

中央大学大学院理工学研究科情報工学専攻  
修士論文

# 煙シミュレーションのための部分空間法的高速化

Accelerated Subspace Method for 3D Smoke Simulation

須之内 俊樹  
Toshiki SUNOUCHI  
学籍番号 23N8100018B

指導教員 森口 昌樹 准教授

2025年3月

## 概 要

**キーワード:** 流体シミュレーション, 部分空間法, Snapshot 固有直交分解, Cubature

# 目次

第1章	序論	1
第2章	関連研究	2
2.1	流体シミュレーション	2
2.2	部分空間法	4
第3章	煙の流体シミュレーション	7
3.1	格子法による離散化	7
3.1.1	ディリクレ境界条件	8
3.1.2	外力項計算	9
3.1.3	移流項計算	10
3.1.4	粘性項計算	11
3.1.5	圧力項計算	12
3.2	ボリュームデータの可視化手法	15
第4章	部分空間法	16
4.1	基底空間構築手法	16
4.2	流体ソルバへの部分空間法の適用	16
4.3	部分空間法の課題	16
第5章	部分空間法的高速化手法の提案	17
5.1	Snapshot の分割による高速化	17
5.2	領域分割による高速化	17
5.3	計算機実験	17
5.3.1	計算条件	17
5.3.2	計算結果	17
5.4	課題	17
第6章	結論	18

謝辭	20
参考文献	21

# 第1章 序論

流体シミュレーションとは、コンピュータを用いて、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。表現できる流体は、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。また、2次元と3次元を扱うことができる。2次元では、水面に広がる波紋や、異なる流体が作り出すマarmor模様などがシミュレーションでき、3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。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、工学分野とコンピュータグラフィックスの分野で活用されている。工学分野では、流体に触れる製品の設計・開発において、製品と流体の相互関係を事前にシミュレーションすることで、実験のコスト削減などに役立っている。コンピュータグラフィックスの分野では、水や煙のそれらしいアニメーションを生成し、映像作品やゲームの演出に役立っている。コンピュータグラフィックスの分野では、現実の物理現象に忠実な計算よりも、それらしい振る舞いを高速に計算することや、ユーザーが表現を操作しやすいことが重要視されている。

近代的な高解像度の流体表現は計算負荷が高く、高速化の研究が盛んに行われている。コンピュータグラフィックスの分野における高速化において、大規模並列計算手法の適用性が重要視されている。流体シミュレーションの計算はGPUによる高速化と相性が良く、高速化手法もGPUに適用できるものが望まれている。また、前処理によってシミュレーション計算に利用する行列の次元を削減し計算負荷を削減する、部分空間法が存在する。これらの手法はシミュレーション計算は高速に行うことができるが、前処理の計算負荷や空間計算量が大きいことに注意が必要である。

本研究では、部分空間法の計算負荷削減により、前処理にかかる時間の短縮を目指す。

## 第2章 関連研究

### 2.1 流体シミュレーション

まず、流体シミュレーションの分野で一般的に使われている表記の説明をする。

- $\mathbf{x}$ : 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2次元, または3次元になる.
- $t$ : 時刻を表す変数. シミュレーション開始の時刻を0とする.
- $\Delta x, \Delta t$ : 離散化する計算格子の格子幅, 次の時刻までの時間幅.
- $\mathbf{u}(\mathbf{x}, t)$ : 位置  $\mathbf{x}$ , 時刻  $t$  での流体の速度ベクトル.
- $p(\mathbf{x}, t)$ : 位置  $\mathbf{x}$ , 時刻  $t$  での流体の圧力値.
- $\rho, \nu$ : それぞれ, 流体の密度, 流体の粘性. ここでは位置や時刻によらない定数とする.
- $\mathbf{f}$ : 位置  $\mathbf{x}$ , 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. シミュレーションの手法によっては外力を位置  $\mathbf{x}$ , 時刻  $t$  によって変化する外力を扱うことができるが, 今回は簡単のため, 重力のみを扱い, どの位置でも, どの時刻でも一定の値として考える.
- $\nabla$ : 空間微分演算子ナブラ. スカラー場に作用させると勾配を表し, ベクトル場に作用させると発散を表す. 3次元空間なので,  $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$
- $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u}(\mathbf{x}, t) \cdot \nabla$ : 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.

下記の式 2.1 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. 下記の式 2.2 は流体の連続の式と呼ばれる, 質量保存や流量保存を表す式である.

