Keywords you should know.　第76回

# 知っておきたいキーワード

# 物理エンジン

関連動画公開中！

長谷川晶一†

† 東京工業大学　精密工学研究所

"Physics Engine, Multi-body Dynamics" by Shoichi Hasegawa (Precision and Intelligence Laboratory, Tokyo Institute of Technology, Tokyo)

キーワード：ゲームエンジン，動力学，シミュレーション，数値計算，モデリング

## 物理エンジンとは

物理法則や物理モデルのような物理学の知識を，動く形にまとめたソフトウェアのことを物理エンジンと呼びます．これは，グラフィクスエンジン，数学エンジン，自然言語処理エンジンなど，物理以外のさまざまな○○エンジンでも同じです．ゲーム開発向けに数多くの○○エンジンがありますが，ゲームだけでなく，科学計算，設計，分析，制御，インターネットやモバイル端末向けのサービスなど，さまざまな目的で○○エンジンが提供，利用されています．

物理エンジンでは，剛体や流体の力学といった力学の一部を対象にしたものが多く，ある時刻の対象の状態からその後の対象の状態を次々に計算することで，対象の運動を再現する計算（＝物理シミュレーション）を行うものが多いです．本稿では，紙面の都合で，剛体のシミュレーションについてお話しします．

力学では，物体の運動が「運動方程式」と呼ばれる微分方程式で書けることが知られています．「運動方程式」は「物体に働く力」と「物体の速度」の関係を表す微分方程式です．この微分方程式を数値積分すれば，時々刻々の物体の位置・速度を求めることができます（**図1**）．コンピュータグラフィクスを用いて，求めた位置に物体を表示すれば，物体の動きを映像にすることもできます．

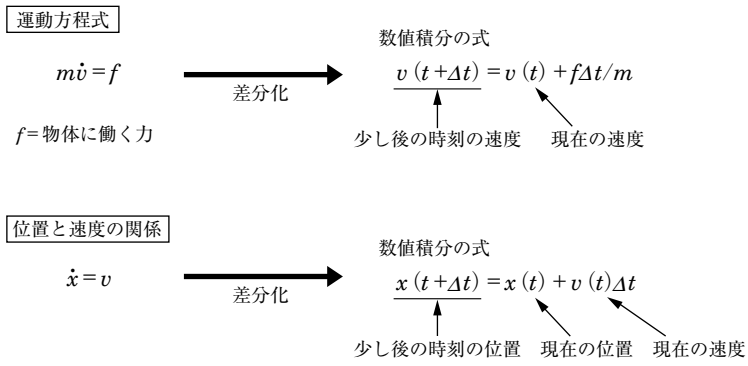


図1　運動方程式と数値積分

運動方程式には物体に働く力 $f$ が含まれますので，この数値積分を行うためには，時々刻々に物体に働く力を求める必要があります．物体に働く力には，重力や磁力のように離れていても働く力と，物体同士が接触している場所で働く力があります．重力や磁力は，物体の位置や速度から簡単に求まります．一方，物体同士が接触すると接触力と呼ばれる力が働きますが，これらを求めるのはなかなか大変で物理エンジンの仕事の中心になります．接触力の接触面に垂直な成分が抗力，水平な成分が摩擦力です（**図2**）．また物体同士が関節でつながっている場合などには，関節を通じて力が働きますがこれも簡単には求まりません．

接触力を求めるためには，まず，どの物体のどこが接触しているのかを調べる接触判定と呼ばれる処理を行います．シミュレーションを行うと物体が動くので，　☞



映像情報メディア学会誌 Vol. 66, No. 5, pp. 394～398（2012）

☞ 積分を1ステップ行う度に接触判定をします．接触力の計算方法によっては，接触した物体と接触位置だけでなく，接触面の向きや接触の深さ，接触部分の形状などの情報も必要になりますので，その場合は接触判定の際にこれらも求めておきます．接触判定を行うために，物理エンジンには物体形状を入力しておく必要があります．形状は球やカプセル型や直方体といった単純な形（プリミティブ）やポリゴンを組合せて表現することが多いです（**図3**）．

