卒業研究論文

流体シミュレーションのFLIP法の粒子半径の解析

須之内 俊樹
学籍番号　19D8102020C

中央大学理工学部情報工学科　形状情報処理研究室

2023年3月

概　要

要約には論文の内容を 1 ページ以内で総括的に述べる．要約の直後に，論文の内容を表す 5 語程度以内のキーワードを記す.

キーワード: 流体シミュレーション，CG，PIC/FLIP,

目 次

第1章　概要

卒業論文は，A4 版の用紙に上下左右とも 2[cm] の余白を設け，横書きに 1 段組で作成する．本文より前に，要約とキーワード，目次を置く．これらが複数ページにわたる場合，ローマ数字のページ番号 i, ii, …を付ける．要約には論文の内容を 1 ページ以内で総括的に述べる．要約の直後に，論文の内容を表す 5 語程度以内のキーワードを記す．本文が開始するページを 1 ページとし，ページ番号 1, 2, …を付ける．卒業論文のファイル名は“学籍番号.pdf”とする．

本文は，序論，本論，結論に分割する．序論は本論への導入部分であり，研究の対象，背景，目的，意義等を述べる．本論は，論旨が明確になるよういくつかの章に分割し，各章に内容が一目でわかるような題名を付ける．各章は，必要ならば，さらに節に分ける．

結論には，研究の成果や意義その他を総括的に過去形で述べる．また，研究の成果にかかわる将来の展望や，後継者に委ねたい今後の課題についても述べる．本文の内容を理解するために不可欠ではあるが，本文に含めると議論の展開がわかりにくくなる内容，たとえば長大な数式の誘導や命題の証明等は，付録に述べる．

卒業論文の書き方は概要のテンプレートに記載されているので，参考にすること．

第2章　序論

流体シミュレーションとは，コンピュータで流体について，流体の位置，速度，圧力などの物理量を計算するものである．ここでいう流体とは，空気，ガスなどの気体や，水，土砂の混ざった水，蜂蜜のような粘性がある液体，砂粒などの固体の粒など多岐にわたる．計算は流体力学の理論を元に，コンピュータで計算できるように離散化して行う．流体力学の理論では，流体の静止状態を計算するものと，運動状態を計算するものに大きく分けることができる．また，二次元と三次元を扱うことができる．三次元では私たちの身の回りの現象を広くシミュレーションすることができる．二次元では，例えば，水面に広がる波紋や，異なる流体が作り出すマーブル模様のシミュレーションなどができる．計算した結果を人間が見やすいように可視化することで，流体力学の理論だけでは把握することが困難な，空間に広がる流体の物理量の分布を把握することができるようになる．流体シミュレーションは，流体に触れる製品の設計・開発において，技術者が製品と流体の様子をシミュレーションして開発に役立てている．また，流体のアニメーションを作ることができるため，コンピュータグラフィックスの分野でも役に立っている．

流体力学は，計算機が登場する以前は理論の確認をするためには，実際の流体を用いて実験を繰り返さなければならなかったが，計算機が登場してからは，シミュレーションをすることが出来るようになった．計算機を使った流体力学を特別に，数値流体力学（CFD: Computational Fluid Dynamics）と呼び分けることも多い．CFD は実験で得ることが困難な，流れ場全体の詳細な情報を得ることができる．CFD は流体に触れる製品の設計・開発に非常に大きな貢献をした．

また，コンピューターグラフィックスにおいても，流体のアニメーションを計算することができる CFD が貢献していて，映像作品やゲームなどで利用されている．コンピューターグラフィックスにおいては，それらしい流体の運動がリアルタイムでシミュレーションできることが重要視される．精度良く計算できることはそこまで重要視されておらず，場合によってはそれらしさのために，現実より大袈裟なシミュレーションをすることもある．

第3章　非圧縮性流体のシミュレーション

3.1　基礎概念

数値流体力学では，シミュレーションする空間を離散化して計算を行う．離散化は，各辺が空間の座標軸に並行な，微小な計算格子を用いて空間を分割する．この計算格子上に物理量を定義して計算を行う．どのように定義して計算を行うかは手法によって異なる．ある時刻で空間の物理量の分布を計算した後，時刻を少し進めて，次の時刻の計算をすることを繰り返してシミュレーションを行う．

3.1.1　ナビエ・ストークス方程式

まず，この後に説明する流体シミュレーションの分野で一般的に使われているシンボルの説明をする．

- $\boldsymbol{x}$ 流体の位置を表すベクトル．シミュレーションする空間によって二次元，または三次元になる．
- t 現在の時刻．シミュレーション開始の時刻を 0 とする．
- Δx,Δt 離散化する計算格子の格子幅，次の時刻までの時間幅.
- $\boldsymbol{u}\,(\boldsymbol{x},t)$ 位置 $\boldsymbol{x}$，時刻 t での流体の速度ベクトル.
- $p\,(\boldsymbol{x},t)$ 位置 $\boldsymbol{x}$，時刻 t での流体の圧力値.
- ρ,ν それぞれ，流体の密度，流体の粘性．ここでは定数とする．
- $\boldsymbol{f}$ 位置 $\boldsymbol{x}$，時刻 t での流体にかかる外力ベクトル．重力などはここに含める．
- ∇ ラプラシアン.
- $\frac{\mathrm{D}}{\mathrm{Dt}} = \frac{\partial}{\partial t} + \boldsymbol{u}\,(\boldsymbol{x},\boldsymbol{t}) \cdot \nabla$ ラグランジュ微分, または物質微分. 流体力学のような，連続体を扱う力学で用いられており，流れに沿って移動する物体と同じように移動する観測者から見た，物理量の時間変化率を表している．

