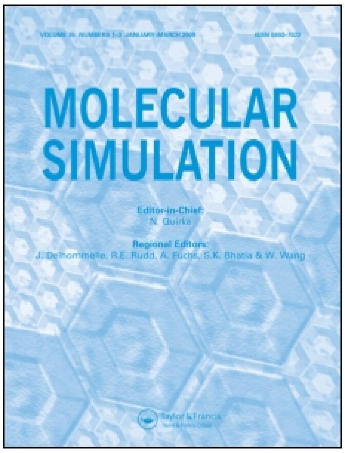

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Algorithm to Compute the Free Area in a 2-D Configuration of Anisotropic Particles

S. M. Ricci[a]; J. Talbot[a]; P. Viot[bc]
[a] School of Chemical Engineering, Purdue University, West Lafayette, IN, USA [b] Laboratoire de Physique Théorique des Liquides, Université Pierre et Marie Curie, Paris, Cedex 05, France [c] Unité de Recherche Associée au CNRS. (URA 765),

To cite this Article Ricci, S. M. , Talbot, J. and Viot, P.(1994) 'Algorithm to Compute the Free Area in a 2-D Configuration of Anisotropic Particles', Molecular Simulation, 13: 1, 61 — 75
To link to this Article: DOI: 10.1080/08927029408022185
URL: http://dx.doi.org/10.1080/08927029408022185

Molecular Simulation, 1994, Vol. 13, pp. 61-75
Reprints available directly from the publisher
Photocopying permitted by license only

ALGORITHM TO COMPUTE THE FREE AREA IN A 2-D CONFIGURATION OF ANISOTROPIC PARTICLES

S.M. RICCI and J. TALBOT

School of Chemical Engineering, Purdue University West Lafayette, IN 47907, USA

P. VIOT

Laboratoire de Physique Théorique des Liquides, Université Pierre et Marie Curie, 4, place Jussieu 75252 Paris Cedex 05 France*

(Received June 1993, accepted July 1993)

Molecular simulations which involve the insertion of particles into bulk system configurations often encounter statistical sampling problems at high density where the free surface fraction is small. This paper details an algorithm to compute the free area in a two-dimensional configuration of spherocylinders. By selecting trial positions for particles only in regions where addition is possible, the efficiency of simulations can be increased by up to two orders of magnitude compared with uniform sampling over the entire surface. An application of the method to the simulation of Random Sequential Adsorption (RSA) processes is presented.

KEY WORDS: Free area, insertion methods, RSA, anisotropic, discretization

1 INTRODUCTION

The concept of free volume has proved of great value in the development of the statistical mechanics of condensed phases. In a system of particles with hard core interactions (i.e., the potential between two particles is infinity if they overlap, zero otherwise) the free volume, V_0, is that volume of the system into which the center of an additional particle may be inserted without causing overlap. In equilibrium systems the free volume is simply related to the chemical potential

$$\mu = \mu^0 - kT\ln\phi \tag{1}$$

where $\phi = V_0/V$,with V the total volume of the system [1]. The concept is also of central importance in the non-equilibrium Random Sequential Addition (RSA) process, in which particles are added randomly and sequentially to a volume without any relaxation. Here, if the number of attempted additions per unit time, k_0, is constant, then

*Unité de Recherche Associée au CNRS.(URA 765).

$$\frac{d\rho}{dt} = k_a \phi \tag{2}$$

As the volume approaches saturation, the available volume fraction becomes vanishingly small. In the saturated state in which it is impossible to add more particles, $\phi(\rho_\infty) = 0$.

The free volume may be estimated from molecular simulations with a fairly obvious sampling procedure: a number of attempts is made to insert a particle into an arbitrary configuration at a position selected from a random uniform distribution. If no overlap would result in the trial position, the number of successes is incremented by one. Note that the particle is never actually inserted but the fraction of successful attempts provides an estimate for ϕ. In RSA, ϕ is estimated by averaging the number of addition attempts required to reach a given density over a number of independent runs. The inverse of the average gives the fractional free volume and thus ϕ.

In both equilibrium and non-equilibrium systems, sampling problems arise at high densities where ϕ is small. Most attempts at placement result in failure and a large number of trials is required to obtain ϕ with reasonable accuracy. From henceforth, we restrict ourselves to a consideration of two-dimensional systems for which "free volume" is replaced by "free area". The reason for this is two-fold. First, two dimensional configurations are most appropriate for the application of RSA to the description of the adsorption of molecules on solid surfaces. Second, the geometrical problems associated with more sophisticated techniques for the evaluation of free volume (area) are less severe in 2-D than in 3-D. Indeed for the 2-D hard disk system one may compute the free area exactly for a given configuration [2, 3]. The programming effort required is non-trivial, though, and there has been no attempt to extend the method to three dimensions.

For equilibrium systems of particles interacting via continuous potentials, the chemical potential is still related to the insertion probability via (1) [4, 5]. The geometrical interpretation of this quantity as a free volume is no longer valid. However, the same sampling problems exist at high densities. By applying an excluded volume map in the calculation of chemical potential, Deitrick *et al* [6] were able to increase the efficiency of the particle insertion method by up to two orders of magnitude compared with uniform sampling. We use a similar construction in the present work.

We are concerned with the development of an efficient algorithm for the evaluation of the free area of systems composed of non-spherical, hard particles. The additional complication in these systems is that the available area depends on the orientation of the trial particle. Unfortunately, the geometrical complications prohibit the exact evaluation of the free area as is possible for hard disks. However, the simple algorithm of uniform sampling may be improved upon considerably. The strategy is to identify regions of the surface which cannot accept a new particle in any orientation. The random number generator is then applied only to that part of the surface which has not been so eliminated.