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2.2の左辺の第二項も0になる。これにより、式2.3と、式2.4を得る。式2.1と式2.4または式2.3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。

$$\frac{\partial}{\partial t} \mathbf{u}(\mathbf{x}, t) = -(\mathbf{u}(\mathbf{x}, t) \cdot \nabla) \mathbf{u}(\mathbf{x}, t) - \frac{1}{\rho} \nabla p(\mathbf{x}, t) + \nu \nabla^2 \mathbf{u}(\mathbf{x}, t) + \mathbf{f} \quad (2.1)$$

$$\frac{D}{Dt} \rho + \nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0 \quad (2.2)$$

$$\frac{D}{Dt} \rho = 0 \quad (2.3)$$

$$\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0 \quad (2.4)$$

一般の非線形微分方程式は厳密な解析的解法が存在せず、様々な条件を設定した上で、数値解法により離散化し、近似解を求めるのが一般的である。ナビエ・ストークス方程式の数値解法は、中間の速度  $\mathbf{u}_i^*(\mathbf{x}, t)$  を用いて、右辺の項について個別に方程式の近似解を求める手法が一般的である。式(2.1)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。

Losasso らが提案した Marker And Cell 法 (MAC 法) [1] と呼ばれる手法は、時間微分の離散化に前進差分を用いてナビエ・ストークス方程式の数値解を求める代表的な手法である。MAC 法は仮の速度  $\mathbf{u}^*$  を用いて以下のようにナビエ・ストークス方程式を分割し、それぞれ個別に計算を行う。ここで、時刻の偏微分に前進差分  $\frac{\partial}{\partial t} \mathbf{u}(\mathbf{x}, t) = \frac{\mathbf{u}(\mathbf{x}, t+\Delta t) - \mathbf{u}(\mathbf{x}, t)}{\Delta t}$  を用いている。

$$\mathbf{u}(\mathbf{x}, t + \Delta t) = \mathbf{u}^* - \Delta t \left( \frac{1}{\rho} \nabla p(\mathbf{x}, t) + \nu \nabla^2 \mathbf{u}(\mathbf{x}, t) + \mathbf{f} \right) \quad (2.5)$$

$$\mathbf{u}^* = \mathbf{u}(\mathbf{x}, t) - \Delta t (\mathbf{u}(\mathbf{x}, t) \cdot \nabla) \mathbf{u}(\mathbf{x}, t) \quad (2.6)$$

ナビエ・ストークス方程式の空間の離散化については、大きく分けて格子法 (Eulerian) と粒子法 (Lagrangian) がある。格子法とは、流体を扱う空間を正方形や立方体の格子に区切り、格子の中心や辺、頂点などに物理量を配置して計算する方法である。区切る格子の数が多ければ、流体の詳細な動きが計算できるが、計算負荷は増加する。格子法の利点は、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ることや、境界条件の設定が容易であることが挙げられる。一方で、格

格子法の欠点は、格子の大きさよりも細かい表現が正確に行えないことや、格子内の流体の粒子の運動を平均化することにより、個々の粒子の詳細な運動を追跡することができないことが挙げられる。これらの欠点により、薄く広がった流体の運動や、水飛沫などを表現することが困難である。

粒子法は、流体の粒子の運動を追跡する手法であり、薄く広がった流体の運動や水飛沫などを表現することが容易である利点がある一方で、計算精度の保証が不十分であることや、境界条件の設定が煩雑になるという欠点がある。

格子法と粒子法は固体にも適用することが可能であり、流体とは異なる支配方程式を用いて変形や移動を計算することができる。流体と固体の相互関係をシミュレーションする際は、格子法と粒子法をそれぞれ個別に適用することが可能である。また、流体の移流項のみを粒子法を用いて計算し、そのほかを格子法で計算するなど、各項の計算について個別に適用する手法が存在する。

## 2.2 部分空間法

部分空間法は、ベクトル空間をより低次元の部分空間に射影し、数値シミュレーションにおける計算負荷を軽減する手法である。流体シミュレーションにおける部分空間法は、モデル縮約やモデル縮退とも呼ばれている。扱うベクトルデータやシミュレーションの計算を低次元の部分空間に射影して行い計算負荷を軽減する。以下では、部分空間法の理論の概要を説明する。