式（3.1）で表される式を，ナビエ・ストークス方程式とよび，これは流体力学の支配方程式である．非線形二階微分方程式となっており，代数的に一般解を求める事ができない．式（3.2）は流体の圧縮性条件である．水や砂粒のように圧縮されない流体を扱うときは，流体の密度が常に一定，つまり密度の時間微分が 0 になり，それに伴い式（3.2 の左辺の第二項も 0 になるという，式（3.3）と，式（3.4）を得る．式（3.1）と式（3.4）または式（3.3）を連立することで非圧縮性流体の速度を求めることができる．ナビエ・ストークス方程式を扱う際は，コンピューターで近似解を解析的に求める手法が用いられる．

$$\frac{\partial}{\partial t}\boldsymbol{u}\left(\boldsymbol{x},t\right)=-\left(\boldsymbol{u}\left(\boldsymbol{x},t\right)\cdot\nabla\right)\boldsymbol{u}\left(\boldsymbol{x},t\right)-\frac{1}{\rho}\nabla p\left(\boldsymbol{x},t\right)+\nu\nabla^{2}\boldsymbol{u}\left(\boldsymbol{x},t\right)+\boldsymbol{f} \tag{3.1}$$

$$\frac{D}{Dt}\rho+\nabla\cdot\boldsymbol{u}\left(\boldsymbol{x},t\right)=0 \tag{3.2}$$

$$\frac{D}{Dt}\rho=0 \tag{3.3}$$

$$\nabla\cdot\boldsymbol{u}\left(\boldsymbol{x},t\right)=0 \tag{3.4}$$

式（3.1）の右辺の第一項を移流項，第二項を圧力項，第三項を粘性項，第四項を外力項とよぶ．移流項は非線形項であり，その他は線形項である．これらを仮の速度 $\boldsymbol{u}^{*}$ を用いて，線形項を式（3.5），非線形項を式（3.6）のように分解し，段階的に計算して，各時刻での流速や圧力を計算する方法が用いられている．

$$\boldsymbol{u}\left(\boldsymbol{x},t+1\right)=\boldsymbol{u}^{*}-\Delta t\left(\frac{1}{\rho}\nabla p\left(\boldsymbol{x},t\right)+\nu\nabla^{2}\boldsymbol{u}\left(\boldsymbol{x},t\right)+\boldsymbol{f}\right) \tag{3.5}$$

$$\boldsymbol{u}^{*}=\boldsymbol{u}\left(\boldsymbol{x},t\right)-\Delta t\left(\boldsymbol{u}\left(\boldsymbol{x},t\right)\cdot\nabla\right)\boldsymbol{u}\left(\boldsymbol{x},t\right) \tag{3.6}$$

解法として，大きく分けて格子法と粒子法がある．

3.1.2 格子法

格子法とは，流体を扱う空間を正方形の格子に区切り，格子の中心や辺，頂点などに物理量をサンプリングして計算する方法である．区切る格子の数が多いほど精度が良くなり，計算負荷は増加する．格子法の利点として，規則正しく並んだ格子で空間を離散化することで，微分演算を差分法などの方法で近似することができ，実装しやすい．

ここで，格子法での移流項の計算に用いられる差分スキームである，風上差分について説明する．まず，時間微分に前進差分，空間微分に後退差分を用いて微分を近似する方法を考える．欠点として，移流項を近似計算する際に数値拡散が生じてしまうため，移流項の精度があまりよくない．粘性項は以下の移流項の計算によって，移流項の数値拡散として考えることができるので，省かれて計算されることが多い．

格子法における移流項の計算

$$\frac{\partial}{\partial t}\boldsymbol{u}\left(\boldsymbol{x},t\right)=-\boldsymbol{u}\left(\boldsymbol{x},t\right)\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{u}\left(\boldsymbol{x},t\right)$$

偏微分演算子を風上差分 $\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{u}\left(\boldsymbol{x},t\right)=\frac{\boldsymbol{u}\left(\boldsymbol{x}+\Delta\boldsymbol{x},t\right)-\boldsymbol{u}\left(\boldsymbol{x},t\right)}{\Delta\boldsymbol{x}}$ を用いて離散化すると,

$$\frac{\partial}{\partial t}\boldsymbol{u}\left(\boldsymbol{x},t\right)=-\boldsymbol{u}\left(\boldsymbol{x},t\right)\frac{\boldsymbol{u}\left(\boldsymbol{x}+\Delta\boldsymbol{x},t\right)-\boldsymbol{u}\left(\boldsymbol{x},t\right)}{\Delta\boldsymbol{x}}$$

二変数のテーラー展開 $f\left(a+h,b+k\right)\fallingdotseq\sum_{t=0}^{n}\frac{1}{t!}(h\frac{\partial}{\partial \boldsymbol{x}}+k\frac{\partial}{\partial y})^{t}f\left(a,b\right)$ を二次まで行うと,

$$\frac{\partial}{\partial t}\boldsymbol{u}\left(\boldsymbol{x},t\right)=-\frac{1}{2!\Delta\boldsymbol{x}}\boldsymbol{u}\left(\boldsymbol{x},t\right)\left(\left(\boldsymbol{u}\left(\boldsymbol{x},t\right)+\Delta t\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{u}\left(\boldsymbol{x},t\right)+\left(\Delta\boldsymbol{x}\right)^{2}\frac{\partial^{2}}{\partial \boldsymbol{x}^{2}}\boldsymbol{u}\left(\boldsymbol{x},t\right)\right)\right.$$

$$\frac{\partial}{\partial t}\boldsymbol{u}\left(\boldsymbol{x},t\right)=-\boldsymbol{u}\left(\boldsymbol{x},t\right)\left(\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{u}\left(\boldsymbol{x},t\right)+\Delta\boldsymbol{x}\frac{\partial^{2}}{\partial \boldsymbol{x}^{2}}\boldsymbol{u}\left(\boldsymbol{x},t\right)\right)$$

$$\frac{\partial}{\partial t}\boldsymbol{u}\left(\boldsymbol{x},t\right)=-\boldsymbol{u}\left(\boldsymbol{x},t\right)\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{u}\left(\boldsymbol{x},t\right)-\boldsymbol{u}\left(\boldsymbol{x},t\right)\Delta\boldsymbol{x}\frac{\partial^{2}}{\partial \boldsymbol{x}^{2}}\boldsymbol{u}\left(\boldsymbol{x},t\right)$$