In section 2 we detail a method which, given a configuration of spherocylinders (discorectangles in 2-D), identifies the portion of surface which is unavailable to the center of a new particle in any orientation and therefore approximates the free area. In section 3, we show how the method is used in RSA and report simulation results.

2 COMPUTATION OF FREE AREA

For anisotropic particles, the region of surface which is excluded to the center of a new particle depends on the particle's orientation. The object of this algorithm is to determine those regions which are unavailable to a new particle in any orientation. Since the range of allowed orientations for a new particle varies from point to point on the surface, it is necessary to test every possible position for the new particle. The only way to accomplish this is to discretize the surface so that there are a finite number of possible positions. The following development focuses on testing a discrete point on the surface for its ability to accept the center of a new particle. If the full range of orientations is excluded to a new spherocylinder placed at the test point by neighboring particles on the surface, then the point is unavailable.

2.1 *Grid Construction*

The test begins by searching the region of surface in the vicinity of the test point for neighboring particle centers. This is accomplished using a simple time saving device that has been used in RSA simulations of hard disks which involves the generation of a systematic grid on the surface such that only one particle center can lie within a grid cell without overlap [9, 11]. For spherocylinders, the grid is constructed by first defining the relative proper area of a particle

$$A_{re} = \frac{\pi b^2 + 4b^2(\alpha - 1)}{A} \tag{3}$$

where A represents the area of the simulation surface, b is the semiminor axis of the sphero-cylinder, and $\alpha = a/b$ is the axial ratio (a is the semimajor axis). The numerator is the area of a spherocylinder with axial ratio α. Since only one particle center must be allowed to fit in a grid cell without overlap, the diagonal of the cell must be approximately equal to, but less than, the width of the particle, $2b$. The length of the side of the cell would then be $s_c \leq \sqrt{2}\mathrm{b}$. The number of cells on a side of the surface is $n_c = S/(2b)$ where S is the length of a side of the simulation box. Since n_c is an integer in the simulation, it is defined in FORTRAN by the statement $NC = INT(S/(SQRT(2.)^*B)) + 1$, where NC, S, and B correspond to n_c, S and b, respectively. Addition of 1 ensures that the diagonal of a grid cell is less than $2b$. The grid construction is useful in that the cells signify the location of individual particles. Suppose the position of particle N is (x,y). We declare a two-dimensional, integer array, $CELL(I,J)$, where I and J denote the Ith and Jth cell positions along the x and y axes, respectively. The cell positions I and J are determined by the following FORTRAN statements

$$\begin{aligned} I &= X(N)^*NC - 1 \\ J &= Y(N)^*NC + 1 \end{aligned} \tag{4}$$

where integer truncation occurs in the multiplication. We then set $CELL(I,J) = N$, meaning that the center of particle N lies in that cell. When $CELL(I,J) = 0$, the cell does not contain a particle center. The grid is used to restrict the search for particles in the vicinity of the test point to neighboring grid cells. In addition, if the position for a trial particle falls in an occupied cell, overlap can be deduced without having to perform an overlap test because this check is always made first. When a cell with a nonzero value is found, the program uses the variable *CELL* (I,J) as the index for the position and orientation of the particle whose center lies therein. We also use the grid to set the scale of discretization by specifying the number of discrete points along the side of each grid cell, n_p. In this way, there are n_p^2 possible positions for a particle within each cell.

2.2 *Excluded Orientation Range*

When a neighboring particle is found, the program must determine the range of orientations which the neighbor would exclude to a trial particle if it were to be placed at the test point. The solution of this problem begins with a convenient mathematical definition of a spherocylinder: the locus of points at a distance $b/2$ from a line segment of length $l = 2(a - b)$. We refer to this as the defining line segment (DLS) of the spherocylinder. See figure 1. The dotted, larger region in the figure is the "limiting" exclusion region of the particle which is inaccessible to the DLS of other spherocylinders. To determine the excluded orientation range, imagine placing the center of a trial particle at the test point, rotating it about the point, and noting where it comes into contact with the neighboring spherocylinder. The two vectors from the center of the trial particle to the points of contact between its DLS and the limiting exclusion surface of the neighbor particle define the excluded orientation range.

Let r_0 be the position of the test point; r be the position of the center of the neighbor particle; and $u = (u_x, u_y)$ be the unit vector defining the orientation of the particle. Since the position and orientation of the neighboring spherocylinder define a line in the plane of the surface, we can introduce a scalar parameter λ which, by the expression $r + \lambda u$, gives the position vector of any point along that line. Note that $-1/2 \leq \lambda \leq 1/2$ denotes points which lie on the DLS of the neighbor. The vector from the test point to any point along the line is just $R + \lambda u$ where $R = r - r_0$. The minimum distance from the test point to the line can be determined using the square of the magnitude of $R + \lambda u$:

$$(R + \lambda u) \cdot (R + \lambda u) = R^2 + 2\lambda R \cdot u + \lambda^2 \qquad (5)$$

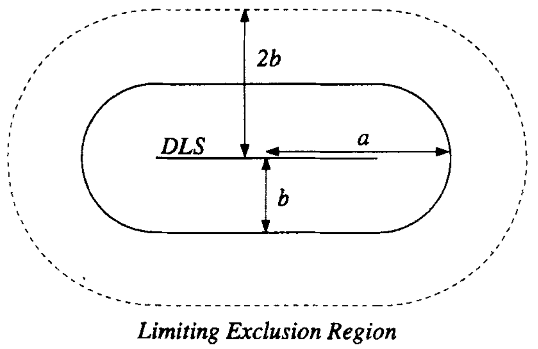

Figure 1 Diagram of a spherocylinder. All points on the particle are a distance b from the "defining line segment" (DLS). The outer region is excluded to the (DLS) of adsorbing spherocylinders.