シミュレーションの前処理として、 $n$  次元ベクトル  $\mathbf{x}$  を  $r$  次元ベクトル  $\mathbf{y}$  に変換する、 $n \times r$  直交行列  $\mathbf{A}$  を考える。以下の 2 式が成り立つ。

$$\mathbf{y} = \mathbf{A}^T \mathbf{x}$$

$$\mathbf{x} = \mathbf{A} \mathbf{y}$$

また、 $\mathbf{y}' = \mathbf{A}^T \mathbf{x}'$  を満たす  $n$  次元ベクトル  $\mathbf{x}'$ 、 $r$  次元ベクトル  $\mathbf{y}'$  について、以下の 2 式が成り立つような、任意の  $m \times n$  行列  $\mathbf{F}$ 、任意の  $m \times r$  行列  $\tilde{\mathbf{F}}$  を考える。

$$\mathbf{x}' = \mathbf{F} \mathbf{x}$$

$$\mathbf{y}' = \tilde{\mathbf{F}} \mathbf{y}$$

$\mathbf{y}' = \mathbf{A}^T \mathbf{x}'$  に、 $\mathbf{x}' = \mathbf{F} \mathbf{x}$  と、 $\mathbf{x} = \mathbf{A} \mathbf{y}$  を代入することで、

$$\mathbf{y}' = (\mathbf{A}^T \mathbf{F} \mathbf{A}) \mathbf{y}$$



が得られる． $\mathbf{y}' = \tilde{\mathbf{F}}\mathbf{y}$  と比較すると，

$$\tilde{\mathbf{F}} = \mathbf{A}^T \mathbf{F} \mathbf{A}$$

が得られる．

以上により，任意のベクトルに対する行列積を， $n \times r$  直交行列  $\mathbf{A}$  を用いて部分空間に射影することができる．流体シミュレーションにおいて，部分空間に射影したベクトルを用いてシミュレーション計算をするためには，射影後のベクトルも流体の非圧縮性条件等の性質を満たさなければならない．そのような射影を達成する行列  $\mathbf{A}$  を得る手法として，Snapshot 固有直交分解がある．行列  $\mathbf{A}$  を既存の流体のデータを主成分分析を用いて計算することで，流体の性質を満たした射影が可能である．以降は Snapshot 固有直交分解の具体的な計算方法について説明する．

時刻  $t \{t \in N, 0 \leq t \leq m\}$  の  $n$  次元ベクトルデータ  $\mathbf{u}_t$  を用いて， $n \times m$  行列  $\mathbf{S}$  を定義する．

$$\mathbf{S} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_{m-1} \\ | & | & & | \end{bmatrix}$$

次に，行列  $\mathbf{S}$  主成分分析を行うため， $n \times r$  行列  $\mathbf{A}$  について以下の最小化問題を考える．ここで  $r$  は抽出したい主成分の数である．

$$\min \|\mathbf{S} - \mathbf{A} \mathbf{A}^T \mathbf{S}\|_F^2$$

この最小化問題の解となる行列  $\mathbf{A}$  は， $\mathbf{S} \mathbf{S}^T$  の固有ベクトル  $\mathbf{v}_i$  ( $0 \leq i \leq r-1$ ) を用いて，以下の行列になる．

$$\mathbf{A} = \begin{bmatrix} | & | & & | \\ \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_{r-1} \\ | & | & & | \end{bmatrix}$$

ここで，固有ベクトル  $\mathbf{v}_i$  に対応する固有値を  $\lambda_i$  は， $\lambda_0 \geq \lambda_1 \cdots \geq \lambda_i \geq \lambda_{n-1}$  を満たすとする． $\mathbf{S} \mathbf{S}^T$  は対称行列になっているため，固有ベクトルは互いに直交し， $\mathbf{A}^{-1} = \mathbf{A}^T$  となる．

次に，行列  $\mathbf{A}$  を用いて  $\mathbf{u}$  を  $\tilde{\mathbf{u}}$  に射影するとき， $\tilde{\mathbf{u}}$  が流体の非圧縮性条件  $\nabla \cdot \tilde{\mathbf{u}} = 0$  を満たしていることを確かめる．行列  $\mathbf{S}$  の各列ベクトル  $\mathbf{u}_i$  は既存のシミュレーションの速度データであるため，流体の非圧縮性条件  $\nabla \cdot \mathbf{u}_i = 0$  を満たす．このことから，

$$\nabla \cdot \mathbf{S} = \sum_{0 \leq i \leq n-1} \nabla \cdot \mathbf{u}_i = 0$$

が成り立つ.