ここで,$-\boldsymbol{u}\left(\boldsymbol{x},t\right)\Delta\boldsymbol{x}=\mu$ とし,$\frac{\partial^{2}}{\partial \boldsymbol{x}^{2}}\boldsymbol{u}\left(\boldsymbol{x},t\right)=\nabla^{2}\boldsymbol{u}\left(\boldsymbol{x},t\right)$ とすれば,

$$\frac{\partial}{\partial t}\boldsymbol{u}\left(\boldsymbol{x},t\right)=-\boldsymbol{u}\left(\boldsymbol{x},t\right)\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{u}\left(\boldsymbol{x},t\right)+\mu\nabla^{2}\boldsymbol{u}\left(\boldsymbol{x},t\right))$$

となり,右辺の二項目が数値拡散となるが、粘性項の形と同じであることがわかる。

格子法における非線形項の計算

式（3.5）の粘性項は先述の通り省略し，外力項は事前に流速に与えておく．残った圧力項の計算について考える．ただし，移流項を先に計算し，$\boldsymbol{u}^{*}$ は既知のものとする．

$$\boldsymbol{u}\left(\boldsymbol{x},t+1\right)=\boldsymbol{u}^{*}-\frac{\Delta t}{\rho}\nabla p\left(\boldsymbol{x},t\right) \tag{3.7}$$

両辺の発散をとると,

$$\nabla\cdot\boldsymbol{u}\left(\boldsymbol{x},t+1\right)=\nabla\cdot\boldsymbol{u}^{*}-\frac{\Delta t}{\rho}\nabla^{2}p\left(\boldsymbol{x},t\right)$$

流体の非圧縮性条件，$\nabla\cdot\boldsymbol{u}\left(\boldsymbol{x},t\right)=0$ により，左辺は 0 になるため,

$$\nabla\cdot\boldsymbol{u}^{*}=\frac{\Delta t}{\rho}\nabla^{2}p\left(\boldsymbol{x},t\right) \tag{3.8}$$

この式を満たすような圧力を求め，式（3.7）に代入することによって，次の時刻の流速 $\boldsymbol{u}\left(\boldsymbol{x},t+1\right)$ が計算できる．式（3.8）のような微分方程式はポアソン方程式と呼ばれ，境界条件が分かれば解が存在することが知られている．ここでいう境界条件とは，扱う領域の境界での $p\left(\boldsymbol{x},t\right)$ の値がわかっていることとなる．

格子法の物理量のサンプリング方法

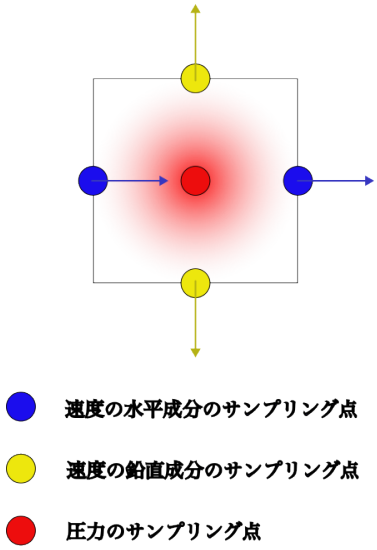


図 3.1: スタッガード格子

CG の流体シミュレーションにおいて，Fedkiew らの，Marker-And-Cell（MAC）法で格子を離散化する方法がよく用いられている．この手法では，扱う領域を正方形の格子に区切り，圧力を格子の中心に配置する．流速は格子面の中心に，その格子面と垂直な流速の成分を配置する．例えば $x-y$ 平面に並行な格子面には，その地点での流速の z 成分を配置する．これはスタッガード格子と呼ばれている．この手法は格子点に全ての物理量を配置するコロケート格子と比べ，境界付近の流速が表現しやすく，それらしいシミュレーションができるだけでなく，ポアソン方程式の境界条件の設定も容易である．具体的には，格子が壁に隣接している場合，その面からの流体の流入出はゼロであるため，壁面方向の圧力差と，壁面と隣接している面の中心に配置されている流速成分は共にゼロにする．

スタッガード格子を用いた，格子法の線形項の計算の離散化方法について考える．

$$\nabla \cdot \boldsymbol{u}^* = \frac{\Delta t}{\rho} \nabla^2 p\left(\boldsymbol{x}, t\right)$$

スタッガード格子では圧力と流速が配置してある位置がずれているため，両辺に同じ差分を適用できない．離散化する方法として，両辺をセルの領域で積分して考える．

$$\int_V \nabla \cdot \boldsymbol{u}^* = \int_V \frac{\Delta t}{\rho} \nabla^2 p\left(\boldsymbol{x}, t\right)$$

両辺にガウスの発散定理を適用すると，周回積分の形で表せる．

$$\oint_V \boldsymbol{u}^* \cdot \boldsymbol{n} = \oint_V \frac{\Delta t}{\rho} \nabla p\left(\boldsymbol{x}, t\right) \cdot \boldsymbol{n}$$

