

球充填シミュレーション高速アルゴリズム

山田修司^{†1} 菅野仁子^{†2} 宮内美樹^{†3}

重力のように単一方向の力ではなく、分子間力のような粒子間引力が働いている環境における球充填シミュレーションのための高速アルゴリズムを考案した。これにより、 10^7 個を超える球の充填シミュレーションを、パーソナルコンピュータを用いて 30 分未満で行うことが可能となる。

Fast Algorithm for Sphere Packing Simulation

SHUJI YAMADA,^{†1} JINKO KANNO^{†2} and MIKI MIYAUCHI ^{†3}

We devised high-speed algorithm for sphere packing simulation in the environment where the gravitation between the particle such as the intermolecular force acted on not the power of the single direction like gravity. Using this algorithm, we can perform a packing simulation of spheres more than 10^7 for less than 30 minutes using a personal computer.

1. Introduction

3次元空間内における球の充填についての研究は数多くある。もっとも古いものの一つは、Scott, Kilgour による [S] であり、20,000 個の鋼球（直径 $\frac{1}{8}$ インチ）を円筒形の容器内に充填し、重力の環境下で容器全体を揺さぶり、その充填密度 0.6366 を求めたものである。このような充填方法で得られる充填は、密充填 (Closed Packing) と呼ばれる。これに対して、小さく軽い粒子を緩やかに充填することで得られるものを、疎充填 (Loose Packing) と呼ぶ。これは、間隙へのナノ粒子の充填など、近年、応用が高まっている研究対象である [D],[K1],[K2]。

^{†1} 京都産業大学

^{†2} ルイジアナ州立工科大学

^{†3} NTT コミュニケーション科学基礎研究所

疎充填は、重力と粒子間の引力との大小関係に依存してその充填の様相が異なってくる。重力が大きいと粒子は最初に接触した位置から外れて、形状的に安定な特殊な位置に留まろうとするし、粒子間引力が大きいと最初に接触した位置、すなわち形状的な安定とは無関係なランダムな位置に留まろうとする。形状的に安定な位置とは、容器壁面や既充填球が形作る窪みのことであり、そこに次充填球が置かれる場合、それは壁面や既充填球と 3 点以上の接触点をもつ場所である。形状的に安定な場所はそうでない場所よりも、他の充填球との接触数が多い場所であるから、そのような場所に球を充填することは充填密度を高める効果を生む。その結果、重力が粒子間引力に比較して大きいと充填密度は大きくなり、反対に粒子間引力が大きいと充填密度は小さくなる。

球の充填については、コンピュータシミュレーションを用いた先行研究も、数多く行われている。それらのシミュレーションモデルの多くは、充填球を容器上部のランダムな位置から落とし込み、既充填球にぶつかった場合、形状的に安定する位置まで転がす、というドロップ・アンド・ロール方式 (drop and roll method) と呼ばれている方法が用いられている [L], [P], [V], [Y]。この方法は、重力のみが作用する環境下での充填をシミュレートするものである。また、これらの研究ではその計算の複雑さから充填される球の個数が制限されるため、数百から数十万という個数の充填が多い。さらに、シミュレーションで得られた充填の様相を解析しようとしても、粒子が置かれる場所の確率分布さえ容易に求めることができないので、その数理的な解析には困難が伴う。

当研究では、分子間力が作用する環境での球充填をシミュレートする、より単純な充填原理をもつモデルを導入した。これにより、より多数の球を高速に充填することができる。さらに、その原理の単純さと合理性とから、シミュレーションで得られた充填結果を数理的に解析することも容易となる。また、このモデルは粒子間引力が充填に及ぼす効果を表すパラメタを含んでおり、それにより、いろいろな環境下での疎球充填を再現することが可能となる。

本稿では、そのモデルに従った充填シミュレーションのための高速アルゴリズムを紹介する。このアルゴリズムを用いることで、数千万個の球を充填するシミュレーションをパーソナルコンピュータ上で 30 分ほどの時間内に行うことが可能となる。

