

DEM を用いた粉粒体シミュレーションの GPU 上への実装

高田 信一郎† 床井 浩平‡

† 和歌山大学 大学院 システム工学研究科

‡ 和歌山大学 システム工学部

E-mail: {s075035, tokoi}@sys.wakayama-u.ac.jp

粉粒体と呼ばれる物質のシミュレーションを行う手法の1つとして、DEM (Distinct Element Method) というものがある。DEM は粉粒体に含まれる粒子の振る舞いを、その粒子に作用する力を元に計算する手法である。本研究では、高速な粉粒体シミュレーションを実現するために、DEM を GPU 上で実装する手法を提案する。GPU で汎用計算を行う場合、入出力データをテクスチャとして扱う必要がある。そこで、DEM を実行する上で必要となる粒子の座標や速度といったデータをテクスチャで用意する。それらのデータを利用して、GPU 上で粒子の振る舞いを計算する。

Powder simulation using DEM on GPUs

Shin-ichiro Takata† and Kohe Tokoi‡

† Graduate School of Systems Engineering, Wakayama University

‡ Faculty of Systems Engineering, Wakayama University

E-mail: {s075035, tokoi}@sys.wakayama-u.ac.jp

There is a method which is named DEM(Distinct Element Method) for a powder simulation. DEM is a method for calculating the movements of particles included a powder model using the forces. In this paper, we present DEM on GPUs to actualize a faster simulation. In case we solve a general purpose computation on GPUs, we have to use some input and output data converted into the textures. Therefore, we have to make a position data and a velocity data and so on into them. We can calculate the movements of particles on GPUs using these texture data.

1. はじめに

近年、砂などの粉粒体の表現に関する研究が精力的に行われている。粉粒体は、非常に多くの粒子を含む物質という特徴を持っているため、その振る舞いの再現には処理時間の高速化、動きのリアリティの確保などの課題が挙げられる。このような課題に対するいくつかのアプローチとして、Yonging らによる粉粒体を流体としてシミュレートした手法[1]や、Onoue らによる堆積形状を、2次元配列に高さの値を格納したデータ構造を用いて再現した手法[2]などが提案されている。前者は、粒子の大きさが極小化された場合、その振る舞いが流体に近づくという性質を利用しており、後者は、粉粒体の堆積形状を2.5次元で表現し、どちらも実際の粒子の挙動を計算しなくてすむようにしている。しかし、

砂粒のように含まれる粒子がある程度の体積を持っている場合、粉粒体を粒子の集合として考え、個々の粒子の振る舞いを計算する必要がある。そこで本研究では、DEM という粉体の数値計算に使用される手法を用いて、2次元での粉粒体シミュレーションを行う。DEM については、次章で詳しく説明する。

DEM は個々の粒子に作用する力を用いてその粒子の振る舞いを計算する手法であり、極めて並列的な処理である。この処理を並列計算に特化した構造を持つ GPU (Graphics Processing Unit) 上に実装することで、処理の高速化を図る。

GPU は本来、3次元画像表示をサポートするハードウェアであるが、その演算性能の高さから、流体シミュレーション[3]や数値計算[4]などを高速化する研究も行われている。

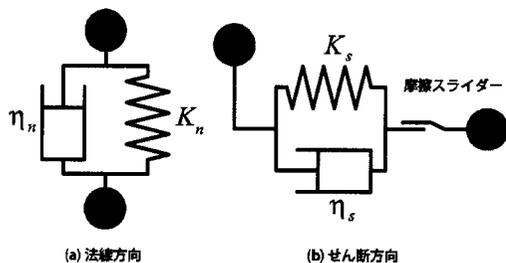


図1 粒子間相互作用力を表すフォークトモデル

2. DEM

DEM(Distinct Element Method)とは、砂や岩石などのような非連続流体の振る舞いを計算する場合に利用される手法で、非連続流体を構成する個々の粒子の運動を、粒子間の接触力などを考慮して計算するというものである[5]。接触粒子間に働く相互作用力は、バネとダッシュポットから構成されるフォークト・モデルを使用して求める。このシミュレーションモデルによって、各粒子の振る舞いを運動方程式で記述することができ、それぞれの粒子に作用する接触力を計算できる。

2.1 フォークト・モデル

粒子間の接触力は様々な方向に作用するため、計算の便宜上、接触2粒子の中心方向（法線方向）成分とそれに垂直なせん断方向成分にわけ、それぞれの成分にフォークト・モデルを適応する（図1）。

