

ボクセルベース衝突判定アルゴリズムを用いた剛体運動シミュレーション

Voxel Based Collision Detection Algorithm for Rigid Body Simulation

鈴木克幸¹⁾、久保田純²⁾、大坪英臣³⁾

Katsuyuki SUZUKI, Jun KUBOTA and Hideomi OHTSUBO

- 1) Ph.D. 東京大学助教授 新領域創成科学研究科 環境学専攻 (〒113-8656 東京都文京区本郷 7-3-1)
- 2) 東京大学 大学院 工学系研究科 環境海洋工学専攻 (〒113-8656 東京都文京区本郷 7-3-1)
- 3) 工博 東京大学教授 工学系研究科 環境海洋工学専攻 (〒113-8656 東京都文京区本郷 7-3-1)

For the simulation of rock fall, the robust rigid body dynamic simulation algorithm based on the technique used in the computer graphics field was developed. The rotation of rigid body was represented using quaternion was used and voxel based collision detection algorithm was developed. The strategy for determining time step was described. Several examples are shown to demonstrate the efficiency, robustness and applicability to the non-convex objects of the method, and compared with polygon based method. It was shown that by changing the level of voxel representation, it was possible to control the accuracy and computational efficiency.

Key Words : Rock Fall, Rigid Body Dynamics, Quaternion, Voxel, Collision Detection

1 緒言

落石の挙動を予測するシミュレーション手法としては、質点系のモデルに基づく解析手法と、剛体を用いたモデルによる解析手法に大別される。質点系のモデルは現実とは異なるため、各種の経験的なパラメータ設定や実験によるパラメータ同定が必要であり、解析手法としては問題が多い。任意形状の落石や相互に干渉しあう複数の落石の解析には、落石の形状を考慮した解析が必須である。代表的な手法として、Cundall らにより開発された、ブロックを剛体として取り扱う DEM(Distinct Element Method)¹⁾、と Shi らにより開発されたブロックを弾性体として取り扱う DDA (Discontinuous Deformation Analysis)²⁾がある。DEM を落石に適用する研究は数多い。また、佐々木ら³⁾、Wu ら⁴⁾は DDA に基づく落石の解析を行っている。一般に、剛体としての落石の解析を正確に行うためには、2次元モデルでは現実との整合性の問題が大きく、3次元モデルが必須である。しかし、2次元から3次元への拡張は、プログラムが極端に複雑になるため、多くの解析は2次元にとどまっているのが現状である。

一方において、コンピュータグラフィックスの分野では、ほとんどすべての計算は3次元的に行われている。特に、近年は物理ベースのシミュレーションに基づくアニメーションの作成が盛んに研究されており、特に剛体運動に関しては様々なアルゴリズムが提案されている。この論文では、コンピュータグラフィックスの分野で開発されたアルゴリズムをベースとして、剛体運動シミュレーションのプ

ログラムを開発し、その落石への適用の可能性を探ることを目的とする。

2章においては、剛体運動シミュレーションの基礎となる運動方程式を定義する。ここでは、回転の表現としてCGの世界で用いられている、クォータニオン(四元数)を用いる。次に、3章において衝突判定のアルゴリズムを検討する。一般に3次元形状はポリゴン(3角形よりなる多面体)で表現されるため、その運動の解析にはポリゴンとポリゴンの衝突を検出し、衝突が生じた際に適当な反発力を定義して、貫入を回避する必要がある。しかし、数多くのポリゴンに対して接触判定を行うのは、非常に計算時間を要する。Baraff ら⁵⁾により凸多面体の効率的な接触判定アルゴリズムおよび反力の定義が提案されているが、これは非凸な形状には適用することができない。また、多面体ベースの接触判定を考えた場合、辺と辺の接触、面と角の接触、面と辺の接触、面と面の接触など、様々なケースがあり、それらをもれなく判定するのは非常に複雑なアルゴリズムになる。また、接触時の反力の計算も、これらの様々な接触ケースに対して同じ式で行うことは疑問が残る。そこで、ここでは Dingliana and O'Sullivan⁶⁾の、衝突判定に Sphere Tree を用いるというアイディアを用い、ボクセルベースの衝突判定および反力の定義法を提案する。この方法により、非凸な形状に対しても衝突判定が可能になり、また反力の定義も非常に簡単に行うことができる。また、この方法を使うことにより、時間積分の刻み幅に関しても明確な指標を示すことができる。

最後に、いくつかの例題により有効性を示す。なお、この論文では剛体を扱っているが、アルゴリズムとしてはDEMに限らず、DDAにおいても利用可能である。

2 クォータニオンを用いた剛体運動解析

3次元の剛体の運動シミュレーションは一般に、剛体の位置、姿勢（回転）、運動量および角運動量を状態変数として微分方程式で表現する。状態変数を以下のように定義する。

$$y(t) = \begin{Bmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{Bmatrix} \quad (1)$$

ただし、 $x(t)$ は重心の位置ベクトル、 $R(t)$ は 3×3 の回転行列、 $P(t)$ は運動量ベクトル、 $L(t)$ は角運動量ベクトルである。この状態変数ベクトルの時間微分は、以下のようになる。

$$\frac{dy(t)}{dt} = \begin{Bmatrix} v(t) \\ \omega(t) \times R(t) \\ \dot{P}(t) \\ \dot{L}(t) \end{Bmatrix} = \begin{Bmatrix} v(t) \\ \omega(t) \times R(t) \\ m\dot{v}(t) \\ I\dot{\omega}(t) \end{Bmatrix} \quad (2)$$

ただし、 $v(t)$ は速度、 $\omega(t)$ は角速度である。ニュートンの運動方程式より、これらには以下のような関係がある。

$$\begin{aligned} F(t) &= m\dot{v}(t) \\ \tau(t) &= I\dot{\omega}(t) \end{aligned} \quad (3)$$

ただし、 $F(t)$ は外力、 $\tau(t)$ は外力のつくるトルク、 m は剛体の質量、 I は慣性モーメント行列である。この微分方程式を数値的に積分することにより、各時間ステップでの剛体の状態を計算することができる。本論文では、数値積分は陽的オイラー法に基づき行う。