2. 球充填シミュレーションモデル

ここでは、球充填の計算実験で用いたシミュレーションモデルを解説する。

容器内に既に充填された球がいくつかあるとき、次に充填される半径 r の球の中心は、容

器外部および既充填球からの距離が r である位置に置かれる。他の多くの論文で用いられているドロップ・アンド・ロール方式では、容器上面内の 1 点を一様な確率分布でランダムに選択し、その地点から充填球を容器内に落とし込み、形状的に安定な位置まで転がすことにより、充填位置を決める、という方法をとっている。これに対し、本論文で用いたモデルでは、選択可能な位置の確率分布 p を直接に決めておき、その分布に応じてランダムに選択するという方法をとる。その確率分布は引力による安定と形状的な安定との関係を表す充填環境パラメタ (environment parameter) $0 \leq \mu \leq 2$ を含んでおり、その値が小さいときは粒子間引力による安定が優位であり、大きいときは形状的安定が優位である確率分布となる。

既充填球および容器壁面からの距離が r 以上である点の全体を R とする。 ∂R が、次充填球の中心が置かれる可能性のある点全体であるので、**充填前面**と呼ぶことにする。 μ を $0 \leq \mu \leq 2$ である実数とし、 $A \subset \partial R$ に対して、 A および R からの距離が両方とも μr である点の全体 $\text{supp}_\mu(A) = \partial N_{\mu r}(A) \cap \partial N_{\mu r}(R)$ を A の μ -支持集合 (support set) と呼ぶ。ここで、 $N_{\mu r}$ は μr -近傍を表す。 $\text{supp}_\mu(A)$ が可測集合であるとき、 A の測度を $m_\mu(A) = \text{area}(\text{supp}_\mu(A))$ と定める。ここで、 area は $\partial N_{\mu r}(R)$ 上での面積測度である。そして、 A 上に次充填球の中心が置かれる確率を

$$p(A) = \frac{m_\mu(A)}{m_\mu(\partial R)} \quad (1)$$

と定める。

3. データ構造とアルゴリズム

ここでは、上記のモデルに従ったシミュレーションを行うためにコンピュータプログラムに実装したデータ構造とアルゴリズムとを解説する。

このシミュレーションを高速に行うためには、「確率密度に従い充填前面上の 1 点をランダムに選択する」ということを高速に行う必要がある。また、球が充填されるたびに充填前面は変化するので、充填前面上の確率分布を表すデータ構造は、その内容を高速に書換ができるものでなければならない。

充填前面は、0,1,2 次元のピットと呼ばれるものに分割して管理される。

充填前面上の 1 点に次充填球を置くと、既充填球および容器壁面 (平面と仮定する) のうち 1 個から 3 個のものに接する。既充填球および容器壁面のうち特定の 1 個のものに接する点全体は、球面あるいは平面状に広がりを持つので、2 次元ピットと呼ぶ。また、特定

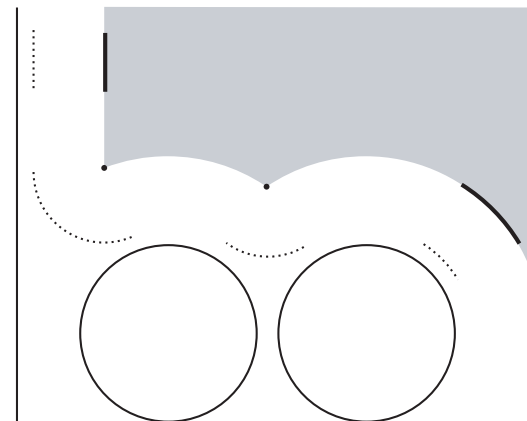


図 1 容器内に既に 2 個の球が充填されている状態の充填前面および支持集合の 2 次元模式図。グレー部分が R であり、太線部分を A とするとき $\text{supp}_\mu(A)$ が点線で表されている。ただし、 $\mu = 0.8$ としている。