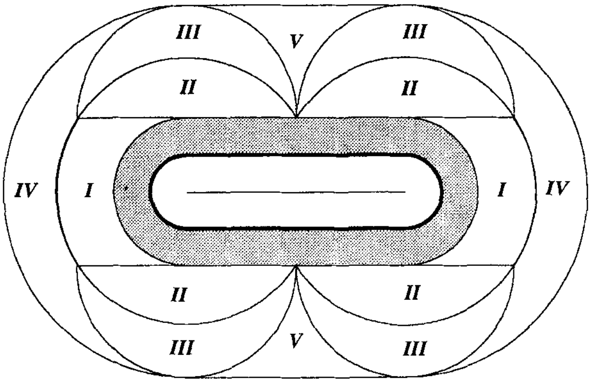

Figure 2 Illustration of the regions about an adsorbed spherocylinder which are used to determine the points of contact between the DLS of a particle placed at a neighboring test point and the surface of the adsorbed particle's limiting exclusion region (shaded).

Differentiating with respect to λ and setting equal to zero gives the value λ* which minimizes (the square of) the distance from the test point to the line:

$$\lambda^* = -R\cdot u \tag{6}$$

New define λ_s as the point along the DLS of the neighboring spherocylinder, i.e. $-l/2 \le \lambda_s \le l/2$, which is closest to the test point. Clearly, if $-l/2 \le \lambda^* \le l/2$, then $\lambda_s = \lambda^*$. If $\lambda^* \ge l/2$, then $\lambda_s = l/2$ and if $\lambda^* \le -l/2$, $\lambda_s = -l/2$. The square of the minimum distances to the points corresponding to λ^* and λ_s can be calculated by substituting them into the right hand side of Equation (5). Define these two distances as d^{*2} and d_s^2 respectively. If d_s^2 the minimum distance to the DLS of the neighbor, is less than $4b^2$, then the point lies within the limiting exclusion region and can be eliminated. If $d_s^2 > (a+b)^2$, then the neighboring spherocylinder could not possibly overlap with a new particle placed at the test point, while for $4b^2 < d_s^2 < (a+b)^2$, the allowed range of orientations is restricted.

Figure 2 shows a spherocylinder with five distinct neighboring regions. The particle in this figure is the emboldened inner spherocylinder and its limiting exclusion region is shaded. The outer boundary is the locus of points which are a distance $(a+b)$ from the DLS of the spherocylinder. The DLS of new particles placed at test points within this outer boundary and outside the exclusion region will, when rotated, contact the limiting exclusion surface at two points. How the points of contact are determined depends upon the regional location of the test point.

Test points which lie in region I are characterized by $d^* < 2b$ and $d_s < (l^2/4 + 4b^2)^{1/2}$. When the center of a new particle is placed in this region with respect to the spherocylinder in the figure and its DLS is rotated, the DLS will contact the semicircular cap of the limiting exclusion surface such that it forms a tangent at both points of contact. Points in region II have $d^* > 2b$ and their distance to the nearest end of the DLS in the figure is less than $(l^2/4 + 4b^2)^{1/2}$. Here, one end of a rotated DLS will contact the side of the exclusion surface and the other will form a tangent to the nearest end cap. Region III is a semicircle of radius $l/2$ whose center is the point on the exclusion surface where its end cap and side join. A DLS whose center lies inside this arc and outside region II will strike the side of the exclusion

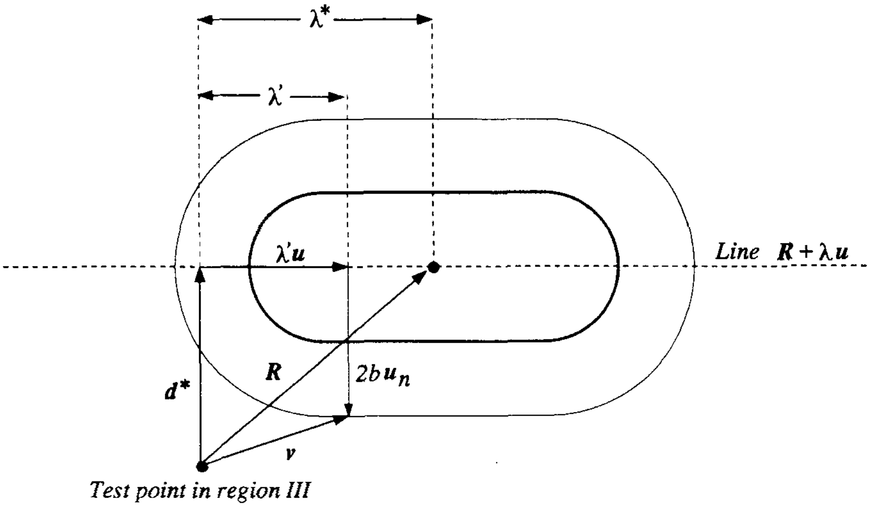

Figure 3 Illustration of the calculation of the point of contact between the DLS of a new particle placed at a test point in region III and the side of an adsorbed spherocylinder's limiting exclusion surface. Recall that λ^* gives the point along the line $R + \lambda u$ which is closest to the test point and λ' is the distance along the line from $R + \lambda^* u$ to the point which is nearest the point of contact between the DLS of a test particle and the "limiting" exclusion surface.

surface at one end and the cap at the other. Points in region III are too far away for a DLS to form a tangent with the end cap. In region IV, both ends of a rotated DLS will strike the cap and in region V both ends will contact the side. Since the area of each region depends upon the half-length of a spherocylinder's DLS, they become larger with α. The area of each of the regions shrinks to zero as $\alpha \rightarrow 1$.

