

SPH (Smoothed Particle Hydrodynamics)法に関する 調査報告

吾妻広夫*

Advanced Algorithm & Systems

November 22, 2011

Abstract

本報告書では、SPH (Smoothed Particle Hydrodynamics) 法と呼ばれる流体力学の数値シミュレーション方法について、その概要を説明する。SPH 法は、シミュレーション計算の対象となる流体の系を、離散化された空間格子ではなく、多数の仮想的な粒子で表現する。これにより、金属の熱による融解等の系の大規模な変形を、比較的容易に再現することが可能である。特に、系が粘性の高い非ニュートン流体と粘性の小さなニュートン流体との間を、相転移しながら時間発展するような場合、SPH 法は有効なシミュレーション方法であると考えられている。なお、本報告書は、文献 [1, 2] を参考にして書かれている。

1 基礎方程式およびアルゴリズムの概要

本報告書では、計算流体力学シミュレーション方法の一つである、SPH (Smoothed Particle Hydrodynamics) 法の概要を説明する。SPH 法は、シミュレーションの対象となる流体を、空間格子ではなく、多数の仮想的な粒子で表現する点に特徴がある。そのため、空間格子では表現しにくかった金属の熱融解等の系の大規模変形を、SPH 法では、比較的容易に再現することが可能である。

従来の三次元空間格子を利用した流体力学的な解析と、仮想粒子を使って材料の変形を表現するSPH 法の違いを、模式的に示したのが図1、2、3 である。三次元空間格子上の各点で与えられるベクトル場で流体の時間発展を記述する場合、空間格子の構造そのものは不変なので、系を構成する金属等の物質の変形を表現するのは、基本的に無理がある。一方、多数の仮想粒子で系を記述する場合、仮想粒子を空間内で自由に移動させることが可能で、これにより物質の変形を柔軟に表現することが出来る。

この節では、SPH 法の出発点となる連立偏微分方程式、SPH 法に特有の仮想粒子の設定方法、および、SPH 法を実現するためのアルゴリズムについて説明する。

まず、系を記述する連立偏微分方程式について説明する。簡単のため、本報告書では、一種類の非ニュートン流体から成る系について考察することにする。流体は、以下の

* Email: h-azuma@asri.jp

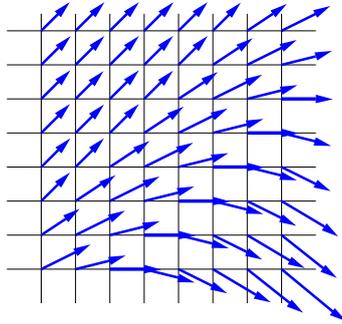


Figure 1: 従来の計算流体力学では、三次元空間格子上的各点で定義された流れの場 (ベクトル場) を解析する手法を採用している。

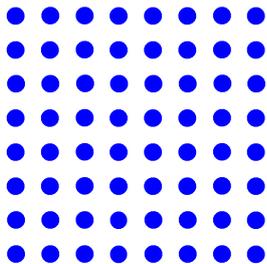


Figure 2: SPH 法では、初期状態において仮想的な粒子を配置する。

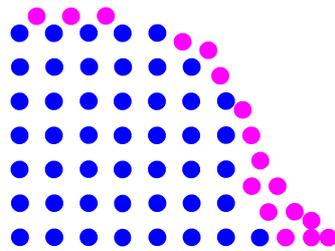


Figure 3: SPH 法においては、材料物質の変形は、仮想的粒子の移動で表現される。

Navier-Stokes 方程式に従うと仮定する。

$$\begin{aligned}\frac{d}{dt}\rho &= -\rho \sum_{i \in \{x,y,z\}} \frac{\partial v_i}{\partial x_i}, \\ \frac{d}{dt}v_i &= -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{1}{\rho} \sum_{j \in \{x,y,z\}} \frac{\partial S_{ij}}{\partial x_j} + g_i \quad \text{for } i \in \{x,y,z\}.\end{aligned}\quad (1)$$

ただし、上式は、Lagrangian specification で記述されていて、 $d/dt = (\partial/\partial t) + \sum_{i \in \{x,y,z\}} v_i (\partial/\partial x_i)$ とする。時間変数は t 、三次元空間直交座標は $x = (x, y, z)$ で与えられるとする。流体の密度は ρ 、速度は $\boldsymbol{v} = (v_x, v_y, v_z)$ 、圧力は p で与えられるとする。さらに、流体の応力テンソル (stress tensor) は $\boldsymbol{S} = S_{ij}$ for $i, j \in \{x, y, z\}$ で与えられるとする。なお、応力テンソル \boldsymbol{S} は、後で説明するが、対称テンソルである。また、重力加速度は $\boldsymbol{g} = (g_x, g_y, g_z)$ で与えられるとする。