また, 行列  $\mathbf{S}\mathbf{S}^T$  の固有ベクトル  $\mathbf{v}_i$  と, それに対応する固有値を  $\lambda_i$  とすると, 固有値と固有ベクトルの定義から以下が成り立つ.

$$\mathbf{S}\mathbf{S}^T \mathbf{v}_i = \lambda_i \mathbf{v}_i$$

両辺に左から  $\nabla \cdot$  を取ることで,

$$(\nabla \cdot \mathbf{S}) \mathbf{S}^T \mathbf{v}_i = \lambda_i \nabla \cdot \mathbf{v}_i$$

$\nabla \cdot \mathbf{S} = 0$  より,  $0$  でない  $\lambda_i$  について,  $\nabla \cdot \mathbf{v}_i = 0$  が成り立つことがわかる.

## 第3章 煙の流体シミュレーション

流体シミュレーションを行う際、境界条件や外力項を適切に設定することで、流体の振る舞いを操作することができる。この章では、文献 [2] を参考に、煙の流体シミュレーションを実装する方法を紹介する。この手法で得られたシミュレーションデータを用いて部分空間法に用いる Snapshot 固有直交分解を行う。

### 3.1 格子法による離散化

格子法による空間の離散化は、扱う格子によって様々な手法が存在する。例えば物理量を格子の中心に配置するコロケート格子や、速度成分を格子の面の中心に配置するスタッガード格子が存在する。コロケート格子は流体の格子の形状によらず適用できるが、境界条件の設定が煩雑である。スタッガード格子は適用が立方体格子などに限定される一方、境界条件の設定が容易である。本論文では、シミュレーション空間をスタッガード格子を用いて離散化する。スタッガード格子は下記の図 3.1 のように、格子の中心を圧力の位置に定義し、格子面の中心に、その格子面と垂直な流速の成分の位置を定義する。例えば  $xy$  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。

図 3.1: スタッガード格子

シミュレーション空間の  $x, y, z$  軸方向の分割数を  $nx, ny, nz$  とする。このとき空間解像度は  $nx \cdot ny \cdot nz$  となり、速度の  $x, y, z$  成分の空間解像度はそれぞれ  $(nx + 1) \cdot ny \cdot nz$ ,  $nx \cdot (ny + 1) \cdot nz$ ,  $nx \cdot ny \cdot (nz + 1)$  となる。本論文では、 $nx = ny = nz$  とし、速度  $\mathbf{v}$  と圧力  $p$  は以下のベクトルに離散化する。

- $\mathbf{v}_x$  :  $(nx + 1) \times ny \times nz$  次元ベクトル.
- $\mathbf{v}_y$  :  $nx \times (ny + 1) \times nz$  次元ベクトル.
- $\mathbf{v}_z$  :  $nx \times ny \times (nz + 1)$  次元ベクトル.

- $\mathbf{v}$  :  $3 \times (nx + 1) \cdot ny \cdot nz$  次元ベクトル.
- $\mathbf{p}$  :  $nx \cdot ny \cdot nz$  次元ベクトル.

ここで  $\mathbf{v}$  の  $v_0$  から  $v_{(nx+1) \cdot ny \cdot nz - 1}$  には速度の  $x$  成分,  $v_{(nx+1) \cdot ny \cdot nz}$  から  $v_{2 \times (nx+1) \cdot ny \cdot nz - 1}$  には速度の  $y$  成分,  $v_{2 \times (nx+1) \cdot ny \cdot nz}$  から  $v_{3 \times (nx+1) \cdot ny \cdot nz - 1}$  には速度の  $z$  成分を保存する. また, 位置  $\mathbf{x} = (i, j, k)$  における速度  $\mathbf{v}(\mathbf{x})$  を, 以下のように表す.

$$\mathbf{v}(\mathbf{x}) = \begin{bmatrix} v_x(i, j, k) \\ v_y(i, j, k) \\ v_z(i, j, k) \end{bmatrix}$$

前進差分を用いた空間の偏微分の離散化の例として, スタッガード格子を用いた以下の偏微分の離散化を上記のベクトルを用いて表現することを考える.

$$\frac{\partial}{\partial x} \mathbf{u}(\mathbf{x}) = \frac{\mathbf{u}(\mathbf{x} + \Delta \mathbf{x}) - \mathbf{u}(\mathbf{x})}{\Delta x}$$