この式はスタッガード格子格子において，周回積分は格子の各面で境界条件を考えて，全ての面の物理量を足すことで表すことができる．ここで，整数 $n = 0, 1, 2, \cdots 5$ までを，それぞれ格子の各面の方向に割り振ると，

$$\sum_n \boldsymbol{u_n^*} D_n F_n \Delta x = \sum_n \frac{\Delta t}{\rho} \nabla p_n(\boldsymbol{x}, t)\ F_n \Delta x$$

それぞれの係数は次のようになっている．

- F_n 格子の面の各方向に隣接しているものが流体なら 1，壁面なら 0 を取るブーリアン値
- D_n 格子の面の各方向に垂直で，格子の外側を向く単位ベクトルを表す係数．向いている方向が軸の正の向きなら$+1$， 負の向きなら -1 を取る整数値．
- p_n 割り振られた方向に隣り合っている格子の圧力値．
- u_n 割り振られた方向の格子面に配置されている流速値．

圧力を計算する格子の添字を (i, j, k) とし，圧力の微分に前方差分 $\nabla \boldsymbol{p_n}(\boldsymbol{x}, t) = \frac{\boldsymbol{p_n}(\boldsymbol{x},t) - \boldsymbol{p_{i,j,k}}(\boldsymbol{x},t)}{\Delta \boldsymbol{x}}$ を適用すると，

$$\sum_n \boldsymbol{u_n^*} D_n F_n \Delta x = \sum_n \frac{\Delta t}{\rho} \frac{\boldsymbol{p_n}(\boldsymbol{x}, t) - \boldsymbol{p_{i,j,k}}(\boldsymbol{x}, t)}{\Delta \boldsymbol{x}} F_n \Delta x$$

式を整理すると，

$$\sum_n \boldsymbol{u_n^*} D_n F_n = \sum_n \frac{\Delta t}{\rho} \frac{\boldsymbol{p_n}(\boldsymbol{x}, t) - \boldsymbol{p_{i,j,k}}(\boldsymbol{x}, t)}{\Delta \boldsymbol{x}} F_n \tag{3.9}$$

この式を満たす圧力を求めれば良い．式（3.9）は未知数 p の連立一次方程式となっている．直接法ではなく反復法を用いて，必要な精度で計算を打ち切って計算し，解法としては前処理付き共役勾配法がよく用いられている．

ディリクレ境界条件

圧力計算の境界条件は，シミュレーションによって追加することもある．ディリクレ境界条件は流体と気体の領域を両方考える際に用いる境界条件で，流体が存在しない格子の圧力値を 0 とする．

3.1.3 粒子法

粒子法とは，流体をいくつかの小さな粒子の集まりとして考え，各粒子に物理量を与えて計算をする方法である．本来は流体は，アボガドロ数（$N_A = 6.02 * 10^{23}$）単位の分子が動いているが，これら全てを計算機で扱うのは不可能であるため，いくつかの分子の動きを一つの粒子にまとめて扱う．粒子の数が多いほど実際の現象に近づき，精度が良くなるが，計算負荷は増加する．利点として，以下の計算によって，移流項の計算に数値拡散が発生しないことが確認できるため，移流項が精度良く計算できる．欠点として，その他の微分演算を単純な方法で近似することができないため，その他の項は計算の精度が悪い．

粒子法における移流項の計算

簡単のため, ナビエストークス方程式から粘性項と外力項を取り除いた, オイラー方程式と呼ばれるものについて考える.

$$\frac{\partial}{\partial t}\boldsymbol{u}\left(\boldsymbol{x},t\right) = -\left(\boldsymbol{u}\left(\boldsymbol{x},t\right)\cdot\nabla\right)\boldsymbol{u}\left(\boldsymbol{x},t\right) - \frac{1}{\rho}\nabla\boldsymbol{p}$$

粒子法はラグランジュ微分を用いることで、移流項の計算が単純になる。時刻 t_0 に位置 $\boldsymbol{x_0}$ にある流体の速度を $\boldsymbol{u}\left(\boldsymbol{x_0},t_0\right)$ とする.Δt 秒後の位置は $\boldsymbol{x_0} + v\left(\boldsymbol{x_0},t_0\right)\Delta t$ になっているので, 時刻 $t_0 + \Delta t$ の流体の速度は、$\boldsymbol{u}\left(\boldsymbol{x_0} + \boldsymbol{u}\left(\boldsymbol{x_0},t_0\right)\Delta t, t_0 + \Delta t\right)$ となる。この間の速度変化 $\Delta\boldsymbol{u}$ をテイラー展開を用いて表すと,

$$\Delta\boldsymbol{u} = \boldsymbol{u}\left(\boldsymbol{x_0} + v\Delta t, t_0 + \Delta t\right) - \boldsymbol{u}\left(\boldsymbol{x_0},t_0\right) = \frac{\partial\boldsymbol{u}\left(\boldsymbol{x_0},t_0\right)}{\partial\boldsymbol{x}}\boldsymbol{u}\left(\boldsymbol{x_0},t_0\right)\Delta t + \frac{\partial\boldsymbol{u}\left(\boldsymbol{x_0},t_0\right)}{\partial t}\Delta t$$

両辺を Δt で割ると,

$$\frac{\Delta\boldsymbol{u}}{\Delta t} = \frac{\partial\boldsymbol{u}\left(\boldsymbol{x_0},t_0\right)}{\partial\boldsymbol{x}}\boldsymbol{u}\left(\boldsymbol{x_0},t_0\right) + \frac{\partial\boldsymbol{u}\left(\boldsymbol{x_0},t_0\right)}{\partial t}$$

$$\frac{\Delta\boldsymbol{u}}{\Delta t} = \left(\boldsymbol{u}\cdot\nabla\right)\boldsymbol{u} + \frac{\partial\boldsymbol{u}\left(\boldsymbol{x_0},t_0\right)}{\partial t}$$

ラグランジュ微分, つまり, 粒子を流れに沿って動かしたときの変化量が, オイラー方程式の移流項になっている．粒子法では, 物理量をもった粒子自体を動かすだけで，移流項の計算をすることができる.

