

粒子法 MPS による流体力学シミュレーションを高速化する試み

粒子達を探索する領域分割法を改良する試み

服部 元史
神奈川工科大学 情報メディア学科
hattori@ic.kanagawa-it.ac.jp

桂畑 司
神奈川工科大学 情報メディア学科
katsurahata@gmail.com

キーワード: 粒子法 Moving Particle Simulation, 領域分割

1 はじめに

砕ける波や飛び散る飛沫(しぶき)など自由境界が複雑に変形する流体領域の運動を数値計算し CG Animation に可視化できる手法として、粒子法 Moving Particle Simulation が開発されている[1]. 自由境界 Navier-Stokes 偏微分方程式を数値解法する事により、圧力分布と流体運動の双方を粒子法 MPS で得ることができる. 粒子法 MPS 半陰解法は圧力勾配項の Poisson 偏微分方程式を求解する際に逆行列を解くため 計算時間が長くなるが 液体の非圧縮性を高く再現できる. 粒子法 MPS 陽解法は 質量密度から直接に陽的に圧力を計算するため 計算時間が短く 若干の圧縮性を認めるが 計算精度について深く検討されている[2].

自由境界を有する複雑な液体の変形運動を粒子法 MPS は数値計算できるが、必要とするメモリ容量や計算時間や膨大となる問題がある. 発散 divergence や勾配 gradient など空間偏微分作用素を 粒子間の相互作用として空間的に離散化する. この時に重み関数 w を使用するが、粒子間の距離が減少するにつれ相互作用の大きさが増加し 粒子間の距離が増加するにつれて相互作用の大きさが減少する ような値が重み関数 w に定められている. 粒子間の距離が一定の値よりも大きくなると、重み関数 w の値は恒等的に 0 となる. 全ての相互作用で重み関数 w の値を乗ずるので、重み関数の値が 0 となる距離にある粒子間では相互作用を計算する必要がない. 一辺の長さを Courant 条件数 0.2 に粒子間の標準距離 l_0 を掛けた長さを 重み関数の影響半径に加えた長さを一辺とする立方体を格子状に敷き詰めて解析領域を覆い尽くす. この立方体に所属する粒子と 隣接する立方体に所属する粒子のみを探索し 粒子間の相互作用を計算することで、探索する回数を削減する. この削減によって粒子法 MPS を数値計算する速度を速めて、Real Time CG Animation 描画が可能であるか否かを検討する.

2 粒子法 MPS 陽解法

粒子法 MPS では、流体領域を粒子の集まりとして捉え、粒子どうしの間の相互作用を計算することにより、流体の動きを求める. 時刻 t で位置 $\mathbf{r}$ に居る流体粒子の速度を $\mathbf{u}(\mathbf{t}, \mathbf{r})$ で表し、質圧力を $P(\mathbf{t}, \mathbf{r})$ で表し、質量密度を $\rho(\mathbf{t}, \mathbf{r})$ で表す. 流体の変形運動は

$$\frac{D\mathbf{r}}{Dt} = \mathbf{u} \quad (1)$$

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla P + \mu \nabla^2 \mathbf{v} + \mathbf{g} \quad (2)$$

で表され、流体の質量保存は

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0 \quad (3)$$

で表される. ここで、空間変数 $\mathbf{r} = (r_x, r_y, r_z)$ による空間偏微分を表す作用素として ∇ は

$$\nabla = (\partial_{r_x}, \partial_{r_y}, \partial_{r_z}) \quad (4)$$

で定義されている.

離散時刻 $k = 0, 1, 2, \dots$ を導入し、時間について式(1)を離散化して

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\Delta t} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u} + \mathbf{g} \quad (5)$$

を得る. 式(5)では圧力が未知数となる. よって圧力を求める前に、粘性項と外力項を加えた後の流速を求めて仮の位置 $\mathbf{r}_i^*$ を

$$\mathbf{r}_i^* = \mathbf{r}_i^k + [\mathbf{u}_i^k + \{\nu \nabla^2 \mathbf{u}_i^k + \mathbf{g}\} \Delta t] \Delta t \quad (6)$$

で求めておく. 式(6)における粘性項の偏微分作用素 ∇^2 は

$$\langle \nabla^2 \mathbf{u} \rangle_i^k = \frac{2d}{\lambda^0 n^0} \sum_{j \neq i} (\mathbf{u}_j^k - \mathbf{u}_i^k) w(|\mathbf{r}_j^k - \mathbf{r}_i^k|) \quad (7)$$

にて空間離散化されている.

このように仮の位置 $\mathbf{r}_i^*$ を求めた後に、非圧縮性を回復するべく 粒子の位置を $\mathbf{r}_i^{k+1}$ へと下記の式(11)によって修正する. 離散時刻 $k+1$ における圧力は

$$P_i^{k+1} = -c^2 \rho \frac{n_i^* - n^0}{n^0} \quad (8)$$

にて求める. 圧力を求めるための粒子数密度は

$$n_i^* = \sum_{j \neq i} w(|\mathbf{r}_j^* - \mathbf{r}_i^*|) \quad (9)$$

にて求める. 圧力勾配項の微分演算子 ∇ は

$$\langle \nabla P \rangle_i^{k+1} = \quad (10)$$

$$\frac{d}{n^0} \sum_{j \neq i} \left[\frac{p_j^{k+1} + p_i^{k+1}}{|r_j - r_i|^2} (r_j - r_i) w(|r_j - r_i|) \right]$$