剛体の姿勢（回転）をオイラー角から計算された 3×3 の回転行列で表すと、シミュレーションが進むにつれて数値誤差が増大し、軸周りの回転に伴う誤差が大きくなってしまふなどの不都合が生じる。この問題は、回転行列のかわりに、ハミルトンによって考案されたクォータニオン（四元数）で表現することで解決する³⁾。クォータニオンは、1つのスカラーと3つのベクトル成分あわせて4成分を持ち、次のように表現される。

$$\begin{aligned} q &= [s, v] = [s, (v_x, v_y, v_z)] \\ &= s + v_x i + v_y j + v_z k \end{aligned} \quad (4)$$

ここで i, j, k は異なる虚数単位で、次のような性質がある。

$$i^2 = -1, j^2 = -1, k^2 = -1, ij = -ji = k, \text{etc}$$

任意の軸 v_{axis} 周りに θ だけ回転させるクォータニオンは以下のように定義される。

$$\begin{aligned} q(\theta) &= [\cos(\theta/2), v_{axis} \sin(\theta/2)] \\ &= [\cos(\theta/2), (v_x \sin(\theta/2), v_y \sin(\theta/2), v_z \sin(\theta/2))] \end{aligned} \quad (5)$$

また、クォータニオンが与えられたとき、回転行列は以下

のように定義され、クォータニオンと回転行列は相互に変換が可能である。

$$R = \begin{pmatrix} 1 - 2v_y^2 - 2v_z^2 & 2v_x v_y - 2sv_z & 2v_x v_z + 2sv_y \\ 2v_x v_y + 2sv_z & 1 - 2v_x^2 - 2v_z^2 & 2v_y v_z - 2sv_x \\ 2v_x v_z - 2sv_y & 2v_y v_z + 2sv_x & 1 - 2v_x^2 - 2v_y^2 \end{pmatrix} \quad (6)$$

このクォータニオンを用いると、運動方程式は以下のよう書くことができる。

$$y(t) = \begin{Bmatrix} x(t) \\ q(t) \\ P(t) \\ L(t) \end{Bmatrix} \quad \frac{dy(t)}{dt} = \begin{Bmatrix} v(t) \\ \omega(t)q(t) \\ m\dot{v}(t) \\ I\dot{\omega}(t) \end{Bmatrix} \quad (7)$$

ただし、 $\omega(t) = [0, (\omega_x, \omega_y, \omega_z)]$ というクォータニオンで、 $\omega(t)q(t)$ はクォータニオンの積である。クォータニオンの積は以下のように定義される。

$$[s_1, v_1][s_2, v_2] = [s_1 s_2 - v_1 \cdot v_2, s_1 v_2 + s_2 v_1 + v_1 \times v_2]$$

3 ボクセルを用いた衝突判定

衝突判定に関しては、コンピュータグラフィックスの分野では様々なアルゴリズムが提案されている。例えば、文献7)に詳しいサーベイがある。一般に多面体同士の衝突判定手法では、点と面の判定や線と線の判定といったように、衝突の種類により衝突判定のアルゴリズムが異なる。また、その衝突の後の挙動のシミュレーションを行うには衝突の際に生じる力を評価する必要があるが、これも当然衝突の種類により異なり、アルゴリズムを非常に複雑にし、計算効率の低下を招く要因となる。また、手法の脆弱性から、しばしば物体が互いに貫通するような現象が生じ、これを防止するためには積分の時間刻みを調整する必要があるが、時間刻みに関しても明確な指針を示すことは困難である。そのためここでは、よりロバストな衝突判定手法として Dingliana and O'Sullivan⁸⁾の考え方を採用し、衝突前後の効率的な判定手法を独自に開発することで、判定の効率化を図るとともに計算時間と精度のコントロールを可能にした。

ここでは手法の解説のため2次元の図で表現するが、ボクセルベースのため3次元への拡張は容易である。衝突判定を行うべき形状は、ポリゴン（3角形よりなる多面体）で表現されているが、その形状を図-1に示すように形状全体を内包するボクセル（立方体形状）に分割する。そして、各ボクセルに対して外接する球を考える。衝突判定においては物体形状を、このボクセルに対する球で近似的に表現し、球と球の衝突判定を行うことによって物体同士の衝突判定を行う。実際の衝突は必ず表面でおこるため、物体表面に対応する球以外は利用しない。形状のボクセル分割のレベルを変更することにより、形状の粗い近似からより正確な近似まで様々なレベルでの表現が可能である。全体を

1つのボクセルで表現した場合を Level 0 とし、ボクセルの1辺の寸法を半分にするに従い Level を増やしていくこととする。図-2 に3次元の場合のボクセル分割と形状表現の例を示す。当然、衝突判定の精度を上げるために細かいボクセル分割を行えば、衝突判定に要する計算時間が増加するため、要求精度に応じた適当なボクセル分割のレベル設定が必要になる。言い方を変えれば、Level を変えることにより精度と計算時間の調整は従来手法よりはるかに容易に行える。

