Nature of packs used in propellant modeling

F. Maggi, S. Stafford, and T. L. Jackson
Center for Simulation of Advanced Rockets, University of Illinois, Urbana, Illinois 61801, USA

J. Buckmaster*
Buckmaster Research, 2014 Boudreau Drive, Urbana, Illinois 61801, USA

(Received 13 August 2007; revised manuscript received 21 January 2008; published 11 April 2008)

In recent years we have constructed closely packed spheres using the Lubachevsky-Stillinger algorithm to generate morphological models of heterogeneous solid propellants. Improvements to the algorithm now allow us to create large polydisperse packs on a laptop computer, and to create monodisperse packs with packing fractions greater than 70% which display significant crystal order. The use of these models in the physical context motivates efforts to examine in some detail the nature of the packs, including certain statistical properties. We compare packing fractions for binary packs with long-known experimental data. Also, we discuss the near-neighbor number and the radial distribution function (RDF) for monodisperse packs and make comparisons with experimental data. We also briefly discuss the RDF for bidisperse packs. We also consider bounded monodisperse packs, and pay particular attention to the near-wall structure where we identify significant order.

DOI: 10.1103/PhysRevE.77.046107 PACS number(s): 89.90.+n

I. MODELING OF COMPOSITE PROPELLANTS: INTRODUCTION

Heterogeneous solid propellants are widely used in the rocket industry, and are likely to play an important role for as long as rockets are built. Fundamentally, they consist of oxidizer particles of order 1–100 μm diam embedded in a rubbery fuel binder. Various choices are possible, but a common one is ammonium perchlorate (AP) in hydroxy-terminated-polybutadiene (HTPB), and we imply no lack of generality if we place our discussion in this context. These components burn in a thin combustion layer, a few hundred microns thick, in the neighborhood of the propellant surface. It is also common to add metal particles to the binder, 10 μm or so in diameter, aluminum being the most common choice. These burn in the chamber gases at distances well removed from the surface.

Designers of rockets are concerned with a number of propellant-related issues, including the burning rate, the thermal and mechanical properties of the propellant, and, for metallized propellants, the behavior of the metal particles at the surface, including agglomeration. A study of these in a virtual engineering framework starts with a model for the morphology, and the relationship of this model to the physical reality is an important matter. This paper is concerned with some aspects of this problem when the model is generated by the dynamical packing algorithm originating with the work of Lubachevsky and Stillinger [1], as discussed in [2,3].

The aforementioned references discuss packs of spheres, and Ref. [3], in particular, describes strategies for accommodating the wide range of particle sizes that are typical of real propellants in the generation of packs suitable for combustion simulations. Real AP particles are not spherical, of course, but a study of packs of spheroids [4] suggests that nonsphericity is not an issue insofar as burning rates are concerned, provided the packs are statistically isotropic.

Burning rates for various AP or HTPB packs are calculated in [5], and comparisons made there with experimental data suggest that the effects of morphology on these rates can be satisfactorily predicted. Because the very smallest AP particles cannot be resolved numerically, and so must be homogenized into the binder, the thermal conductivity of the blend must be calculated and this is discussed in [6]. We have not had the need to examine the mechanical properties and behavior of such blends—e.g., effective Young's modulus, stress augmentation, von Mises stress, etc.—but the homogenization literature makes it clear that these kinds of things can depend on the fine statistical details of the packs, e.g., [7].

When aluminum is an ingredient of the propellant, agglomeration can be an important issue. Agglomeration occurs when the particles come to the surface, reside there for a while, and during this residence adhere to other particles. Agglomerates of an order of 100 μm diam are formed in this fashion, and are subsequently carried into the chamber with negative consequences, in addition to the positive energy addition. It would be of great value if the designer could fine-tune the propellant morphology to generate an agglomerate size distribution, which minimizes the negative consequences.[1] No completely predictive strategy has yet been developed, but there is good evidence that knowledge of the pack morphology and of the mean agglomerate size permits the prediction of the size distribution (i.e., the standard deviation, should the distribution be lognormal). This prediction is derived from a proximity model—roughly speaking, particles that are sufficiently close to each other within the solid will agglomerate—discussed in Ref. [8]. Use

*limey@uiuc.edu

[1]Frictional energy losses, accumulation in submerged nozzles, nozzle impaction, and exhaust signature.

of this model to explore the effects of morphology requires the construction of large packs, packs with hundreds of thousands of particles, and these need to be generated in, at most, a few hours of CPU time. Also, in this as in all combustion applications, the packing fraction (volume fraction of AP) must be large.

These remarks should make it clear that there is a need for the numerical construction of large polydisperse packs that can be generated quickly and have properties that are close to those of real packs. This paper discusses the issue. We know of only one other packing strategy that is used for propellant modeling, one developed at ATK-Thiokol [9], and at various places we examine the results of this work.

II. PACKING ALGORITHM

The algorithm is described in [1] and the application to propellant packs in [2,3]. For the most part, Stillinger and his co-workers have only been interested in monodisperse packs, but the propellant application is primarily concerned with polydisperse packs, with the largest particles being one or two orders of magnitude larger than the smallest particles.

The algorithm begins with an infinite computational domain defined by the periodic continuation of a cube (or cuboid) in three-dimensional space. Points are randomly assigned to this domain at time $t=0$ with random velocities. For $t>0$ these points grow linearly with time, to generate spheres. The growth rates vary amongst the spheres and their distribution defines the final distribution of diameters (of AP and aluminum particles) in the propellant. The largest particles have a radius $r(t)=at$, where t is time, capped at t_{end}. Typically, t_{end} is not chosen but is defined when the desired packing fraction is achieved, or when the computation terminates because the pack is "jammed." We shall call a the growth rate; smaller particles grow at a slower rate. Overlapping is prevented by including collisions in the algorithm. Jamming arises when the time between collisions becomes too small, but we note here and discuss later that, strictly speaking, within this dynamic framework jamming never occurs as it does in real packs.

Our first version of the algorithm was in no way optimal, and used a parallel platform message passing interface (MPI) to generate packs within an acceptable time frame. But recently we have had reason to improve our algorithm so that it can generate large packs on a laptop in decent time, and these improvements are described here. We claim no global superiority of our strategy, only that it is vastly superior to our old strategy, and enables us to achieve our application goals.

III. IMPROVED ALGORITHM

The improved algorithm has two major advantages.

(1) It enables us to construct large polydisperse packs on a laptop computer within an acceptable time frame.

(2) It accurately depicts rigid sphere packing, and both packing and statistical results compare favorably with experimental hard sphere packs.

These improvements arise from the use of strategies adopted (in some cases, for the first time) from the molecular dynamics (MD) literature.

Modern implementations of rigid sphere packing take advantage of an event-driven molecular dynamics (EDMD) approach. In EDMD, particles are advanced between "events," where an event is loosely defined as anything that changes a particle's state. The event could be a binary collision, a collision between a particle and a domain boundary, or a transfer of a particle across an internal or external boundary. Instead of advancing the particles by a fixed time step as in time-driven MD (TDMD), the particles are always advanced to the next event time. Each EDMD time step therefore requires two tasks.

(1) Find and execute the next predicted event.

(2) Update all event predictions influenced by this event.

These two tasks may be optimized separately. Optimization techniques used for the first task usually rely on an efficient priority queue algorithm. Optimization of the second task is more difficult and can involve neighbor lists or cell methods. A typical EDMD method will use a min heap to find the next event, an upper or lower triangular matrix to store collision time estimates for binary pairs, and a neighbor list or cell for each particle to reduce the number of binary collision prediction calculations. There are many excellent texts that describe these processes in detail [10,11]. For a concise description of a modern EDMD implementation using min heaps and neighbor lists, see Donev *et al.* [12].

Our EDMD-based packing method uses a priority queue for event handling and a hierarchical cell scheme to reduce the number of binary collision prediction calculations for polydisperse packs. We use a fully object-oriented implementation in C^{++} that provides complete encapsulation of the sphere, boundary, and cell objects. With our method it is not necessary to perform the collision validation checks necessary for our older strategy, nor is it necessary to store a triangular collision time matrix. In practice, our method has demonstrated $O(N)$ run times (for monodisperse packs) and $O(N)$ memory requirements over $10^3 \leq N \leq 10^6$. This makes it possible to generate the very large polydisperse packs we need in less than a day using a serial code on a laptop.

Our priority queue implementation uses a binary heap to process events. Binary heaps are outperformed by Fibonacci heaps for the find_min function, which has $O(\ln N)$ complexity for a binary heap, but $O(1)$ for a Fibonacci heap. However, processing each collision requires one find_min, one delete_min, and one insert operation, returning the overall complexity to $O(\ln N)$ for both heap methods. A parallel algorithm might further reduce the run-time cost of the heap operations [13]; unfortunately, the method is based on a triangular storage matrix for the collision times and thus has $O(N^2)$ memory requirements. A more sophisticated parallel algorithm using multiple heaps achieves $O(N)$ memory requirements [14] and the additional complexity could be justified by future requirements for larger packs. However, in our current applications with $N \leq 10^6$ the computation time is dominated by the binary collision prediction calculation. Thus, for the foreseeable future a binary heap will be sufficient for our purposes.

We use a hierarchical cell structure to reduce the number of binary collision prediction calculations that are necessary when updating the collision times. With the cell method, the particles' centroids are contained within cuboid cells whose

maximum enclosed spheres have larger radii than the contained particles. The cell keeps a list of all contained particles and a list of its boundaries, which may be internal (local cell boundaries) or external (domain boundaries). When predicting collisions, an optimized cell method can reduce the prediction computation time per particle to constant time; this is because only collisions between particles in neighboring cells need be computed. There are additional, inexpensive calculations to determine when the particle passes out of the cell into a neighboring cell or when a particle interacts with a boundary. Both cell transfers and boundary interactions are treated as events, and thus participate in the priority queue.

Although hierarchal cell methods have been implemented in two dimensions [15], we are not aware of any other analysis or implementation of the three-dimensional case. In our implementation, the cells are arranged in a three-dimensional tree structure. When particles outgrow their current cell, they are transferred up the tree to their cell's parent. In the transfer event, the particle is removed from its previous level so that it is always associated with a unique level. Since the binary collision calculation is required for a given particle with all other particles in the neighboring cells, the volume searched is proportional to r^D, where r is the cell radius and D is the dimensionality. Thus, it is advantageous to have each particle in the smallest cell that is practical in order to reduce the volume (and thereby reduce the number of binary pairs). Because of this, our implementation does not require the tree to be an octree. In our implementation, the number of cells at each level and in each dimension need only be integer multiples of the level immediately above it. However, in practice we have found octrees to perform well for very polydisperse packs, so the remainder of the results and discussion in this paper pertain to the octree implementation.