Simple geometry can be used to determine the coordinates of the points of contact when the rotated DLS strikes the end caps. When contact is with the nearest side, however, the calculation is slightly more involved. In this case, the vector v from the test point to the point of contact will have magnitude $l/2$ and its tip will be a distance $2b$ away from some point along the DLS of the neighboring spherocylinder. The vector v can be written as follows

$$v = d^* + \lambda' u + 2bu_n \tag{7}$$

where λ' is the distance along the line $R + \lambda u$ from the tip of d^* to the point along the DLS which is a distance $2b$ from the tip of v, and u_n is the unit vector normal to the orientation u of the spherocylinder (see figure 3). It is necessary to solve for λ' such that $|v| = l/2$. Using the fact that d^* and u_n are both perpendicular to u

$$v^2 = \frac{l^2}{4} = d^{*2} + \lambda'^2 + 4b^2 + 4bd^* \cdot u_n \tag{8}$$

We choose u_n to point toward the test point so that $d^* \cdot u_n = -d^*$. Rearranging Equation (8) gives

$$\lambda' = \pm\ (l^2/4 - d^{*2} - 4b^2 + 4bd^*)^{1/2} \tag{9}$$

The two resulting vectors v can then be written

$$v = R + (\lambda^* \pm \lambda')u + 2bu_n \tag{10}$$

where we have used $d^* = R + \lambda^* u$. When the test point lies in regions II or III, only one end of the rotated DLS contacts the side. Thus we are interested in only one of the two vectors v corresponding to $\pm\lambda'$, in particular the one for which $|\lambda^* \pm \lambda'| \leq l/2$ since this gives a point along the DLS of the spherocylinder. When the test point is in region V, both $|\lambda^* + \lambda'|$ and $|\lambda^* - \lambda'|$ will be less than $l/2$.

2.3 *Bookkeeping*

Once the two vectors from the test point to the points of contact are determined, they are normalized and their components are stored in arrays which are indexed by a neighbor counter. The orientation range which is excluded to the test point by a neighbor is the angle between the two unit vectors. Excluded ranges from a set of two or more neighbors may overlap to form a larger range. When enough of them overlap to form a range which is greater than π, the test point can be eliminated.

For every neighbor n after the first, the program loops through all neighbors $i = 1, \ldots, n - 1$ to check for range overlap. A state array, *STATE*(I), is used to signify whether range i has been included in a larger range due to a previously determined overlap. A value of 0 for *STATE*(I) means that it has not and 1 means that it has and that it is not necessary to check for overlap with range i. This should become clear in the following development.

Upon entering the loop, a variable *RANGE* is set equal to the angle β_n between the two unit vectors v_{n1} and v_{n2} corresponding to neighbor n. A unit circle construction (see figure 4) is used to detect overlap between range n and range i. The state variable is first checked to determine if it is necessary to check for overlap. If the state variable has a value of zero, the program performs an overlap test as follows. The equation of the secant lines connecting the tips of each of the vector pairs (v_{n1}, v_{n2}) and (v_{i1}, v_{i2}) are determined as well as their mutual point of intersection. If the intersection point lies outside the unit cirle, the two ranges do not overlap and the program moves on to the next neighbor in the loop. If the intersection point lies inside the circle, the two ranges must overlap. The new, larger excluded range will be composed of one each of the two vector pairs. The angle β_{ni} between the vector from range n which is *not* part of the larger range and the vector from range i which *is* part of the larger range is the additional excluded angle to be added to the variable *RANGE*. To determine these two vectors, the program first calculates the point of intersection between secant line i and the line along vector v_{n1}. If the intersection occurs between the origin and the tip of v_{n1}, then v_{n1} is the vector *not* to be included in the new, larger range and is used to calculate the angle β_{ni}. Otherwise, secant line i must intersect the line along v_{n2} between the origin and its tip and v_{i1} is used. Next, the intersection of secant line n and the line along vector v_{i1} is calculated. If the intersection point lies inside the circle and along v_{i1}, then vector v_{i1} is the one which is *not* part of the larger range and v_{i2} is used to determine β_{ni}. Otherwise, v_{i1} is used. The angle β_{ni} is then computed and added to *RANGE*. In figure 4, the vectors v_{n2} and v_{i2} are used to determine β_{ni},