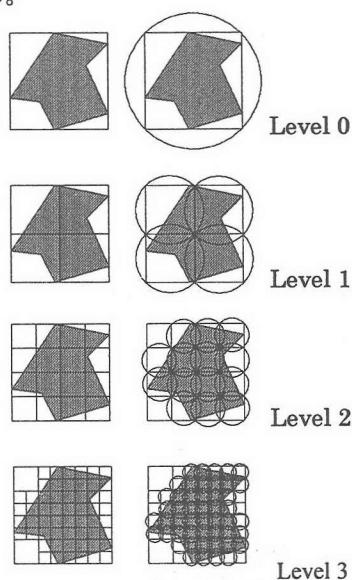


図-1 物体のボクセル分割（2次元）

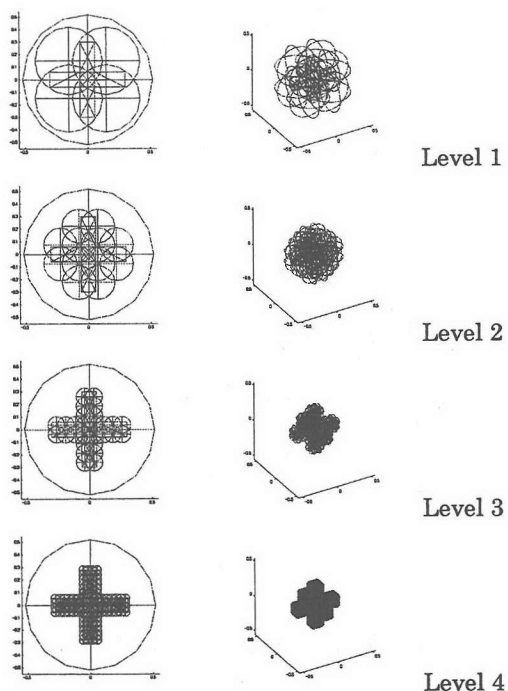


図-2 物体のボクセル分割（3次元）

衝突判定の流れは以下になる。まず、物体の重心間距離を求めて物体の半径と比較することにより、2物体が衝突している可能性を調べる。このときに用いる物体の

半径は図-1 の Level 0 の球のものではなく、物体に外接する球の半径を用いる。図-1 の Level 0 の球は物体を内包する立方体に外接する球であるため、この半径を用いると計算の効率が悪化する。この判定で衝突している可能性がある」と判明した場合は Level 1 の球同士で衝突の判定を行う。そして、衝突している Level 1 の球があれば、そのボクセルをさらに8分割して Level 2 のボクセルにする。このように、衝突しているボクセルのみを細かくしてゆくことで、判定の効率を上げている。

以上の操作を図-3 および図-4 に示す。図-3 は衝突判定を Level 4 まで行う場合のフローチャートである。どの Level まで球を細かくするかあらかじめ決定しておき、その Level まで分割しても球同士が衝突していれば、衝突が起こったと判定する。

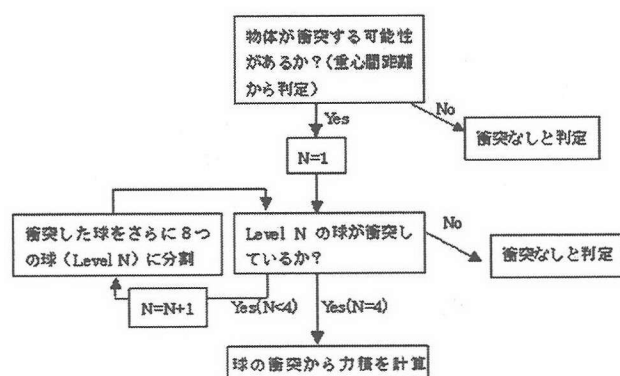


図-3 衝突判定のフローチャート

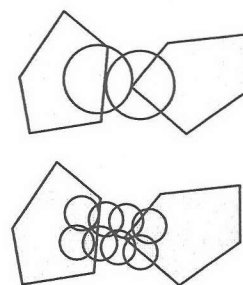


図-4 段階的衝突判定

衝突がおこったと判定された場合には、衝突後の運動を計算するために衝突した物体の相互に作用する力（衝撃力）を求める必要がある。この力は非常に短時間に作用するため、力の時間変化を考慮することはあまり意味がなく、力積と考える。

衝突点は2つの球中心の midpoint とする。図-5 のように球の中心を x_1 、 x_2 として、物体の重心を x_{g1} 、 x_{g2} とする。図-5 のように物体1と物体2が衝突する場合、物体同士の摩擦を考えなければ物体1が受ける力の向きは図-5 の n_{21} であり、式 (8) で表される。摩擦が生じる場合は、この力に対して、運動に応じた適当な摩擦力を垂直力に対して摩擦係数を乗ずることにより求める。

$$n_{21} = \frac{\mathbf{x}_1 - \mathbf{x}_2}{|\mathbf{x}_1 - \mathbf{x}_2|} \quad (8)$$

衝突直前の2つの球の相対速度の大きさを v_{rel}^- 、衝突直後を v_{rel}^+ とする。衝突直前の球の中心の位置を $\mathbf{x}_1^-$ 、 $\mathbf{x}_2^-$ 、衝突直後の位置を $\mathbf{x}_1^+$ 、 $\mathbf{x}_2^+$ と表すと、式(9)および式(10)が成り立つ。

$$v_{rel}^- = n_{21} \cdot (\dot{\mathbf{x}}_1^- - \dot{\mathbf{x}}_2^-) \quad (9)$$

$$v_{rel}^+ = n_{21} \cdot (\dot{\mathbf{x}}_1^+ - \dot{\mathbf{x}}_2^+) \quad (10)$$

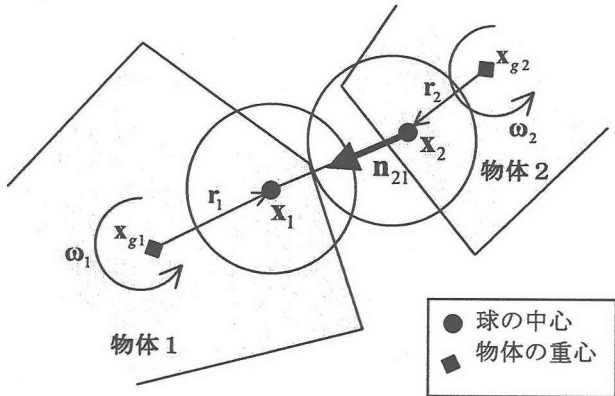


図-5 衝突力の位置と方向

また、反発係数 ε は式(11)である。

$$v_{rel}^+ = -\varepsilon v_{rel}^- \quad (11)$$

角速度 ω も速度と同様に表記すると、球の中心の速度は物体重心の速度と角速度および物体重心から球中心までのベクトル $\mathbf{r}$ により式(12)のように表される。

$$\mathbf{x}_1^- = \mathbf{x}_{g1}^- + \omega_1^- \times \mathbf{r}_1 \quad (12)$$