ここで,  $\mathbf{u}(\mathbf{x})$  は位置  $\mathbf{x}$  での速度を表す三次元ベクトルである. スタッガード格子の一辺の長さを  $\Delta L = \Delta x$ , 流体の位置ベクトル  $\mathbf{x}$  を自然数  $i$  ( $0 \leq i \leq nx - 1$ ),  $j$  ( $0 \leq j \leq ny - 1$ ),  $k$  ( $0 \leq k \leq nz - 1$ ) を用いて,  $\mathbf{x} = (i\Delta L, j\Delta L, k\Delta L)$  とすると, 以下ようになる.

$$\frac{\mathbf{u}(\mathbf{x} + \Delta \mathbf{x}) - \mathbf{u}(\mathbf{x})}{\Delta x} = \begin{bmatrix} \frac{v_x(i+1, j, k) - v_x(i, j, k)}{\Delta L} \\ \frac{v_y(i, j+1, k) - v_y(i, j, k)}{\Delta L} \\ \frac{v_z(i, j, k+1) - v_z(i, j, k)}{\Delta L} \end{bmatrix}$$

### 3.1.1 ディリクレ境界条件

流体シミュレーションにおいて, 流体と流体以外の境界面の物理量の条件を設定する必要がある場面がある. これを境界条件と呼び, 文献 [2] では境界における物理量を事前に一定の値に設定するディリクレ境界条件が用いられており, 境界における物理量は 0 と設定する. ディリクレ境界条件は速度成分を都度初期化することで適用できるほか, 境界成分に対応する成分を 0 とした単位行列を用いた表現も可能である. ディリクレ境界条件を表現する行列を  $\mathbf{D}$  とすると,  $\mathbf{D}$  の  $i, j$  成分は以下のように定義される.

$$\mathbf{D}_{i, j} = \begin{cases} 1 & i = j \text{ かつ } i \text{ は境界成分でない} \\ 0 & \text{その他} \end{cases} \quad (3.1)$$

この行列表現は通常の流体シミュレーションでは不要であるが, 部分空間法を用いた流体計算では境界成分に対応する値を変更することが困難なため, 境界条件を行列積を用いて

表現する手法が用いられている．以降は気体の流体シミュレーションにおけるナビエ・ストークス方程式の計算方法を説明する．

$$\mathbf{u}(\mathbf{x}, t + \Delta t) = \mathbf{u}^* - \Delta t \left( \frac{1}{\rho} \nabla p(\mathbf{x}, t) + \nu \nabla^2 \mathbf{u}(\mathbf{x}, t) + \mathbf{f} \right) \quad (3.2)$$

$$\mathbf{u}^* = \mathbf{u}(\mathbf{x}, t) - \Delta t (\mathbf{u}(\mathbf{x}, t) \cdot \nabla) \mathbf{u}(\mathbf{x}, t) \quad (3.3)$$

上記の式 3.2 と式 3.3 を以下のように各項ごとに分割して計算する．

$$\mathbf{u}_0 = \mathbf{u}(t) - \Delta t \mathbf{f} \quad (3.4)$$

$$\mathbf{u}_1 = \mathbf{u}_0 - \Delta t (\mathbf{u}_0 \cdot \nabla) \mathbf{u}_0 \quad (3.5)$$

$$\mathbf{u}_2 = \mathbf{u}_1 - \Delta t \nu \nabla^2 \mathbf{u}_1 \quad (3.6)$$

$$\mathbf{u}(t + \Delta t) = \mathbf{u}_3 = \mathbf{u}_2 - \Delta t \frac{1}{\rho} \nabla p \quad (3.7)$$

ここで、式 3.4 は外力項、式 3.12 は移流項、式 3.6 は粘性項、式 3.6 は粘性項、式 3.7 は圧力項を計算する式である．

### 3.1.2 外力項計算

外力項では、再現したい現象になるような外力を与えなければならない．コンピュータグラフィックスにおける気体の外力は、気体にかかる重力と、温度差による対流を発生させる力、気体が渦巻くように与える力などをユーザーがパラメータによって調整できるようにする計算方法が一般的である．本実験では重力と、対流を発生させる力を加える．よりそれらしい振る舞いを再現するため、他の計算では定数として扱う密度や温度を、ここでは位置と時刻によって変化するデータとして計算する．位置  $\mathbf{x}$  における気体の密度、温度をそれぞれ  $\rho(\mathbf{x})$ ,  $T(\mathbf{x})$ , 環境温度を  $T_{amb}(\mathbf{x})$ , 重力の方向ベクトルを  $\mathbf{d}$  とすると、気体に働く外力は以下ようになる．

$$\mathbf{f}_{buoy}(\mathbf{x}) = \alpha \rho(\mathbf{x}) \mathbf{d} + \beta (T(\mathbf{x}) - T_{amb}(\mathbf{x})) \mathbf{d} \quad (3.8)$$