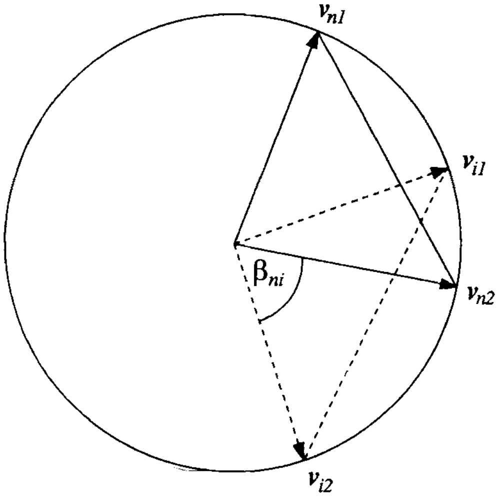

Figure 4 Unit circle construction used to detect overlap of excluded orientation ranges created by adsorbed particles in the neighborhood of a test point. The solid lines represent range n (most recently encountered neighbor) and the dooted lines range i (previously encountered neighbors) β_{ni} is the additional excluded orientation angle which results from the overlap of the two ranges.

while v_{n1} and v_{i2} form the new, larger excluded orientation range. If *RANGE* is less than π, the two vectors which form the larger range (v_{n1}, v_{i2}) replace both vector pairs i and n. A variable i^* is set equal to i to be used as the index of the vectors which contain the largest excluded range determined in the loop. Range i^* retains state 0 so that in the future, when new neighbors are found and the loop is entered again, the largest exclusion range found in the previous trip through the loop can be checked for overlap. Range n is given state 1; meaning that it has been included in range i^* and does not need to be checked for overlap when additional neighbors are found. Range n also becomes the larger range so that the largest excluded range found in the current loop is tested for overlap with the others. Recall that only range n is compared to ranges $i < n$. In addition, each time a subsequent overlap is detected and an even larger range is determined in the loop, the vectors corresponding to i^* are again replaced with those that form the new range and those ranges $i > i^*$ involved in overlap are given state 1. If, after all neighbors have been found, there is no excluded range greater than p, then it is still possible to place the center of a new spherocylinder at the test point and that point remains a part of the free area.

There are a number of ways to keep track of the free area. The choice depends on the system size, scale of discretization, and the memory capacity of the computer. The best method, which requires the most memory, is to declare a state array, say *POINT*(I, J), which signifies whether point (I, J) is part of the free area. Once all points have been tested, the free points can be numbered 1, . . ., m, where m is the number of free points, and their coordinates stored accordingly. The free area can then be sampled uniformly by selecting a point at random and using its coordinates for the center of a new particle. Another method, which requires less memory but eliminates less space, is to include all grid cells which contain one

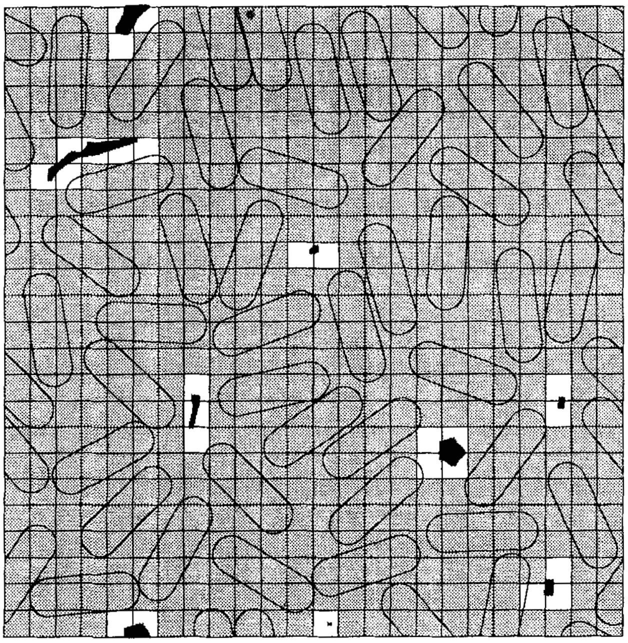

Figure 5 Configuration of spherocylinders with the grid construction shown. The shaded cells have been eliminated by the algorithm. The free points within the remaining cells are shown in black.

or more free points in the free area. These can be numbered and sampled uniformly in the same way. After randomly selecting a cell, a trial position for a new particle can be chosen at random therein.

Figure 5 displays a small configuration of spherocylinders for which the surface discretization density is $\rho_S \approx 1250$. The free grid cells which have been eliminated by the algorithm are shaded. The clusters of free points within the remaining grid cells are colored black.

3 APPLICATION TO RSA

Random sequential adsorption is a model for the irreversible, monolayer adsorption of large molecules from solution to a solid surface [7–20]. It is a simple model which incorporates only the blocking effect of previously adsorbed particles. The kinetics of the process for anisotropic particles are written in a "generalized Langmuir" form

$$\frac{d\theta}{dt} = k_a \int d\Omega f(\Omega) \Phi(\theta(t), \Omega) \tag{11}$$

where k_a is. the "bare" rate of adsorption, $f(\Omega)$ is the distribution of orientations, Ω, of adsorbing particles (assumed uniform, i.e. $f(\Omega) = 1/2\pi$), and $\Phi(\theta(\mathrm{t}), \Omega)$ represents the probability of adding a particle with orientation Ω to the surface when the coverage is θ. $\Phi(\theta(\mathrm{t}), \Omega)$ is essentially the fractional free area, i.e. the portion of surface which is accessible to the center of a new particle with orientation Ω. Since there is no exact solution to Equation (11) for two-dimensional RSA, studies of 2-D RSA have involved the extensive use of computer simulation.

RSA simulations have employed the following general algorithm:

- The position and orientation of a trial particle are selected randomly from uniform distributions.
- The trial particle is tested for overlap with neighboring, previously adsorbed particles.
- If overlap is found, the trial particle is discarded. Otherwise it is added to the surface.

