コンピュータグラフィクスアニメーション課題

05242628 三田村彰大

概要　粒子法、特に MPS 法を実装し、実際に流体シミュレーションを行った。また異なる手法の比較という観点から、半陰解法（semi-implicit）および陽解法（explicit）を用いた場合についてそれぞれ計算を行い挙動の違いを観察した。その上で、粒子法が波面のダイナミックな動きを再現することができることや、境界条件の設定を非常に自由に行えることなどを体感した。また、陽解法が数値不安定であることや、安定性を達成できても非圧縮性をうまく再現できないことなどを実際に計算を可視化することで理解した。

※ コード置き場：　https://github.com/ramutami/mps_particle_fluid

※ 可視化動画置き場：　https://shota-mitamura.com/works/ryuushi/ryuushhou/

※ AI との会話ログ：　https://chatgpt.com/g/g-p-6a31a79525788191ac9f0113ba5c1b07-yang-jie-fa-li-zi-fa/project

1　はじめに

流体のシミュレーションの主な手法としては、空間を格子に分割し各格子に物理量を割り当てることで流れを計算する Euler 的な手法と、流体を粒子の集まりとして表現し各粒子の位置を計算点と考える Lagrange 的な手法がある。前者は一般に格子法などと呼ばれ、後者は一般に粒子法などと呼ばれる。[2]

理論が整備されており微分の離散化が比較的単純な格子法に比べると、粒子法は理論がまだ成熟しておらずまた精度や数値安定性に問題を抱える手法である。しかし、格子法が苦手としている波面の表現や、連続体の変形・破断などは表現においては粒子法は非常に優れている。(Figure1参照。) そのためゲームや CG など、物理的な正確性よりも見栄えの美しさなどが優先される場面においては粒子法の利用が非常に盛んとなっている。

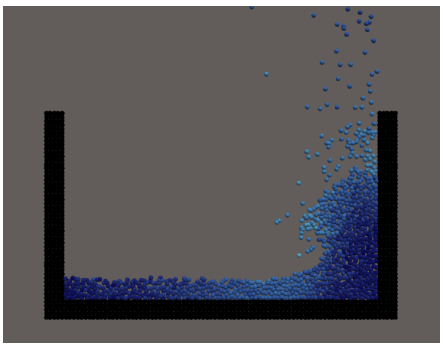

Figure1: 粒子法が得意とする波面の表現の一例。今回の数値計算の結果を示している。

さて、授業では粒子法の手法の一つとして SPH 法が紹介されたが、今回は MPS 法というこれとは異なる手法を用いた粒子法を実装することを考える。両者の違いとしては、SPH 法が圧縮性流体の計算手法として開発された手法であるのに対し MPS 法は非圧縮流体の計算手法として開発されたという経緯がある。そのため SPH 法が安定性に優れる反面物理的に正確な挙動を表現することを苦手としているのに対し、MPS 法は非圧縮性を保ちながら計算するため SPH 法よりかは厳密であり、工学的な応用が多い（参考：[3]）[*1]。

その上で、今回は [1] を参考に Fortran を用いて MPS 法を用いた水柱崩壊を実装し、粒子法の性質について観察することを試みた。また、計算の途中で必要となる圧力を implicit に解いた場合と explicit に解いた場合でどのような性質の違いがあるのかについても観察した。

コードはgithubにアップロードしており、本レポートには書ききれなかった詳細な理論背景を README にまとめている。また、可視化ページに計算結果の動画を置いている。

※ 参考図書の利用・スパコンの利用・AI の利用について

今回、MPS 法の実装にあたって丸善出版より発売されている『粒子法入門：流体シミュレーションの基礎から並列計算と可視化まで』[1] および付随しているサンプルコードを大いに参考にした。ただしコードは基本的に一から自分の手で書いている。[*2]