Figure 1 is a one-dimensional cartoon of the cell hierarchy showing search paths for sphere 1 interacting with spheres 2 and 3. Shading denotes cells that must be checked as well as the pathways that are traversed to find these relevant cells: (a) sphere 1 is located in a cell at the bottom of the tree and must search upwards through parent cells recursively; (b) sphere 1 is located at an intermediate level and must search upwards and downwards recursively; and (c) sphere 1 is moving in the direction indicated, and the trajectory calculation is used to recursively rule out subtrees that are not relevant. Were the trajectory method not used, all of the cells in (c) would need to be checked. In three dimensions, the number of ruled-out cells typically exceeds the number of checked cells by a factor of 3.

The computational efficiency of the basic cell hierarchy scheme breaks down when the particles have very different radii. This is due to the r^D search volume needed by the larger particles. For example, the smallest particles need only search their neighboring cells and their parents' neighboring cells, recursively all the way up the tree, with computation time $t_{\text{small}} \propto N_{\text{small}} 3^D h$, where N_{small} is the number of small particles and h is the height of the tree. At worst, t_{small} grows linearly with h, resulting in a very efficient search. By comparison, the largest particles must search their subtree along with all of their neighbors' subtrees, with $t_{\text{large}} \propto N_{\text{large}} 3^D (1 - 2^{Dh})/(1-2^D)$. The search time is proportional to a geometric series and is very expensive, even for $h=2$. The t_{large} could be partially offset in an optimistic scenario where there is one large particle per cell at the top of the tree. In that case, we might expect $N_{\text{large}}/N_{\text{small}} \propto 2^{-D(h-1)}$, leading to overall computational time

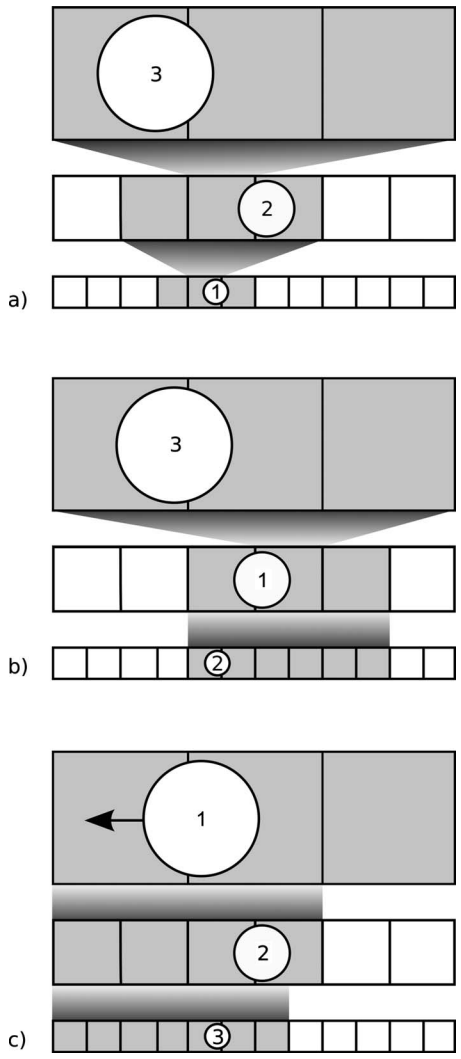

FIG. 1. A one-dimensional cartoon of the cell hierarchy search paths (see the text).

$$t_{\text{overall}} \propto N_{\text{small}} 3^D \left(h + \frac{2^{-D(h-1)} - 1}{1 - 2^D} \right). \tag{1}$$

This situation would arise, for example, if a pack was made by first completely filling the domain (until jammed) with the largest particles and then packing the smallest particles into the voids. The second term in parentheses in Eq. (1) would then be very small, and the run time would be negligibly affected by the polydispersity. In most relevant packings, however, we have fewer large particles than large cells. It is also unlikely that many, if any, of the particles' final radii will exactly match their container cells' radii (perfectly optimized cell sizes) or that any particle size modes will have radii that are a multiple of 2 times any other particle size mode (the best case scenario for an octree). Thus, in more

realistic polydisperse packs the overall run time is dominated by the calculations for large particles with their smaller neighbors. Because of this we construct the octrees by first attempting to fit the final radii of the largest particles as close as possible. After the largest cells are constructed, they are subdivided recursively. Subdivision is ended when the next subdivision would create cells too small to contain the smallest particles.

We use a large particle trajectory cell selection method to avoid the geometric series dependence of the overall computation time. In our method, the subtree cells are treated as (stationary) spheres themselves, and collision times are computed between the cells and the large particle of interest. The cell sphere is constructed as the smallest cell-centered sphere that encloses both the cell and any particle it could possibly contain. If the large particle will not collide with the cell, then any particles in the cell are ruled out. The procedure is recursive through the cell's offspring, so that the complete subtree is instantly also ruled out. To further optimize the calculation, the trajectory of the large particle that is used in the cell collision calculation is bounded by the current best estimate of the next collision time for the large particle. This straightforward trajectory selection process reduced the overall computation time by a factor of 4 for the polydisperse packs in this paper.

The cell scheme has the additional advantage of being well suited for handling various boundary geometries. We use three types of boundary conditions: rigid planar, periodic planar, and cylindrical. Rigid planar boundaries are easily handled by computing the binary collision between a particle and its reflection across the plane. For simulating the very large packs needed for energetic material modeling, we use periodic boundaries. A periodic cuboid has 26 neighboring images which must be included in the calculation, which could increase the computational burden 27-fold. Fortunately, with the cell hierarchy, boundary cells are computed as if they were inner cells, and the computational time is negligibly larger than the equivalent rigid planar calculation.

We also implemented a cylindrical boundary that more accurately reflects the geometry used in many experiments and real-world packs. The cylinder is bounded at each end by either rigid planar or periodic planar boundaries. Collisions with the cylinder are calculated by projecting the sphere onto a plane perpendicular to the cylinder axis and then computing the collision time between the two circles. This boundary is also easy to implement in the cell hierarchy simply by linking the boundary to the cells at the periphery of the cylinder. Figure 2 shows an example of a polydisperse cylindrical pack; it contains 70 008 particles with a ratio of largest diameter to smallest of 29.3 and was constructed on a laptop in 20 h. Note, however, that computational times are strongly dependent on the stopping criterion, the definition of jamming. Thus Kansai *et al.* [16] report the construction of a bidisperse pack of 10 000 particles, size ratio 10, that took 48 h on a 1 GHz Pentium machine, but do not specify the stopping criterion, and so a comparison with our calculation is not possible, although we can note that for their code the scaling with particle number is roughly N^2. However, our old code, similar in most important respects to that used in [16], is not suitable in a serial version for the generation of the packs needed for our combustion simulations, and the improvement we have been able to achieve is substantial. Much of this improvement is due to the use of a cell hierarchy, rather than neighbor lists.

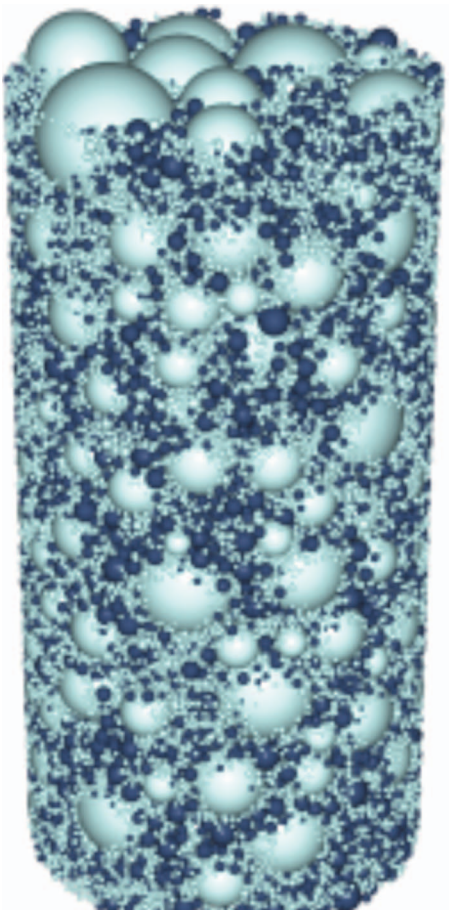

FIG. 2. (Color) A polydisperse cylindrical pack, periodic along the axis, size distributions from Table I.

Note that although in our application we are interested in large polydisperse packs such as the one shown in Fig. 2, experimental data with which we can compare the properties of our numerically generated packs is only available for relatively simple packs, and these comparisons occupy the significant part of our discussion. All of the packs generated in the paper have periodic planar boundary conditions, unless explicitly specified differently.

IV. COLLISIONS

As we noted earlier, collisions must be accommodated in order to avoid particle overlap. Moreover, these must be pointwise-in-time events to fit within the general framework of the algorithm. Because each particle is growing and there is an outward surface velocity relative to the centroid, a classical elastic collision does not guarantee a pointwise event, and in some cases, depending on the collision speeds and the growth rates, it is necessary to add extra impulses, generating increments to the rebound velocities. The original Lubachevsky-Stillinger algorithm uses the relative velocity

TABLE I. Left columns: coarse and fine AP distributions used in Fig. 2. Right columns: aluminum distribution.

Number	Diameter	Number	Diameter
2	348.320	2	99.685
3	324.110	6	92.135
4	301.580	12	85.155
6	280.615	23	78.705
8	261.110	39	72.745
11	242.960	61	67.235
13	226.070	92	62.140
16	210.355	134	57.340
18	195.730	186	53.080
21	182.125	252	49.060
22	169.465	329	45.345
23	157.685	418	41.910
24	146.725	516	38.735
23	136.525	619	35.800
22	127.035	723	33.090
20	118.205	823	30.585
18	109.990	912	28.265
15	102.345	985	26.125
13	95.230	1037	24.150
10	88.610	1064	22.320
8	82.450	1065	20.630
6	76.720	1039	19.065
4	71.390	989	17.620
3	66.425	918	16.285
2	61.805	831	15.050
1	57.510	733	13.910
1	52.156	632	12.855
2	24.000	530	11.885
44	30.160		
184	26.975		
540	24.125		
1312	21.575		
2792	19.295		
5316	17.260		
9180	15.435		
14464	13.805		
20879	12.350		

of the particle surfaces when computing impulses, and energy is added at each collision; our old version uses the centroid velocities with rebound additions for every collision that depend only on the growth rates, and not on the collision speeds. Of particular relevance is the ratio of the growth rate a_i ($=a$ for the most rapidly growing particles) to the speed $\|\vec{v}_i\|$. After many collisions in which the kinetic energy is enhanced this ratio can become very small so that little growth occurs between collisions, and jamming cannot be achieved within a reasonable time frame. Also, it can become impossible to generate low-density jammed packs. Because of this, it is necessary to manage the kinetic energy of the pack and, in some sense, minimize its growth. The qualification is necessary because it is only the increase in energy that arises from changes in speed that is of concern, not from the changes in mass.