せん断方向成分には、粒子間での摩擦相互作用が考慮できるように、摩擦スライダを挿入する。質量 m をもつ2粒子間に働く力 F による接触粒子の並進変位 u と回転変位 ψ は、バネの弾性係数を K 、ダッシュポットの粘性係数を η として次式で表される。添え字 n, s はそれぞれ法線およびせん断方向を表す。

$$\begin{aligned} m \frac{d^2 u_n}{dt^2} + \eta_n \frac{du_n}{dt} + K_n u_n &= 0 \\ m \frac{d^2 u_s}{dt^2} + \eta_s \frac{du_s}{dt} + K_s u_s &= 0 \\ I \frac{d^2 \psi}{dt^2} + \eta r^2 \frac{d\psi}{dt} + Kr^2 \psi &= 0 \end{aligned} \quad (1)$$

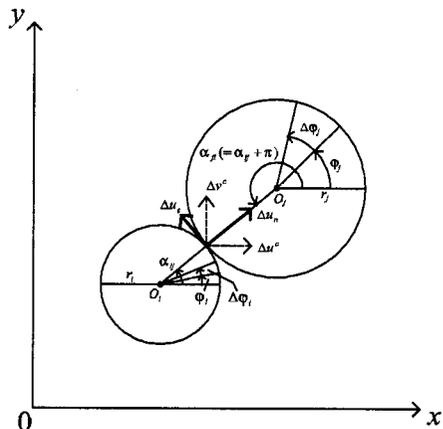


図2 2粒子の接触

ここで、 I は慣性モーメント ($= \rho_p \pi r^4 / 2$) を示し、 ρ_p は粒子密度、 r は粒子半径である。

2.2 粒子間の相対変位増分

粒子 i が微小時間 Δt の間に位置 $A(x_i, y_i)$ から位置 $B(x_i + \Delta u_i, y_i + \Delta v_i)$ に移動するものとする。このとき変位増分の x, y 成分をそれぞれ $\Delta u_i, \Delta v_i$ とし、その変位にともなう粒子の回転変位増分を $\Delta \phi_i$ とする。粒子半径がそれぞれ r_i, r_j である2つの球形粒子 i, j は次式で満たされる条件を満足するときに接触する。

$$r_i + r_j \geq L_{ij} \quad (2)$$

L_{ij} は粒子 i, j の中心間距離である。また、 i, j 粒子の接触点の位置を識別するために、 i, j 粒子の中心を結ぶ共通法線が x 軸となす角 α_{ij} を用いる。この角度 α_{ij} と粒子の座標との関係は次式で表せる。

$$\begin{aligned} \sin \alpha_{ij} &= -(y_i - y_j) / L_{ij} \\ \cos \alpha_{ij} &= -(x_i - x_j) / L_{ij} \end{aligned} \quad (3)$$

接触する2粒子 i, j の微小時間 Δt における相対変位量の法線方向とせん断方向成分は図2のようになり、角度 α_{ij} を用いて次式で与えられる。

$$\begin{aligned} \Delta u_n &= (\Delta u_i - \Delta u_j) \cos \alpha_{ij} + (\Delta v_i - \Delta v_j) \sin \alpha_{ij} \\ \Delta u_s &= -(\Delta u_i - \Delta u_j) \sin \alpha_{ij} + (\Delta v_i - \Delta v_j) \cos \alpha_{ij} \\ &+ (r_i \Delta \psi_i + r_j \Delta \psi_j) \end{aligned} \quad (4)$$

2.3 粒子間の作用力

様々な方向に作用する粒子間の接触力を、法線方向成分 f_n とせん断方向成分 f_s に分けて考える。

2.3.1 法線方向成分

時刻 t における作用力は、時刻 $t - \Delta t$ の作用力に時間 Δt の間の粒子 i, j の相対変位増分 Δu_n による力を加えたものであり、次式で表される。

$$\begin{aligned}\Delta e_n &= K_n \Delta u_n \\ \Delta d_n &= \eta_n \frac{\Delta u_n}{\Delta t}\end{aligned}\quad (5)$$

時刻 t における法線方向の弾性力 Δe_n と粘性抵抗力 Δd_n の和が、その時刻での2粒子間に働く接触力の法線方向成分となる。

2.3.2 せん断方向成分

せん断方向成分についても同様に、相対変位増分 Δu_s による力は次式で表せる。

$$\begin{aligned}\Delta e_s &= K_s \Delta u_s \\ \Delta d_s &= \eta_s \frac{\Delta u_s}{\Delta t}\end{aligned}\quad (6)$$