粒子の位置の補正

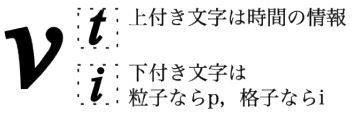
粒子を流速に沿って移動させた際，境界をすり抜けてしまうことがある．この対策として，各時刻で，飛び出してしまった粒子を境界内に戻す手法がある．戻す方向や距離は，陰関数を用いたレベルセットサーフェスを考えることで，比較的簡単に実装できる．

粒子法で流体の非圧縮性条件を満たすようなシミュレーションをするには，大きく分けて二つの方法がある．1 つはポアソン方程式を解く方法である．MPS 法などの粒子法は，格子法でも登場したポアソン方程式を解く方法を取っているが，離散化が格子法のように簡単には行えず，精度や計算時間に問題がある．2 つ目は，粒子の密度を計算し，密度が一定になるように粒子を移動させたり，粒子を増減させる方法である．SPH 法などはこのような方法をとっている．この方法はナビエストークス方程式の計算過程とは関係ないため，粒子の分布の変化に伴って粒子の物理量も補正する必要がある．一つ目の方法は流体の非圧縮性条件として，式（3.4）を用いているのに対し，二つ目の方法は，式（3.3）を使っていると考えることもできる．式（3.3）を利用する手法は確実に密度が保存されるため，密度に関しては頑健であるのに対し，圧力が振動してしまう．それに対し式（3.4）を用いる方法は密度の計算誤差が蓄積していってしまうのに対し，圧力の振動を防ぐことができる．

3.2 関連研究

3.2.1 PIC（Particle In Cell）

粒子法は移流項の計算の精度がよく，他の線形項の精度が悪い．格子法は移流項の精度が悪く，他の線形項の精度が良い．OO らの手法は，粒子法と格子法の利点を組み合わせた手法である．粒子と格子の間で物理量を受け渡し，移流項の計算を粒子で行い，その他の計算を格子で行う．この分野でよく使用されるシンボルの説明をする．



流速などの物理量は，粒子が持っているのか，格子が持っているのか，また，その時刻によって異なるため，上付き文字として時刻を，下付き文字として物理量を持っているの

が粒子なのか格子なのかを表記する．例えば時刻 t において，格子の物理量が格納されている位置を $\boldsymbol{x}_i^t = (x_i^t, y_i^t)$，粒子の位置を $\boldsymbol{x}_p^t$，格子の流速を $\boldsymbol{v}_i^t$，粒子の流速を $\boldsymbol{v}_p^t$，格子の質量を m_i^t，粒子の質量を m_p^t とする．ただし，基本的には粒子の質量は時刻によらず一定である．また，格子の中に入っている粒子についての物理量は，v_{ip} のように i と p を両方つけて表記する．

粒子と格子での物理量の受け渡し

また，格子の一辺の長さを h とする．まず，格子と粒子の位置による重みを以下のように定義する．

$$w_{ip}^t = N_i^t(\boldsymbol{x}_p^t) = N\left(\frac{1}{h}(x_p^t - x_i^t)\right) \cdot N\left(\frac{1}{h}(x_p^t - x_i^t)\right) \tag{3.10}$$

$N(x)$ はカーネル関数であり，以下のように定義する．

$$N(x) = \begin{cases} \frac{1}{2}|x|^3 - |x|^2 + \frac{2}{3} & (0 \leqq |x| < 1) \\ \frac{1}{6}(2 - |x|)^3 & (1 \leqq |x| \leqq 2) \\ 0 & else. \end{cases} \tag{3.11}$$

カーネル関数は，引数の絶対値が 0 に近いほど大きく，離れるに従って小さくなっていく，微分可能な関数である．粒子と格子の距離が格子の幅の二倍以上離れていると重みは 0 になるため，その範囲内の粒子についてのみ考えればよい．

次に格子の質量について，以下のように定義する．

$$m_i^t = \sum_p m_p^t w_{ip}^t \tag{3.12}$$

式（3.12）を使って，粒子の速度を格子の速度に以下のように受け渡す．

$$\boldsymbol{v}_i^t = \sum_p w_{ip}^t \boldsymbol{v}_p^t / m_i^t \tag{3.13}$$

式（3.13）は，両辺に m_i^t をかけることで運動量についての式になり，シミュレーション全体では運動量が保存しているように振る舞う．

PIC のアルゴリズム

PIC の大まかなアルゴリズムは以下のようになる．

- 粒子の流速 $\boldsymbol{v}_p^t$ を式（3.13）に従って，格子の流速 $\boldsymbol{u}(\boldsymbol{x}, t)$ に受け渡す．

- 格子の流速に重力等の外力の情報を反映させる．例えば重力は $\boldsymbol{u}\,(\boldsymbol{x}, t)$ $+= \boldsymbol{f}\Delta t$ のように反映させる．
- 3.1.2 のように，格子法と同じ方法で格子の圧力項を計算する．ここで，格子の次の時刻での流速 $\boldsymbol{u}_t(\boldsymbol{x}, t+1)$ が求められる．
- 格子の次の時刻での流速 $\boldsymbol{u}\,(\boldsymbol{x}, t+1)$ を粒子に受け渡し，粒子の次の時刻の流速 $p\boldsymbol{v}_{t+1}$ を求める．この際も式（3.13）と同様の重みを使用する．
- 粒子の位置を $\boldsymbol{v}_p^{t+1}$ を使って，$\boldsymbol{x}_p^{t+1} = \boldsymbol{v}_p^{t+1}\Delta t$ のように更新する．