Since the position of adsorbed particles remain fixed for the duration of the simulation, all properties are time dependent and RSA is a non-equilibrium process. The dimensionless time is defined as $A_{rel}N_A$, where N_A is the total number of attempts to add particles to a surface of area A. Multiple runs, with independent random number sequences, are required to reduce statistical errors in the measured quantities. Initially, the probability of placing a new particle is high and the surface fills rapidly. As the surface coverage increases, the adsorption probability decreases due to the exclusion effects from previously adsorbed particles until, eventually, it is impossible to place another particle without overlap. This condition, associated with a zero probability of adsorption (zero free area) is known as the jamming limit.

Kinetic studies of RSA have divided the process into "regimes" for which exact results can be derived. In the limit of low coverage, the time evolution of the coverage can be expressed as a virial-like series, whose coefficients are related to the equilibrium virial coefficients. At high density, the kinetics are fundamentally different. In the approach to saturation, the free surface becomes broken into very small, isolated regions (known as target areas) which can accept, at most, the center of one additional particle. For spherical particles, the rate at which these targets disappear, or equivalently, the rate at which the density approaches its saturation value, has been shown to follow a power law [7, 8]

$$\theta(\infty) - \theta(t) \sim t^{-p} \tag{12}$$

where $p = 1/D$ and D is the dimensionality of the space ($D = 2$ for disks adsorbing on a surface and $D = 3$ for the addition of spheres to a volume). Simulation studies confirmed this result for both disks [9] and spheres [12]. It was originally conjectured that Equation (12) would apply to any system in D dimensions. Talbot *et al* [14] tested this result for anisotropic particles in their 2-D RSA simulations of ellipses. Their results showed that the kinetics still follow a power law, but that the power law exponent was not 1/2 as expected. In fact, the exponent was significantly less than 1/2 and as low as 1/4 for ellipses with an axial ratio of 5:1. They also proposed a plausible explanation. At very high coverage, a non-spherical particle can fill a target only if its orientation lies within a restricted range. This additional degree of freedom results in an exponent of 1/3 instead of 1/2 (see references [14, 15, 19]). However, because of the difficulty in obtaining reliable data in simulations of anisotropic particles at high coverage, their simulation results were inconclusive. Subsequent RSA studies involving unoriented squares [15], ellipses [16], and rectangles [17, 18] all appeared to confirm the $t^{-1/3}$ power law. The errors in the estimates were typically large, however, and the studies were generally confined to relatively short particles ($1 < \alpha < 5$). Any simulations involving longer particles were, by necessity, very short due to the added computational effort required for long particles. In addition, the effect of particle shape and elongation on the exponent remained unclear. We [19] then decided to employ the method outlined here for spherocylinders to perform the most careful, extensive, and accurate simulations to date. We were able to simulate very large systems ($A_{\text{rel}} = 0.0002$),make a large number of addition attempts per independent run (10^8), and simulate systems involving very elongated particles ($\alpha = 15$). The increase in efficiency was as high as two orders of magnitude over that of the conventional RSA algorithm.

The simulations were performed as follows. The surface was discretized such that the side of each grid cell was composed of 25 points. Each grid cell then contained 625 discrete points. The width of the spherocylinder for all aspect ratios was fixed at unity, i.e. $b = 1/2$. Since the diagonal of each grid was slightly less than $2b$, the area of each was ≈ 0.5. Thus, the number of points per unit area of surface was $\rho_s \approx 1250$.

The first 10^6 attempts were made in the usual way, i.e. by selecting positions uniformly over the entire surface. Note that even at low coverage, positions were constrained to the discrete points. Particle positions and orientations were then saved in new arrays and every grid cell was then tested for its ability to accept a new particle center. Because of the large system size, the memory requirement was too great to keep track of individual free points. The free cells were numbered and the fraction of free cells, f, was calculated. f is used to scale the number of attempts and thus the adsorption probability for the remainder of the run. Each attempt in the free area would then correspond to $F = 1/f$ attempts made uniformly over the whole surface.

After the first 10^6 attempts, F was anywhere from 30–70. The simulation then proceeded by selecting trial positions only in free cells and counting each as $F = 1/f$ attempts. *Since the addition of each new spherocylinder necessarily decreases the fraction of free surface, the free cells in the vicinity of each newly added particle were tested thus increasing F and accelerating the computing efficiency.* By the end of a run, F was sometimes as high as 600: a dramatic improvement in the adsorption per CPU time ratio. Since the routine to determine unavailable regions of surface is complicated and time consuming, the configuration saved after the first 10^6 attempts was then recalled to serve as the initial state for a new run. 50 runs were performed in this way before a new configuration was generated from a clean

Table 1 Power law exponents p_α and saturation coverage $\theta_\alpha(\infty)$ characterizing the asymptotic regime of spherocylinders from reference [19]. The results were obtained from the simulation data by least square fit procedures based on Equation (12).

	spherocylinders		
α	p_α	$\theta_\alpha(\infty)$	*Runs*
1.25	0.421 ± 0.006	0.569 ± 0.001	750
1.5	0.334 ± 0.006	0.580 ± 0.001	750
2	0.338 ± 0.006	0.581 ± 0.001	650
3	0.335 ± 0.006	0.569 ± 0.001	650
4	0.337 ± 0.006	0.554 ± 0.001	650
6	0.337 ± 0.006	0.524 ± 0.001	700
10	0.330 ± 0.006	0.482 ± 0.001	750
15	0.334 ± 0.007	0.445 ± 0.001	500

Table 2 Same as table 1 for ellipses.