このとき、クーロンの摩擦法則から得られるせん断力より大きな力は作用しないので、せん断力には次の条件を加える。

$$\begin{aligned}\Delta e_n &\longrightarrow \Delta e_s = \Delta d_s = 0 \\ |\Delta e_s| &\geq \mu \Delta e_n \longrightarrow |\Delta e_s| = \mu \Delta e_n \times \text{SIGN}(\Delta e_s), \\ \Delta d_s &= 0\end{aligned}\quad (7)$$

ここで、 μ は粒子間摩擦係数、SIGN は Δe_s の正負を表す記号関数である。法線方向成分と同様に、時刻 t におけるせん断方向の弾性力 Δe_s と粘性抵抗力 Δd_s の和が、その時刻での2粒子間に働く接触力のせん断方向成分となる。

2.4 数値積分計算

一般的な粉粒体では、1つの粒子の周りに多数の粒子が接触しているので、それらすべての接触点に対して式(1)が成立し、着目粒子の運動を求めるために接触点と同じ数の方程式を連立して解かなければならない。この問題を避けるために、実際の計算では式(1)を時間増分 Δt で数値積分して計算する。

時刻 t での着目粒子 i に接触するすべての粒子 j からの接触力 f_n, f_s を求めると、粒子 i に作用する力の x 方向分力 X_i と y 方向分力 Y_i 、ならびに粒子 i の中心周りのモーメント M_i を次式により求めることができる。

$$\begin{aligned}X_i(t) &= \sum_j \{-f_n(t) \cos \alpha_{ij} + f_s(t) \sin \alpha_{ij}\} \\ Y_i(t) &= \sum_j \{-f_n(t) \sin \alpha_{ij} - f_s(t) \cos \alpha_{ij}\} \\ M_i(t) &= -r_i \sum_j \{f_s(t)\}\end{aligned}\quad (8)$$

Σ は粒子 i に接触するすべての粒子 j についての総和を表し、 m_i は粒子 i の質量である。粒子 i に関する運動方程式を時間増分 Δt で差分近似すると、作用力は式(8)で与えられるから、時刻 t における粒子 i の加速度は次式で求められる。

$$\begin{aligned}a_{xi}(t) &= X_i(t) / m_i \\ a_{yi}(t) &= Y_i(t) / m_i \\ a_{\phi i}(t) &= M_i(t) / I_i\end{aligned}\quad (9)$$

$a_{xi}, a_{yi}, a_{\phi i}$ はそれぞれ粒子 i の x 軸方向、 y 軸方向、回転成分の加速度を表し、 I_i は慣性モーメントを表す。

次に、時刻 t における粒子の速度は式(9)を時間増分 Δt で積分することで求められる。計算式は以下ようになる。

$$\begin{aligned}v_{xi}(t) &= v_{xi}(t - \Delta t) + a_{xi}(t) \times \Delta t \\ v_{yi}(t) &= v_{yi}(t - \Delta t) + a_{yi}(t) \times \Delta t \\ v_{\phi i}(t) &= v_{\phi i}(t - \Delta t) + a_{\phi i}(t) \times \Delta t\end{aligned}\quad (10)$$

$v_{xi}, v_{yi}, v_{\phi i}$ はそれぞれ粒子 i の x 軸方向、 y 軸方向、回転成分の速度を表す。

さらに速度を時間増分 Δt で積分することで、時刻 t における粒子 i の位置を求めることができる。

$$\begin{aligned}p_{xi}(t) &= p_{xi}(t - \Delta t) + v_{xi}(t) \times \Delta t \\ p_{yi}(t) &= p_{yi}(t - \Delta t) + v_{yi}(t) \times \Delta t \\ p_{\phi i}(t) &= p_{\phi i}(t - \Delta t) + v_{\phi i}(t) \times \Delta t\end{aligned}\quad (11)$$

$p_{xi}, p_{yi}, p_{\phi i}$ はそれぞれ粒子 i の x 座標、 y 座標、回転変位を表す。

式(9)から式(11)の計算を繰り返すことで、時間増分 Δt ごとの粒子の運動が計算できる。

3. GPU上への実装

本研究では、DEM を用いた粉粒体シミュレーションの GPU 上への実装方法を提案する。GPU は並列データ処理に特化したアーキテクチャを持っている。そのため、各入力データの処理において、前後のデータ処理との依存関係を実現することは難しい。また、命令実行の際に自由に読み書きが行えるメモリ空間が存在しないため、テクスチャメモリから入力データを読み出し、処理結果をフレームバッファに書き出すという処理手順になる。