また、今回計算に当たっては東大の Wisteria Odyssey スーパーコンピュータシステムを利用した。たまたま工学部の計算機系の授業をとっていた関係で学生一人一人に計算資源（8 ノード・48 コア・一計算 15 分）が割り当てられていたため、そちらを利用した。[*3]

また今回粒子法の実装にあたって AI をデバッグや並列化手法の提案、サンプルコードの解読などに利用した。またレポートの誤字脱字のチェックや内容の査読にも AI を用いている。詳しくは本文冒頭の会話ログのリンクを参照。

[*1] SPH は宇宙物理やゲーム、MPS は製品開発におけるシミュレーションなどの工学分野、といった雰囲気の棲み分けがあるらしい…？

[*2] もともと C 言語で書かれたサンプルコードを Fortran に書き写す遊びを半年ほど前に行っており、今回はそれを再開する形で実装を行った。

[*3] 利用にあたっては授業用計算資源の趣旨に反しない範囲であることを意識した。件の工学部の授業でも粒子法は一つの大きなトピックとして扱われており実際にその並列化方法をコードを動かして学ぶことが求められていたため、実質的に授業の内容が被っていると判断し計算資源を利用した。

2　理論

流体のシミュレーションにおいて基本となるのは次に示すナビエ・ストークス方程式である。

$$\frac{D\boldsymbol{u}}{Dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2\mathbf{u} + \boldsymbol{g} \tag{1}$$

ただし以下では流体の流れを $\boldsymbol{u}$ で表すことにする。また、上式において P は圧力、ρ は密度、ν は流体の粘性を決める動粘性係数、$\boldsymbol{g}$ は重力加速度である。以下では、粒子法、特に MPS 法において (1) がどのように数値的に解かれるかを見ていく。以下の内容は基本的に [1] を参考にしている。

2.1　粒子法の概要

最初に述べたように、粒子法においては流体を粒子の集まりとして表現し各粒子 i の位置において物理量を評価する。よって、i 番目の粒子位置 $\boldsymbol{r}_i$ においてナビエストークス方程式の右辺を評価し、それを用いて次のように流速 $\boldsymbol{u}$ を更新する。

$$\boldsymbol{u}_i^{k+1} = \boldsymbol{u}_i^k + \Delta t \cdot \left(-\frac{1}{\rho}\nabla P + \nu\nabla^2\mathbf{u} + \boldsymbol{g}\right)_i^k \tag{2}$$

ただし $k, k+1$ は時間ステップを表す。また、流速が求まれば粒子の位置は $\boldsymbol{r}_i^{k+1} = \boldsymbol{r}_i^k + \Delta t \cdot \boldsymbol{u}^k$ として更新することができる。　よって、(1) 式の離散化をどのように行うかということが本質であり、MPS や SPH といった手法の違いを理解する上でも重要である。

2.2　MPS 法における微分演算子の計算スキーム

ナビエストークス方程式の右辺を離散化するということはつまり微分演算子 ∇, ∇^2 を離散化するということである。MPS 法においては、粒子からの距離に応じて減衰する重み関数を用いてある粒子の近傍に存在する粒子からの微分への寄与を足し込むことでこれらの値を計算する。具体的には、着目している粒子からの距離が近ければ近いほど大きくなり、同時に一定距離 R_e より外側では重みが 0 となるような重みを考える。(Figure2参照。)

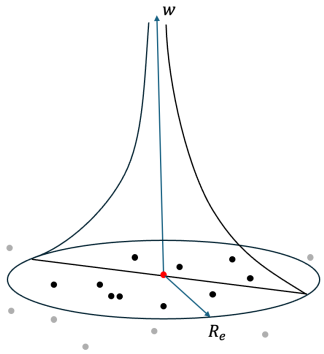

Figure2: 重み関数。赤点が着目粒子であり、影響半径 R_e より内側の粒子の情報がその近さに応じて足し込まれる。

以下に MPS 法で用いられる重み関数の一例を示す。この重み関数はあくまで一例であり、今回の場合は半陰解法を用いた手法で用いられているが例えば陽解法においては別の重み関数を用いることになる。(詳しくはコード置き場のREADMEを参照。)

$$w(r) = \begin{cases} \dfrac{r_e}{r} - 1 & (r < r_e) \\ 0 & (r > r_e) \end{cases} \tag{3}$$