$$\mathbf{x}_1^+ = \mathbf{x}_{g1}^+ + \omega_1^+ \times \mathbf{r}_1$$

衝突による運動量の変化を j 、物体の質量を M 、物体の慣性モーメント行列を I とすると式(13)および式(14)が成り立つ。

$$\mathbf{x}_{g1}^+ = \mathbf{x}_{g1}^- + \frac{j n_{21}}{M_1} \quad (13)$$

$$\omega_1^+ = \omega_1^- + I_1^{-1} (\mathbf{r}_1 \times j n_{21}) \quad (14)$$

式(12)に式(13)および式(14)を代入すると $\mathbf{x}_1^+$ が $\mathbf{x}_1^-$ で表され、式(9)、式(10)および式(11)と合わせることで力積の大きさは式(15)のように得られる。

$$j = \frac{-(1+\varepsilon)v_{rel}^-}{\frac{1}{M_1} + \frac{1}{M_2} + n_{21} \cdot \left\{ (I_1^{-1} (\mathbf{r}_1 \times n_{21})) \times \mathbf{r}_1 + (I_2^{-1} (\mathbf{r}_2 \times n_{21})) \times \mathbf{r}_2 \right\}} \quad (15)$$

力積ベクトルは式(16)となる。

$$\mathbf{J} = j \cdot n_{21} \quad (16)$$

4 時間積分の刻み幅

時間積分は陽的オイラー法により行う。ここで扱うよう

な衝突が頻繁におこる問題に対しては衝突による不連続性が発生するため、Runge Kutta 法等の高次の積分法を用いるよりも、時間刻みを十分に小さくすることが重要である。その際、時間刻み Δt は衝突に対する精度および安定性より決定することになる。ポリゴンによる衝突判定においては、この時間刻みの決定が難しく、これを大きく取りすぎると物体同士が貫入してしまうことがあるが、経験的な値を使うしかなく、明確な指針はなく、しばしば不安定な解が生じることがある。一方ボクセルによる手法は、ボクセルの Level、速度、角速度により以下のように決定することができる。

物体の速度が速い場合には、球が互いに貫通する可能性があるため、短い時間刻みをとる必要がある。図-6に最も貫通しやすい状況を示す。前のステップで、2つの球が衝突していないが非常に接近している状況である。この場合、 Δt の間に進む距離が球の半径よりも小さければ球が互いに貫通することはない。このことから、 Δt が式(17)を満たせば、この問題は発生しない。ここに、 v_{max} は球の中心における速度、 r は球の半径である。

$$\Delta t \cdot v_{max} \leq r \quad (17)$$

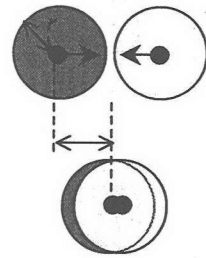


図-6 球同士の貫通

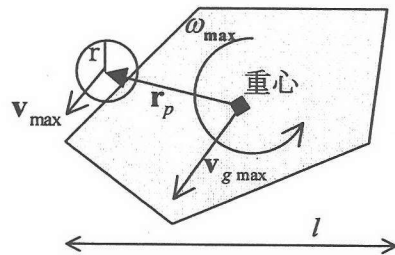


図-7 v_{max} の算出

図-7に示すように、物体は回転しながら運動しているので、 v_{max} は式(18)のように表される。

$$\begin{aligned} v_{max} &= \max(|\mathbf{v}_g| + |\mathbf{r}_p \cdot \omega|) \\ &= |\mathbf{v}_{g \max}| + |\mathbf{r}_{p \max} \cdot \omega_{\max}| \\ &= v_{g \max} + \frac{l}{2} \cdot \omega_{\max} \end{aligned} \quad (18)$$

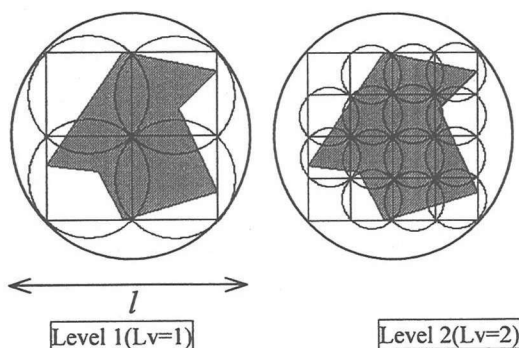


図-8 球の半径とレベルの関係

また図-8 より、球の半径 r 、物体の直径 l とレベル (Lv) の間には式(19)の関係が成立する。

$$r = \frac{l}{2^{Lv+1}} \quad (19)$$

以上より、シミュレーションを行う際に許される Δt の最大値は式(20)で与えられる。

$$\Delta t_{\max} = \frac{l}{2^{Lv+1} \cdot \left(v_{g \max} + \frac{l}{2} \cdot \omega_{\max} \right)} \quad (20)$$

5 精度検証および適用例

例題 1 では、時間刻みを変えることで、計算時間やシミュレーション結果がどの程度異なるかを調べた。また、例題 2 から例題 4 では、ボクセルを用いた衝突判定を用いることにより計算効率が改善されることを検証するため、Barraf らの文献⁵⁾に基づくポリゴンによる衝突判定（以下、旧手法と書く）を用いた場合との計算時間および解析結果の比較を行った。例題を解く際に、衝突判定アルゴリズム以外はすべて同じアルゴリズムを用いた。

例題 5 および例題 6 には、本論文で提案する手法を適用した例を示す。

なお、計算時間とは、各ステップの物体の座標を求めるためにかかった時間であり、描画時間は含まない。計算には MathWork 社の MATLAB を用いた。この計算に用いた PC のスペックは CPU が Pentium4 2.4GHz、メモリ 1GB である。

例題 1 から例題 5 まではすべて物体の密度を $0.8 [g/cm^3]$ （木材）とし、例題 6 では物体の密度を $2.5 [g/cm^3]$ （岩石の標準的な値）とした。また、例題 2 から例題 6 において、反発係数は物体同士の場合 0.5、床と物体の場合 0.3、動摩擦係数は物体同士の場合 0.3、床と物体の場合 0.8 とした。例題 1 では床と物体の間の動摩擦係数を 0.5 とした。