		ellipses	
α	p_α	$\theta_\alpha(\infty)$	*Runs*
1.25	0.43 ± 0.05	0.568 ± 0.001	200
1.5	0.33 ± 0.03	0.580 ± 0.001	200
2	0.32 ± 0.03	0.583 ± 0.001	200
3	0.32 ± 0.03	0.569 ± 0.001	200
4	0.32 ± 0.03	0.552 ± 0.001	200
6	0.35 ± 0.03	0.536 ± 0.001	200

Table 3 Same as table 1 for rectangles.

		rectangles	
α	p_α	$\theta_\alpha(\infty)$	*Runs*
1	0.34 ± 0.01	0.530 ± 0.001	35
1.5	0.34 ± 0.02	0.552 ± 0.002	40
1.618	0.32 ± 0.02	0.553 ± 0.002	40
2	0.34 ± 0.04	0.548 ± 0.002	35
5	0.33 ± 0.04	0.510 ± 0.005	35
15	0.32 ± 0.04	0.483 ± 0.005	35

surface. This should present no problem in determining the form of the kinetics and estimating the jamming coverage as long as the system is large enough so as to contain a representative sample of targets.

Tables 1, 2, and 3 display the estimates of the power law exponent, p_α, and jamming limit, $\theta_\alpha(\infty)$, for spherocylinders, ellipses, and rectangles, respectively, from reference [19]. The conventional method was used for ellipses and rectangles. For ellipses, 200 independent runs of 10^7 attempts were performed but the system size was 10 times smaller ($A_{\text{rel}} = 0.002$). The simulations of rectangles used the same system size as was used for spherocylinders ($A_{\text{rel}} = 0.0002$) but the number of independent runs was significantly lower. Note how much more precise are the estimates of the power law exponents for spherocylinders.

In order to compare the computational efficiencies of the two methods, new simulations were run for spherocylinders of axial ratio $\alpha = 2$. All simulations were run on a Sun Sparc station. For these, the system size was $A_{\text{rel}} = 0.002$, and a total of 50 independent runs of 10^7 addition attempts each were performed using both methods. In the runs using the new method, the first 10^5 addition attempts were made using the conventional algorithm, and 5 runs were performed from that point before starting over from a clean surface. This was repeated 10 times to complete the 50 independent runs. Six simulations using the new algorithm were run, each with a different scale of discretization, ρ_s. Table 4 lists the results. The power law exponent and particularly the jamming limit coverage are insensitive to the value of ρ_s approximately $\rho_s = 200$ (the estimates all have large errors because these were short runs). Differences begin to appear for coarser discretizations ($\rho_s < 200$). The results clearly indicate that the mesh spacing employed in reference [19] ($\rho_s \approx 1250$) adequately represents a continuous surface. Table 4 also shows that even with a very fine mesh of discrete points, the new algorithm finished in 9% of the time required for the old method.

Table 4 Comparison of calculated power law exponents p_α and saturation coverages $\theta_\alpha(\infty)$ as well as required CPU time for various discretization densities ρ_S using the new method with that for the old method. The new method required significantly less CPU time, even with a very fine mesh of discrete points. The results are the same within statistical uncertainties for $\rho_S \geq 200$.

method	ρ_S	p_α	$\theta_\alpha(\infty)$	*CPU time (hours)*
new	50	0.08±0.07	0.57±0.10	0.13
	200	0.31±0.07	0.58±0.10	0.23
	800	0.34±0.07	0.58±0.10	0.42
	1800	0.28±0.07	0.58±0.10	0.74
	3200	0.37±0.07	0.58±0.10	1.15
	5000	0.31±0.07	0.58±0.10	1.70
old	∞	0.37±0.07	0.58±0.10	19.67

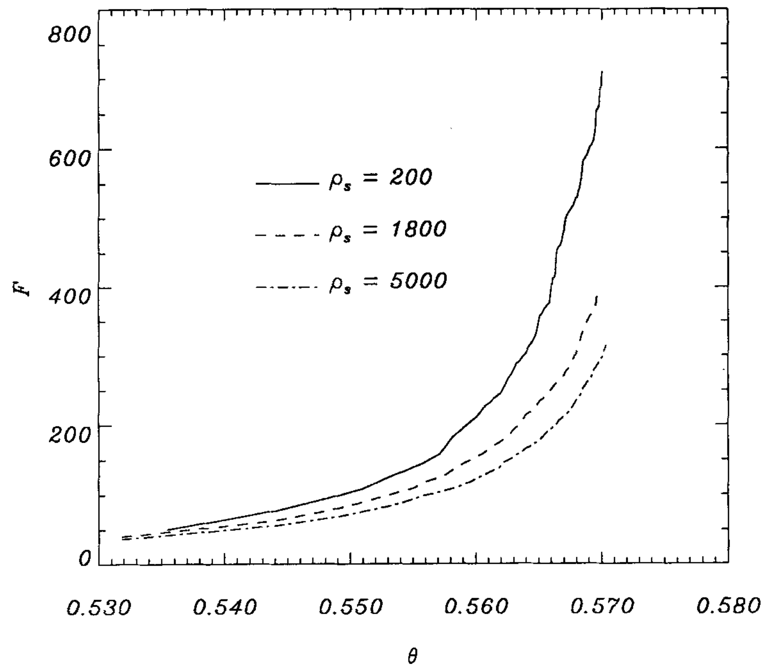

Figure 6 Averaged scale-up factor F versus coverage θ from the simulations. This plot illustrates the "acceleration" of the efficiency of the new method at high coverage.