流体の運動方程式は、式 (1) の Navier-Stokes 方程式で与えられるとした。ここで、さらに、流体に温度パラメータを与え、その熱伝導を考えることにする。流体の熱伝導方程式は、以下の式で与えられることにする。

$$\frac{d}{dt}T = k\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)T. \quad (2)$$

ただし、 $T = T(x, y, z, t)$ は、時刻 t における空間座標 (x, y, z) での流体の温度を表しているとする。

式 (1) における応力テンソル \boldsymbol{S} は、考察の対象となる流体が Newton 的か否か、さらには流体の粘性がどのような性質かを定める重要な物理量である。ここでは、応力テンソル \boldsymbol{S} を、以下のような手順で与えることにする。まず、歪みテンソル (deformation tensor) $\boldsymbol{D} = D_{ij}$ for $i, j \in \{x, y, z\}$ を以下のように定義する。

$$D_{ij} = \frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}, \quad (3)$$

上式より、歪みテンソル \boldsymbol{D} は、明らかに対称テンソルであることが分かる。歪みテンソル \boldsymbol{D} を使って、応力テンソル \boldsymbol{S} を、次のように定める [3]。

$$\boldsymbol{S} = \eta(\boldsymbol{D})\boldsymbol{D}, \quad (4)$$

$$D = \frac{1}{\sqrt{2}}|\text{trace}(\boldsymbol{D})|, \quad (5)$$

$$\eta(D) = (1 - e^{-[J(T)+1]D})\left(\frac{1}{\sqrt{D}} - \frac{1}{D}\right). \quad (6)$$

上式で定義される η は、粘性率に対応する物理量である。式 (6) 中に現れる $J(T)$ は、jump number と呼ばれる量で、物体が温度 T によって、固体、粘性率の高い流体、粘性率の低い流体と変化するに従って、 η の値を変化させるためのパラメータの役割を果たす。

ここでは、なるべく簡単な取り扱いで済ませるため、パラメータ $J(T)$ を、以下の形で与えることにする。

$$J(T) = (1 - u)J_{\max} + uJ_{\min}, \quad (7)$$

$$u = \frac{T - T_{\min}}{T_{\max} - T_{\min}}, \quad (8)$$

ただし、 $0 < J_{\min} < J_{\max}$ 、 $0 < T_{\min} < T_{\max}$ を仮定する。式 (7)、(8) より、温度 T が高くなると jump number $J(T)$ は減少し、温度が低くなると $J(T)$ は増大する。これは、式 (6) において、温度が高くなると粘性率 η が減少し、温度が低くなると粘性率 η が増大することを意味する。

さらに、仮想粒子の受ける圧力は以下で与えられるとする。仮想粒子の空間座標および時間変数を (x, y, z, t) として、圧力 $p(x, y, z, t)$ および密度 $\rho(x, y, z, t)$ の間には、

$$p(x, y, z, t) = c^2[\rho(x, y, z, t) - \rho_0], \quad (9)$$

という関係が成立するとする。ただし、 c は系を構成する物質を媒体と見なした場合の音速、 ρ_0 は時刻 $t = 0$ での粒子の密度とする。

以上で、系の時間発展を決定する偏微分方程式がそろったことになる。これにより、例えば、高温熱源に接触する金属が熱伝導方程式に従って温度上昇、融点に達した部位から溶け出した粘性の高い流体が重力落下、よって物体の大規模変形が進行する、といった現象をシミュレートすることが可能となる。

次に、SPH 法特有の仮想粒子の考え方について説明を行う。SPH 法は、既に述べたように、連続的に分布する固体や流体を、仮想的な粒子の集まりで表現する。各仮想粒子は、偏微分方程式に従って、位置の移動、温度変化等の物理的な作用を受ける。連続的な固体、流体に対して定義される物理量は、多数の仮想粒子の平均値として得られる。従って、SPH 法では、仮想粒子の物理量の平均値を取る操作のための、一種の重み関数を定義しておく必要がある。そして、この重み関数の定義の仕方が、SPH 法の数値シミュレーション方法としての優劣を決定すると考えて差し支えない。

SPH 法で使われる平均操作のための重み関数は、smoothing kernel と呼ばれ、関数 $W(\mathbf{x} - \mathbf{x}', h)$ で与えられる。重み関数に含まれるパラメータ h は、局所的な物理量を得るための平均値算出の対象となる微小体積要素を特徴付ける長さである。以下に、物理量の平均値の具体的な算出方法を示す。