5.1 立方体の投射による精度検証（例題 1）

(1) 解析条件

この例題は、本論文で提案する手法について、衝突判定に用いる球の Level を変化させた場合、結果にどの程度影

響するかを調べることを目的とする。一辺の長さが $30 [cm]$ の立方体を 2 つ用いた。立方体 1 つを重心の座標が $(0, 0, 0.15)$ となるように（単位は $[m]$ ）静止させておき、もう一方の立方体を重心座標 $(1.00, 0.20, 0.30)$ から回転をさせず、速度 $(-5.00, 0, 0)$ で投射した。（速度の単位は $[m/s]$ ）投射後 2 秒たった状態でどの場合も 2 つの立方体が静止した。この時点での重心の位置を参照解と比較した。参照解は Level 6 の球を用い、時間刻みを $1/2160 [s]$ とした場合の結果とした。時間刻みはそれぞれの Level において式(20)から求められる値とほぼ同じ値を用いた。

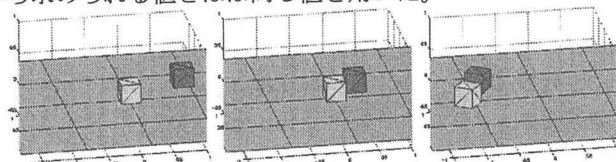


図-9 立方体の投射（例題 1）

(2) 解析結果

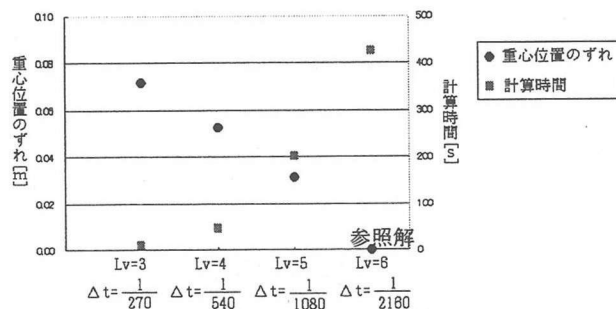


図-10 参照解との重心位置のずれ

図-10 における重心位置のずれは、2 つの立方体の重心それぞれについて参照解の重心位置との距離を求め、その値を平均したものである。Level を 6 としたときの値が真の値であるとは言えないが、Level を上げることによって真の値に近づいていくことが予想される。また、Level を上げることにより、球の数が増え、さらに式(20)により時間刻みを小さくする必要があるため、計算時間が増加する。

また、以上の結果で物体同士の貫入は起こらなかった。時間刻みは式(20)を満たせば十分であると考えられる。

5.2 ブロックの投射（例題 2）

(1) 解析条件

床にいくつかのブロックを投射するシミュレーションを行った。このようなシミュレーションは従来の手法でも作成可能であったが、物体が密集した状況ができると貫入が起りやすいという問題があった。

この例題で用いたブロックの詳細と、各レベルにおける衝突判定の球を図-11 に示す。図-11 は左からブロックのポリゴン形状、Level 2, 3, 4 の球である。新手法では Level 2, 3,

4 の各レベルで計算時間を計測した。ブロックはポリゴン数 12、頂点数 8、直径 0.5[m] で、質量 15[kg] である。

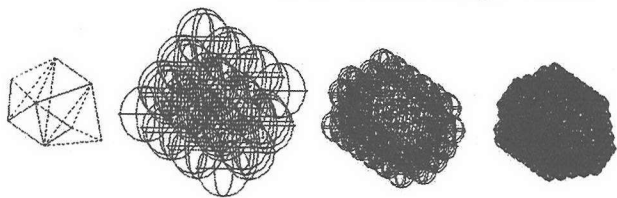


図-11 物体の形状 (例題 2)

時刻 $t=0[s]$ から $t=4[s]$ までの間に、ブロックを 10 個投射し、5 秒間のシミュレーションを行った。投射は、高さ 1[m] の地点から水平方向に速度 2[m/s]、鉛直上向きに速度 0.5[m/s] で行った。式(20)において $L_v=4$ とすると $\Delta t_{\max}=1/113 [s]$ となるため、新手法では $\Delta t=1/120 [s]$ とした。また、旧手法では Δt を決定することが困難であるため、結果を見ながら Δt を変え、適当な値を探った。

(2) 解析結果

旧手法で $\Delta t=1/120 [s]$ から小さくしていったところ、 $\Delta t=1/2400 [s]$ ではじめて貫入の起こらない結果が得られた。新手法では $\Delta t=1/120[s]$ で貫入は起こらなかった。旧手法で貫入が起こったものを図-12 に示す。計算時間の比較を図-13 に示す。貫入が起こったものは図-13 の中で白抜きで表した。また、作成した動画の一部を図-20 に示す。

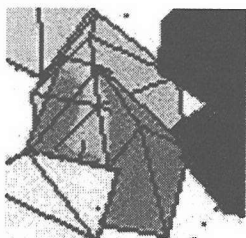


図-12 旧手法で生じた貫入の例

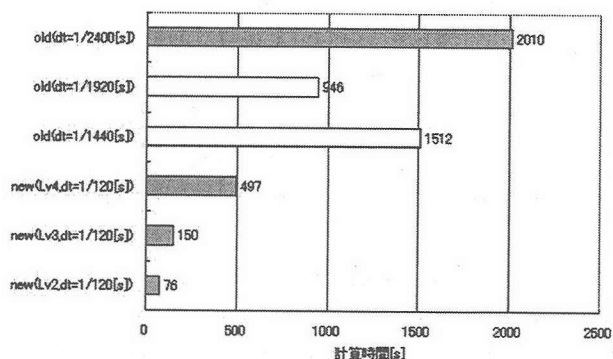


図-13 計算時間の比較 (例題 2)

図-13 の旧手法において $\Delta t=1440 [s]$ の場合の方が $\Delta t=1/1920[s]$ の場合よりも計算時間がかかっている原因は貫入にある。貫入が多くの物体間で起こることにより無

駄な判定の回数が増加する。旧手法においては、 Δt を小さくしてもより精度の高いシミュレーションが行われることが保障されない。一方、新手法では Level を変えることで精度の向上と計算時間の増大がどの程度起こるかを予想しやすく、これも新手法のメリットである。