の 2 個のものに接する点全体は円周の一部となるので 1 次元ピットと呼び、特定の 3 個のものに接する点全体は 1 点集合であるので 0 次元ピットと呼ぶ。(図 2 参照)

充填前面は、これらの有限個のピットから構成される。その個数を n とし、その全体を $\{P_1, P_2, \dots, P_n\}$ とする。次充填球の中心位置の選択は、すべてのピットの中からその一つ P_i をその支持集合 $\text{supp}_\mu(P_i)$ の面積に比例する確率で選択した後、 P_i の上から一様な確率で 1 点を選択することで行われる。そこで、ピットの情報を保存するためのデータ構造と、その中の一つをそこに書かれた確率に応じて高速で選び出すアルゴリズムが必要となる。また、ピット情報の追加削除および変更も高速で行える必要がある。ここで紹介するのは、選択と追加削除および変更が、ともに $\log n$ のオーダーで行えるものである。

ピット情報を保存するために用いたデータ構造には、配列を用いたヒープ構造を応用したものをを用いる。配列を用いたヒープ構造とは、配列 $a[1], a[2], \dots, a[n]$ 上に二分木の構造を実現するものである。 $a[1]$ が根であり、ノード $a[i]$ の子ノードは $a[2i]$ および $a[2i+1]$ であり、ノード $a[i]$ の親ノードは $a[i/2]$ である。

ここでは、各ノード $a[i]$ はピット P_i に対応し、 $a[i]$ には自身を含めたすべての子孫ノードに対応するピットの支持面積の総和が格納されているものとする。したがって、ピット P_i の支持面積は $a[i] - a[2i] - a[2i+1]$ で与えられる。ただし、 $i > n$ の場合は $a[i] = 0$ とみなす。特に、根である $a[1]$ には、全てのピットの支持面積の総和が格納されている。

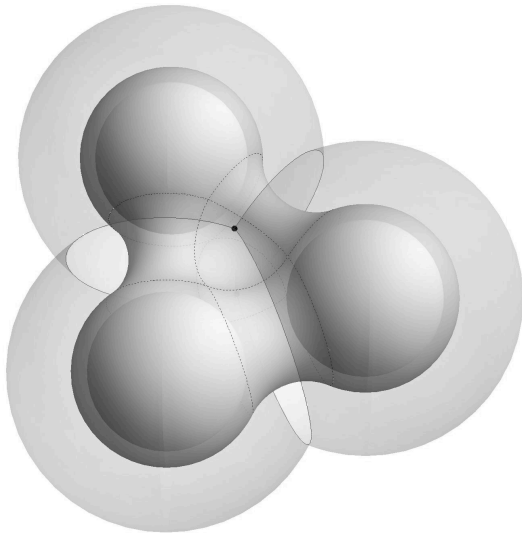


図2 既充填球が3個（一番内側にある小球）ある場合の、充填前面を構成するピットおよびその支持集合。図の中央付近にある黒点が0次元ピット（反対側にもう1点ある）で、そこから出ている3本の円弧が1次元ピット。1次元ピット間に張られた大きな球面（充填球と同心で半径が2倍の球面）の一部が2次元ピット。これらのピットを併せた物が充填前面である。充填球と充填前面との間にある曲面が支持集合であり、対応するピットごとに破線で区切って表示してある。0次元ピットの支持集合は、0次元ピットを中心とする半径 μr の球面の一部。1次元ピットの支持集合は、1次元ピットを管の中心弧とする管半径 μr のトーラスの一部。2次元ピットの支持集合は、充填球と同心で半径が $(2 - \mu)r$ の球面の一部。ただし、 $\mu = 0.8$ としている。