三次元空間座標中において定義されているスカラー場 $A(\mathbf{x})$ の、仮想粒子による平均値の算出方法は、以下の式で与えられる。

$$A(\mathbf{x}) = \sum_{n=1}^N A(\mathbf{x}_n) \frac{m}{\rho_n} W(\mathbf{x} - \mathbf{x}_n, h). \quad (10)$$

上式において、添え字 $n \in \{1, 2, \dots, N\}$ は、仮想粒子に付けられた識別番号である。なお、これより、本報告書では、仮想粒子を識別するための index は l, n を、 $x_i \in \{x, y, z\}$ のように空間直交座標成分を区別するための index は i, j を使用することにする。今の場合、仮想粒子は系全体で N 個としている。個々の仮想粒子に対して、質量 m_n 、密度 ρ_n が与えられていて、 m/ρ_n は、仮想粒子の占める体積を表現している。

式 (10) での、重み関数 (smoothing kernel) $W(\mathbf{x} - \mathbf{x}_n, h)$ の役割は、次の通りである。関数 $W(\mathbf{x} - \mathbf{x}_n, h)$ は、 n 番目の粒子が、空間座標 \mathbf{x} を中心とする半径 h の球の内側に存在するとき、ゼロで無い値を取る。しかし、識別番号 n の粒子が、球の外側にある場合、 $W(\mathbf{x} - \mathbf{x}_n, h)$ はゼロの値を取る。これにより、式 (10) は、図 4 に示されるように、空間座標 \mathbf{x} 付近の仮想粒子について、物理量の平均を取ることが可能となる。

長さの次元を持つパラメータ h の決め方は、特に定まっていない。全計算領域の体積と全仮想粒子数 N から、適切に決める必要がある。なお、仮想粒子の濃度が疎な領域で

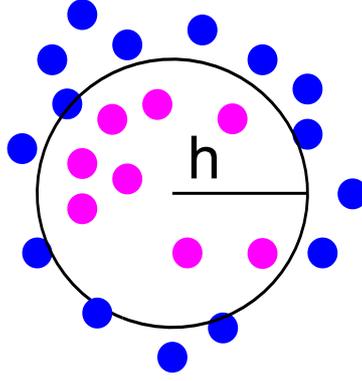


Figure 4: 重み関数 (smoothing kernel) $W(\mathbf{x} - \mathbf{x}_j, h)$ は、半径 h の球の内部の仮想粒子について、物理量を足し上げる働きを持つ。

は h を大きく、密な領域では h を小さくというように、シミュレーションの進行に合わせて h を変化させるという考え方も有り得る。そのような場合、

$$h_n = h_0 \left(\frac{\rho_0}{\rho_n} \right)^{1/3}, \quad (11)$$

とするのが一般的である。

なお、スカラー場 $A(\mathbf{x})$ を空間座標変数に関して微分した量 $\nabla A(\mathbf{x})$ は、式 (10) からの類推により、以下の式に従って平均値を計算するものとする。

$$\nabla A(\mathbf{x}) = \sum_{n=1}^N A(\mathbf{x}_n) \frac{m}{\rho_n} \nabla W(\mathbf{x} - \mathbf{x}_n, h). \quad (12)$$

前の説明では、重み関数 (smoothing kernel) $W(\mathbf{x} - \mathbf{x}_n, h)$ を、半径 h の球の内部と外部を区別する関数として説明した。このような、いわば階段型の関数は、領域内の仮想粒子数が十分多く、かつ、半径 h を十分小さく取れる極限では、正しい結果を与えるが、そうでない場合には、良い計算精度が得られない可能性がある。そこで、実際の数値シミュレーション計算においては、様々な工夫を凝らした smoothing kernel が使われることになる。文献 [4] では、以下の形の $W(\mathbf{x} - \mathbf{x}', h)$ を使用することが推奨されている。

$$W(\mathbf{x} - \mathbf{x}', h) = \frac{1}{\pi h^3} \begin{cases} 1 - (3/2)q^2 + (3/4)q^3 & \text{for } 0 \leq q < 1, \\ (1/4)(2 - q)^3 & \text{for } 1 \leq q < 2, \\ 0 & \text{for } 2 \leq q. \end{cases} \quad (13)$$