3.1 アルゴリズムの概要

本研究では、1 タイムステップ内で、大きく分けて4つの処理を行う。

- ①近傍領域に存在する粒子を探索
- ②境界との接触力の計算
- ③近傍領域に存在する粒子同士の接触力の計算
- ④データテクスチャの更新

必要となる入力テクスチャは、「座標テクスチャ」、「速度テクスチャ」、「粒子-境界間に作用する法線方向の弾性力テクスチャ」、「粒子-境界間に作用するせん断方向の弾性力テクスチャ」、「接触粒子間に作用する法線方向の弾性力テクスチャ」、「接触粒子間に作用するせん断方向の弾性力テクスチャ」、「着目粒子に作用する接触力テクスチャ」、そして近傍領域に存在する粒子の番号を格納した「近傍粒子テクスチャ」の計8枚で、すべてのテクスチャのサイズは、粒子の総数と等しいピクセル数を持つ2次元テクスチャである。座標テクスチャはRGBの3つの要素を持ったテクスチャで、各要素にx座標、y座標、粒子の回転変位 ϕ を保持している。速度テクスチャも同様にRGBにx軸方向の速度、y軸方向の速度、回転速度が格納されている。粒子-境界間の弾性力テクスチャ2枚は、RGBの各要素に3つの境界面から受ける弾性力がそれぞれ格納される。着目粒子に作用する接触力テクスチャは中間データとして利用され、RGBの各要素に一時的なx軸方向の合力、y軸方向の合力、力のモーメントが格納される。近傍粒子テクスチャは、RGBAの4要

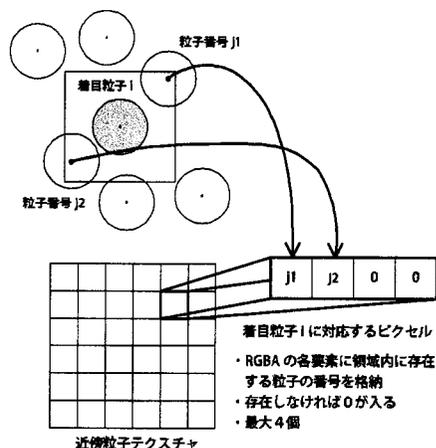


図3 近傍領域内の粒子を格納

素に着目粒子の近傍領域に存在する粒子の粒子番号が格納される。粒子間の弾性力テクスチャ2枚は、各RGBA要素に近傍領域テクスチャに保持された粒子番号の粒子から受ける弾性力を保持している。

今回利用したGPUには、同時に出力できるデータが4種類という制限があるため、②の処理が終了後、一度中間データをCPU側に出力している。以下で、これらのテクスチャを使用した提案手法の実装方法を詳しく説明する。

3.2 近傍領域に存在する粒子の探索

GPUで各粒子の振る舞いを計算する前に、計算の対象となる粒子に衝突する可能性のある粒子(近傍領域に存在する粒子)の探索を行う。この処理はCPU側で行い、近傍領域に存在する粒子の番号を、近傍粒子テクスチャに格納する(図3)。

近傍領域の大きさは、図3のように着目粒子*i*を中心とした1辺が粒子直径の2倍となる正方形とした。その領域内に存在する粒子の番号を近傍粒子テクスチャの着目粒子に対応するピクセルのRGBAに格納する。各要素にはそれぞれ1つの粒子番号を格納することができる(最大で4つ)。粒子が存在しない場合には、その要素に0が入る。

この処理を全ての粒子に対して行うことで、入力データの1つである近傍粒子テクスチャを作成できる。

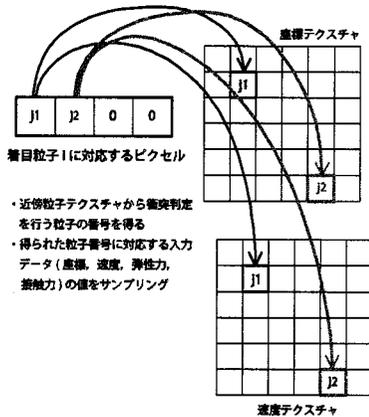


図 4 近傍粒子テクスチャからのサンプリング

3.3 境界との接触力の計算

全ての粒子に対して、境界との距離を利用した衝突判定を行う。座標テクスチャ、速度テクスチャ、粒子-境界間に作用する法線方向の弾性力テクスチャ、粒子-境界間に作用するせん断方向の弾性力テクスチャを入力データとし、2章で説明した DEM を用いて境界との接触力を計算する。