The literature abounds with methods to counter the increase in kinetic energy. The simplest method is to periodically renormalize the velocities of all particles either by rescaling [11] or by setting the velocities to zero (this method is only relevant when the pack is nearly jammed, see, for example, [1]). In either case, the renormalization requires restarting the priority queue, which is an expensive operation. Rescaling the velocities has also been shown to violate energy equipartition [17], which could be problematic. Another interesting technique borrowed from molecular dynamics is the use of stochastic thermostats that place the pack into contact with an imaginary heat bath. Some of these, such as the Andersen thermostat [10], can be implemented as random collisions with ghost particles, and can thus be integrated into the EDMD scheme as a single-particle event. However, the Andersen thermostat in particular has been shown to pollute the transport coefficients [10] and thus can reasonably be expected to reduce the efficiency of sampling the configuration space during packing. The Lowe-Andersen thermostat removes this shortcoming by considering particle pairs, and it appears to be a promising alternative for efficient packing [18]. Unfortunately, all stochastic thermostats become inefficient as the collision frequency increases because they require additional event predictions at each thermostat collision and also because they require several independent random numbers. When packing spheres, high quality random number generators such as the Mersenne Twister [19] produce a double precision number in the time it takes two binary collisions to be calculated. Thus for high growth rates and the corresponding need for many thermostat collisions, the additional events and random number generation can dominate the computation time.

Our approach is to simply minimize the amount of energy added during each binary collision. When none is needed, none is used; when it is needed an amount is added sufficient merely to cause the surfaces to move away from each other at a minimal speed, say 10^{-30}. The amount of energy added in this way is monitored by defining and calculating a pseudotemperature, a substitute for the set of growth-rate or speed ratios. This pseudotemperature also plays a role in the specification of the initial velocity distribution at the start of the calculation, and links the algorithm more closely to a molecular dynamics framework. In our earlier calculations the velocity components were randomly sampled from the interval $[-1,1]$; here we sample from a Maxwellian (normal) distribution.

We are only interested in situations when N, the number of particles in the pack, is large, and it makes the discussion of our pseudothermodynamics more agreeable if we suppose that N is an asymptotically large parameter. Then the classical definition of temperature is related to the mean kinetic energy of the particles by

$$\frac{3}{2}kT = \frac{1}{N}\sum_i \frac{1}{2}m_i\|\vec{v}_i\|^2, \tag{2}$$

where k is Boltzmann's constant. Each mass is given by

$$m_i = \frac{4}{3}\rho\pi r_i^3 = \frac{4}{3}\rho\pi(a_i t)^3, \tag{3}$$

and it is this t^3 dependence that is not relevant to us. And so we define a pseudotemperature by

$$T = \frac{1}{3N}\sum_i \frac{a_i^3}{a}\|\vec{v}_i\|^2, \tag{4}$$

nondimensional when the velocities are appropriately defined. Initial values are defined by

$$\vec{v}_{i,0} = \sqrt{\frac{T_0}{a_i^3}}(\alpha_1, \alpha_2, \alpha_3), \tag{5}$$

where T_0 is the initial temperature, and the α_i are sampled from independent normal distributions.

Thus the system starts out in thermal equilibrium (small particles travel faster, on average, than large particles) and would remain so were it not for the inelastic collisions. Note that the Maxwellian is consistent with Eq. (2). Note also that this approach could also be applied to systems of particles with rotational degrees of freedom to ensure the packing initially satisfies equipartition of energy.

We use a simple method for determining when a pack has jammed. During the run, the packing fraction is computed after each "pass," where a pass is defined as a set of N sequential interparticle or solid boundary collisions (particle transfers and other events are not counted). It is of course unlikely that all N particles will participate in collisions in any given pass; nevertheless, the concept of a pass allows us to discuss jamming independently of the number of particles N. We simply stop the pack when the change in the packing fraction over a pass is less than some specified limit value, e.g., the jamming criterion after the ith pass is $(\rho_i - \rho_{i-1})/\rho_i < \varepsilon$, where ε is the limit value.

For monodisperse packs, ε is closely related to the "distance to jamming" $(1-\rho/\rho_J)$, where ρ_J is the jamming density (see [20] for a detailed discussion of pack properties as this distance approaches zero). To see this, we start with the fact that ρ is proportional to t^3 so that in the last pass,

$$\Delta\rho/\rho = 3\Delta t/t \sim \varepsilon, \tag{6}$$

where Δt is the length of the pass, and we have linearized, since ρ is close to the limit value ρ_J. At the beginning of the pass, when the particle diameter is d, the gap between each particle is $\sim(d_J - d)$, so that since ρ is proportional to d^3,

$$(\text{gap}) \sim (d/3)(1-\rho/\rho_J) = (at/3)(1-\rho/\rho_J). \tag{7}$$

Thus the time interval before a particle collides with its neighbor is

$$\sim (at/3c)(1-\rho/\rho_J), \tag{8}$$

where c is the representative surface speed, the sum of the growth rate a, and a positive translational speed. But in this interval $\sim N$ collisions occur, so that it is the length of the pass, and comparing Eqs. (6) and (8) we have

$$\varepsilon \sim (a/c)(1-\rho/\rho_J). \tag{9}$$

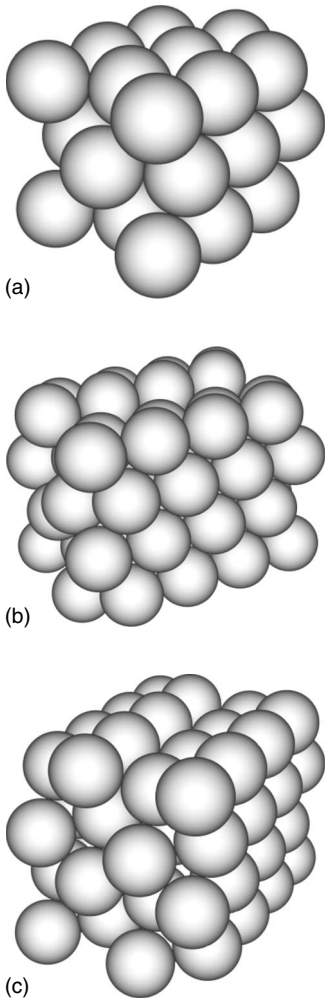

FIG. 3. Lattice packs. (a) Double Nested, (b) Close Pack (Side 1), (c) Close Pack (Side 2)

Throughout the paper we generate most of the packs using the improved algorithm, a serial code with temperature-based initial conditions, what we shall call the T algorithm; growth rates are normalized with a so that the maximum growth rate is 1 and the velocity–growth-rate ratio is controlled by T_0. But there are a few results obtained using the old parallel code for which velocity components are sampled on the interval $[-1,1]$ and the ratio is controlled by a; we shall call this the a algorithm.

V. MONODISPERSE PACKS

In this section we examine monodisperse packs. There is a maximum packing fraction for a monodisperse pack of spheres, well defined for lattice packs, less so for random packs. Lattice packs are characterized by regular repetitive structures. Thus (see [21]) we have cubic lattice—53.36%, orthorhombic—60.46%, double-nested—69.81%, and close-packed—74.05%, the largest attainable for monomodal spheres. The last two arrangements are shown in Fig. 3. As we shall see, the close-packed lattice is relevant to order that we obtain for high-density packs, and near rigid boundaries.

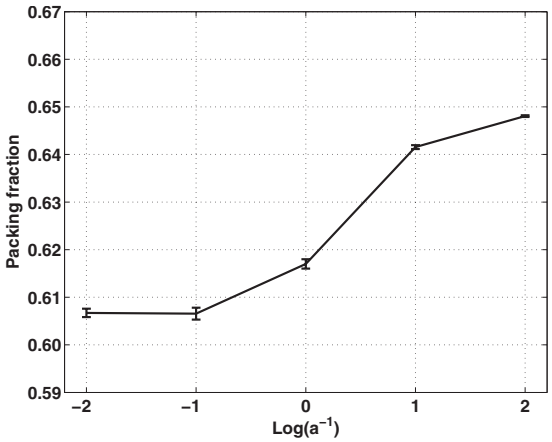

FIG. 4. Final packing fraction as a function of inverse growth rate (a algorithm).

The very concept of random packs is not well defined, because all packs, whether generated experimentally or numerically, have some degree of order. Equally, maximum packing fractions are not well defined. But attempts to create random packs typically lead to packing fractions in the range of 59%–64% where values at the low end are characteristic of what are called loose packs (LRP), and values at the high end are characteristic of what are called dense packs (DRP) [21,22]. (It is a minor syntactical misfortune that dense packs are not called tight packs.)

The a algorithm is capable of modeling both jammed LRP and DRP by varying the growth rate; except for extremely small growth rates, the smaller the growth rate the higher the packing fraction. Figure 4 shows this for three monomodal packs of 3000 spheres, the packs differing because of different initial conditions. The bar for each choice of a shows the maximum, minimum, and average fractions. End values are 60.74% for $a=100$ (an LRP) and 64.78% for $a=0.01$ (a DRP). The ballistic-deposition algorithm used by Webb and Davis [9] yields a value of 60%. Experimental data obtained using mechanical shaking can be found in [21] (62.5%) and [22] (64%).

Figure 5 shows the packing fractions that can be achieved with the T algorithm, to be compared to Fig. 4. We are not

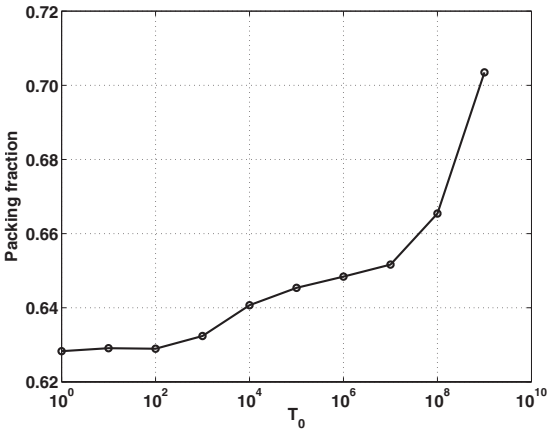

FIG. 5. Packing fraction of a 10 000 particle pack as a function of the initial temperature T_0 (T algorithm).