のように空間離散化する．time step を一つ進めた後の時間における粒子の座標を

$$r_i^{k+1} = r_i^* + \left[-\frac{1}{\rho} \langle \nabla P \rangle_i^{k+1} \Delta t \right] \Delta t \quad (11)$$

で求める．以上の過程を 1time step とし，次の時刻へ進めた後に繰り返すことで流体の変形をシミュレートする．

粒子間の標準的な距離を l_0 で表したうえで，1 frame 時間刻み幅 Δt を，陽解法 MPS では以下の Courant 条件

$$u_{max} \Delta t < 0.2 l_0 \quad (8)$$

$$c \Delta t < l_0 \quad (9)$$

を共に満たすように定める．ここで液体の物理的な音速よりもかなり小さな値を c に設定する．物理的な音速に近い大きな値を c に用いると，液体の非圧縮性が高く再現されるが，1 frame の時間刻み幅 Δt を非常に小さな値にしなければならず，計算時間が半陰解法よりも長く成ることがある．音速の値はある程度小さくても十分な精度で計算できるため計算時間を抑えるために小さい値を選ぶ．

3 領域分割探索による計算時間の短縮

粒子法 MPS における空間偏微分作用素を計算するにあたり 重み関数 w を乗じている．そのため，重み関数 w が zero 0 となる粒子同士の相互作用は計算する必要が無い．この時に重み関数 w が有効となる範囲は近傍粒子のみに限定される．この近傍粒子を探索することで不要な粒子間の探索を省くことができ，計算時間を短縮することができる．

式(8)の右辺 $0.2 l_0$ を重み関数 w の影響半径に加えた値を一边の長さとする立方体を格子状に敷き詰めて解析領域全体を覆い，それぞれの立方体に所属する粒子達を求め リストで管理する．立方体の一边の長さを上記のように定める理由は，圧力を計算する過程で粒子を移動させても同じ立方体に粒子が居続けるためである．この配慮のお陰で，それぞれの粒子が所属する立方体を計算する処置を 1 time step ごとに一度だけ行えば良く成る．近傍粒子の探索は 重み関数 w の値が有効となる範囲のみを対象とするためのものである．重み関数 w が有効となる範囲が粒子の属する領域と隣接する領域のみとなるように立方体の一边の長さを定めているため，粒子が属する領域と隣接する領域のリストを網羅するだけで足りる事に成り，計算不要な粒子を探索せずに済む．

粒子番号を網羅し，粒子ごとに属する領域を求めた後に近傍粒子を網羅をする Algorithm が文献[1]で解説されている．この探索手順と比較するべく，「立方体を網羅し，立方体に所属する粒子達を網羅し，近傍粒子を網羅して行く」Algorithm を本研究で提案した．これらの Algorithm に基づいて 粒子達が移動して行く様子を Real Time に CG 描画しつつ 粒子法 MPS で数値計算を実施した．その CG 描画を Figure 1, Figure 2 に示す．

4 数値計算を実施した検証結果

近傍粒子の探索なし，既存の手順，提案する手順でそれぞれ一定の時間までの計算時間と最大消費メモリ量を計測して比較した．その結果については，発表当日に詳しく報告する．

5 おわりに

粒子法 MPS で計算時間を短縮するために 既に利用されている Algorithm[1]や本研究で提案した Algorithm が，計算時間を大幅に短縮する事を実計算で確認した．しかし，低スペック PC で稼働するゲームで使用できる位に 粒子法 MPS を高速に数値計算するためには，粒子数を減らす必要があり 解析領域を小さく設定する必要があると考えられる．つまり，規模を縮小することでゲームへの導入は可能である．ただし，3 次元空間を用いたゲームに粒子法 MPS を利用するには，それ相応に高いスペックの PC で稼働する事が現段階では必要である．

粒子法 MPS で計算時間を短縮するには，領域分割において time step あたりの時間刻み幅を短くすることよりも Courant 条件などが関係する時間刻み幅を大きく設定できる事の方が有効であると考えられる．

参考文献

- [1] 越塚誠一,柴田和也,室谷浩平,“粒子法入門”,丸善 2014
- [2] 大地雅俊,越塚誠一,酒井幹夫,自由表面流れ解析のための MPS 陽的アルゴリズムの開発”, Transactions of JSCES, Paper No.20100013, Vol.32-4, 104 (2013)
- [3]室谷浩平,大地雅俊,藤澤智光,越塚,吉村,“ParMETIS を用いた MPS 陽解法の分散メモリ型並列アルゴリズムの開発”, Transactions of Japan Society for Computational Engineering and Science, Paper No.20120012 (2012)



Figure 1 水柱崩壊 CG Animation 描画

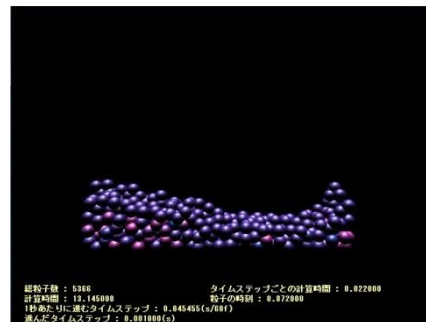


Figure 2 水柱崩壊 CG Animation 描画