この処理の後、中間データとして粒子に働く接触力を出力する。また、次のステップで利用するデータとして、粒子-境界間の弾性力テクスチャも同時に出力する。

3.4 近傍領域に存在する粒子同士の接触力の計算

近傍粒子テクスチャを利用し、全ての粒子に対して、その粒子の近傍領域内に存在する粒子との接触力の計算を行う。座標テクスチャ、速度テクスチャ、接触粒子間に作用する法線方向の弾性力テクスチャ、接触粒子間に作用するせん断方向の弾性力テクスチャ、中間データとして出力した着目粒子に働く接触力テクスチャ、そして近傍粒子テクスチャを入力データとして使用する。

まず、接触力計算の対象となる i 番目の粒子を現在のフラグメントの座標から求める[6]。次に、 i 番目の粒子の近傍領域に存在する粒子の番号を、近傍粒子テクスチャからサンプリングする。格納されている粒子の番号を利用して、 i 番目の粒子と同様に座標テクスチャ、速度テクスチャ、

弾性力テクスチャから必要な値をサンプリングする(図4)。

サンプリングした値から、2章で説明した DEM を用いて各粒子に働く接触力を計算する。その接触力から加速度を求め、数値積分を繰り返すことで粒子の運動を求めることができる。

3.5 データテクスチャの更新

最終的に求めた粒子の座標、速度、粒子間の法線方向とせん断方向の弾性力を出力データとし、次のステップで利用できるようにテクスチャを更新する。粒子のレンダリングは、更新された座標テクスチャの値を利用して行っている。

4. 実行結果と考察

実行結果を図5に示す。シミュレーション環境は、Pentium4 3.40GHz, ATI RADEON X300 を搭載した PC で行った。粒子の数は 1024 個、差分時間は 0.00015 である。

GPU を利用したプログラムとそうでないものとで実行速度の比較を行った。表1にその結果を示す。

表 1 実行速度

	CPU	GPU
256 個	108 fps	6 fps
1024 個	27 fps	3 fps
4096 個	6 fps	1 fps

GPU 上へプログラムを実装することで、処理全体をより高速に実行できることを期待したが、処理速度向上には至らなかった。原因としては、テクスチャを多く使用しているため、CPU-GPU 間のデータのやり取りで時間を費やしてしまうということが考えられる。

5. おわりに

本研究では、2次元での DEM を用いた粉粒体シミュレーションを GPU 上へ実装した。しかしながら、提案手法では8種類のデータテクスチャを使用する必要があり、あまり効率のよいアルゴリズムを実装することができなかった。

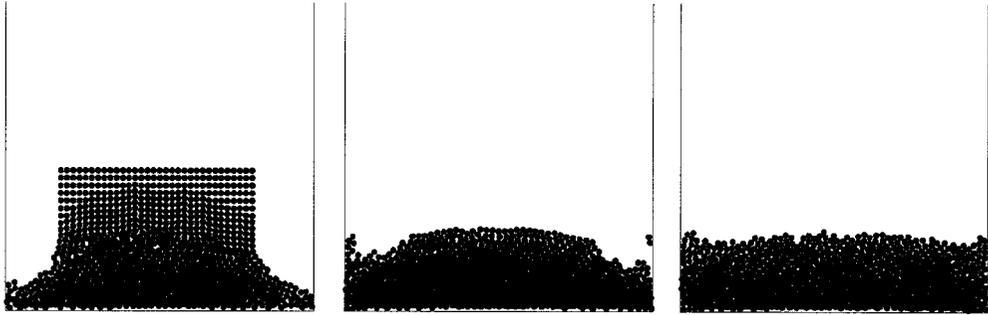


図 5 実行結果

今後の課題としては、処理速度を向上させるために、使用するテクスチャの数を少なくできるようなアルゴリズムで実装する必要がある。また、3次元への対応や、より粉粒体らしく見えるようなレンダリングを行う必要があると考えられる。

参考文献

- [1] Yongning Zhu, Robert Bridson, “Animating Sand as a Fluid”, SIGGRAPH 2005, pp.965-972, 2005-4
- [2] Kouichi Onoue, Tomoyuki Nishita, “An Efficient Method for Displaying Marks on Soft Grounds Created by Objects”, IIEEJ 2003 (Journal of IIEEJ) ,Vol.32.No.4, 2003-7
- [3] Randima Fernando, “GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics”, Addison-Wesley Professional, 2004
- [4] Matt Pharr, Randima Fernando, “GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation”, Addison-Wesley Professional, 2005
- [5] 粉体工学会, “粉体シミュレーション入門”, 産業図書, 1998
- [6] Randi j. Rost, “OpenGL Shading Language”, Addison-Wesley Professional, 2004