ピットを選択するためには、0以上 $a[1]$ 以下の実数から一様な確率で一つの実数 r を選ぶことから始める。そして、次のように再帰的に定義された関数`selectPit`を、`selectPit(1, r)`として呼び出すことで、支持面積に応じた確率分布に従って選択されたピットの番号が返される。この関数は、引数の r の値（ $0 \leq r \leq a[i]$ の範囲にある）に応じて、再帰呼び出しを行う（図3）。

関数 `selectPit(i, r)` は、ノード $a[i]$ の（自身を含めた）子孫の中から、その

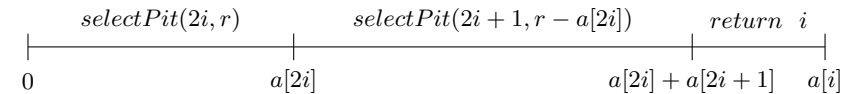


図3 $0 \leq r \leq a[i]$ の値に応じた `selectPit(i, r)` の再帰呼び出し

値に比例した確率分布でランダムにインデックスを1つ選択し、それを返す。

```
function selectPit(i, r) :=
    if (r < a[2*i]) then return selectPit(2*i, r);
    else if (r < a[2*i] + a[2*i+1]) then return selectPit(2*i+1, r - a[2*i]);
    else return i;
```

ピット P_i の支持面積が b だけ増加して変更になった場合には、次の関数を `addPitArea(i, b)` として呼び出す。以降、 $i/2$ は除算 $i/2$ の商（整数部分）とする。

関数 `addPitArea(i, b)` は、ノード $a[i]$ の（自身を含めた）すべての祖先の値を b だけ増やす。

```
function addPitArea(i, b) :=
    if (i > 0) then
        a[i] := a[i] + b;
        addPitArea(i/2, b);
```

ノードへのピットの割当状況を管理するために、 $m[1], m[2], \dots, m[n]$ という配列を用いる。 i 番目のノードを含めてその子孫に割り当てられたピットの総数を $m[i]$ に格納することとする。 i 番目のノードがピットに割り当てられているかいないかは、 $m[j] - m[2*j] - m[2*j+1]$ の値が1であるか0であるかで判断できる。なお、プログラムとして実装する際には、 $a[i]$ と $m[i]$ はピットの実体へのポインタとともに、構造体として一つにまとめられるべきものである。

支持面積 b をもつピット P_i の追加あるいは削除は、次の関数を `addPitNum(i, 1)` あるいは `addPitNum(i, -1)` として呼び出し、その後 `addPitArea(i, b)` あるいは `addPitArea(i, -b)` を行うことでなされる。

関数 `addPitNum(i, f)` は、ノード $m[i]$ の（自身を含めた）すべての祖先の値を

f だけ増やす。

```
function addPitNum(i, f) :=  
  if (i > 0) then  
    m[i] := m[i] + f;  
    addPitNum(i/2, f);
```

また、未使用の配列要素のインデックスは次の関数を `vacantIndex(1)` として呼び出すことで得られる。支持面積 b の新しいピットを作成するには、 $i := \text{vacantIndex}(1)$ とした後、`addPitNum(i, 1)` および `addPitArea(i, b)` を実行すればよい。

関数 `vacantIndex(i)` は、ノード $m[i]$ の（自身を含めた）子孫の中から、ピットに割り当てられていないノード ($m[j] - m[2*j] - m[2*j+1] == 0$ であるもの) のインデックスを返す。

```
function vacantIndex(i) :=  
  if (m[i] - m[2*i] - m[2*i+1] == 0) then return i;  
  else if (m[2*i] <= m[2*i+1]) then return vacantIndex(2*i);  
  else return vacantIndex(2*i+1);
```