次に，接触部分や関節に働く力を求めます．接触力のうち接触面に垂直な抗力は，物体同士が重なることを防ぐように働きます（**図4**）．摩擦力は物体同士が滑ることを防ぎます．関節に働く力は関節が外れないように保ちます．このように，接触部分や関節で働く力は，何らかの拘束条件を満たすように働くので拘束力と呼びます．拘束力を求めるには，拘束条件を満たすような力を見つければ良いのですが，ランダム探索では時間がかかってしまいます．拘束条件が満たされるように，条件を式にして運動方程式と連立させて解く解析法と呼ばれる手法や，拘束条件の違反量（重なりの量や関節のずれの量）に比例した力を拘束力とすることで，拘束違反が減少することを期待するペナルティ法と呼ばれる手法が知られています．

こうして物体に加わる力が求まると，運動方程式を差分化した式に代入することで，少し後の時刻の物体の速度を求めることができ，シミュレーションを1ステップずつ進めることができます．

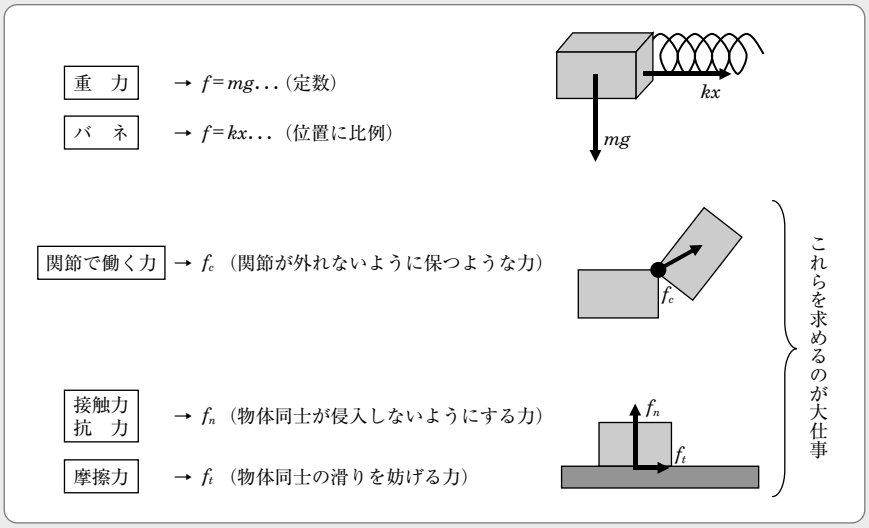


図2　物体に加わる力

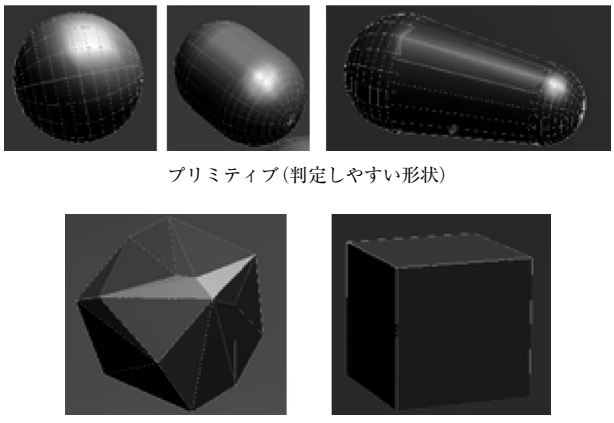


図3　物体形状の例

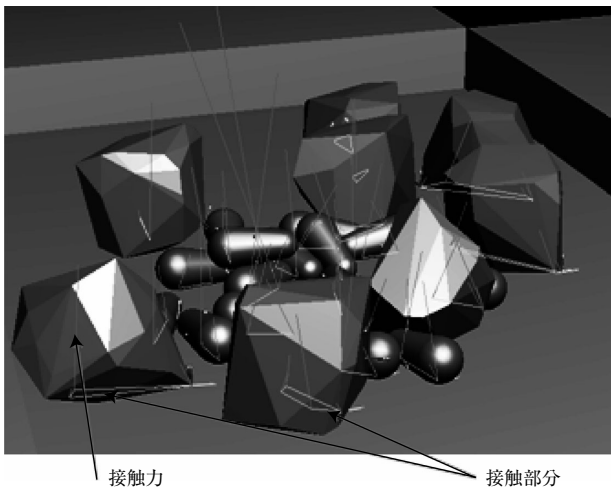


図4　接触判定と接触力