ただし、上式では、 $q = |\mathbf{x} - \mathbf{x}'|/h$ としている。

式 (13) で与えられる重み関数 (smoothing kernel) $W(\mathbf{x} - \mathbf{x}', h)$ が、具体的にはどのような形の関数かを調べたのが図 5 である。なお、式 (13) で定義される $W(\mathbf{x} - \mathbf{x}', h)$ は規格化されており、

$$\int_{-2h}^{2h} dx \int_{-2h}^{2h} dy \int_{-2h}^{2h} dz W(\mathbf{x}, h) = 1 \quad \text{for } 0 < \forall h < +\infty, \quad (14)$$

が成立する。

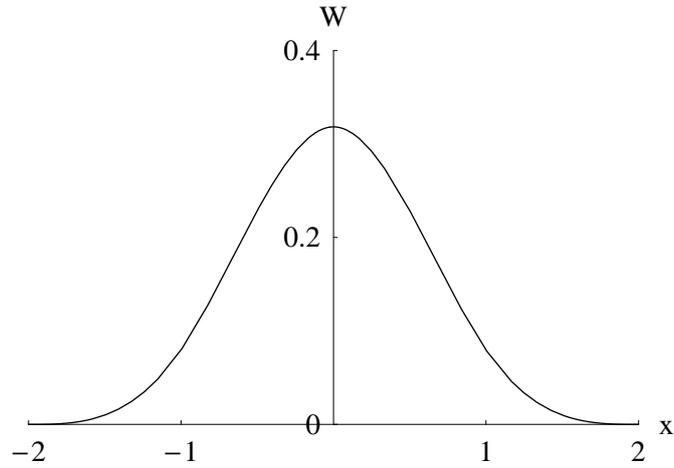


Figure 5: 式 (13) で与えられる重み関数 (smoothing kernel) $W(x, 1)$ を、 $-2 \leq x \leq 2$ の範囲でプロットしたグラフである。

Algorithm 1 SPH method

```

repeat
  for  $n = 1$  to  $N$  do
    Update  $(d\rho_n/dt)$  for the  $n$ -th particle
    Update  $(d\mathbf{v}_n/dt)$  for the  $n$ -th particle
    Update  $(dT_n/dt)$  for the  $n$ -th particle
  end for
  for  $n = 1$  to  $N$  do
    Update  $\mathbf{x}_n$  for the  $n$ -th particle
    Update  $\mathbf{v}_n$  for the  $n$ -th particle
    Update  $T_n$  for the  $n$ -th particle
    Update  $\rho_n$  for the  $n$ -th particle
    Update  $p_n$  for the  $n$ -th particle
    Update  $\eta(D)_n$  for the  $n$ -th particle
  end for
  Update particle neighbours
  Update  $\Delta t$ 
   $time = time + \Delta t$ 
until  $time < time_{\text{end}}$ 

```

Table 1: 仮想粒子 $n(\in \{1, 2, \dots, N\})$ の保持する局所的 (local) な物理量

変数 (記号)	物理量としての意味
\mathbf{x}_n	空間座標
\mathbf{v}_n	速度ベクトル
$\mathbf{a}_n (= d\mathbf{v}_n/dt)$	加速度ベクトル
\mathbf{D}_n	歪みテンソル
ρ_n	密度
η_n	粘性率
T_n	温度

Table 2: 系全体として保持する大域的 (global) な物理量

変数 (記号)	物理量としての意味
h	the smoothing length
m	(仮想粒子の) 質量
c	音速

次に、SPH法を実行するための、大まかな手順を、アルゴリズム1に示しておく。アルゴリズム1中での、個々の命令文の実行の詳細については、次の節以降に説明される。

アルゴリズム1を実行する際、 N 個の仮想粒子が初期状態において設定される。識別番号 $n(\in \{1, 2, \dots, N\})$ の仮想粒子が保持する局所的 (local) な物理量は、表1の通りである。一方、系全体として保持する大域的 (global) な物理量は、表2の通りである。これら表2に含まれる大域的な物理量は、基本的に時間変化しないと考える。しかし、前に述べたように、例えば式(11)を使って h を ρ_n に依存して変化させる場合は、 h_n は各仮想粒子の保持すべき局所的な量となる。

実際のプログラミング作業では、表1に示された属性を保持する仮想粒子を、一まとまりのクラスまたはオブジェクトと見なして、オブジェクト指向でコーディングすることが可能である。また、オブジェクト指向とは行かないまでも、構造体の機能を利用して、C言語等の古い言語であっても、見通しの良いソースコードを作成することが可能である。Fortran90等でも、このようなコーディングは可能である。