以上のどの関数も、二分ヒープの根にあるいは葉に向かう再帰呼び出しを用いているので、計算量は $\log n$ のオーダーであることは容易に分かる。

以上で、ピットを支持面積に従ってランダムに選択する方法について述べた。次は、選択されたピットが 1 次元あるいは 2 次元ピットである場合に、そのピット中から 1 点を一様な分布で選ばなければならない。しかしながら、2 次元ピットの場合は、球面の一部分であるピットの境界が複雑な形をしているものがあり、その場合、支持面積の計算が複雑で、さらに一様な分布で 1 点をその中から選ぶことが困難となる。そのため、ピットの実際の形や支持面積を無視して、ピットの支持集合が乗っている球面（支持球面）の面積をそのまま支持面積とする。そのピットが選ばれた後にそのピット上から 1 点を選択するときに、その支持球面上から一様な確率で 1 点を選択し、それが他の既充填球と干渉しないときに限りその選択を実際に行い、もしも干渉するならば、そのピットの選択をキャンセルして、再びピット選択の作業からやり直すこととする。

これにより、若干の計算のオーバーヘッドが生じるが、それは高々定数倍以下の変化であ

り、計算量が $\log n$ のオーダーであることに変わりはなく、支持集合が複雑な形をしていても、一様な分布での 1 点の選択が容易となる。また、他の既充填球との干渉を調べるために必要な時間は、空間をメッシュに区切り、小さな直方体ごとに充填球を管理することにより、定数のオーダーとすることが可能である。

4. 結 論

ここで紹介したアルゴリズムは、充填前面にある充填球の個数が n であるとするとき、 m 個の球面を $O(m \log n)$ のオーダーの時間で充填することができる。これを用いることにより、 10^7 個を超すような非常に多数の球の充填シミュレーション実験を行うことができる。その結果から、少数の充填実験の時には見られなかった興味深い現象を見ることが可能となる。たとえば、充填過程において充填前面の形状が変化することによる充填密度のゆらぎ現象などである。

現実の充填においては、球を充填するスペースが残っていても、その回りが既に充填された球で囲まれて充填されない空間が残る場合がある。当シミュレーションモデルにおいては、そのような状況は考慮していない。それを考慮するためには、充填前面のトポロジーを調べる必要が生じるため非常に複雑な計算が必要となる。計算速度を優先するためと数学的な解析を容易にするために、今回はその状況は無視しているが、それを考慮に入れることは今後の課題である。

参 考 文 献

- [D] Dong, K. J., Yang, R. Y., Zou, R. P., Yu, A. B., “Role of Interparticle Forces in the Formation of Random Loose Packing,” Physical Review Letters, 14 April 2006.
- [K1] Kanno, J., Richardson, N., Phillips, J., Kupwade-Patil, K., Mainardi, D.S. and Cardenas, H.E., “Modeling and simulation of electromutagenic processes for multi-scale modification of concrete,” Journal of Systemics, Cybernetics and Informatics, 7(2): 69-74, (2009).
- [K2] Kupwade-Patil, K., “Chloride and sulfate based corrosion mitigation in reinforced concrete via electrokinetic nanoparticle treatment,” Ph.D. dissertation, Louisiana Tech University, Ruston, Louisiana (May 2010).
- [L] Guangli Liu, Karsten E. Thompson, “Influence of computational domain boundaries on internal structure in low-porosity sphere packings,” Powder Technology 113, 2000, 185-196.
- [P] Phillips, J., “A Modeling and Simulation Framework for Electrokinetic Nanopar-

title Treatment,” Ph.D. dissertation, Louisiana Tech University, Ruston, Louisiana (May 2011).

- [S] Scott, G. D., Kilgour, D. M., “The density of random close packing of spheres,” J. Phys. D: Appl. Phys. 2 863. (1969).
- [V] Visscher, W. M., Bolsterli, M., “Random packing of equal and unequal spheres in two and three dimensions,” Nature 239, 504 - 507, (27 October 1972).
- [Y] Shi, Y., Zhang, Y., “Simulation of random packing of spherical particles with different size distributions,” Applied Physics A: Materials Science & Processing, Volume 92, Number 3, 621-626 (2008).