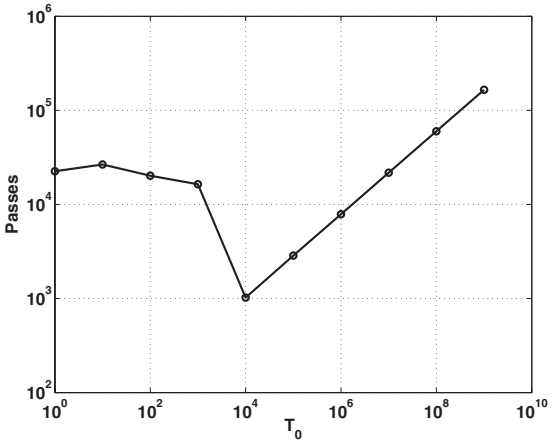

FIG. 6. Number of passes used to generate Fig. 5, $\varepsilon=10^{-7}$.

aware of any other reports of packing fractions greater than 0.7 obtained using the Lubachevsky-Stillinger algorithm. The corresponding number of passes before jamming is achieved are plotted in Fig. 6; ε is equal to 10^{-7}. High temperatures generate packs of higher density than that of a double-nested lattice, implying significant order. Later, we shall see that these ordered packs have many of the statistical characteristics of a close-packed lattice. That packs generated using low temperatures have less order than those generated using high temperatures is clear from Figs. 7 and 8, the former (packing fraction equal to 0.6283) calculated for $T_0=1$, the latter (packing fraction equal to 0.7035) for $T_0=10^9$. Figure 8 looks very much like a lattice pack with dislocations, whereas Fig. 7 is far less regular.

VI. BIDISPERSE PACKS

Typical packing fractions for propellants are in the neighborhood of 78%, and so for this purpose monodisperse packs have little relevance. Here we discuss bidisperse packs, the simplest of the polydisperse variety, and packs for which there is experimental data. The fine component can fill some of the spaces between spheres of the coarse component, and therefore generate higher packing fractions. Not surprisingly, the greater the ratio between the size of the coarse component and size of the fine component, the greater is the maximum packing fraction. The maximum is attained at a

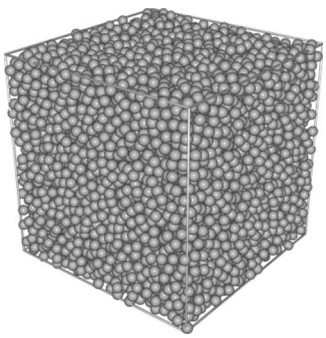

FIG. 7. A 10 000 particle pack for $T_0=1$, packing fraction 0.6283.

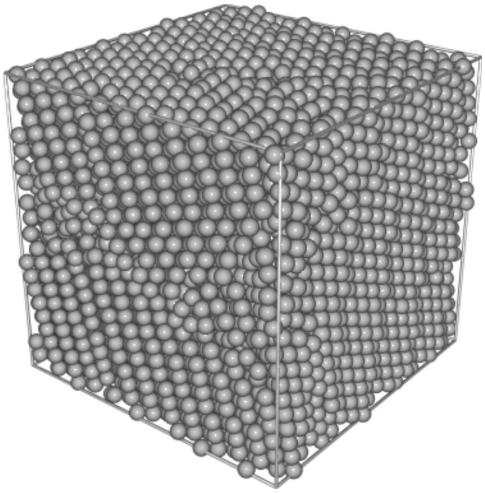

FIG. 8. A 10 000 particle pack for $T_0=10^9$, packing fraction 0.7035.

uniquely defined value of the ratio of the volume fraction of coarse to that of fine.

The earliest reports of polydisperse packs obtained using the Lubachevsky-Stillinger algorithm may be found in [2,3,16], the latter a discussion of bidisperse packs only. Here we compare numerical results for bidisperse packs with experimental data of McGeary [21].

McGeary created a number of bidisperse packs. The size of the particles was defined by using meshes, a standard procedure in the particle sorting business. He used seven-mesh spheres as a coarse component together with several finer components, achieving a maximum packing fraction of approximately 84% for the finest. We compare McGeary's data with numerical results for two growth rates, $a=0.2$, and $a=1$ (a algorithm). We also plot results from [9] generated using a ballistic-deposition strategy.

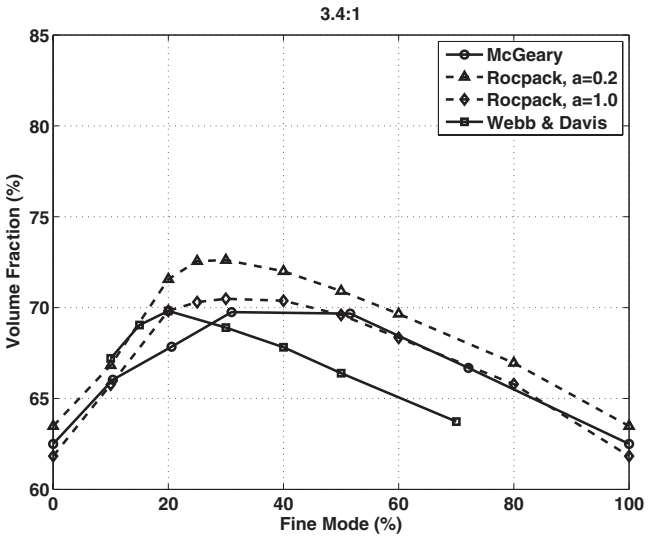

FIG. 9. Packing fraction vs fine-mode percentage for a 3.4:1 coarse-to-fine size ratio.

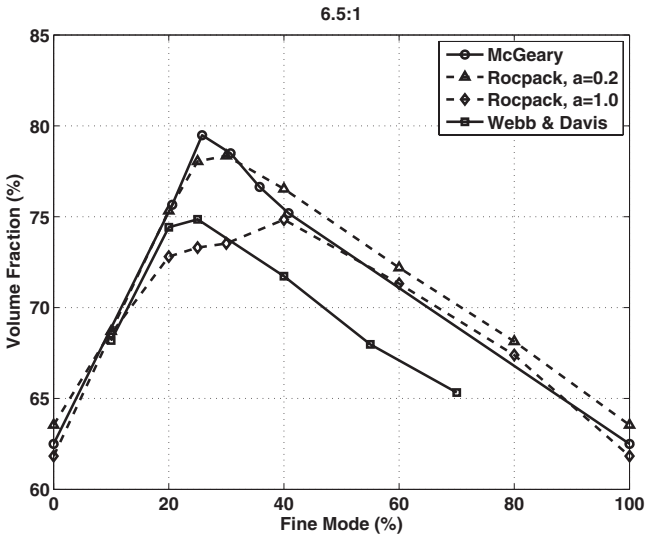

FIG. 10. Packing fraction vs fine-mode percentage for a 6.5:1 coarse-to-fine size ratio.

VII. COARSE-TO-FINE SIZE RATIO 3.4:1

This models the experiments with seven-mesh and 20-mesh, and 20 000 spheres were used in the simulations; comparisons are shown in Fig. 9. Except at a fine-mode fraction of 20, the results for $a=1$ are in excellent agreement with McGeary's data. It is difficult to pin down with precision the location of the maxima, but for $a=1$ the two largest calculated packing fractions correspond to the coordinates (30, 0.7049) and (40, 0.7038); for $a=0.2$ they are (25, 0.7255) and (30, 0.7262). The Webb and Davis results peak early, and then underpredict.

VIII. COARSE-TO-FINE SIZE RATIO 6.5:1

This models the 7/40 mesh study of McGeary, using 50 000 spheres, and the results are shown in Fig. 10. The slower growth rate ($a=0.2$) gives the closest agreement with experiment, with the largest calculated packing fraction of 78.35% at a fine-mode percentage of 30. When $a=1$ the results deviate significantly from the experimental values in the fine-mode percentage interval (20,40)%; the largest calculated value is 74.84% at a fine-mode percentage of 40%.

IX. COARSE-TO-FINE SIZE RATIO 16.5:1

The largest size ratio is achieved with 7/80 meshes and the simulations use 80 000 spheres; only results for $a=0.2$ are shown (Fig. 11). Agreement with experiment is very close, with a largest calculated value of 82.01% at a fine percentage of 25. In all of the cases we discuss here, the ballistic-deposition algorithm typically yields packing fractions significantly lower than the experimental values. In such an algorithm there is, of course, no tuning parameter comparable to a. Whether this matters depends on the application. In the modeling of a propellant with a significant fraction of very fine AP, so fine as to be unresolvable numerically, a relatively low packing fraction for the resolvable

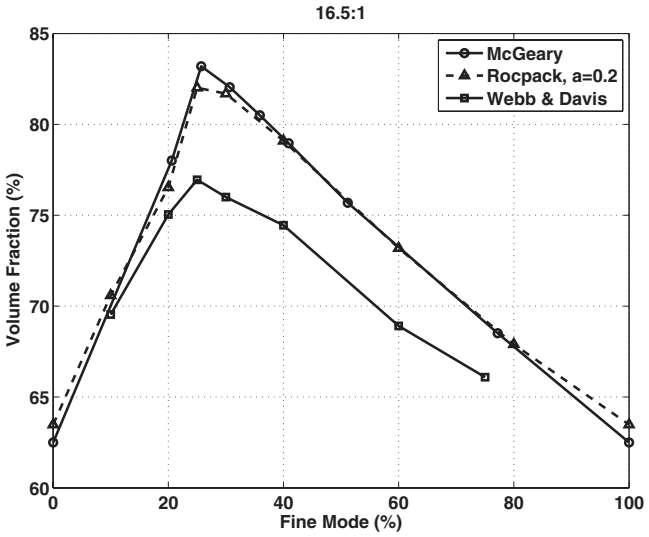

FIG. 11. Packing fraction as a function of fine-mode percentage for a 16.5:1 coarse-to-fine size ratio.

particles might be all that is needed, with the remaining fraction homogenized into the binder. But in other situations, the inability to reach experimentally attainable packing fractions could mean that the strategy is not useful.

It is important to note that the construction of polydisperse packs is much more computationally intensive than the construction of monodisperse packs, and the use of hierarchal cells rather than neighbor lists plays a crucial role in increasing the efficiency of the algorithm so that a serial platform can be used. All of the old work cited in this section [2,3,16], uses neighbor lists, and as a consequence it was necessary for us to partially parallelize the code (not the heaps) in order to generate packs for our combustion studies [5,6].

X. STATISTICAL ANALYSIS OF PARTICLE PACKINGS

The previous section focused on the packing algorithm and the products of this algorithm were characterized merely by the packing fraction and by two related concepts, loose and dense packings (LRP and DRP). The algorithm generates different jammed-state packing fractions according to the choice of the growth parameter a, or the initial temperature T_0, and variations in the fraction correspond to variations in the inner structure. Thus experimentalists report that a DRP can only be achieved with strong shaking or shocking, whereas this is not necessary for an LRP. The nature of the inner structure is of intrinsic interest, but also, from the propellant modeling perspective, it is relevant to the simulations. Certain thermal and mechanical properties of the pack are likely to depend on it, and bounds and estimates for these quantities in the homogenization literature often depend on its statistics. Thus it is relevant to discuss the statistics and, where possible, compare them with experimental data.