5.3 ポリゴン数の多いブロックの投射 (例題 3)

旧手法において、衝突判定は点・線や面の衝突として判定されていたため、計算時間はポリゴン数に大きく依存していた。このため、ポリゴン数の大きな物体や、複雑な形状の物体を扱うことが非常に困難であった。新手法では物体をボクセルで表現するため、衝突の精度と計算時間は Level のみに依存する。つまり、対象物体がどんなにポリゴン数が多くても、精度と計算時間は任意に設定できる。

以下の例題では、ポリゴン数の大きな物体も新手法では容易に扱えることを示す。

(1) 解析概要

図-21 に示すブロックを投射する。このブロックはポリゴン数が 192、頂点数が 98 であり、重量 670 [kg] である。

初期状態でこのブロックが床に 2 つ置いてある状況で、時刻 $t=1, 2, 3, 4[s]$ のときに物体をひとつずつ投射する。シミュレーションの時間は $t=6[s]$ までとする。

式(20)において $L_v=4$ とすると $\Delta t_{\max}=1/80 [s]$ となるが、 Δt を $1/100 [s]$ 以上にすると床との衝突判定が粗くなり、床の上で微少な振動をするため、新手法では $\Delta t=1/120[s]$ とした。旧手法では $\Delta t=1/120, 1/600[s]$ を用いた。

(2) 解析結果

計算時間比較の結果を図-14 に示す。なお、横軸は対数で表記してある。

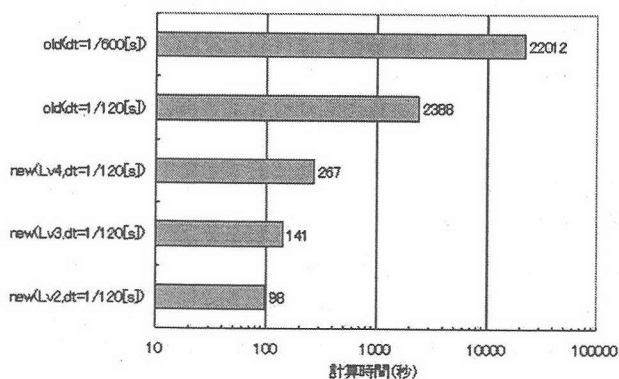


図-14 計算時間の比較 (例題 3)

このように、新手法でははるかに短い時間で計算が終了している。旧手法により得られたアニメーションを見ると、貫入があるようには見えないが、これは、一部の頂点や点が貫入しても、ポリゴン数が多いことにより他の部分が正

常な衝突判定を行うため、結果的に2つの物体が離れる方向に力が働くためであると考えられる。シミュレーション結果の一部を図-21 に示す。

5.4 ブロックの積み重ね（例題4）

例題1では、旧手法を用いた際の貫入の例を示したが、ブロックの積み重ねのようなアニメーションを作成する場合にはこの点が非常に重要になる。物体が積み重なる状況では貫入が起きやすい。しかし、新手法を用いるとこのような状況でも貫入が回避できる。

この例題ではブロックの積み重ねを行っても、新手法では貫入が起こらないことを示す。また、ブロックの個数を変えると計算時間がどのように変化するかという点に関しても比較を行った。

(1) 解析概要

高さ $0.5[m]$ の2箇所の位置から交互にブロックを自由落下させる。ブロックのポリゴン数は24、重量は $44[kg]$ であり、ブロックははじめ1個で、 $t=10[s]$ までの間に10個、20個、30個落とした場合の計算時間を比較した。解析時間は $12[s]$ である。

式(20)において $L_v=4$ とすると $\Delta t_{\max}=1/150[s]$ となるため、新手法では $\Delta t=1/150[s]$ とした。旧手法では $\Delta t=1/600[s]$ と $\Delta t=1/1200[s]$ の2種類の時間刻みで計算を行った。

(2) 解析結果

旧手法の場合は $\Delta t=1/600[s]$ 、 $\Delta t=1/1200[s]$ とともに、すべてのブロック数において貫入が発生した。貫入が起こった様子を図-15 に示す。また、計算時間の比較を図-16 に示す。

旧手法においては、どの程度 Δt を小さくすれば貫入を回避できるか決定するのは困難である。また、図-16の結果からもわかるように、新手法よりも多くの計算時間を費やしてもなお貫入を起こしている。ブロック数の増加に対する計算量の増加も、旧手法に比べて新手法ではそれほど大きく増えないことが分かる。

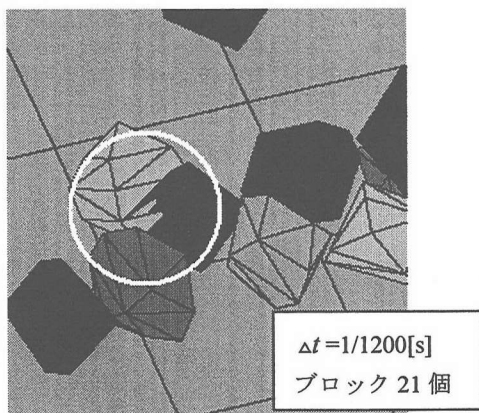


図-15 旧手法を用いた際に起こった貫入（例題4）

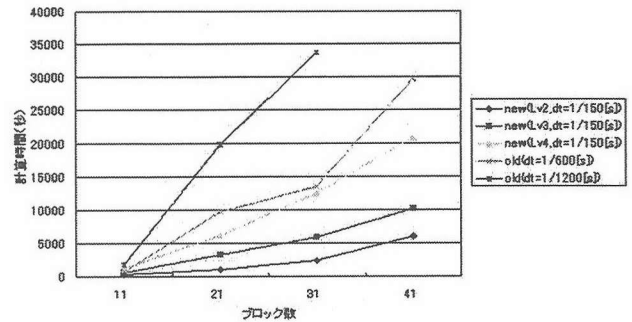


図-16 計算時間とブロック数（例題4）

5.5 非凸形状の物体への適用（例題5）

旧手法では判定手法が凸形状であることを前提に作成されているため、非凸形状物体を扱うことは困難である。また、非凸形状を扱おうとすると計算量が凸形状を扱う場合より遥かに多くなるため、実際に適用している例はほとんどない。新手法では、凸形状であろうと、非凸形状であろうと全く同じアルゴリズムを適用できる。