ここで、 $\alpha$ ,  $\beta$  はユーザーが設定するパラメータであり、それぞれ重力と対流の力を調整する．最終的な外力項計算は以下ようになる．

$$\mathbf{u}_0 = \mathbf{u} - \Delta t \mathbf{f}_{buoy} \quad (3.9)$$

### 3.1.3 移流項計算

移流項は流体の物理量が流体の速度場によって移動する様を計算することができる．移流項は非線形方程式であり，様々な計算方法が存在する．文献 [2] の Semi-Lagrangian 法を紹介する．Semi-Lagrangian 法は計算負荷が小さく，陰解法により安定した計算結果を得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広く用いられている．

Semi-Lagrangian 法は，格子に配置されている計算点上の物理量を計算する格子法（Eulerian）の特徴と，移動する計算点を追跡し計算を行う粒子法（Lagrangian）の特徴を両方持っている計算手法である．移流項の計算

$$\mathbf{u}_1 = \mathbf{u}_0 - \Delta t (\mathbf{u}_0 \cdot \nabla) \mathbf{u}_0$$

を変形し，時刻の偏微分に対し前進差分  $\frac{\partial \mathbf{u}}{\partial t} = \frac{\mathbf{u}_1 - \mathbf{u}_0}{\Delta t}$  すると，以下の式が得られる．

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}_0 \cdot \nabla) \mathbf{u}_0 = 0$$

この式は以下の輸送方程式と呼ばれる式の特殊な場合であり，輸送方程式は一般の物理量  $\phi$  に関して適用可能である．

$$\frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla) \phi = 0 \quad (3.10)$$

この輸送方程式が成り立つとき，位置  $\mathbf{x}$ ，時刻  $t$  における物理量  $\phi(\mathbf{x}, t)$  について，以下の式が成り立つ．

$$\phi(\mathbf{x}, t) = \phi(\mathbf{x} - \Delta t \mathbf{u}, t) \quad (3.11)$$

Semi-Lagrangian 法を以下のように流速に適用することで移流項の計算を行う．また，密度や温度などに対して適用することで，空間の速度ベクトル場に沿って気体が運動する様子を再現することができる．

$$\mathbf{u}_1(\mathbf{x}) = \mathbf{u}_0(\mathbf{x} - \Delta t \mathbf{u}_0) \quad (3.12)$$

以下では，輸送方程式 3.10 が成り立つとき，式 3.11 が成り立つことを確認する． $\mathbf{x}$  にある物理量は  $\mathbf{u}(\mathbf{x})$  に沿って  $\Delta t \mathbf{u}(\mathbf{x})$  移動すると仮定すると，移動前の物理量は  $\phi(\mathbf{x} - \Delta t \mathbf{u}(\mathbf{x}))$  となる．この間の物理量の変化  $\Delta \phi$  は以下ようになる．

$$\Delta \phi = \phi(\mathbf{u}(\mathbf{x}), t) - \phi(\mathbf{x} - \Delta t \mathbf{u}(\mathbf{x}), t)$$

これを一次のテイラー展開を用いて表すと，

$$\Delta \phi = \frac{\partial \mathbf{u}(\mathbf{x})}{\partial \mathbf{x}} \phi(\mathbf{x}, t) \Delta t + \frac{\partial \phi(\mathbf{x}, t)}{\partial t} \Delta t$$

$\frac{\partial \mathbf{u}(\mathbf{x})}{\partial \mathbf{x}} \phi(\mathbf{x}, t) = (\mathbf{u}(\mathbf{x}) \cdot \nabla) \phi(\mathbf{x}, t)$  より,

$$\frac{\Delta \phi}{\Delta t} = (\mathbf{u}(\mathbf{x}) \cdot \nabla) \phi(\mathbf{x}, t) + \frac{\partial \phi(\mathbf{x}, t)}{\partial t}$$

右辺が輸送方程式 3.10 になっていることから,  $\Delta \phi = 0$  が成り立つ.  $\Delta \phi = \phi(\mathbf{u}(\mathbf{x})) - \phi(\mathbf{x} - \Delta t \mathbf{u}(\mathbf{x})) = 0$  より, 式 3.11 が成り立つことが確認できる.