一般的に粒子の数が格子の数よりも多いため，粒子から格子に物理量を受け渡す際に，1 つの格子に対して複数の粒子の物理量を集める．この際に，通常の粒子法と比べると粒子の粘性が高くなるような効果がある．これを人工粘性と呼ぶ．格子法とは異なり，移流項を粒子法で計算すると粘性項を別途計算する必要があるが，PIC は人工粘性があるため，粘性項は計算しない．しかし，意図しない粘性が含まれてしまうため，粘性の小さな流体のシミュレーションには不向きである．この人工粘性を解決するためのさまざまな手法が提案されている．

3.2.2 FLIP（Fluid Implicit Particles）

PIC のアルゴリズムを少し変更し，格子の速度の変化分を粒子に受け渡す．アルゴリズム内では，線形項の計算に入る前に現在の流速を保存しておく．圧力項の計算が終わってから，格子の速度を粒子の速度に受け渡す際に，粒子の速度を直接計算するのでなく，変化分を重みづけして計算し，現在の粒子の速度に加算する．この方法を取ることで，PIC のように，格子の影響の範囲内にある粒子の速度が同じような方向に向いてしまうことがなくなる．これにより，PIC の人工粘性を解消することができる．問題点として，数値誤差が蓄積し続ける，粒子の勢いがほとんど減衰しないなどがある．実際の流体は粘性が少なからずあるため，粘性を別で考える必要がある．

PIC/FLIP

FLIP の粘性を考える際，PIC で計算した速度の影響を少しだけ入れることで，人工粘性を持たせる方法である．PIC で計算した速度を $\boldsymbol{v}_{\mathrm{PIC}}$，FLIP で計算した速度を $\boldsymbol{v}_{\mathrm{FLIP}}$ とすると，$\boldsymbol{v}_{\mathrm{PIC/FLIP}}$ は以下のように計算する．

$$\boldsymbol{v}_{\mathrm{PIC/FLIP}} = \alpha\boldsymbol{v}_{\mathrm{FLIP}} + (1-\alpha)\,\boldsymbol{v}_{\mathrm{PIC}} \tag{3.14}$$

ここでαは0から1までの係数で，小さくするほどPICの速度の割合が大きくなり，粘性が大きくなる．しかし，FLIPの数値誤差が蓄積する問題は解決しておらず，PICの角運動量や運動エネルギーが失われてく問題も残っている．

PIC/FLIPの粒子位置の修正

3.1.3で粒子法における粒子位置の修正法を紹介したが，PIC/FLIPは圧力項を粒子法で計算しないため，ポアソン方程式を解く方法を使用するのは手間である．そのため，式（3.3）のように，密度が一定になるように直接補正する方法を紹介する．まず，全ての粒子について，粒子の密度分布を修正する方向ベクトルΔp_iを計算する．

$$W_{smooth}(\boldsymbol{r}, h) = max\left(0, 1 - ||r||^2/h^2\right) \tag{3.15}$$

$$W_{sharp}(\boldsymbol{r}, h) = max\left(0, h^2/||r||^2 - 1\right) \tag{3.16}$$

$$\Delta p_i = -\gamma \Delta t d_i \sum_j \frac{p_j - p_i}{||p_j - p_i||} W_{smooth}(p_j - p_i, d_i) \tag{3.17}$$

ここで，W_{smooth}は平滑化カーネルで，W_{sharp}は中心で値が無限大になるようなカーネルである，γは修正度合いを表す係数で，d_iは粒子半径である．p_i，p_jは粒子の位置である．このベクトルに従って粒子の位置を$x_p+ = \Delta p_i$のように修正する．次に，粒子の物理量をそのままに，粒子の位置を修正すると，粒子によって表現されている流体の物理量の分布が歪んでしまうため，粒子の物理量も計算し直す必要がある．粒子iの物理量q（$\boldsymbol{p_i}$）は以下のように修正する．

$$q\left(\boldsymbol{p_i}\right) = \frac{\sum_j m_j q_j W_{sharp}(p_j^* - p_i, d_i)}{\sum_j m_j W_{sharp}(p_j^* - p_i, d_i)} \tag{3.18}$$

ここで，q_j, m_j, p^*はそれぞれ，粒子の物理量，粒子の質量，修正前の粒子の位置である．この手法による粒子位置の修正は，PICのように角運動量や運動エネルギーが失われてく性質があるため，視覚的には人工粘性が大きくなるような効果がある．

3.2.3 MPS法における不自然な数値振動の抑制

3.1.3で述べたように，流体の非圧縮性条件に従った粒子位置の修正方法には種類があり，それぞれ利点と欠点がある．PICが格子法と粒子法を混ぜ合わせて計算しており，

PIC/FLIP が PIC と FLIP を混ぜ合わせて計算しているように，粒子位置の修正方法を混ぜ合わせた手法となっている．圧力振動がある手法は密度変化をよく防止するため，緩和して用いることができるこの手法によって，既存の他の手法よりも安定した計算と，なめらかな圧力分布を実現している．

第4章　提案手法

4.1　FLIP の実装

まず，PIC/FLIP の実装方法について考える．本手法では三次元のデータを扱うため，c++の線形代数ライブラリ，Eigen を用いる．また，Eigen の連立方程式のソルバーを用いて，ポアソン方程式を解くことができる．スタッガード格子の水平方向の数を Nx，鉛直方向の数を Ny とすると，スタッガード格子は水平方向の速度，鉛直方向の速度，圧力を三次元配列に保存することで実装できる．その際の配列のサイズは以下のようになる．