その上で、この重み関数を用いて微分演算子は次のように離散化することができる。

$$\nabla\phi = \frac{d}{n^0}\sum_{i\neq j}\frac{\phi(\boldsymbol{X}_j) - \phi(\boldsymbol{X}_i)}{|\boldsymbol{X}_j - \boldsymbol{X}_i|}\frac{\boldsymbol{X}_j - \boldsymbol{X}_i}{|\boldsymbol{X}_j - \boldsymbol{X}_i|}w(|\boldsymbol{X}_j - \boldsymbol{X}_i|) \tag{4}$$

$$\nabla^2\phi = \frac{2d}{\lambda_0 n^0}\sum_{i\neq j}(\phi(\boldsymbol{X}_j) - \phi(\boldsymbol{X}_i))w(|\boldsymbol{X}_j - \boldsymbol{X}_i|) \tag{5}$$

ただし ϕ は着目している粒子 i での物理量を表し、$\boldsymbol{X}_i, \boldsymbol{X}_j$ は i 番目、j 番目の粒子の位置を表す。また d は空間の次元（例えば 3 次元の水柱の崩壊を計算するなら $d = 3$）、n^0 は初期状態における流体の数密度、λ^0 は n^0 を用いて計算されるパラメタ[*4]である。(詳しい導出は [1] を参照。)

ここでは詳しい導出は行わないが、本質的に大事なのはそれぞれの微分演算子が実質的に「影響半径内に存在する粒子からの寄与の重み付きの足し込み」によって計算されているということである。[*5] これが後ほど並列化による計算高速化を考える上で重要となる。

また、(4) (5) 式には初期状態での流体の数密度 n^0 が現れている。今一般の時刻における粒子 i 近傍の数密度 n_i は

$$n_i = \sum_{i\neq j} w(|\boldsymbol{X}_j - \boldsymbol{X}_i|) \tag{7}$$

によって計算されるが、(4) (5) の導出にあたっては $n_i \sim n^0$ という近似を用いている。これは

「流体の密度は初期基準値からずれない」

という非圧縮性を MPS 法が織り込んでいることを意味する。実際、流体の密度変化を解消するように圧力で粒子を押し戻すという計算によって MPS 法は非圧縮性を実現している。

さて、微分演算子の離散化スキームがもとまったので、我々が (2) を計算するために知りたかった、「粒子 i の位置においてのナビエストークス方程式の右辺の値」を計算することができる。ただし、MPS 法においては先にまずナビエストークス方程式の右辺第二項（粘性項）と右辺第三項（重力項）のみを計算する。そしてそれらを用いて流速を更新したのち、改めて右辺第一項（圧力項）を計算して流速を再び更新する。式に示すと次のよう。

$$\frac{\boldsymbol{u}_i^{k+\frac{1}{2}} - \boldsymbol{u}_i^k}{\Delta t} = \nu\left(\nabla^2\boldsymbol{u}\right)_i^k + \boldsymbol{g} \tag{8}$$

$$\frac{\boldsymbol{u}_i^{k+1} - \boldsymbol{u}_i^{k+\frac{1}{2}}}{\Delta t} = -\frac{1}{\rho^0}\left(\nabla P\right)_i^{k+\frac{1}{2}} \tag{9}$$

この時、粘性項（と重力項）は粒子の時刻 k での情報から計算することができるが、圧力項を計算するには時刻 k における圧力を知っていなければならない。しかし流体方程式は圧力の時間発展を記述する式を持たないので、前の時刻の圧力をステップ更新して現在の時刻における圧力を求めるといった手法を取ることができない。