先に示したアルゴリズムでは、各仮想粒子について計算される物理量の微小な時間当たりの変位 $\{(d\rho_n/dt), (d\mathbf{v}_n/dt), (dT_n/dt) : n \in \{1, 2, \dots, N\}\}$ から、各時刻での物理量 $\{\mathbf{x}_n, \mathbf{v}_n, T_n, \rho_n, p_n, \eta(D)_n : n \in \{1, 2, \dots, N\}\}$ を算出する必要がある。それには、系を支配する連立偏微分方程式を基にした各物理量の差分表現が必要である。通常、SPH法においては、leapfrog法による差分表現が使われる場合が多い。以下に、必要な差分表現をまとめておく。

時刻 $t = t_0$ から計算を開始し、時間ステップ幅 Δt で時間発展を追うとする。まず、次

の物理量を計算する。

$$\begin{cases} \rho_n(t_0 + (\Delta t/2)) &= \rho_n(t_0) + (\Delta t/2)(d\rho_n/dt)|_{t=t_0}, \\ \mathbf{v}_n(t_0 + (\Delta t/2)) &= \mathbf{v}_n(t_0) + (\Delta t/2)(d\mathbf{v}_n/dt)|_{t=t_0}, \\ \mathbf{x}_n(t_0 + (\Delta t/2)) &= \mathbf{x}_n(t_0) + (\Delta t/2)\mathbf{v}_n(t_0), \\ T_n(t_0 + (\Delta t/2)) &= T_n(t_0) + (\Delta t/2)(dT_n/dt)|_{t=t_0}. \end{cases} \quad (15)$$

次回ステップ以降、 $t = t_0 + n\Delta t + (\Delta t/2)$ for $n = 1, 2, 3, \dots$ での、 ρ_n 、 \mathbf{v}_n 、 T_n の値は、次の式で計算する。

$$\begin{cases} \rho_n(t + (\Delta t/2)) &= \rho_n(t - (\Delta t/2)) + (\Delta t/2)(d\rho_n/dt)|_{t-\Delta t}, \\ \mathbf{v}_n(t + (\Delta t/2)) &= \mathbf{v}_n(t - (\Delta t/2)) + (\Delta t/2)(d\mathbf{v}_n/dt)|_{t-\Delta t}, \\ T_n(t + (\Delta t/2)) &= T_n(t - (\Delta t/2)) + (\Delta t/2)(dT_n/dt)|_{t-\Delta t}. \end{cases} \quad (16)$$

そして、 $t = t_0 + n\Delta t$ for $n = 1, 2, 3, \dots$ での、 ρ_n 、 \mathbf{v}_n 、 \mathbf{x}_n 、 T_n の値は、次の式で計算する。

$$\begin{cases} \rho_n(t + (\Delta t/2)) &= \rho_n(t - (\Delta t/2)) + \Delta t(d\rho_n/dt)|_t, \\ \mathbf{v}_n(t + (\Delta t/2)) &= \mathbf{v}_n(t - (\Delta t/2)) + \Delta t(d\mathbf{v}_n/dt)|_t, \\ \mathbf{x}_n(t + (\Delta t/2)) &= \mathbf{x}_n(t - (\Delta t/2)) + \Delta t\mathbf{v}_n(t), \\ T_n(t + (\Delta t/2)) &= T_n(t - (\Delta t/2)) + \Delta t(dT_n/dt)|_t. \end{cases} \quad (17)$$

なお、時刻 $t = t_0 + n\Delta t$ for $n = 1, 2, 3, \dots$ での、 D_n 、 η_n は、時刻 t での $\{\mathbf{v}_n(t) : n \in \{1, 2, \dots, N\}\}$ および $\{T_n(t) : n \in \{1, 2, \dots, N\}\}$ より算出可能である。

2 密度 ρ_n 、速度 \mathbf{v}_n 、温度 T_n の時間微分の求め方

第1節で示したアルゴリズムおよび leapfrog 法による差分方程式によって、各物理量の時間発展を追跡するには、各時刻において、 N 個の仮想粒子の保持する $d\rho_n/dt$ 、 $d\mathbf{v}_n/dt$ 、 dT_n/dt を計算する必要がある。この節では、これらの時間微分量を計算する方法について説明する。

これより先の議論では、次の計算テクニックがしばしば使われる。ある物理量に微分演算子 ∇ を作用させる場合、重み関数 (smoothing kernel) $W(\mathbf{x} - \mathbf{x}', h)$ 特有の性質より、以下に示す事情が発生する。まず、定数 1 に微分演算子 ∇ を作用させることを考える。これは、重み関数 $W(\mathbf{x} - \mathbf{x}', h)$ を使って、次のように表現される。

$$\begin{aligned} \nabla 1 &= \nabla \sum_{n=1}^N \frac{m}{\rho_n} W(\mathbf{x} - \mathbf{x}_n, h) \\ &= \sum_{n=1}^N \frac{m}{\rho_n} \nabla W(\mathbf{x} - \mathbf{x}_n, h). \end{aligned} \quad (18)$$