### 物理エンジンは物理の問題を解いてくれるものではない

力学の練習問題，例えば「時刻$t$=0に鉛直上方向に初速度10m/sで投げ上げた質点の運動を求めよ」といった時に求められる答えは質点の位置の数値ではなく，時々刻々の質点の位置を求める式「$y=10t-\frac{1}{2}gt^2$」が答えです．これに対して物理エンジンが求めてくれるのは，「0, 1.8, 3.2, 4.2, 4.9, 5.1, 4.9, 4.4, 3.5, 2.1, 0.4, −1.7」といった数値です（**図5**）．コンピュータグラフィクスで映像にするときに必要なのは，時々刻々の位置・姿勢の数値ですので，数値が求まれば充分です．

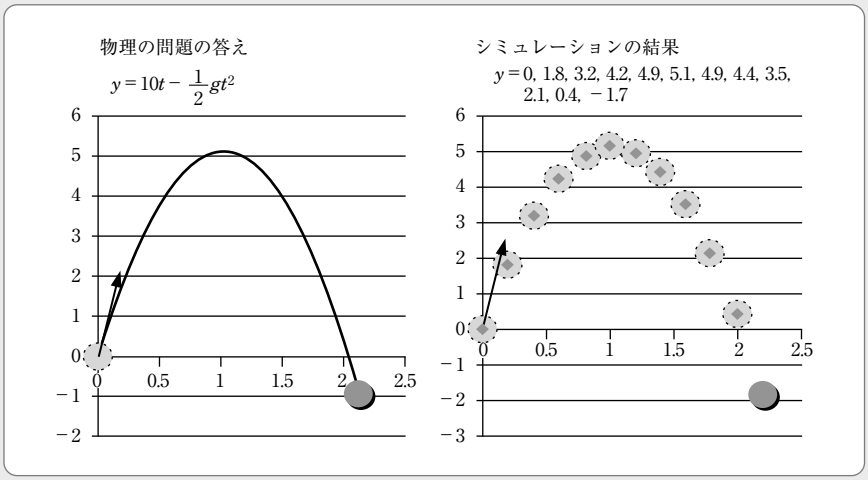


図5　物理の問題とシミュレーション

### 物理エンジンを使うと，運動方程式を立てなくてよい

機構の動作をシミュレーションする場合は，まずラグランジュの運動方程式など，その機構の運動を表す運動方程式を立て，次にその運動方程式を数値積分するという手順を思い浮かべる方が多いと思います．実際には，物理エンジンは普通，式を立てる部分も自動化してくれます．

物理エンジンには，質量・慣性テンソルといった動力学のパラメータや位置・姿勢といった初期値の情報に加え，物体の形や複数の剛体がどのように関節やバネ・ダンパなどでつながっているのかといった情報を与えます．物理エンジンは，後者の情報を用いて運動方程式を作り，自動的にシミュレーションしてくれます（**図6**）．内部的にラグランジュの運動方程式を求めるような物理エンジンもありますが，普通は関節を通じて剛体間に働く力を求め，これを各剛体の運動方程式に代入し，剛体毎に運動をシミュレーションします．