[*4] λ^0 は具体的には次のようん計算される。

$$\lambda_0 = \frac{1}{n^0}\sum_{i\neq j}|\boldsymbol{X}_j - \boldsymbol{X}_i|^2 w(|\boldsymbol{X}_j - \boldsymbol{X}_i|) \tag{6}$$

[*5] $w(|\boldsymbol{X}_j - \boldsymbol{X}_i|)$ が影響半径の外側で 0 となるので $\sum_{j\neq i}$ は実質的に影響半径内の粒子についての足しあわせと考えることができる。

すなわち、現在時刻における流速や密度の情報を使って都度その時刻における圧力を計算する必要がある。今回はその手法として次に述べる半陰解法と陽解法を利用する。

2.3　圧力項の計算：半陰解法

半陰解法とは、圧力をポアソン方程式の解として求める手法である。今（8）（9）式と、非圧縮条件

$$\frac{D\rho_i}{Dt} + \rho_i \nabla \cdot \boldsymbol{u}_i = 0 \tag{10}$$

などを用いることで次を得ることができる。[*6]

$$-\frac{1}{\rho_0}\left(\nabla^2 P\right)_i^{k+\frac{1}{2}} = \frac{1}{\Delta t^2}\frac{n_i^{k+\frac{1}{2}} - n^0}{n^0} \tag{11}$$

よってこのポアソン方程式を解けば圧力 P を求めることができる。この時境界条件として、液面に存在する粒子の圧力を 0 とする。（Figure3参照。）

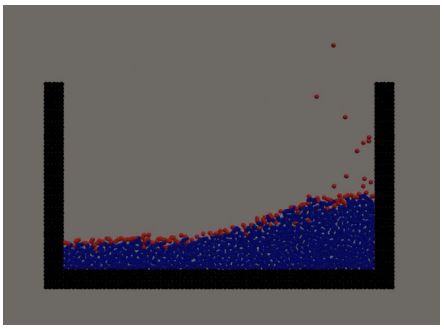

Figure3: ポアソン方程式の境界条件に用いられる自由表面粒子の図。粒子数密度 n_i が一定値を下回った粒子を液面に存在する自由表面粒子と判定する。（図中赤色で示された粒子が自由表面粒子。）

2.4　圧力項の計算：陽解法

半陰解法はポアソン方程式を解くことを必要としており、計算コストが重くなりがちである。そこでより explicit な方法として圧力 P を方程式の解ではなく直接計算する手法が知られており、陽解法と呼ばれる。

今、音速の定義 $c^2 = \partial P/\partial\rho$ に従って圧力を $P_i \sim P^0 + c^2\left(\rho_i - \rho^0\right)$ テイラー展開し、$\rho_i \sim \rho_0 \frac{n_i}{n^0}$ 、$P^0 = 0$ と考えることによって

$$P_i^{k+\frac{1}{2}} = c^2 \frac{\rho^0}{n^0}\left(n_i^{k+\frac{1}{2}} - n^0\right) \tag{12}$$

という式を得ることができる。よってこの式を用いれば、方程式を解くことなく粒子数密度の情報から直接粒子 i の位置における圧力を計算することが可能である。ただしこの陽解法を用いることによる計算コストの削減は数値安定性や非圧縮性の再現度とトレードオフの関係にあり、後々実装結果を議論する際に注目される。