しかし、上式の左辺は明らかにゼロなので、次の公式を得る。

$$\sum_{n=1}^N \frac{m}{\rho_n} \nabla W(\mathbf{x} - \mathbf{x}_n, h) = 0. \quad (19)$$

これより先の計算では、しばしば式 (19) を使う。

これより、各仮想粒子の保持する時間微分された物理量の評価方法について考える。まず、 $d\rho_n/dt$ for $n = 1, 2, 3, \dots, N$ の評価方法について考察する。時刻 t における識別番号 n の仮想粒子の密度 $\rho_n(t)$ を、次のように定義する。

$$\rho_n(t) = \sum_{l=1}^N mW(\mathbf{x}_n - \mathbf{x}_l, h). \quad (20)$$

式 (20) による $\rho_n(t)$ の定義は、前に説明したスカラー場 $A(\mathbf{x})$ の局所平均値操作である式 (12) と合致している。

式 (20) より、時刻 t での各仮想粒子の密度 $\rho_n(t)$ が決定される。従って、 $d\rho_n/dt$ は、式 (1) を使って次のように書き下せる。

$$\begin{aligned} \frac{d}{dt}\rho_n(t) &= -\rho_n(t)\nabla \cdot \mathbf{v}_n(t) \\ &= -\rho_n(t)\nabla_n \cdot \sum_{l=1}^N \frac{m}{\rho_l(t)} \mathbf{v}_l(t) W(\mathbf{x}_n - \mathbf{x}_l, h) \\ &= -\rho_n(t) \sum_{l=1}^N \frac{m}{\rho_l(t)} \mathbf{v}_l(t) \cdot \nabla_n W(\mathbf{x}_n - \mathbf{x}_l, h). \end{aligned} \quad (21)$$

上式の右辺に、先に導出した公式 (19) を加えて、以下を得る。

$$\frac{d}{dt}\rho_n(t) = -\rho_n(t) \sum_{l=1}^N \frac{m}{\rho_l(t)} (\mathbf{v}_l(t) - \mathbf{v}_n(t)) \cdot \nabla_n W(\mathbf{x}_n - \mathbf{x}_l, h). \quad (22)$$

上の関係式を使って、時刻 t での仮想粒子 n の保持する $d\rho_n/dt$ が算出できる。

次に、 dv_n/dt for $n = 1, 2, 3, \dots, N$ の計算方法について考察する。まず、式 (1) より、 dv_n/dt を以下のように書き下す。

$$\frac{d}{dt}v_{i,n}(t) = -\frac{1}{\rho_n(t)} \frac{\partial p_n}{\partial x_i} + \frac{1}{\rho_n(t)} \sum_{j \in \{x,y,z\}} \frac{\partial S_{ij,n}}{\partial x_j} + g_i. \quad (23)$$

そこで、 $-(1/\rho_n)(\partial p_n/\partial x_i)$ および $(1/\rho_n)(\partial S_{ij,n}/\partial x_j)$ の評価方法について考える。

まず、 $-(1/\rho_n)(\partial p_n/\partial x_i)$ については、あらかじめ、次のように式変形しておくことが有効である。

$$-\frac{1}{\rho_n(t)} \frac{\partial p_n}{\partial x_i} = -\frac{\partial}{\partial x_i} \left(\frac{p_n}{\rho_n} \right) - \frac{p_n(t)}{\rho_n(t)^2} \frac{\partial \rho_n}{\partial x_i}. \quad (24)$$

上の式を、重み関数 $W(\mathbf{x} - \mathbf{x}', h)$ を使って書き下すと以下を得る。

$$\begin{aligned} -\frac{1}{\rho_n(t)} \frac{\partial p_n}{\partial x_i} &= -\frac{\partial}{\partial x_{i,n}} \sum_{l=1}^N \frac{m}{\rho_l(t)} \frac{p_l(t)}{\rho_l(t)} W(\mathbf{x}_n - \mathbf{x}_l, h) \\ &\quad - \frac{p_n(t)}{\rho_n(t)^2} \frac{\partial}{\partial x_{i,n}} \sum_{l=1}^N \frac{m}{\rho_l(t)} \rho_l(t) W(\mathbf{x}_n - \mathbf{x}_l, h) \\ &= -\sum_{l=1}^N m \left(\frac{p_l(t)}{\rho_l(t)^2} + \frac{p_n(t)}{\rho_n(t)^2} \right) \frac{\partial W(\mathbf{x}_n - \mathbf{x}_l, h)}{\partial x_{i,n}}. \end{aligned} \quad (25)$$