- x 方向の速度 $u[Nx+1][Ny][Nz]$
- y 方向の速度 $v[Nx][Ny+1][Nz]$
- z 方向の速度 $v[Nx][Ny][Nz+1]$
- 圧力 $p[Nx][Ny][Nz]$

粒子から格子に速度を受け渡す際や，外力の影響を考える際に，格子の質量が必要である．この質量は，水平方向と鉛直方向それぞれの速度がサンプリングされている位置と同じ位置に配置する．また，外力は質量と同じ位置での圧力値を二次元配列で与えるが，重力のような保存力を考える場合，格子にかかる力は格子の質量によって異なるため，加速度のみを二次元配列で与え，必要に応じて重力を乗算する．また，後に FLIP で流速の変化量を計算するために，変化前の流速を保存する配列を同時に用意する．粒子については，1 つの粒子を，粒子の位置，速度，PIC で計算した速度，FLIP で計算した速度をメンバ変数にもつ構造体で実装し，その構造体の配列で全粒子を管理する．粒子は，実装するシミュレーションの次元を d とすると，1 つの格子に 2^d 個配置することが多い．つまり，粒子数は二次元の場合だと 4NxNy となる．ここまでをシミュレーションの前に初期化する．

粒子-格子での物理量の受け渡しの際に，格子のサンプリング点の周辺にある粒子を探索する必要がある．粒子の位置を，格子の数と格子の一辺の長さをかけた数で割ることで，粒子がどの格子の近くにあるか判定することができるが，ナイーヴに格子ごとに近傍の粒子を探索して実装すると，計算量が大きくなってしまう．この問題は，格子ごとに

どの粒子が入っているのかを保存するようなデータ構造を，各タイムステップで更新することで解決できる．各タイムステップの最後で粒子の位置を移動させるため，各タイムステップの最初でこのデータ構造を更新する．本手法では structured grid を使用している．c++では unorderedmap のキーを格子の位置，値を格子に格納されている粒子にすることで実装できる．

次は粒子の速度を格子にわたす．格子点ごとに，カーネル関数の値が非ゼロになるような範囲内の粒子を探索し，式（3.13）を用いて格子の速度を計算する．例えばカーネル関数として，式（3.11）を利用するとすると，粒子との距離が格子幅の二倍以内の粒子を計算すれば良い．このとき，粒子の速度のうち，速度を保存している配列の，保存している速度成分のみを受け渡す．同時に格子の質量も計算される．FLIP で，後に速度の変化量を計算するため，この段階で計算した速度を保存する．

4.2 圧力項の離散化と実装

$$\sum_n \boldsymbol{u_n^*} D_n F_n = \sum_n \frac{\Delta t}{\rho} \frac{\boldsymbol{p_n}(\boldsymbol{x},t) - \boldsymbol{p_{i,j,k}}(\boldsymbol{x},t)}{\Delta \boldsymbol{x}} F_n \tag{4.1}$$

圧力項は圧力を変数とした，上記の連立方程式を解く．格子の総数を n とすると，全ての格子で連立方程式を考えるため，合計 n 本の連立方程式を解くことになる．そこで，$\boldsymbol{Ax} = \boldsymbol{b}$ に対応させたものを解く．$\boldsymbol{A}$，$\boldsymbol{b}$ それぞれのサイズは，$\boldsymbol{A}$ は $n \times n$ であり，$\boldsymbol{b}$ は n である．この際，三次元に分布する圧力をベクトルにうつす．具体的には，圧力の (i,j,k) 成分は，ベクトルの $iNyNz+jNx+k$ 成分にうつす．また，$\boldsymbol{A}$ の $(iNyNz+jNx+k, lNyNz+mNx+n)$ の成分には，位置 (i,j,k) の格子についての連立方程式の，圧力の (l,m,n) 成分にかかる係数を保存すれば良い．このままだと計算量が非常に大きくなってしまうが，$\boldsymbol{A}$ のほとんどの成分は 0 であることを利用し，Eigen の疎行列ライブラリ Sparse を利用して疎行列をつくり，前処理付き共役勾配法の関数 ConjugateGradient を用いて連立方程式を解く．$\boldsymbol{A}$ と $\boldsymbol{b}$ の各成分は以下のように計算する．

- $\boldsymbol{A}$，$\boldsymbol{b}$ を初期化する．この際，$\boldsymbol{b}$ の成分は全て 0 にしておく．
- 全粒子について，ディリクレ境界条件を考え，粒子が存在しない空間の圧力を 0 で固定する．これは，standerdgrid の (i,j,k) に格納されている粒子の数を見て，0 なら $\boldsymbol{A}$ の $(iNyNz+jNx+k, iNyNz+jNx+k)$ 成分を 1 にして，処理を終了すれば良い．

- 粒子が入っていたセル (i,j,k) について，式（3.9）に従って係数を計算する．セル (i,j,k) の周囲の 6 方向に格納されている流速について，シミュレーションの境界で，壁に隣接していない方向についての流速の周回積分を計算する．周回積分の値を，$\boldsymbol{b}$ の $(iNyNz+jNx+k)$ 成分に代入する．
- セル（i,j,k）の周囲の 6 方向のセルについて，粒子が入っていないセルの方向，または，壁に隣接している方向なら疎行列には何もせず，それ以外なら疎行列の $iNyNz+jNx+k$ 行の，それぞれの方向に対応した対応した成分に $\frac{\Delta t}{\rho \Delta x^2}$ を代入する．