3　実装

ここまでで MPS 法の基本となる理論の解説を行ってきた。以下では実際に実装する上でプログラムがどのように構築されるかについて簡単に解説する。

[*6] 詳しい導出は誌面の都合上割愛するが、README にまとめてある。

3.1　プログラムの概要

今回用いるプログラムの概要を Figure4に示す。

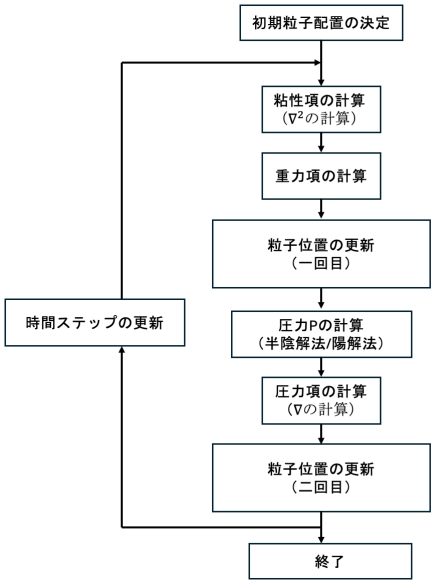

Figure4: プログラムの概要

まず初めに粒子の初期配置を行う。この時、壁も流体と同じように粒子で表現する。この時この壁粒子に流体粒子が接近すると粒子数密度が上昇し非圧縮条件を満たすように圧力による押し戻しが発生することにより、「流体粒子を跳ね返す」という壁として満たしてほしい性質が満たされることになる。（Figure5参照。）

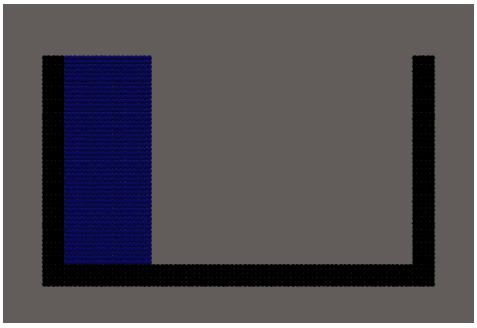

Figure5: 水柱崩壊における粒子の初期配置。黒が壁粒子、青が流体粒子。

粒子の初期配置が決定したら実際の計算に進む。まず初め先に示した ∇^2 の離散化スキームに従って粘性項を計算し、重力項による加速と足し合わせて得られる加速度で粒子の一回目の位置更新を行う。その後、半陰解法・陽解法のいずれかの手法により圧力を計算し、先に示した ∇ の離散化スキームに従って圧力項を計算する。その後、この圧力項による加速を元に粒子位置を今一度更新すれば、一ステップの計算が終了する。詳しいプログラムの内容についてはREADMEを参照。

3.2　半陰解法における共役勾配法を用いた行列計算

さて、半陰解法においては（11）に示されたポアソン方程式を解くわけであるが、先ほど示した ∇^2 に関する離散化スキームを用いればこれは次のように離散化できる。

$$-\frac{1}{\rho_0}\frac{2d}{\lambda^0 n^0}\sum_{j\neq i}\left(P_j - P_i\right)w(|\boldsymbol{r}_j - \boldsymbol{r}_i|) = \frac{1}{\Delta t^2}\frac{n_i - n^0}{n^0} \tag{13}$$

よってポアソン方程式は $A\boldsymbol{P}=\boldsymbol{b}$ という形の行列方程式に帰着されることがわかる。特に今、A が正定値実対称行列となっていることを示すことができるので[*7]、共役勾配法によって P を求めることができる。

この時、「近傍粒子を辿っていっても自由表面粒子に辿りつかないような内部粒子」が存在してしまうと A の正定値性は弱くなってしまうことがわかり、安定に計算するためには行列の正定値性を強くするような工夫が必要となる。今回の場合においては

$$A\boldsymbol{P}=\boldsymbol{b} \rightsquigarrow \left(A+\frac{\kappa}{\Delta t^2}I\right)\boldsymbol{P}=\gamma\boldsymbol{b} \tag{14}$$

のようにして A の対角成分を大きくする工夫を行っている。

3.3 近傍粒子探索・陽解法・並列化による計算量の削減

さて、半陰解法において共役勾配法を用いることができるとわかったので、一番計算が重くなる可能性のあったポアソン方程式の求解は粒子数 N に対し $\mathcal{O}(kN^2)$ となる。よって、それ以外の粘性項の計算や数密度の計算などに現れる $\sum_{i\neq j}$ が i 及び j の二重ループとして現れ $\mathcal{O}(N^2)$ となることを考えると、ポアソン方程式の計算が計算量を決定することがわかる。