Here and earlier we wrote of "jamming." From an algorithmic point of view this can refer simply to the state in which the time between collisions is so small that the calculation is effectively stalled, but more precise discussions are possible. Indeed, Donev *et al.* define the concepts of local jamming, collective jamming, and strict jamming [23]. Local jamming occurs when each particle in a subset of the pack is locally trapped by its neighbors and is unable to translate. Such particles are necessarily touched by at least $d+1$ peers not in the same hemisphere where d is the spatial dimension. In three dimensions this condition is fulfilled for spheres with more than three contacts. Particles with fewer contacts are called rattlers, since they can move.

Donev *et al.* note that computer-generated packs often do not satisfy the local jamming criterion. That is certainly true of packs generated using the Lubachevsky-Stillinger code and its variations, as a small gap exists between all particles when the computation ceases. However, given enough computer time, most gaps can be made arbitrarily small, and the number of contact points between particles can be well defined [20]. For those concerned with the important mathematical questions associated with packing, and for whom packs of modest size are sufficient, this is important. We, however, are concerned with generating large packs in reasonable times, and are not concerned with sophisticated mathematical questions, but with whether the pack statistics match those of experimental packs. In the determination of experimental morphology (by X-ray tomography, for example) there is error, both from measurement uncertainties and the reconstruction algorithm. Then contact is only defined within the constraint of a tolerance. Thus, to make comparisons with the numerical results we also use a tolerance, and particles are said to be in contact if their closest surface separation is smaller than some assigned constant. This is not a new idea; see, for example, Refs. [22,24,25].

Packing structure as well as particle organization can be characterized using spatial statistics, something that has been done for a long time. Most recently, Aste *et al.* generated a number of experimental packs of up to 150 000 monomodal spheres inside a cylindrical container and then analyzed them using XCT (x-ray computed tomography) to identify the placement of centers; statistical analyses were then possible [22]. A number of theoretical studies on ideal or modeled packs are reported in [26,27,24,20,9].

Here we discuss pair correlation and coordination number for monodisperse packs of varying packing fraction; one would expect that the degree of order is related to the packing fraction. We also discuss high-density lattices, experimental results, and correlate certain features of the statistics with certain kinds of order. And we briefly discuss pair correlation for bidisperse packs.

XI. COORDINATION NUMBER

It is known that a single sphere can touch at most 12 equal spheres [28], a condition satisfied by a close-packed lattice. If all spheres in a pack are jammed, each must be in contact with at least four others. The mean number of contacts is called the coordination number (or contact number or kissing number).

Experimental results of Bernal and Mason for monodisperse packs [29] reveal an average number of contacts of 6.4 for a dense packing (packing fraction 62%) and 5.5 for a

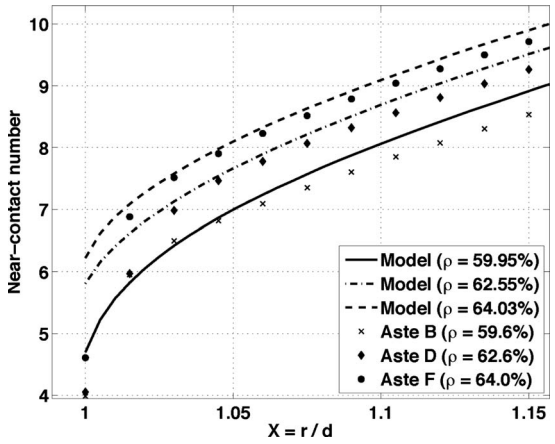

FIG. 12. Average near-contact number as a function of $X \equiv 1+\epsilon$ (35 000 particles) for three packing fractions, compared with data of Aste *et al.*

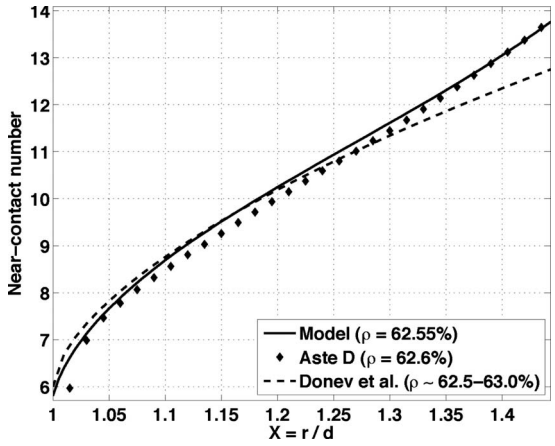

FIG. 13. Average near-contact number as a function of X (35 000 particles) for a single packing fraction, compared with data of Aste *et al.* and results of Donev *et al.* [20].

loose packing (packing fraction 60%). Also, a near-neighbor (or near-contact) number was determined for these packs, where a near-neighbor to a reference sphere is defined as a sphere whose center is separated by no more than 1.05 diameters from the center of the reference. These numbers are 8.5 and 7.1. In general, near neighbors can be defined using $1+\epsilon$ instead of 1.05, and it is common, if ϵ (the tolerance) is small (0.01 for example), to identify these near-neighbor numbers with the contact numbers. Clearly the contact number is a function of ϵ, and if ϵ is smaller than ε the Lubachevsky-Stillinger algorithm will generate packs with zero contact number.

Figure 12 shows variations in the contact number with $1+\epsilon \equiv X$ for three different packs of 35 000 particles, the packs differing one from the other because of different choices of a. Similar curves are reported in [30,24], but here we include comparisons with experimental data obtained by Aste and his colleagues. They have studied monodisperse packs using x-ray tomography [31,22] and have provided us with their raw data so that we might make comparisons. Single pack results obtained by us and by Donev *et al.* [20] are shown in Fig. 13.

Table II lists the contact number for some experimental data and for packs that we have generated (the inhouse name for our code is Rocpack) for various values of ϵ. Of particular interest is the data of Gotoh as he used a large number of particles, as needed for results to be statistically significant.

XII. RADIAL DISTRIBUTION FUNCTION (RDF)

The concepts of touching number and rattlers (which we do not discuss) are nearest-neighbor concepts, but the statistics of particles at greater distances are of importance, and this brings us to the radial distribution function, also known as the pair distribution or pair correlation. It is defined as the probability of finding a particle center at a distance between r and $r+\Delta r$ from the center of a reference sphere [25]. The discrete definition is

$$g(r,\Delta r) = \frac{V}{N}\frac{n(r,\Delta r)}{4\pi r^2 \Delta r}, \tag{10}$$

where N is the number of particles in the pack, V is the pack volume, and $n(r,\Delta r)$ is the number of particles in the shell of inner radius r and thickness Δr. This is averaged over all particles; it asymptotes to 1 as $r \to \infty$.

It is obvious that some separation distances are more likely than others. For example, in a monodisperse pack

TABLE II. Average contacts n_c of Rocpack and experimental packings (data from [29,30,32]). Whether the smallest value of X is an appropriate value for the first column data of Bernal and of Mason is not clear from their reports.

Source	Fraction	N	$n_c(1.005)$	$n_c(1.02)$	$n_c(1.05)$	$n_c(1.1)$
Bernal-LRP	60%	420	5.5	N.A.	7.1	N.A.
Bernal-DRP	62%	476	6.4	N.A.	8.5	N.A.
Mason	N.A.	536	4.7	6.8	8.0	8.9
Gotoh	63.6%	7934	N.A.	7.05	8.0	9.0
Rocpack-LRP	59.95%	35000	5.22	6.04	7.00	8.06
Rocpack-Mid	62.55%	35000	6.15	6.81	7.67	8.69
Rocpack-DRP	64.03%	35000	6.62	7.27	8.10	9.09

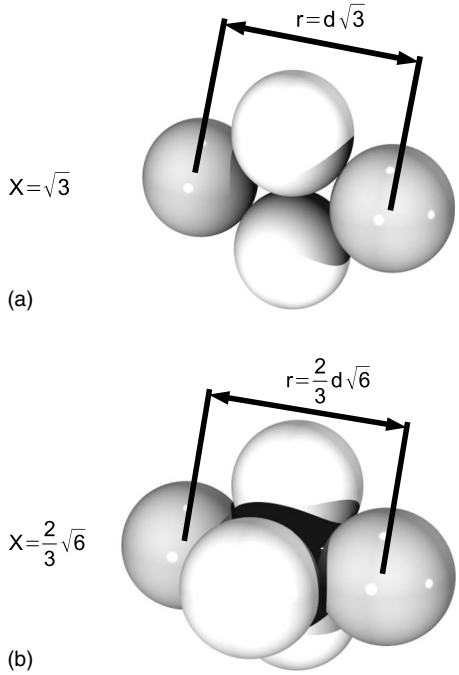

FIG. 14. Possible sphere configurations and the location of corresponding peaks in the RDF. (a) Four close spheres, (b) Tetrahedron configuration.

contact particles will be separated by the diameter d so that we would expect the RDF to have a sharp peak at $X \equiv r/d = 1$. The location of other peaks at $X = \sqrt{3}$ and $\frac{2}{3}\sqrt{6}$ can be understood by referring to the arrangements drawn in Figs. 14(a) and 14(b). The peak at 2 corresponds to three spheres in a row.

XIII. RDF OF MONODISPERSE PACKS

In calculating the RDF for a prescribed pack, it is necessary to make a choice of Δr. If too large, the function is smoothed and peaks are obscured; if too small the noise generated because only a finite number of particles are used can also obscure the peaks. And so it is necessary to make some trial runs to optimize the choice. We do not show the results

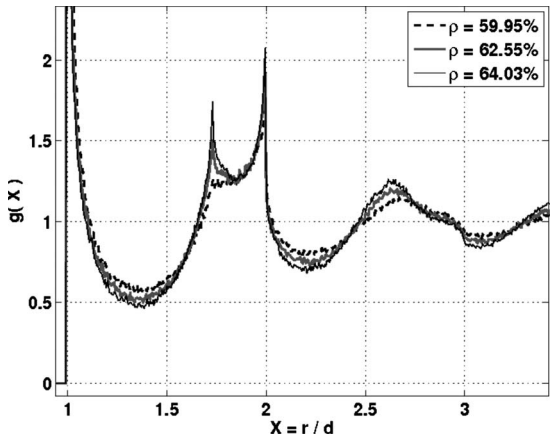

FIG. 15. RDF for different, relatively low, packing fractions.