### 3.1.4 粘性項計算

粘性項は流体が徐々に周囲に広がっていく様を計算することができる. 粘性項計算は  $\Delta t \nu \nabla^2$  を行列を用いて表現することで, 以下のように行列積を用いて表すことができる.

$$\mathbf{u}_2 = \mathbf{u}_1 - \Delta t \nu \nabla^2 \mathbf{u}_1 = (\mathbf{I} - \Delta t \nu \mathbf{L}) \mathbf{u}_1 = \mathbf{W} \mathbf{u}_1 \quad (3.13)$$

ここで,  $\nabla^2$  の離散化を行列  $\mathbf{L}$  を用いて表すことを考える. 行列  $\mathbf{L}$  を  $v_0$  から  $v_{n-1}$  の  $n$  個の頂点を持つ無向グラフに対して離散ラプラシアンと呼び,  $n \times n$  行列  $\mathbf{L}$  の成分  $\mathbf{L}_{i,j}$  は以下のように定義される. ただし, 無向グラフは自己ループを持たないとする.

$$\mathbf{L}_{i,j} = \begin{cases} \deg(v_i) & i = j \\ -1 & v_i \text{ と } v_j \text{ が接続している} \\ 0 & \text{その他} \end{cases} \quad (3.14)$$

スタaggerド格子を用いた流体シミュレーションにおいて, 無向グラフは流体のそれぞれの速度成分ごとの連結成分を持つとする. つまり, 本論文のシミュレーションにおいては連結成分数は 3 であり, それぞれの頂点数は  $(n_x + 1) \times n_y \times n_z$  である. また, シミュレーション境界を除く速度に対応する次数は 6 であり, シミュレーション境界上の成分はディリクレ境界条件によって速度成分が 0 になるため, 任意の値でよい. よって粘性項計算を表現する行列  $\mathbf{W}$  の成分  $\mathbf{W}_{i,j}$  は以下のように定義される.

$$\mathbf{W}_{i,j} = \begin{cases} 1 - 6\Delta t \nu & i = j \\ \Delta t \nu & v_i \text{ と } v_j \text{ が接続している} \\ 0 & \text{その他} \end{cases}$$

### 3.1.5 圧力項計算

圧力項計算は、流体の非圧縮性を満たすために流体に圧力勾配による外力を与える計算である。

$$\mathbf{u}(t + \Delta t) = \mathbf{u}_2 - \Delta t \frac{1}{\rho} \nabla p$$

両辺の発散をとると、

$$\nabla \cdot \mathbf{u}(t + \Delta t) = \nabla \cdot \mathbf{u}_2 - \Delta t \frac{1}{\rho} \nabla^2 p$$

が得られる。流体の非圧縮性条件、 $\nabla \cdot \mathbf{u}(t + \Delta t) = 0$ により、左辺は0になるため、

$$\nabla \cdot \mathbf{u}_2 = \frac{\Delta t}{\rho} \nabla^2 p \quad (3.15)$$

が得られる。この式を圧力に関するポアソン方程式として解き、 $\mathbf{u}(t + \Delta t) = \mathbf{u}_2 - \Delta t \frac{1}{\rho} \nabla p$ に代入することで圧力項計算が完了する。ポアソンの境界条件にはディリクレ境界条件を適用する。

図 3.2: 圧力計算のモデル

$$\nabla \cdot \mathbf{u}_2 = \frac{\Delta t}{\rho} \nabla^2 p$$

スタaggerド格子では圧力と流速の配置されている位置が異なるため、空間微分に対して同じ差分は適応できない。そこで両辺をセルの領域で積分することを考える。

$$\int_V \nabla \cdot \mathbf{u}_2 dV = \int_V \frac{\Delta t}{\rho} \nabla^2 p dV$$

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。

$$\oint_V \mathbf{u}_2 \cdot \mathbf{n} dV = \oint_V \frac{\Delta t}{\rho} \nabla p \cdot \mathbf{n} dV$$

左辺から空間微分が消去できたため、右辺の空間微分に対して適切な微分スキームを適用することで、周回積分を離散化することができる。ここで離散化のため、 $x$ 軸方向に $i$ 番目、 $y$ 軸方向に $j$ 番目、 $z$ 軸方向に $k$ 番目の格子を $g_{i,j,k}$ 、圧力を $p_{i,j,k}$ と定義し、図3.2のようなモデルを考える。