ただしこの時、先に述べたように $\sum_{i\neq j}$ はほとんど全ての場合において重み関数をかけながら足し込むので、影響半径の外側で重みが 0 となることを踏まえると実質的な計算は着目粒子 i についてのループといくつかの近傍粒子 j についての足し合わせになり $\mathcal{O}(N)$ となる。よって適切に近傍粒子のリストを保持することができれば、ポアソン方程式の求解の計算量は $\mathcal{O}(kN)$ となり、これが支配的な計算オーダーになる。よって近傍粒子探索によって全体の計算量を抑えることができる。[*8]

いずれにせよ、半陰解法において計算のボトルネックとなるのはポアソン方程式の共役勾配法による求解の $\mathcal{O}(kN)$ であり、計算量の削減にはこの圧力計算を軽量化する必要がある。この時、(12) 式に示した explicit な方法によって圧力を求めれば、この圧力計算を $\mathcal{O}(N)$ で実現することが可能になり、陽解法を用いることにより計算量が削減できる。

※ ただし今回の計算においては自分のキャパの限界もあり近傍粒子探索を実装することができなかった。すなわち圧力の計算以外の諸計算が $\mathcal{O}(N^2)$ で実現されており、陽解法を用いた計算の軽量化を確認することはあまり望めなかった。

また、今回は計算を東大の Wisteria Odyssey スーパーコンピュータシステムを利用して行っており、8 ノード 48 コアを用いることによって大幅な計算高速化を達成している。具体的には MPI と openMP を用いて Fortran コードの並列化を行った。

3.4 陽解法の不安定性と圧縮性

[1] によれば、MPS 法における数値安定性条件は次のようになる。

$$\frac{u_{\max}\Delta t}{l^0}<0.2 \tag{15}$$

$$\frac{c\Delta t}{l^0}<1.0 \tag{16}$$

上が半陰解法と陽解法両方が満たさなければいけない数値安定性条件で、下は陽解法のみが満たさなければいけない数値安定性条件である。ただし上式において l^0 は初期粒子間距離である。

この時、物理的な音速 c の大きさと粒子の小ささを考えると、下の条件を満たすために陽解法においては Δt は相当小さな値を取らなければならない。言い換えれば、陽解法においては計算コストが低くなる代わりに数値安定性が悪くなってしまう。

このようにして生じる数値不安定性を回避する手段の一つとして、音速 c を実際の値よりも小さくとるというものがある。すなわち $c=u_{\max}/0.2$ などと置くことで、(15) が満たされるときに (16) が自動的に満たされるようにしてしまう。これにより Δt を小さくとらずとも陽解法の解は (16) を満たし安定となることができる。

ただしこの場合、音速を本来の物理的な値よりも小さく見積もっていることになってしまい、これは実質的に圧縮性を大きく見積もっていることに相当する。すなわち非圧縮条件が崩れ解が非圧縮解からズレることになる。よって陽解法による計算コストの削減は「数値不安定性」か「非圧縮条件からのずれ」のいずれかを必ず代償として受け入れなければならないことを意味する。

4 実装結果

実際に MPS 法により水柱崩壊を計算した動画を動画ページに示している。

4.1 半陰解法

まず初めに半陰解法による二次元水柱崩壊の計算を行った。(Figure6参照。)

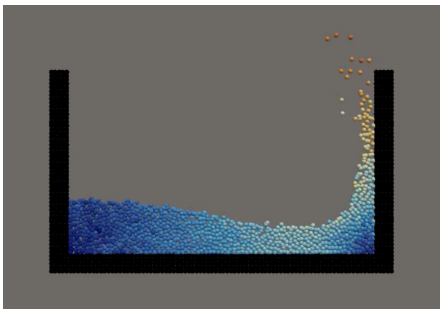

Figure6: 半陰解法を用いた二次元水柱崩壊の数値計算結果。色は速度場を示している。

その結果、粒子法による計算では波面の波飛沫が非常にダイナミックに再現できること、8 ノード 48 コアのスパコン計算を用いても 2 分近くかかり、適切な近傍粒子探査による計算量削減を実施しないと粒子法の計算は非常に重くなってしまうことなどがわかった。