The real power of the new method becomes apparent when the number of addition attempts per independent run is increased. With the conventional method, increasing the number of attempts per run by a factor of ten roughly increases the required CPU time by the same amount. Since the new method counts attempts by the scale-up factor F, and F is increased with each successful placement of a particle, increasing the number of addition attempts by a factor of ten would require only a small amount of additional CPU time. Figure 6 shows F, averaged over the 50 independent runs, versus coverage from the simulations using ρ_s = 200, 1800, and

5000. The scale-up is obviously greater for smaller discretization densities. Note how the efficiency of the method "accelerates" as the coverage increases.

4 NOTE ADDED IN PROOF

The procedure for determining overlap of excluded orientation ranges outlined in section 2.3 may be simplified by viewing the process as a 1-dimensional addition process with periodic boundary conditions. One simply defines the center angle of each range and its angular diameter. Overlap occurs when the angular distance between range centers is less than half the sum of their diameters. Keeping track of and combining ranges is done in the same manner as prescribed in section 2.3.

It also occurred to us that the efficiency may be improved upon even further if, instead of leaving available entire grid cells which contain open points, one stores the minimum and maximum x and y coordinates of the open points in a cell and uses those to encase the points in rectangle. The distance between two discrete points would be added to the maxima and subtracted from the minima so long as the resulting rectangle does not penetrate neighboring cells. The width, height, and coordinates of the bottom left-hand corner of each rectangle would be stored and positions generated within them in the same way as previously discussed.

5 DISCUSSION

In this paper, we have shown how the efficiency of simulations of spherocylinders which involve insertion of particles into bulk configurations can be significantly enhanced by identifying those regions of surface where insertion is not possible. The same method can clearly be applied to anisotropic objects of different geometries, although the overlap criteria may vary significantly from case to case. Spherocylinders are particularly convenient in this regard. Extension of the method to three dimensions is a considerably more challenging problem.

Acknowledgement

We thank NSF (CTS-9011240) for financial support.

References

[1] H. Reiss and A.D. Hammerich, "Hard spheres: scaled particle theory and exact relations on the existence and structure of the fluid/solid phase transition", *J. Phys. Chem.* **90**, 6252 (1986).

[2] W.G. Hoover, N.E. Hoover, and K. Hanson, "Exact hard disk free volumes", *J. Chem. Phys.* **70**, 1873 (1979).

[3] J. Zolweg, "Chemical potential of a slute at infinite dilution in a hard disk fluid by monte carlo simulation", *J. Chem. Phys.* **72**, 6712 (1980).

[4] B. Widom, "Some topics in the theory of fluids", *J. Chem. Phys.* **39**, 2808 (1963).

[5] J.L. Jackson and L.S. Klein, "Potential distribution method in equilibrium statistical mechanics", *The Physics of Fluids*, **7**, 228 (1963).

[6] G.L. Deitrick, L.E. Scriven, and H.T. Davis, "Efficient molecular simulation of chemical potentials", *J. Chem. Phys.* **90**, 2370 (1989).

[7] Y. Pomeau, "Some asymptotic estimates in the random parking problem", *J. Phys.* **A13**, L193 (1980).

[8] R.H. Swendsen, "Dynamics of random sequential adsorption", *Phys. Rev.* **A24**, 504 (1981).

[9] E.L. Hinrichsen, J. Feder, and T. Jøssang, "Geometry of random sequential adsorption", *J. Stat. Phys.* **44**, 793 (1986).

[10] P. Schaaf and J. Talbot, "Kinetics of random sequential adsorption", *Phys. Rev. Lett.* **62**, 175 (1989).

[11] P. Schaaf and J. Talbot, "Surface exclusion effects in adsorption processes", *J. Chem. Phys.* **91**, 4401 (1989).

[12] J. Talbot, P. Schaaf, and G. Tarjus, "Random sequential addition of hard spheres", *Mol. Phys.* **72**, 1397 (1991).

[13] G. Tarjus, P. Schaaf, and J. Talbot, "Random sequential addition: a distribution function approach", *J. Stat. Phys.* **63**, 167 (1991).

[14] J. Talbot, G. Tarjus, and P. Schaaf, "Unexpected asymptotic behavior in random sequential adsorption of nonspherical particles", *Phys. Rev.* **A40**, 4808 (1989).

[15] P. Viot and G. Tarjus, "Random sequential addition of unoriented squares: breakdown of Swendsen's conjecture", *Europhys. Lett.* **13**, 295 (1990).

[16] J.D. Sherwood, "Random sequential adsorption of lines and ellipses", *J. Phys.* **A23**, 2827 (1990).

[17] R.D. Vigil and R.M. Ziff, "Random sequential adsorption of unoriented rectangles onto a plane", *J. Chem. Phys.* **91**, 2599 (1989).

[18] R.D. Vigil and R.M. Ziff, "Kinetics of random sequential adsorption of rectangles and line segments", *J. Chem. Phys.* **93**, 8270 (1990).

[19] P. Viot, G. Tarjus, S. Ricci, and J. Talbot, "Random sequential adsorption of anisotropic particles. I. Jamming limit and asymptotic behavior", *J. Chem. Phys.* **97**, 5212 (1992).

[20] S. Ricci, and J. Talbot, G. Tarjus, and P. Viot, "Random sequential adsorption of anisotropic particles. II. Low coverage kinetics", *J. Chem. Phys.* **97**, 5219 (1992).