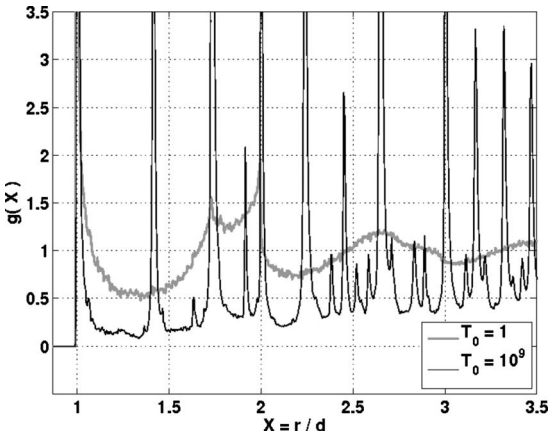

FIG. 16. RDFs for two T-algorithm packs, $T_0 = 1$ and $T_0 = 10^9$.

of these trials, but we came to the conclusion that $\Delta r = 0.005d$ lies within an optimal band. We make the same choice for bidisperse packs when d is taken to be the diameter of the smaller spheres.

The growth rate a is an important parameter in the a algorithm; large values prevent the spheres from distributing themselves in a tight fashion; small values permit such a distribution. Similar consequences arise using the T algorithm when T_0 and the tolerance are varied. Figure 15 shows the RDFs for three packs of mass fractions typical of those that can be generated using the a algorithm. It is noteworthy that the peak at $\sqrt{3}$ apparent at the highest packing fraction is completely lost at the lowest. It is also noteworthy that for no pack do we see a peak at $(2/3)\sqrt{6} = 1.633\ldots$ [see Fig. 14(b)]. The tetrahedron arrangement is characteristic of the close-packed lattice and does not appear to arise in low-density random packs.

A similar exercise using the full power of the T algorithm yields far more striking results. Figure 16 shows the RDFs for the two packs of Figs. 7 ($T_0 = 1$) and 8 ($T_0 = 10^9$). The many peaks for the high-temperature result (including one at 1.633…) reveal the high degree of order in this pack. Later we shall compare it with the RDF of a close-packed lattice.

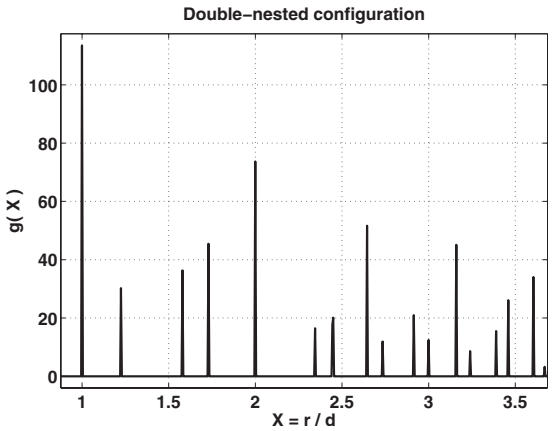

FIG. 17. RDF of double-nested lattice.

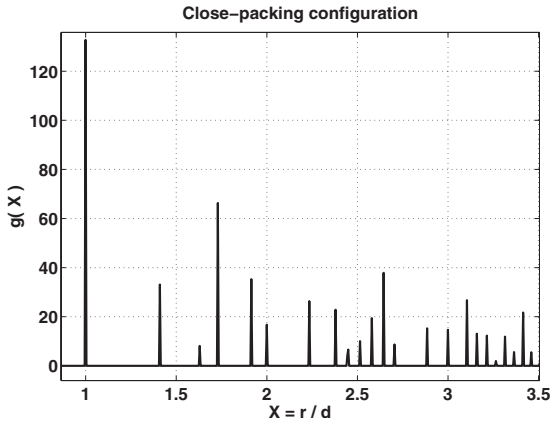

FIG. 18. RDF of close-packing lattice.

XIV. RDFs FOR LATTICES

We have noted that the peaks in figures such as 15 arise because of specific nonrandom structures. Lattices consist entirely of nonrandom structures, and it is of interest to examine the peaks that then arise; in principal, all can be explained by geometry. Since this has been discussed before, we shall just provide a brief description of double-nested and close-packed lattices. The former has a packing fraction typical of packs generated numerically; the latter has a significantly higher packing fraction, but has order that is related to that of high-density numerical packs.

A. Double-nested lattice

The double-nested lattice is shown in Fig. 3(a), and the RDF in Fig. 17. The first six peaks, easily identified from the lattice geometry, are at 1, $\sqrt{\frac{3}{2}}=1.224\ldots$, $\sqrt{\frac{5}{2}}=1.581\ldots$, $\sqrt{3}=1.732\ldots$, 2, and $\sqrt{\frac{15}{2}}=2.738\ldots$.

B. Close-packing lattice

The close-packed lattice is shown in Figs. 3(b) and 3(c), and the following peaks can easily be identified: 1, $\sqrt{2}=1.414\ldots$, $\frac{2}{3}\sqrt{6}=1.632\ldots$, $\sqrt{3}=1.732\ldots$, 2, and $\sqrt{5}=2.236\ldots$.

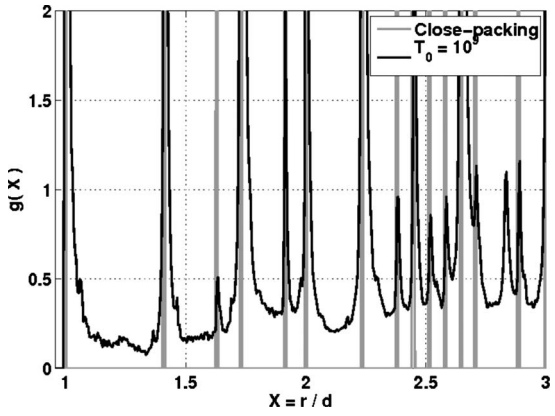

FIG. 19. RDF of the pack of Fig. 8, and close-packing values.

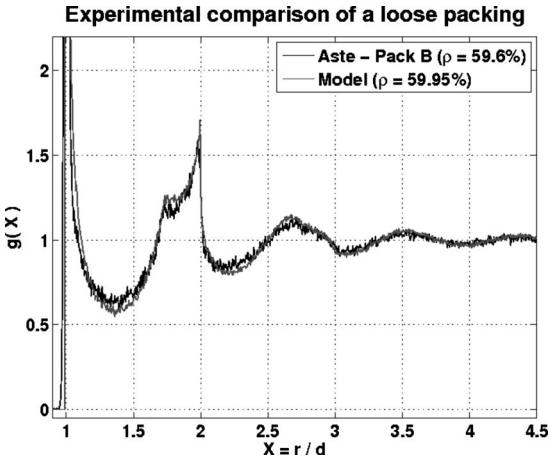

FIG. 20. RDF comparison with experimental data, (a).

Note that the peak at 1.632… arises from the tetrahedron arrangement of Fig. 14(b).

The RDF is shown in Fig. 18. And in Fig. 19 we show the values along with the RDF of the *T*-algorithm pack of Fig. 8. In view of the packing fraction of the pack (70.35%), it might be thought that we should compare its RDF with that of the double-nested lattice (packing fraction 69.81%), but there is little correlation between the two. On the other hand, there is clearly a strong correlation for the close-packing lattice. Thus, although we are still significantly shy of the close-packing density (74.05%), the pack must have a morphology that has many of the characteristics of the lattice.

XV. COMPARISONS WITH THE MONODISPERSE DATA OF ASTE *et al.*

RDFs for packs generated using the Lubachevsky-Stillinger algorithm have been reported before, e.g., [20], but here we compare the calculated results with experimental data. For it is, we believe, important that the packs used in a virtual engineering framework have the same statistics as the real packs. It is possible, of course, that real packs will have different statistics according to the manner in which they are

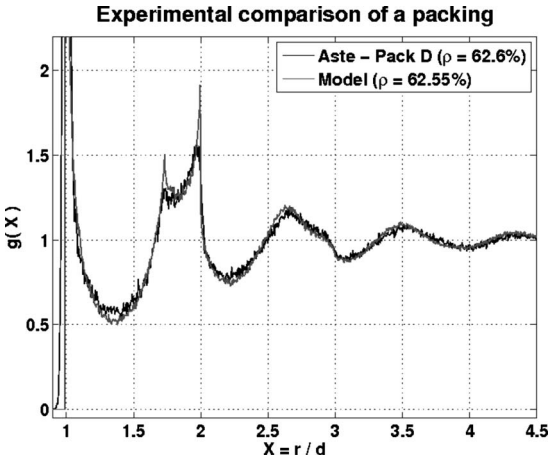

FIG. 21. RDF comparison with experimental data, (b).

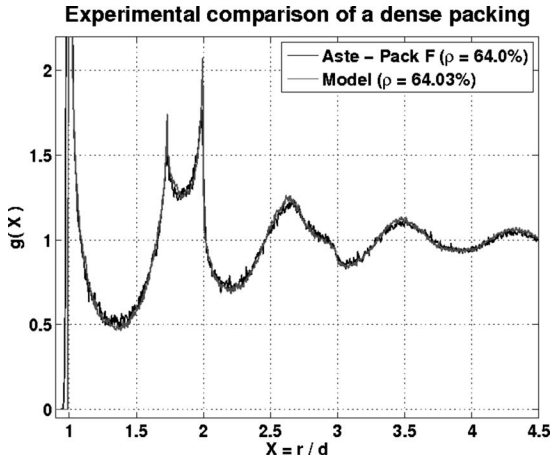

FIG. 22. RDF comparison with experimental data, (c).

generated, a matter that we are not in a position to address at the present time, but Aste and his colleagues at the Australian National University in Canberra have measured the RDFs for a number of monodisperse packs using x-ray tomography [31,22], and have provided us with their raw data so that we might make comparisons, and we do that here. Our packs use 35 000 spheres in a periodic cube and are generated using the T algorithm with packing fractions of 59.95%, 62.55%, and 64.03%; the experimental packing fractions are 59.6%, 62.6%, and 64.0%. The comparisons are shown in Figs. 20–22.