また Figure7に示すように高い粘性を持つ流体でも水柱崩壊の計算を行った結果、粒子法は流体の粘性をよく捉えることができることがわかった。(物理的な正確性は別として、流体の「粘り気をよく表現することがわかった。」)

[*7] README参照。

[*8] ただし $\mathcal{O}(N^2)$ の計算を行わないようにするには、「初めに各粒子の近傍粒子をリスト化するルーチン」を i,j について走査して $\mathcal{O}(N^2)$ で計算してはいけない。実際には、空間をパケット化することによってこの近傍粒子探索を $\mathcal{O}(N)$ で実現することができることが知られている。

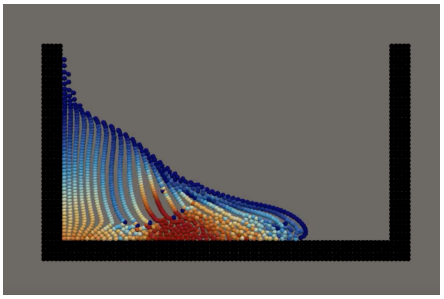

Figure7: 高粘性流体における半陰解法を用いた二次元水柱崩壊の数値計算結果。色は圧力場を示している。

4.2　陽解法

次に、陽解法による二次元水柱崩壊の計算を行った。

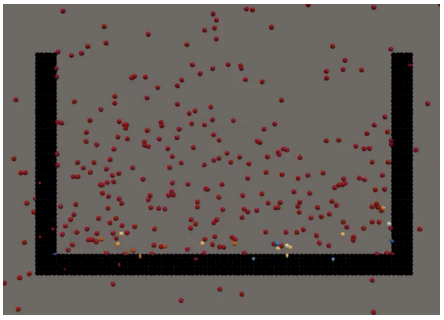

Figure8: 陽解法を用いた二次元水柱崩壊の計算結果。数値不安定により発散してしまっている。

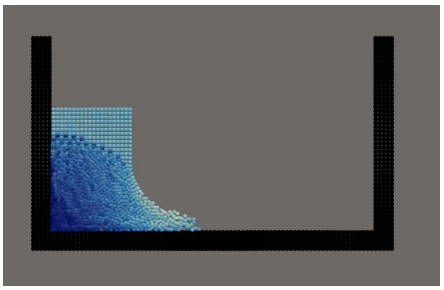

Figure9: 安定性条件を満たすように時間ステップ幅を小さくして計算した図。流体粒子同士が密に重なり合って圧縮されてしまっていることがわかる。

その結果、Figure8に示すように数値不安定により計算が爆発してしまうことがわかった。より詳しく動画を見るとわかるが、これは流体の重力によって加速し続け速度が一定値を超えた瞬間起こっていることがわかる。すなわち先に示した不安定条件が効いていることがわかる。また、この際の計算時間は 1 分半程度に抑えられており先ほどよりも早くなっていた。よって陽解法が計算量の削減と数値安定性のトレードオフの関係にあることを実際に確認できた。

また、時間ステップ幅を小さく取ることにより陽解法における数値不安定性を抑えた実験では流体が圧縮や膨張をしてしまうことがわかった。（Figure9参照。）すなわち、陽解法において安定性を実現しようとすると今度は物理的な精度がトレードオフされてしまうということがわかり、先の章で見た通り陽解法の計算量削減が必ず何がしかの悪影響とトレードオフになってしまっていることがわかった。