4.3 シミュレーション領域の定義

本手法では，シミュレーションは正方形の内部の領域内で行う．正方形の一辺の長さ L は，スタッガード格子の水平方向の数を N，スタッガード格子の一辺の長さを dx とすると，$L=Ndx$ である．4 点 $(0,0)$, $(L,0)$, $(0,L)$, (L,L) を頂点とするような正方形のレベルセットサーフェス $\varphi(x,y)$ を，以下のように定義する．

$$\varphi(x,y)\begin{cases}(x,y) & x\leq 0, y\leq 0\\ (L-x,y) & L\leq x, y\leq 0\\ (x,L-y) & x\leq 0, L\leq y)\\ (L-x,L-y) & L\leq x, L\leq y)\\ (0,0) & else\end{cases} \tag{4.2}$$

このようにして定義されるレベルセットサーフェスは，粒子の座標を引数とし，4 点 $(0,0)$, $(L,0)$, $(0,L)$, (L,L) を頂点とするような正方形から外に出ていだ場合は，正方形領域に最短距離で戻すようなベクトルを返り値として返す．粒子に対して移流計算や粒子位置の補正を行うと，シミュレーション領域から粒子が出ていってしまうことがあるため，そのような計算を行った後に全粒子に対して領域内部に戻す処理を行う．

4.4 シミュレーションの定数や，カーネル関数の決定

流体シミュレーションにおけるさまざまな定数や，カーネル関数は，シミュレーションしたい流体や，シミュレーションの手法，目的によってどのような値を用いるのが最適か

は異なっている．例えば，PIC/FLIP 法での α の値が大きいほど，数値粘性が抑えられる．その一方で，流体の勢いがほとんど減衰しない FLIP の欠点が目立つ．粘性の少ない流体で，激しい流れや流体の運動を表現するには α の値は大きい方が良いため，最大値の 1 でもそれらしい動きが表現できる．本実験では，初期条件として水に見立てた，粘性が小さく，非圧縮性である流体を，水の柱のように特定の領域に配置する．それが重力に従って，決められた空間内でどのように運動するのかのシミュレーションを行う．このようなシミュレーションは，ダム崩壊シミュレーション（Dam Break Simulation）とも呼ばれている．ダム崩壊シミュレーションでは，水の粘性が小さいため，ダムの崩壊直後は水の勢いは激しい．一方で，重力以外の外力は働かないため，水の動きは次第に穏やかになり，最終的には静止する．また，水は非圧縮性の流体なので，シミュレーションの中で密度や体積が保存される方が望ましい．

第5章　計算機実験

MeshLab を用いて，陰関数の関数値から等値面を計算することで，水面のメッシュを生成することができる．MeshLab には HausDorff 距離を計算する機能もあり，幾何誤差を計算することもできる．一例として，和文の著書 [1]，和文の論文誌 [?]，英文の編書 [2]，英文の論文誌 [3]，国際会議 [4]，修士論文 [5]，電子雑誌 [6]，Web ページ [7] を，2 ページの参考文献の節に載せる．**参考文献には信頼性が高く，後世に残るものを載せるように注意せよ．**

書くべき情報は以下のとおりである．

- 和文の著書: 著者，書名，シリーズ名（あれば），発行所，都市，年.
- 和文の論文誌: 著者，題名，誌名，巻，号，頁，年.
- 英文の編書: 編者，書名，発行所，都市，年.
- 英文の論文誌: 著者，題名，誌名，巻，号，頁，年.
- 国際会議: 著者，題名，予稿集名，都市，コード等，頁，年.
- 修士論文: 著者，題名，機関名，年.
- 電子雑誌: 著者，題名，誌名，巻，号，頁（オンライン），DOI，西暦年.
- Web ページ: 著者，Web ページの題名，Web サイトの名称（オンライン）（ただし，著者と同じ場合は省略可），入手先〈URL〉（参照日付）.

英語の文献はすべて半角文字で書く．参考文献には本文で引用した文献のみ載せる．情報処理学会の論文誌の原稿執筆案内 [7] も参考になる．

通し番号は，引用順または著者名のアルファベット順に付ける．文献の引用のしかたは分野ごとに異なるので，**自己流では書かず，当該分野の論文誌などを参考にする**こと．

第6章　結論

結論には，研究の成果や意義その他を総括的に**過去形**で述べる．

謝辞

本研究を進めるにあたり，大変多くのご指導，ご助言を頂いた中央大学理工学部情報工学科の中央太郎教授に深く感謝いたします．また，多大なるご助言，ご協力を頂いた形状情報処理研究室の皆様には大変お世話になりました．心から感謝いたします．

参考文献

[1] 近藤雅裕，越塚誠一，MPS 法における不自然な数値振動の抑制，日本計算工学会論文集，Paper No.20080015，2008.

[2] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, Andrew Selle, *A material point method for snow simulation*, University of California Los Angeles , ACM Transactions on GraphicsVolume 32, Issue 4, July, 2013.

[3] W. Rice, A. C. Wine, and B. D. Grain, Diffusion of impurities during epitaxy, *Proc. IEEE*, vol. 52, no. 3, pp. 284–290, 1964.

[4] L. J. Guibas and R. Sedgewick, A dichromatic framework for balanced trees, *Proc. 19th IEEE Sympos. Found. Comput. Sci.*, Ann Arbor, pp. 8–21, 1978.

[5] 中大次郎，マルチメディアと数理工学，中央大学大学院理工学研究科情報工学専攻修士論文，1998.

[6] K. Iwama, A. Kawachi, and S. Yamashita, Quantum biased oracles, *IPSJ Digital Courier*, vol. 1, pp. 461–469 （online）, DOI: 10.2197/ipsjdc.1.461, 2005.

[7] 情報処理学会，論文誌ジャーナル（IPSJ Journal）原稿執筆案内，情報処理学会（オンライン），入手先〈`https://www.ipsj.or.jp/journal/submit/ronbun_j_prms.html`〉（2022-04-25）.