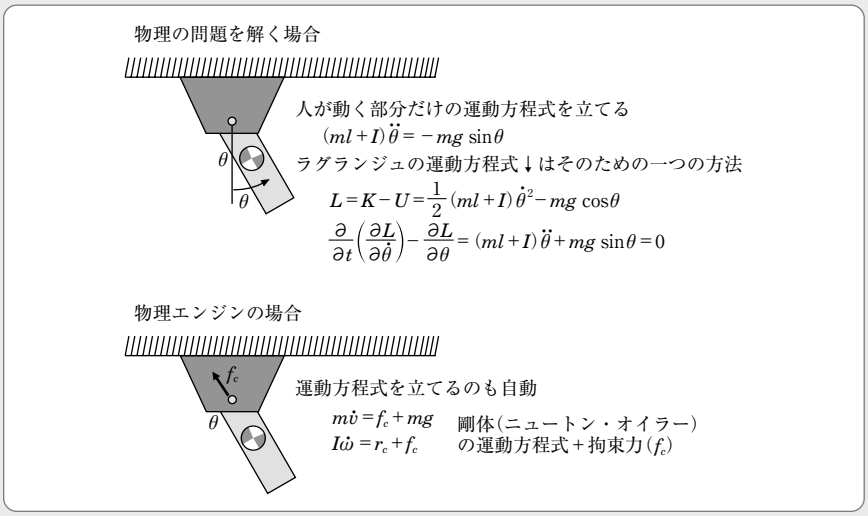


図6　運動方程式を誰が立てるか

### 3次元の回転は簡単ではない

3次元の位置は速度の積分で，どちらも3次元ベクトルで表せます．一方，3次元の角速度と姿勢（向き）の関係は簡単ではありません．角速度は3次元ベクトルで表すことができます．ベクトルの向きは回転軸の向きを表し，長さはその軸周りの角速度を表します．また，ベクトルの $x, y, z$ 成分は，それぞれ $x$ 軸，$y$ 軸，$z$ 軸周りの角速度を表します．

角速度を積分すると3次元ベクトルが得られますが，これは向きを表すものにはなりません．3次元の回転では時間が経つと回転軸が変化してしまうことがありますが，角速度を単純に積分するとその変化を無視して足しこんでしまうことになるのです（**図7**）（回転軸がずっと変化しなければ，積分した3次元ベクトルは回転の角度になります．角速度の向きが一定ならば，回転軸の向きも変わりませんので，積分できます）．☞

☞　角速度を積分するための一つの方法は，姿勢を3次元の回転行列で表すことにし，角速度も回転行列の形に変換し，足しこむ代わりに回転行列の合成を繰り返す方法です．この方法は，合成（＝行列の掛け算）を繰り返すうちに，回転行列が正規直交行列でなくなってきてしまうので，時々正規化をしなければなりません．クォータニオンと呼ばれる4変数で姿勢を表す方法を用いると，回転の合成も正規化も計算が簡単になるため，物理エンジンでは，クォータニオンで物体の姿勢を表すことが多いです（**図8**）．