この例題には図-17 に示すクロス形状のブロックを用いた。ポリゴン数は60であり、頂点数は32である。ボクセル分割は3章の図-2 のようになる。

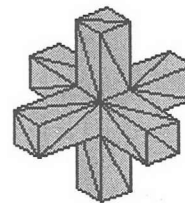


図-17 例題5で用いたブロック

このブロックを高さ $1.5[m]$ の位置から、時刻 $t=12[s]$ までに合計26個の物体を落とした。解析時間は $18[s]$ である。Level2 の場合は $\Delta t=1/120[s]$ とすると、計算時間は約 $1500[s]$ （25分）であった。この例題の結果を図-23 に示す。

また、新手法では樽にブロックを投げ入れるといったシミュレーションも可能である。この例を図-24 および図-25 に示す。

5.6 崖を転がる岩石のシミュレーション（例題6）

最後に、崖のモデルを作成し、物体と崖との衝突に床との衝突アルゴリズムを応用する。図-18 のように、崖をモデリングする際にXY座標を格子状にすることで、衝突判定の高速化を図った。

岩石は5.4の例題4で用いた、ポリゴン数24、頂点数14の物体を使用した。崖の詳細を図-19 に示す。

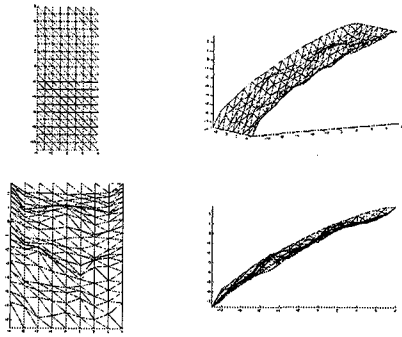


図-18 崖のモデリング

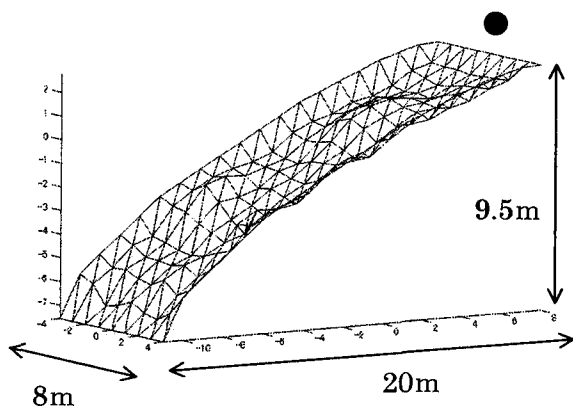


図-19 解析条件

崖の高さは 9.5[m] である。物体は崖の最も高い部分から 0.5[m] 上方で、崖の中央部分である図-19 の●印から投射した。投射速度は水平方向に 0.5[m/s]、鉛直上向きに 2[m/s] である。ただし、投射方向は水平面内で毎回変化させる。

以上のシミュレーションの結果を図-26 に示す。地形に沿って岩石が落ちていく様子が再現された。

6 結論

落石などの動的な離散体の解析のため、コンピュータグラフィックスの技術をベースとして、剛体運動シミュレーションプログラムの開発を行った。クォータニオンによる剛体回転の表現を用いた展開を行い、衝突判定にはこれまでのポリゴンを用いた手法に代わりボクセルをベースとしたアルゴリズムを開発した。時間積分のタイムステップの決定も、ボクセルベースの手法ではより容易かつロバストな値の設定が可能であることを示した。

いくつかの例題により、ポリゴンに基づく手法よりも効率的に衝突判定が行えることを示した。また、ボクセル分割のレベルを変更することにより精度と計算効率の調整が可能であることを示した。さらに、これまでのポリゴンに基づく手法では非常に計算が複雑になった、非凸な形状に対しても効率的に衝突判定が行えることを示した。

様々なシミュレーションにおいて、手法の高度化および汎用化が行われている。近年の計算力学という分野の展開も、このような背景が元になっているが、今後、ますます様々な分野の融合が重要になっていくと思われる。その意味で、より広い分野との技術の交流がますます重要になって行くであろう。

今後は、実際の地形データを用いて、手法の有効性の検証を行っていく予定である。

参考文献

- 1) Cundall, P.A.: A discrete numerical model for granular assemblies, *Geotechnique*, 29, No.1, pp.47-65, 1979
- 2) Shi, G. H.: Block system modeling by Discontinuous Deformation Analysis, Univ. of California, Berkeley, Dept. of Civil Eng. August, 1989
- 3) 佐々木、他：不連続変形法(DDA)による落石問題の検討、*計算工学講演会論文集* Vol.7 pp.423-426, 2002
- 4) Wu, et. al: Slope Failure Simulation by Three-Dimensional Discontinuous Deformation Analysis, *Proc. Computational Engineering Conference* Vol 7, pp. 427-430, 2002
- 5) Baraff, D.: Physically Based Modeling, *SIGGRAPH '99 Course Notes*, 1999
- 6) Shoemake, K. Animating rotation with quaternion curves. In *Computer Graphics (Proc. SIGGRAPH)*, 19, pp. 245-254. ACM, 1985
- 7) P. Jimenez et al.: 3D collision detection: a survey, *Computers & Graphics* 25, pp. 269-285, 2001
- 8) Dingliana, J and O'Sullivan, C.: Graceful Degradation of Collision Handling in Physically Based Animation, *Computer Graphics Forum* Vol.19 Issue3, pp. 239-248, 2000