4.3　３次元の水柱落下

最後に三次元の水柱落下の計算を行った。Figure10、Figure11参照。

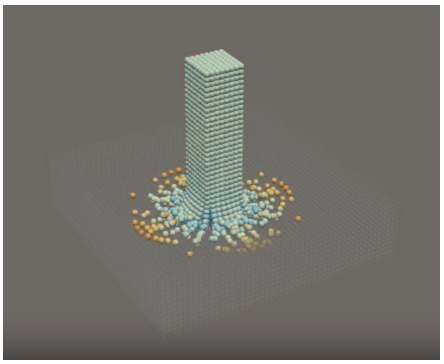

Figure10: 初め空中にあった水柱が浅い水槽に落下する様子。（1）

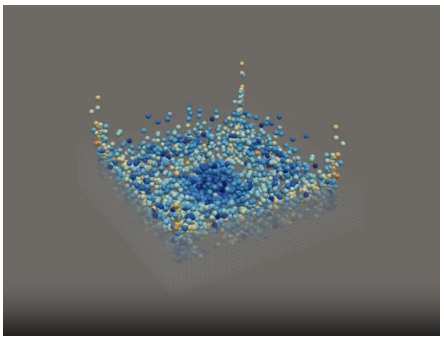

Figure11: 初め空中にあった水柱が浅い水槽に落下する様子。（2）

特に今回、水槽を記述するための粒子数が次元の増加とともに大きく増えること、それに伴い２次元から３次元に変わると爆発的に計算時間が伸びることなどがわかった。先の二次元の水柱崩壊では 2 分程度で 20000 ステップの計算が終わったが、３次元の場合においては 20000 ステップの計算の終了までにおおよそ 2 時間近い時間がかかることがわかった。[*9]

5　結果とまとめ

今回、粒子法の中でも特に MPS 法を実装し、二次元と三次元での水柱崩壊・水柱落下のシミュレーションを行った。その結果として、

- 粒子法は波面の運動をダイナミックに表現することができる。
- 粒子法は粒子数の増大とともに計算コストが重くなってしまいがちであり、近傍粒子探索や並列化を効率よく用いる必要がある。
- 陽解法は計算コストが小さくて済む代わりに不安定であり、その不安定性を抑えると今度は非圧縮性の喪失という形で別の問題が現れてしまう。

といったことがわかった。特に陽解法の計算コスト削減と諸々のトレードオフは実験前から予想できていたことであり、それを実際に確認できたことは非常に良い成果であった。

また陽解法による計算コストの削減については、今回はポアソン方程式の求解の $\mathcal{O}(kN^2)$ が陽解法の $\mathcal{O}(N)$ に置き換わった程度であり、計算が全体として $\mathcal{O}(N^2)$ に支配されていた為体感できる計算

[*9] 今回は計算資源の都合上 15 分で計算できる最初の数ステップだけ計算を行った。

量削減は小さかった。しかし、適切な近傍粒子探索を実装して半陰解法の支配的な計算オーダーを $\mathcal{O}(kN)$ にすることができれば、陽解法によって計算オーダーが $\mathcal{O}(N)$ に落ちた時の変化をより強く感じることができると期待される。

6　おわりに

今回の粒子法の実装は、サンプルコード付きの教科書があったとはいえそれなりに骨の折れる作業であった。特に MPI で並列化する際に計算のどのタイミングでデータを送受信しなければならないかを考えるのが非常に難しかった。結果的にコードを完成させるまでに（休み休みであったとはいえ）半年ほどかかってしまったが、最終的にこのように授業の課題に還元することができ、良い思い出になったと感じている。[*10]

参考文献

[1] 越塚誠一，柴田和也，室谷浩平，『粒子法入門：流体シミュレーションの基礎から並列計算と可視化まで』，丸善出版，2014.

[2] https://www.jstage.jst.go.jp/article/jjws/80/6/80_534/_pdf

[3] https://www.jepoc.or.jp/tecinfo/library.php?_w=Library&_x=detail&library_id=436

[*10] 期日を授業日の木曜だと勘違いしており、気付いてから慌ててレポートを始めた結果 1 時間ほど遅れての提出になってしまいました。本当に申し訳ございません。