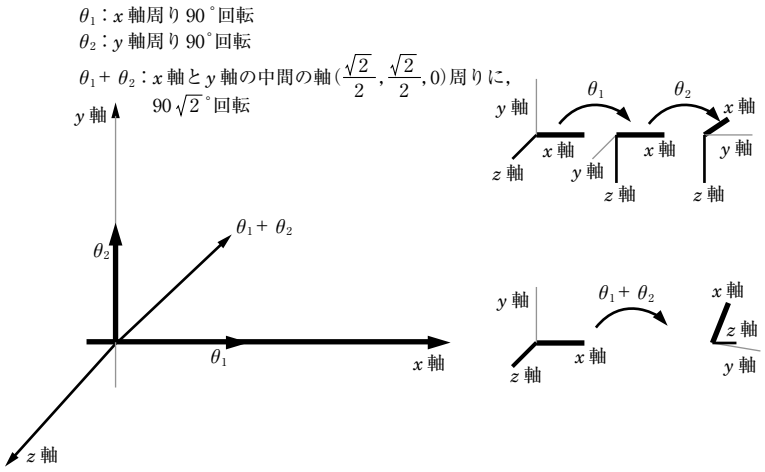


図7　回転は足せない

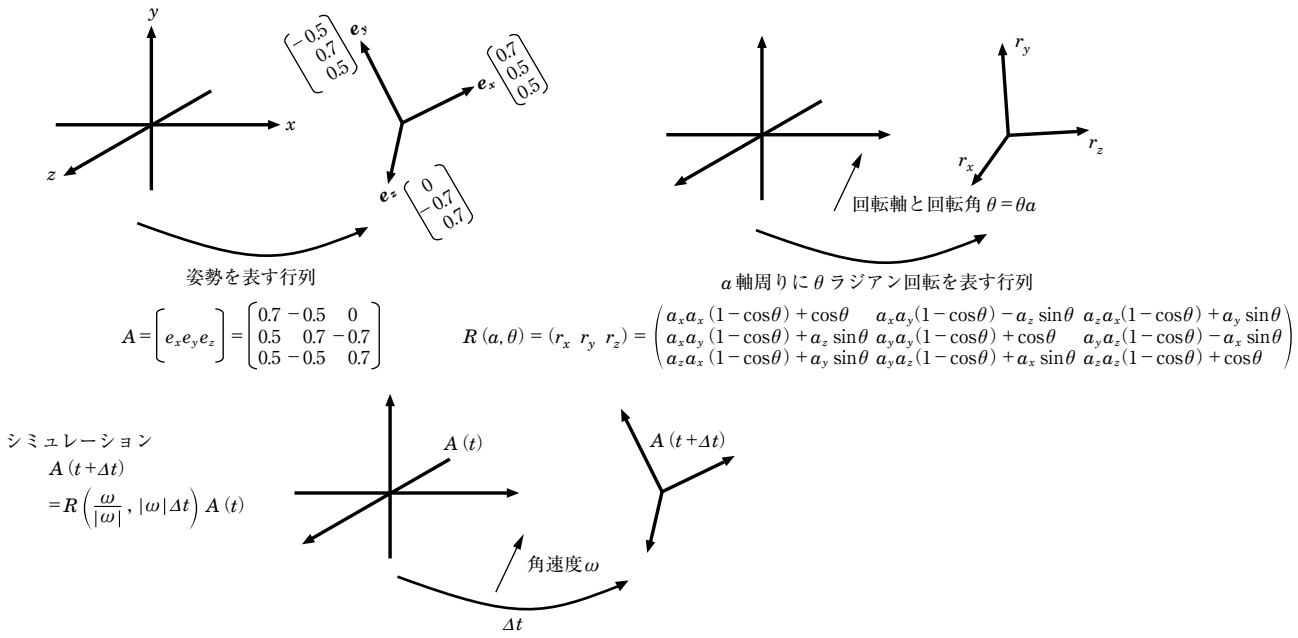


図8　行列による回転のシミュレーション

## 剛体以外の物理エンジン

髪の毛や衣服などの糸や布状の物，生き物の体やゴムやゼリーのような柔軟物，水や油のような流体など，実世界には剛体以外にもさまざまな物があります．これらの動きも物理エンジンの対象です．糸状のものは鎖だと考えれば剛体の輪や剛体と関節の組合せでも表現できますが，糸専用にプログラムを書いたほうが効率が良いため，専用のアルゴリズムでシミュレーションすることが多いです．布も同様です．糸や布は，折れ曲がりにくさを与えるためのバネ・ダンパと，長さがかわらないという制約条件で結ばれた質点の集まりと考えてシミュレーションすることが多いです．

ゴムやゼリーのような柔軟物のシミュレーションでは，有限要素法やマス・バネ・ダンパ（MSD）法といった手法が知られています．有限要素法は，まず物体を連続体だと考え，物体内部の微小な要素の運動と周りの要素から受ける力を考える連続体力学に物体の変形と運動が従うと考えます．次に，物体をある程度の小ささの有限個の要素（四面体など）に区切り，要素毎に連続体の運動方程式を積分します．☞

☞ これにより，要素の頂点（節点）についての運動方程式が求まりますので，これをシミュレーションして節点の運動を求めます．一方，マス・バネ・ダンパ（MSD）法は，物体をバネ・ダンパでつながれた質点だと考えて，質点の運動をシミュレーションします（**図9**）．

流体は流れてしまうため，小さな要素に分けて考えた場合，隣の要素がどんどん入れ替わってしまいます．そのため，自由に動き回れる粒子で流体を表す粒子法や，空間を細かく区切って各区画の流体の出入りを追う格子法でシミュレーションします（**図10**）．粒子法では，一つの粒子がある量の流体を表します．粒子はその量の流体の質量を持ち，粒子間には流体内で流体同士に働く力をまとめた力が働くようにします．粒子の運動をシミュレーションすれば，流体の動きを追うことができます．コンピュータグラフィクスで映像にするためには，各粒子からある距離の場所に面を張るなどの方法で，流体の表面形状を作る必要があります．格子法では，格子内の流体の速度を運動方程式に基づいて更新するとともに，格子間の流体の移動量を速度を用いて計算します．流体の表面は格子内の流体の量が一定値になるところに面を張ることで作ることができます．