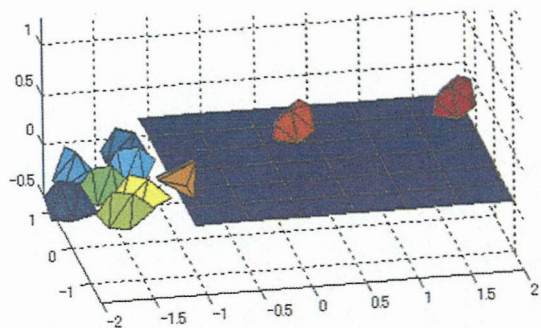


図-20 例題 2

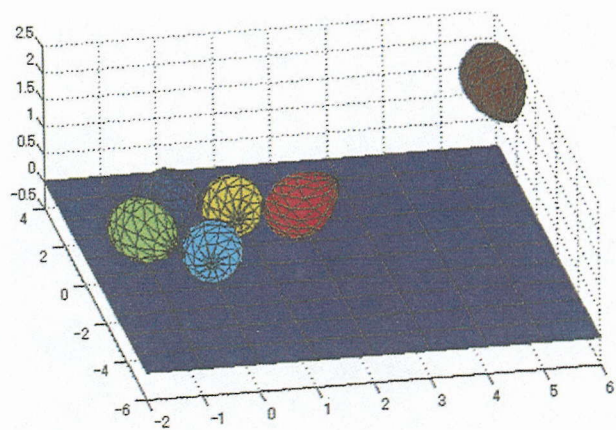


図-21 例題 3

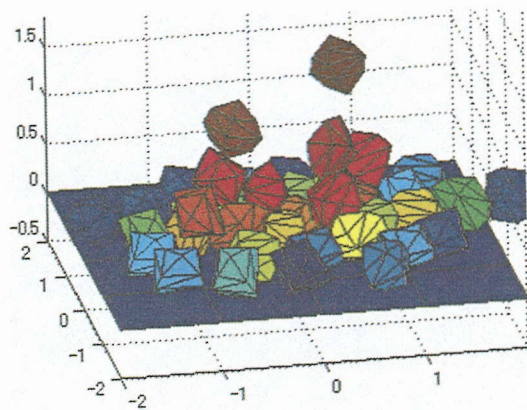


図-22 例題 4

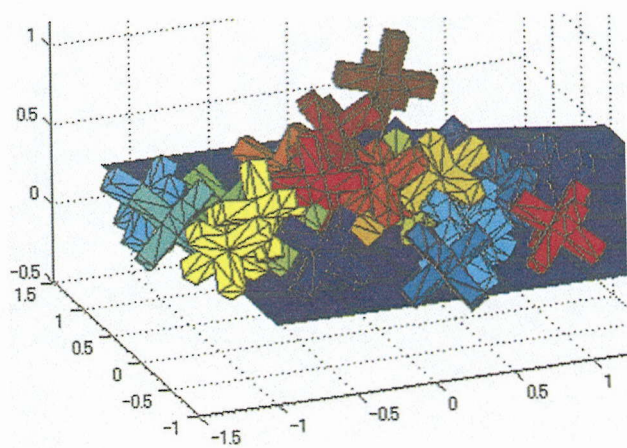


図-23 例題 5

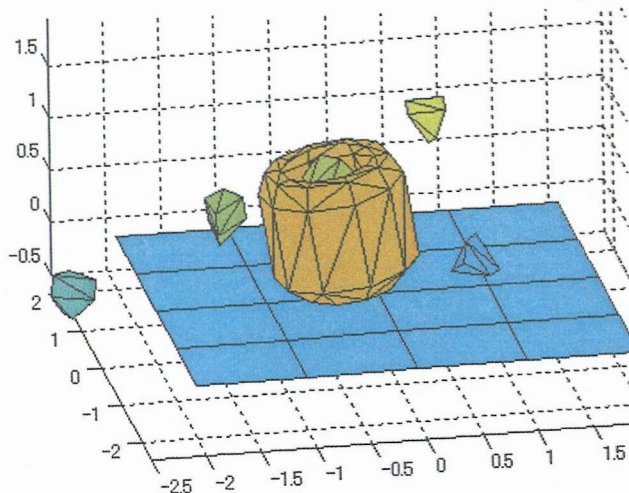


図-24 樽にブロックが入る瞬間

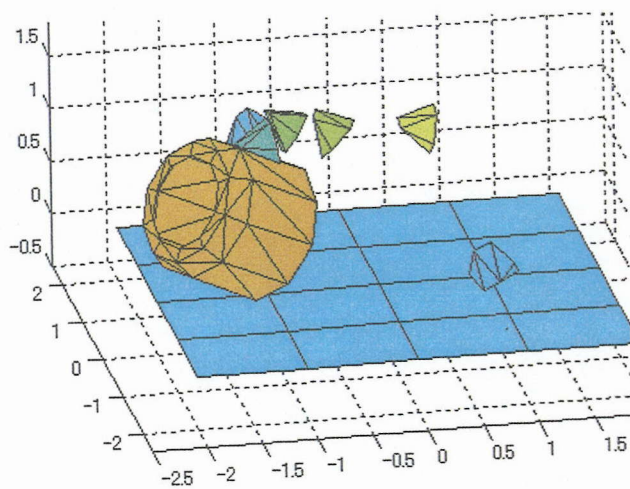


図-25 ブロックにより樽が倒れる瞬間

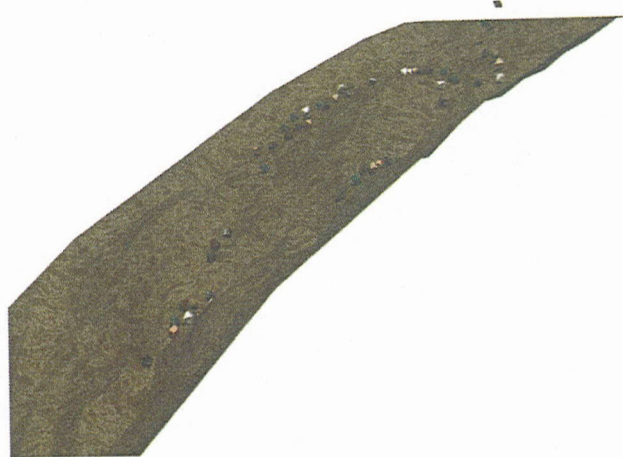


図-26 例題 6

(2003 年 4 月 18 日 受付)