また、 $(1/\rho_n)(\partial S_{ij,n}/\partial x_j)$ については、以下のように計算する。

$$\begin{aligned} \frac{1}{\rho_n(t)} \frac{\partial S_{ij,n}}{\partial x_j} &= \frac{1}{\rho_n(t)} \frac{\partial}{\partial x_{j,n}} \sum_{l=1}^N \frac{m}{\rho_l} S_{ij,l} W(\mathbf{x}_n - \mathbf{x}_l, h) \\ &= \sum_{l=1}^N \frac{m S_{ij,l}(t)}{\rho_n(t) \rho_l(t)} \frac{\partial W(\mathbf{x}_n - \mathbf{x}_l, h)}{\partial x_{j,n}}. \end{aligned} \quad (26)$$

ところで、式 (19) より、

$$\sum_{l=1}^N \frac{m S_{ij,n}(t)}{\rho_n(t) \rho_l(t)} \frac{\partial W(\mathbf{x}_n - \mathbf{x}_l, h)}{\partial x_{j,n}} = 0, \quad (27)$$

を得る。そこで、式 (26) の右辺に式 (27) を加えて、

$$\frac{1}{\rho_n(t)} \frac{\partial S_{ij,n}}{\partial x_j} = \sum_{l=1}^N \frac{m}{\rho_n(t) \rho_l(t)} [S_{ij,n}(t) + S_{ij,l}(t)] \frac{\partial W(\mathbf{x}_n - \mathbf{x}_l, h)}{\partial x_{j,n}}, \quad (28)$$

を得る。

以上で、式 (23)、(25)、(28) により、 $d\mathbf{v}_n/dt$ の評価方法が得られたことになるのだが、数値計算の安定性を確保するため、以下の補正を行うのが良いことが、経験的に知られている。

$$\frac{d\mathbf{v}_n}{dt} \longrightarrow \frac{d\mathbf{v}_n}{dt} - \sum_{l=1}^N \frac{m}{\rho_n(t)} \Pi_{nl} \nabla_n W(\mathbf{x}_n - \mathbf{x}_l, h), \quad (29)$$

ただし、

$$\Pi_{nl} = \begin{cases} (1/2)(-\alpha\mu_{nl} + \beta\mu_{nl}^2)/(\rho_n + \rho_l) & \text{for } (\mathbf{v}_n - \mathbf{v}_l) \cdot (\mathbf{x}_n - \mathbf{x}_l) < 0, \\ 0 & \text{for } (\mathbf{v}_n - \mathbf{v}_l) \cdot (\mathbf{x}_n - \mathbf{x}_l) \geq 0, \end{cases} \quad (30)$$

$$\mu_{nl} = \frac{h(\mathbf{v}_n - \mathbf{v}_l) \cdot (\mathbf{x}_n - \mathbf{x}_l)}{|\mathbf{x}_n - \mathbf{x}_l|^2 + (h^2/100)}, \quad (31)$$

とする。

最後に、 dT_n/dt for $n = 1, 2, 3, \dots, N$ の計算方法について考察する。まず初めに、式 (2) より、

$$\frac{dT_n}{dt}(t) = k \sum_{l=1}^N \frac{m}{\rho_l(t)} T_l(t) \nabla^2 W(\mathbf{x}_n - \mathbf{x}_l, h), \quad (32)$$

を得る。これを、式 (19) のテクニックを使って、以下のように書き換える。

$$\frac{dT_n}{dt}(t) = k \sum_{l=1}^N \frac{m}{\rho_l(t)} [T_l(t) - T_n(t)] \nabla_n^2 W(\mathbf{x}_n - \mathbf{x}_l, h). \quad (33)$$

これが、SPH 法における一般的な dT_n/dt の求め方なのだが、実は、上式を使うと、系全体の熱量が保存しない傾向が有ることが知られている。そこで、以下の補正を行うと良いことが経験的に知られている [5]。

$$\frac{dT_n}{dt}(t) = k \sum_{l=1}^N \frac{m}{\rho_l(t)} \left[\frac{4\rho_n(t)}{\rho_n(t) + \rho_l(t)} \right] [T_n(t) - T_l(t)] \frac{(\mathbf{x}_n - \mathbf{x}_l) \cdot \nabla_n W(\mathbf{x}_n - \mathbf{x}_l, h)}{|\mathbf{x}_n - \mathbf{x}_l|^2 + (1/100)h^2}. \quad (34)$$