（2012年3月15日受付）

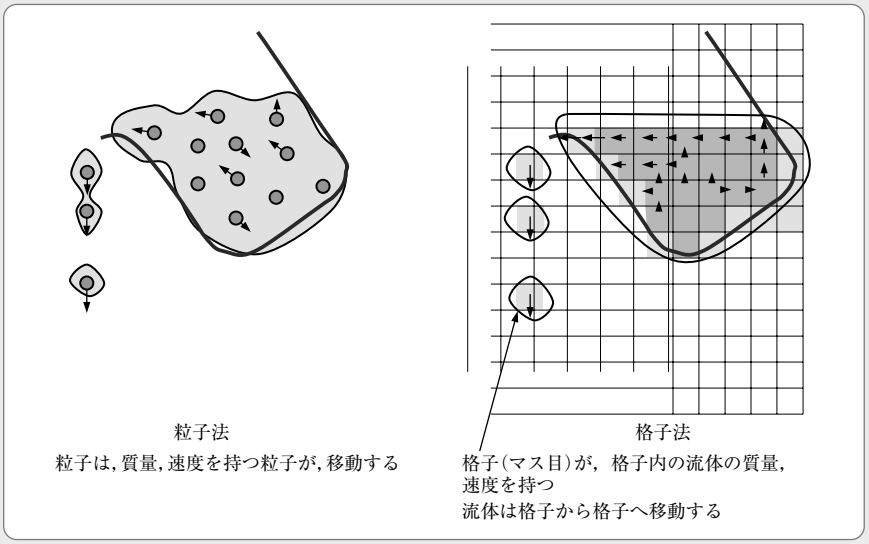


図10　粒子法と格子法

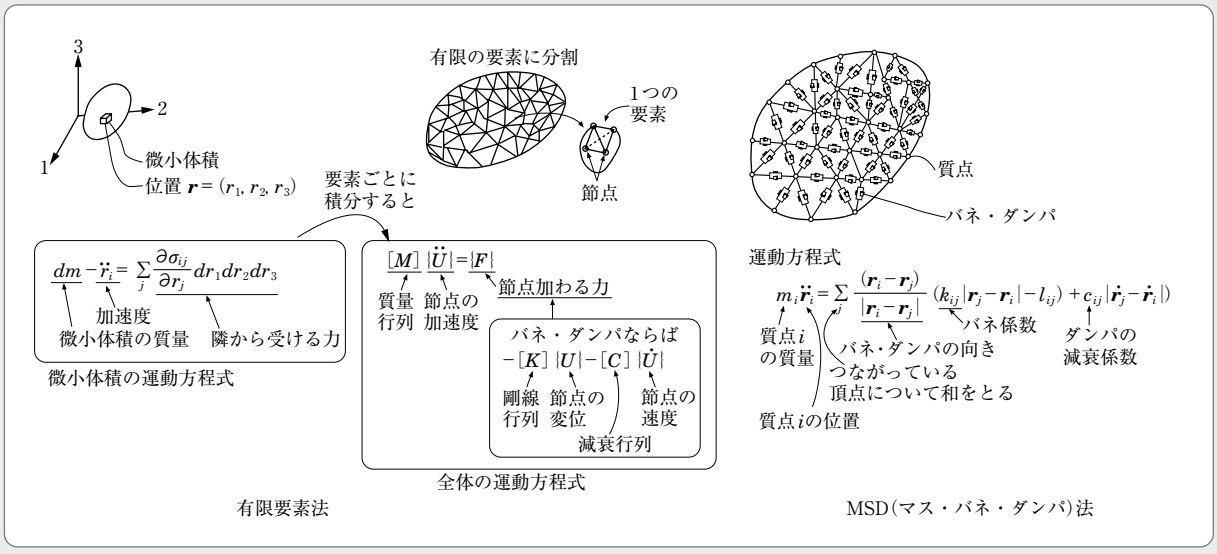


図9　有限要素法とMSD法

長谷川晶一（はせがわしょういち）　1997年，東京工業大学工学部電気電子工学科卒業．1999年，同大学大学院知能システム科学専攻修士修了．同年，ソニー（株）入社．2000年，東京工業大学精密工学研究所助手．2007年，電気通信大学知能機械科准教授．2010年，東京工業大学精密工学研究所准教授現在に至る．