We note the coordinates of the first two peaks lying to the right of $X=1.5$, the model coordinates first, the experimental coordinates second: {Pack B: peak 1 (1.735,1.26), (1.730,1.22); peak 2 (1.995,1.71), (1.980,1.59)}; {Pack D: peak 1 (1.730,1.50), (1.720,1.30); peak 2 (1.995,1.91), (1.985,1.55)}; {Pack F: peak 1 (1.730,1.74), (1.730,1.50); peak 2 (1.995,2.07), (1.985,1.77)}. Note that the model peak values are always greater than the experimental ones, significantly so in some cases. But these differences are consistent with errors in the experimental measurements. Aste *et al.* assume a Gaussian uncertainty in the center distance for two touching spheres, and estimate that the average standard deviation is 0.015 diameters. If we randomly adjust the model center coordinates accordingly, and recalculate the RDFs, the

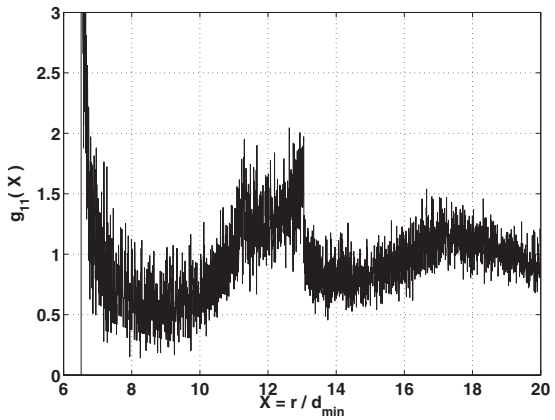

FIG. 23. g_{11}-6.5:1 Bidisperse packing, case A.

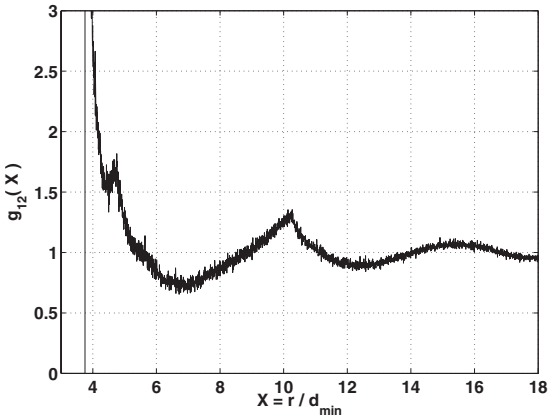

FIG. 24. g_{12}-6.5:1 Bidisperse packing, case A.

model peaks are decreased and more closely match the experimental values. Similar smoothing would occur if we allowed for a distribution in the sphere diameters.

XVI. RDF OF BIDISPERSE PACKS

For polydisperse structures the counterpart to the RDF function is the partial pair correlation function ($g_{ij}(X)$), the number density of particles of type j that are within shells r to $r+\Delta r$ centered on particles of type i. The function is normalized so that it has the value 1 as $r\rightarrow\infty$, and it defines a symmetric matrix. In its determination there is a problem if the size ratio of the particles is large. For example, for a 16.5:1 ratio in a pack of 50 000 spheres there are only 45 large spheres vs 49 500 small spheres. In this case the statistics involving only the coarse component are nonrepresentative.

A. Pack with size ratio 6.5:1, case A

We consider a pack with 50 000 particles, a 9 to 1 coarse-to-fine mass ratio (i.e., 90% of the mass comprises coarse particles), and a packing fraction of 68.7%. It is close to one of McGeary's data points [21]. Figure 23 is a plot of g_{11} where the index 1 refers to the larger particles. Because, as we have already noted, there are so few of these, the function

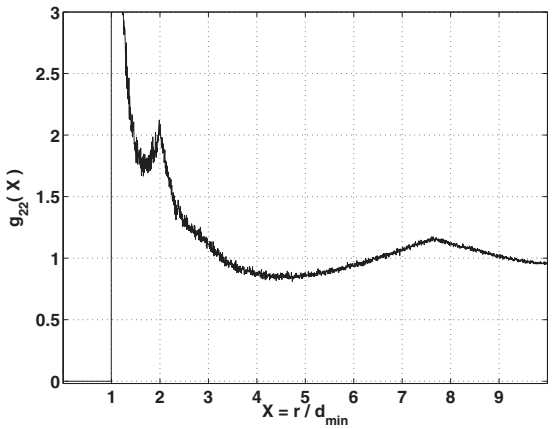

FIG. 25. g_{22}-6.5:1 Bidisperse packing, case A.

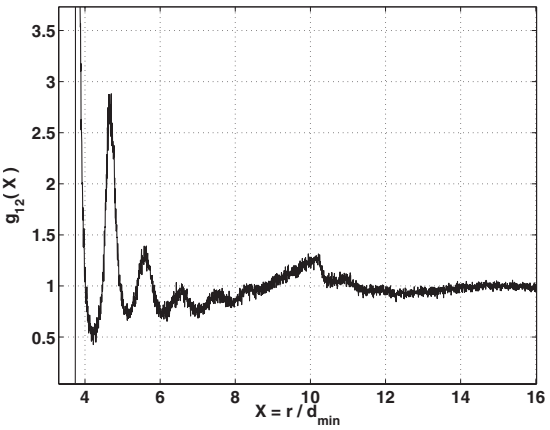

FIG. 26. g_{12}-6.5:1 Bidisperse packing, case B.

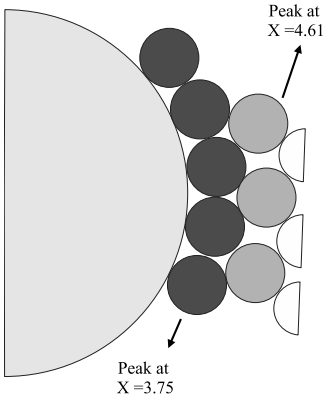

FIG. 28. A possible local configuration for a 6.5:1 bidisperse pack, (a).

is noisy, but trends are apparent. They are similar to those of a monodisperse pack, not a surprising observation. Recall that d is the diameter of the smaller particles. Figure 24 is a plot of g_{12} and reveals structure in the neighborhood of $X=4.5$ and 10. And Fig. 25, a plot of g_{22}, reveals order at $X=2$ and in the neighborhood of $X=7.7$.

B. Pack with size ratio 6.5:1, case B

Here we change the mass ratio to 75% coarse, 25% fine, achieving a 73.1% packing fraction. Now there are so few coarse particles that it is difficult even to observe trends in g_{11}, and so we do not show it. Figure 26 displays five well-defined peaks for X smaller than 8 for g_{12}, and Fig. 27 shows a number of peaks for g_{22}. Some of these can be understood within the monodisperse framework, but others are a consequence of the bidisperse morphology. Thus a layer of small particles in contact with a large one will generate a peak in g_{12} at $X=3.75$, Fig. 28. The closest small particles in a second layer beyond the first define a peak at 4.61, Fig. 28, the furthest at 4.75, Fig. 29, and so one might expect a single averaged peak between these two limits.

XVII. BOUNDED MONODISPERSE PACKS

In the earlier sections we only considered unbounded periodic packs, but in applications there are always boundaries, and it is of interest to examine the effects of rigid boundaries on the pack morphology. We have examined three different pack geometries: a bounded cube, a bounded right-circular cylinder of infinite length, periodic along the axis, and a finite right-circular cylinder, bounded on all sides, but only present results for the latter configuration.

In the generation of such packs using the Stillinger algorithm, collisions must be accounted for between the spheres and the walls, and the code is easily modified to accommodate these. Our discussion is concerned only with the results. That there will be wall effects (a wall layer) is recognized, for example, in the experimental work of Aste and his colleagues [22]; because they are interested in the properties of unbounded packs, they randomly glue spheres to the walls in an attempt to eliminate the wall layer. Our purpose here is to identify the thickness of the wall layer, and some understanding of its morphology. We only consider monodisperse packs.

There are various ways of characterizing the local pack structure, and we choose to do it in the following fashion. We start by defining a sampling volume or bin. Thus for a cube, for example, a sensible choice would be a sheet of finite thickness parallel to two of the bounding faces; for a cylinder it would be either a cylindrical shell coaxial with the cylinder, or a finite thickness circular disk perpendicular to the axis. In each bin of volume V we count the number of sphere

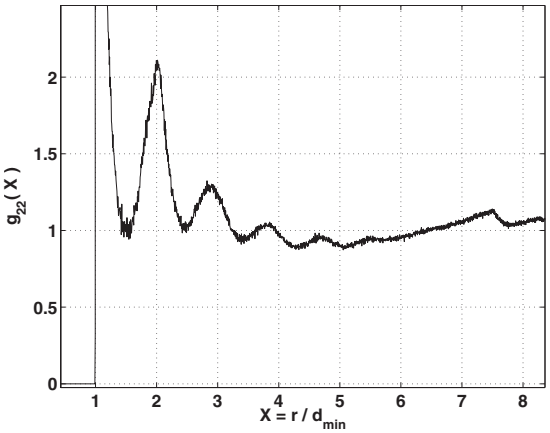

FIG. 27. g_{22}-6.5:1 Bidisperse packing, case B.

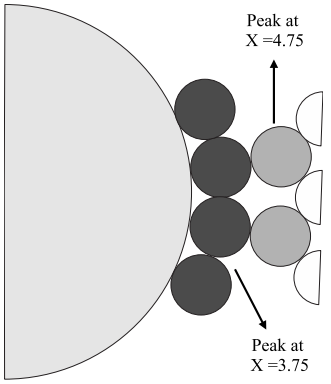

FIG. 29. A possible local configuration for a 6.5:1 bidisperse pack, (b).

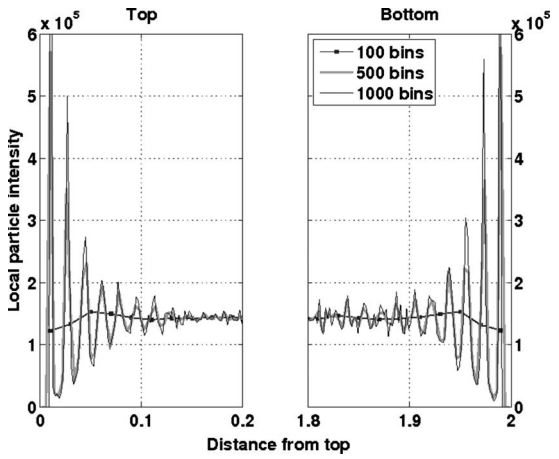

FIG. 30. Center-number density along the axial direction calculated for different bin numbers.

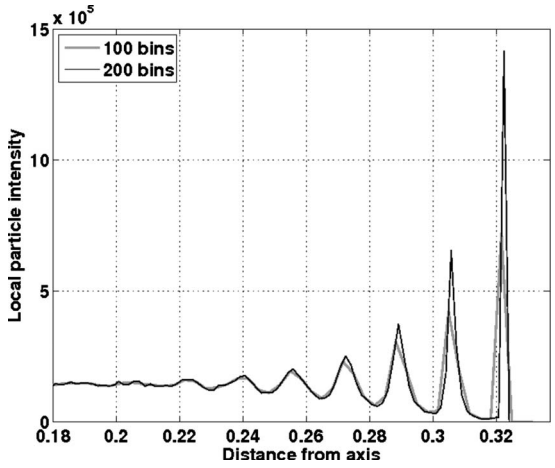

FIG. 31. Center-number density along the radial direction.

centers located within it, and use this number (N) as a measure of the number of particles within the bin, defining a center-number density by

$$\lambda = \frac{N}{V}. \tag{11}$$

Thus consider a bounded cylinder of height 2, diameter 2/3, containing 100 000 particles. The packing fraction is 62.57%, the overall number density is 143 239.4, and each particle has a diameter of 0.0202. Figure 30 shows the axial variation of center-number density for different choices of the number of bins. The axial distance is defined by the center of each bin. Note that for 1000 bins each bin has thickness 0.002, for 500 bins the thickness is 0.004, and for 100 bins the thickness is 0.02, essentially the particle diameter.