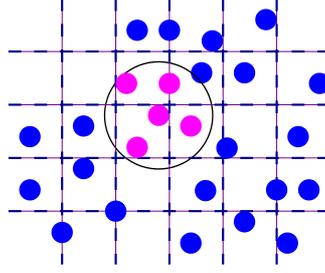


Figure 6: 三次元空間領域を、あらかじめ一辺の長さが h 程度の立方体ブロックに分割し、各ブロックに含まれる仮想粒子を登録しておいて、隣接粒子の検索に役立てる。これにより、計算時間の短縮が期待できる。

これまでの議論では、アルゴリズムを時間ステップ毎に進めるに際して、その時間刻み幅 Δt を一定と、暗に仮定していた。しかし、実際の数値計算では、系の時間発展の挙動が安定するように、以下のように Δt を補正しつつ、時間ステップを進めた方が良いことが、経験的に知られている。

$$\Delta t = \frac{1}{10} \min \left[\frac{h}{|\mathbf{v}_{\max}| + c}, \frac{h^2}{6\eta_{\max}} \right]. \quad (35)$$

さらに、実際のシミュレーション計算において、以下の点に注意を払うべきである。アルゴリズム 1 を実行する際、最も計算時間を消費するのは、各仮想粒子の隣接粒子を探す、Update particle neighbours の部分である。そこで、この部分の計算時間を抑える工夫として、図 6 のように、三次元空間の領域を一辺の長さが h 程度の立方体ブロックに分割し、あらかじめ各ブロックに所属する仮想粒子を登録しておくことにする。これにより、隣接粒子の検索に消費される時間を、効果的に短縮することが可能となる。

なお、仮想粒子間の相互浸透 (inter-penetration) 効果を取り入れるための補正として、以下の式が使われることがある。

$$\mathbf{v}_n \longrightarrow \mathbf{v}_n + \epsilon \sum_{l=1}^N \frac{m}{(1/2)[\rho_n(t) + \rho_l(t)]} (\mathbf{v}_l - \mathbf{v}_n) W(\mathbf{x}_n - \mathbf{x}_l, h) \quad \text{for } \forall \epsilon \in [0, 1]. \quad (36)$$

3 重み関数 $W(\mathbf{x} - \mathbf{x}', h)$ の微分公式

数値計算の過程で、重み関数 $W(\mathbf{x} - \mathbf{x}', h)$ の微分を計算する必要がある。そこで、ここに、 $W(\mathbf{x} - \mathbf{x}', h)$ の微分の公式をまとめておく。

$$\frac{\partial}{\partial x_i} W(\mathbf{x} - \mathbf{x}', h) = -\frac{3x_i}{\pi h^5 q} \begin{cases} q[1 - (3/4)q] & \text{for } 0 \leq q < 1, \\ (1/4)(2 - q)^2 & \text{for } 1 \leq q < 2, \\ 0 & \text{for } 2 \leq q, \end{cases} \quad (37)$$

$$\nabla^2 W(\mathbf{x} - \mathbf{x}', h) = -\frac{3}{\pi h^5} \begin{cases} 3(1 - q) & \text{for } 0 \leq q < 1, \\ (1/q)(2 - 3q + q^2) & \text{for } 1 \leq q < 2, \\ 0 & \text{for } 2 \leq q, \end{cases} \quad (38)$$

ただし、 $q = |\mathbf{x} - \mathbf{x}'|/h$ とする。

References

- [1] G. R. Liu and M. B. Liu, *Smoothed particle hydrodynamics: a meshfree particle method* (World Scientific, Singapore, 2003).
- [2] A. Paiva, F. Petronetto, T. Lewiner and G. Tavares, ‘Particle-based non-Newtonian fluid animation for melting objects’, Proceedings of the Sibgrapi 2006 (IEEE Press) pp. 78–85.
- [3] J. R. R. Siffert, P. R. de Souza Mendes and E. S. S. Dutra, ‘Gas displacement of viscoplastic liquids in capillary tubes’ in XII CREEM 2005, Ilha Solterea, XII Congresso Nacional de Estudantes de Engenharia Mecânica.
- [4] J. J. Monaghan and R. A. Gingold, ‘Shock simulation by the particle method SPH’, *Journal of Computational Physics*, **52**, 374–389 (1983).
- [5] S. J. Cummins and M. Rudman, ‘An SPH Projection Method’, *Journal of Computational Physics* **152**, 584–607 (1999).