorithm in Fig. 12 processes 150–450 pairwise collisions per second, depending on the disk density. Processing is faster, the higher the density. It takes less than 2 hr of CPU time of VAX8550 to reach the state presented in Fig. 5 starting from the initial configuration in Fig. 2.

APPENDIX B. A CONCRETE EXPRESSION FOR ASSIGNMENT (A.1)

We have

$$time \leftarrow t_* + t$$

where

$$t_* = \max(time1, time2)$$

and

$$t = \begin{cases} [-B - (B^2 - AC)^{1/2}]/A & \text{if } (B \leqslant 0 \text{ or } A < 0) \text{ and } B^2 - AC \geqslant 0 \\ +\infty & \text{otherwise, i.e., if } (B > 0 \text{ and } A \geqslant 0) \text{ or } B^2 - AC < 0 \end{cases}$$

with

$$A = |\mathbf{v}|^2 - a_0^2, \quad B = \mathbf{r} \cdot \mathbf{v} - a_0 a(t_*), \quad C = |\mathbf{r}|^2 - a(t_*)^2$$

$$\mathbf{r} = \mathbf{r}_{20} - \mathbf{r}_{10}, \qquad \mathbf{v} = \mathbf{v}_2 - \mathbf{v}_1$$

$$\mathbf{r}_{10} = \mathbf{r}_1 + \mathbf{v}_1(t_* - time1), \qquad \mathbf{r}_{20} = \mathbf{r}_2 + \mathbf{v}_2(t_* - time2)$$

The value t is the least positive real solution of the equation $At^2 + 2Bt + C = 0$, which is derived from $|\mathbf{r} + \mathbf{v}t|^2 = [a(t_*) + a_0 t]^2$. The latter represents the condition that the distance between the centers of spheres 1 and 2 equals the current diameter.

REFERENCES

1. B. J. Alder and T. E. Wainwright, *J. Chem. Phys.* **31**(2):459–466 (1959).
2. H. C. Andersen, *J. Chem. Phys.* **72**:2384 (1980).
3. C. J. Bashe *et al.*, *IBM's Early Computers* (MIT Press, Cambridge, Massachusetts, 1986).
4. C. H. Bennett, *J. Appl. Phys.* **43**:2727 (1972).

5. J. D. Bernal, *Proc. R. Soc. A* **280**:299 (1964).
6. F. P. Buff and F. H. Stillinger, *J. Chem. Phys.* **39**:1911 (1963).
7. S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases* (Cambridge University Press, Cambridge, 1953), Chapters 5 and 6.
8. K. E. Davis, W. B. Russel, and W. J. Glantschnig, *Science* **245**:507 (1989).
9. J. J. Erpenbeck and W. W. Wood, Molecular dynamics techniques for hard-core systems, in *Statistical Mechanics.* Part B: *Time-Dependent Processes*, B. J. Berne, ed. (Plenum, New York, 1977).
10. J. L. Finney, *Nature* **266**:309 (1977), and references therein.
11. H. L. Frisch, *Adv. Chem. Phys.* **6**:229 (1964).
12. P. H. Gaskell, in *Glassy Metals*, Vol. II, H. Beck and H.-J. Güntherodt, eds. (Springer, Berlin, 1983), pp. 5–49.
13. K. Huang and C. N. Yang, *Phys. Rev.* **105**:767 (1957).
14. M. G. Kendall, *A Course in the Geometry of n Dimensions* (Hafner, New York, 1961), p. 36.
15. D. E. Knuth, *Art of Computer Programming*, Vol. 3, *Sorting and Searching* (Addison-Wesley, 1973).
16. T. D. Lee, K. Huang, and C. N. Yang, *Phys. Rev.* **106**:1135 (1957).
17. P. W. Leung, C. L. Henley, and G. V. Chester, *Phys. Rev. B* **39**:446 (1989).
18. B. D. Lubachevsky, *J. Comp. Phys.* (1990).
19. G. Mason and W. Clark, *Nature* **207**:512 (1965).
20. C. A. Murray and D. H. van Winkle, *Phys. Rev. A* **34**:562 (1986).
21. D. R. Nelson and B. J. Halperin, *Phys. Rev. B* **19**:2457 (1979).
22. R. Pindak, D. J. Bishop, and W. O. Sprenger, *Phys. Rev. Lett.* **44**:1461 (1980).
23. C. A. Rogers, *Packing and Covering* (Cambridge University Press, Cambridge, 1964), p. 3.
24. G. D. Scott, *Nature* **194**:957 (1962).
25. N. J. A. Sloane, *Sci. Am.* **250**(1):116 (1984).
26. W. A. Steele, *J. Phys. Chem.* **69**:3446 (1965).
27. F. H. Stillinger, E. A. DiMarzio, and R. L. Kornegay, *J. Chem. Phys.* **40**:1564 (1964).
28. F. H. Stillinger and Z. W. Salsburg, *J. Stat. Phys.* **1**:179 (1969).
29. F. H. Stillinger and T. A. Weber, *J. Chem. Phys.* **83**:4767 (1985).
30. J. Vieillard-Baron, *J. Chem. Phys.* **56**:4729 (1972).
31. M. Widom, K. J. Strandburg, and R. H. Swendsen, *Phys. Rev. Lett.* **58**:706 (1987).