A couple of observations may be made.

(1) If the particle centers were randomly distributed, the center-number density would be constant. This is approximately true at a distances greater than 0.1 from each end, but not within the wall layer which, accordingly, is approximately 5 particle diameters in thickness.

(2) The large oscillations within the wall layer imply order imposed by the boundary constraint. Indeed, there are 5–6 peaks in the layer, correlating with the particle diameter. Thus, for example, for a cubic lattice (for which the following estimates are easily made) there would be approximately 890 particles in each layer near the end walls, and for a bin thickness exactly equal to the particle diameter the center-number density would be constant with a value of approximately 1.28×10^5. For a bin of vanishing thickness the center-number density would be zero except at distances $d/2$, $3d/2$, $5d/2$,… from an end wall ($d=0.0202$ is the particle diameter) where the values would be approximately $2550/\mu$ where μ is the bin thickness, yielding 12.75×10^5 for the 1000 bin case, 6.38×10^5 for the 500 bin case.

Figure 31 shows the radial distribution of the center-number density for 100 bins (shell thickness 0.0033) and 200 bins (shell thickness 0.0017) and, as expected, here also the wall layer is 5–6 particle diameters thick. The expected peaks for a local cubic lattice structure are approximately 7.73×10^5 for 100 bins, and 15.27×10^5 for 200 bins.

To gain more insight into the structure, we have replotted Fig. 31 in Fig. 32 using 333 bins (shell thickness 0.0010) and measuring the distance from the side wall in units of the particle diameter (scaled shell thickness 0.0496). Also the number of sphere centers is plotted, rather than their density. For a double-nested lattice the peaks would be at $\{0.5, 1.366, 2.232, 3.098, 3.964\}$; for a close-packed lattice they would be at $\{0.5, 1.317, 2.133, 2.950, 3.766\}$. We shall compare our peaks with the latter.

Thus the first peak (the 11th bin) is located in the interval $[0.496, 0.546]$, of which 0.5 is an interior point. The second peak is at the 28th bin, and the one to its immediate left (the 27th) is located in the interval $[1.290, 1.339]$, which includes 1.317. The third peak is at the 45th bin, and the one to its immediate left (the 44th) is located in the interval $[2.133, 2.182]$, which includes 2.133. The fourth peak is at the 61st bin, and the one to its left is located in the interval $[2.926, 2.976]$, which includes 2.950. The fifth peak is at the 78th bin, and the one to its left is located in the interval $[3.769, 3.819]$, which barely excludes 3.766. Thus, as would be expected for an imperfect lattice, there is a small outward displacement in the mean particle locations from what one

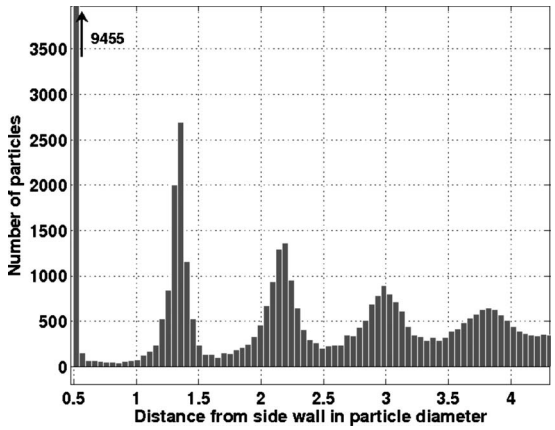

FIG. 32. Particle distribution vs wall distance (in particle diameters).

would get for a perfect lattice, but the correlation with the close-packed lattice is very strong.

XVIII. CONCLUSIONS

In this paper we have used a modified version of the Lubachevsky-Stillinger code to examine sphere packings in a context relevant to the modeling of rocket-propellant morphology. Various algorithmic refinements enable us to pack large numbers of spheres, of different sizes, in an efficient manner. Run times on a laptop computer are sufficiently modest that, with the addition of a graphic user-interface (GUI) (presently under development), the code could be of value to both the university and the industrial community. Packs can be generated in periodic cuboids, bounded cuboids, and in cylinders that are bounded or periodic along the axis.

The initial conditions include the specification of the velocities of randomly distributed points, and we do this in one of two ways: for the T algorithm, for which the maximum growth rate is 1 and the velocity or growth-rate ratio is controlled by a pseudotemperature T_0, the initial velocity components are proportional to $\sqrt{T_0}$ and are sampled from a normal distribution; for the a algorithm, the ratio is controlled by a and the velocity components are sampled on the interval $[-1,1]$. The T algorithm is capable of generating monodisperse packs of high density, higher than that of a double-nested lattice, and these high-density packs display significant order.

For bidisperse packs, using the a algorithm, we have compared the variations in packing fraction with the fine-mode percentage with experimental data of McGeary. For the larger coarse-to-fine size ratios (6.5 and 16.5) excellent agreement is achieved with the choice $a=0.2$. For the smaller size ratio (3.5) the choice $a=1$ yields better results, with the smaller a leading to significant overprediction. Apparently, mechanical shaking is not particularly effective as a packing tool when the size ratio is small.

A study of the radial distribution function for monodisperse packs reveals sharp peaks corresponding to local order. Only a modest number of such peaks are generated using the a algorithm, or for modest values of T_0 using the T algorithm, but when T_0 is large a large number of peaks can be generated. When the latter are compared with the peaks defined by a lattice pack, a strong correlation is achieved with the close-packed lattice. For modest packing fractions (small number of peaks) there exist experimental measurements of the radial distribution function, and we obtain excellent agreement with this data. For bidisperse packs for which the corresponding statistical metric is the partial pair correlation function, certain peaks can be correlated with certain expected arrangements of large and small particles.

For bounded packs (rigid boundaries) it is intuitively clear that the presence of the boundary introduces local order. Indeed, there is a boundary-layer effect, and for monomodal packs the layer thickness is approximately 5 particle diameters. If we examine the pdf of the sphere-center distribution there are 5+ peaks in the neighborhood of a boundary that correlate approximately with integer multiples of the sphere diameter. When examined more closely, the peaks correlate closely with those that would be expected for a close-packed lattice.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy through the University of California under Contract No. B523819, and by the Air Force Research Laboratory under Contract No. FA9550-06-C-0078, program manager Dr. A. Nachman. We are grateful to T. Aste of the Australian National University for providing us with the pack morphology data used in the comparisons of Figs. 11, 12, and 20–22. Ownership of this data resides with ANU.

[1] B. D. Lubachevsky and F. H. Stillinger, J. Stat. Phys. **60**, 561 (1990).
[2] G. M. Knott, T. L. Jackson, and J. Buckmaster, AIAA J. **39**, 678 (2001).
[3] S. Kochevets, J. Buckmaster, T. L. Jackson, and A. Hegab, J. Propul. Power **17**, 883 (2001).
[4] X. Wang, J. Buckmaster, and T. L. Jackson, J. Propul. Power **22**, 764 (2006).
[5] L. Massa, T. L. Jackson, and J. Buckmaster, J. Propul. Power **21**, 914 (2005).
[6] L. Massa, T. L. Jackson, J. Buckmaster, and M. Campbell, Proc. Combust. Inst. **29**, 2975 (2002).
[7] R. Lipton, J. Mech. Phys. Solids **47**, 1699 (1999).
[8] T. L. Jackson, F. Najjar, and J. Buckmaster, J. Propul. Power **21**, 925 (2005).
[9] M. D. Webb and I. L. Davis, Powder Technol. **167**, 10 (2006).
[10] D. Frenkel and B. Smit, *Understanding Molecular Simulation* (Academic Press, New York, 2002).
[11] M. P. Allen and D. J. Tildesley, *Computer Simulations of Liquids* (Oxford University Press, New York, 1987).
[12] A. Donev, F. H. Stillinger, and S. Torquato, J. Comput. Phys. **202**, 737 (2005).
[13] M. Mu, J. Comput. Phys. **179**, 539 (2002).
[14] S. Miller and S. Luding, J. Comput. Phys. **193**, 306 (2004).
[15] M. Wackenhut, S. McNamara, and H. Herrmann, Eur. Phys. J. E **17**, 237 (2005).
[16] A. R. Kansai, S. Torquato, and F. H. Stillinger, J. Chem. Phys. **117**, 8212 (2002).
[17] S. C. Harvey, R. K. Tan, and T. E. Cheatham III, J. Comput. Chem. **19**, 726 (1998).
[18] C. P. Lowe, Europhys. Lett. **47**, 145 (1999).
[19] M. Matsumoto and T. Nishimura, ACM Trans. Model. Comput. Simul. **8**, 3 (1998).
[20] Aleksandar Donev, Salvatore Torquato, and Frank H. Stillinger, Phys. Rev. E **71**, 011105 (2005).
[21] R. K. McGeary, J. Am. Ceram. Soc. **44**, 513 (1961).

[22] T. Aste, M. Saadatfar, and T. J. Senden, Phys. Rev. E **71**, 061302 (2005).
[23] Aleksandar Donev, Salvatore Torquato, Frank H. Stillinger, and Robert Connelly, J. Appl. Phys. **95**, 989 (2004).
[24] Alexander Bezrukov, Monika Bargiel, and Dietrich Stoyan, Part. Part. Syst. Charact. **19**, 111 (2002).
[25] K. Lockmann, L. Oger, and D. Stoyan, Solid State Sci. **8**, 1397 (2006).
[26] William M. Visscher and M. Bolsterli, Nature (London) **239**, 504 (1972).
[27] G. Mason and W. Clark, Nature (London) **211**, 957 (1966).
[28] John Leech, Math. Gaz. **40**, 22 (1956).
[29] J. D. Bernal and J. Mason, Nature (London) **188**, 910 (1960).
[30] G. Mason, Nature (London) **217**, 733 (1968).
[31] T. Aste, M. Saadatfar, A. Sakellariou, and T. J. Senden, Physica A **339**, 16 (2004).
[32] Keishi Gotoh and John L. Finney, Nature (London) **252**, 202 (1974).