まず、整数 $n = 0, 1, 2, \dots, 5$ までを、それぞれ格子の各面の方向に割り振る。図3.2の場合、 $n = 0$ は $x$ 軸正の方向、 $n = 1$ は $y$ 軸正の方向、 $n = 2$ は $x$ 軸負の方向、 $n = 3$ は $y$ 軸負の方向、 $n = 4$ は $z$ 軸正の方向、 $n = 5$ は $z$ 軸負の方向に割り振る。ここで、この整数 $n$ を用いて、以下の配列や変数を定義する。



- $F_n$  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.

図 3.2 の場合,  $n = 0$  に対応する  $x$  軸正の方向と,  $n = 5$  に対応する  $z$  軸負の方向は壁に隣接していて格子がないため,  $F_0, F_5 = 0, F_1, F_2, F_3, F_4 = 1$  となる.

- $D_n$  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. 向いている方向が座標軸の正の向きなら +1, 負の向きなら -1 を取る整数値.

図 3.2 の場合, 座標軸の正の方向に向いているのは  $n = 0, n = 1, n = 4$  に対応する方向で, それ以外は負の方向に向いているため,  $F_0, F_1, F_4 = 1, F_2, F_3, F_5 = -1$  となる.

- $p_n$  割り振られた方向に隣り合っている格子の圧力値. 図 3.2 の場合,  $p_0 = p_{i+1,j,k}, p_1 = p_{i,j+1,k}, p_2 = p_{i-1,j,k}, p_3 = p_{i,j-1,k}, p_4 = p_{i,j,k+1}, p_5 = p_{i,j,k-1}$  となるが, 境界条件を考えると, 壁に隣接している格子との圧力値の勾配を 0 にするため,  $p_0 = p_{i,j,k}, p_5 = p_{i,j,k}$  である.
- $\nabla p_n$  割り振られた方向の圧力勾配. 前方差分によって,  $\nabla p_n(\mathbf{x}, t) = \frac{p_n - p_{i,j,k}}{\Delta x}$  と近似できる.
- $u_n$   $g_{i,j,k}$  に配置されている流速値で, 割り振られた方向の格子面に配置されている流速値. 壁に隣接している方向は流速を 0 にするため,  $u_0 = 0, u_5 = 0$  である.

定義した配列や変数を用いて, 周回積分の形で表された式を離散化することで,

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

が得られ, 圧力勾配を  $\nabla p_n(\mathbf{x}, t) = \frac{p_n - p_{i,j,k}}{\Delta x}$  で近似すると,

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

が得られる. この式を整理すると,

$$\sum_n \mathbf{u}_n^* D_n F_n = \sum_n \frac{\Delta t}{\rho} \frac{p_n(\mathbf{x}, t) - p_{i,j,k}(\mathbf{x}, t)}{\Delta x} F_n \quad (3.16)$$

が得られる. 式 (3.16) は  $p$  についての連立一次方程式となっている. 全ての格子に対して式 (3.16) を考えることになるため, 連立方程式は規模が大きくなる. 従って, 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する．連立方程式の解法としては、前処理付き共役勾配法がよく用いられている．

$$\sum_n \mathbf{u}_n^* D_n F_n = \sum_n \frac{\Delta t}{\rho} \frac{\mathbf{p}_n(\mathbf{x}, t) - \mathbf{p}_{i,j,k}(\mathbf{x}, t)}{\Delta \mathbf{x}} F_n \quad (3.17)$$

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、 $\mathbf{Ax} = \mathbf{b}$  に対応させたものを解く． $\mathbf{A}$ 、 $\mathbf{b}$  それぞれのサイズは、 $\mathbf{A}$  は  $n \times n$  であり、 $\mathbf{b}$  は  $n$  である．この際、3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  $iN_yN_z + jN_x + k$  成分にうつす．また、 $\mathbf{A}$  の  $(iN_yN_z + jN_x + k, lN_yN_z + mN_x + n)$  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．通常の行列のままだと空間計算量や時間計算量をが非常に大きくなってしまいが、 $\mathbf{A}$  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を用いることで、空間計算量や時間計算量を削減できる．連立方程式の解法は、未知数が多いため、直接法で計算するのは現実的ではない．反復法では、必要な精度に解が収束したら計算を打ち切るため、解の収束が速い方が望ましい．前処理をすることで解の収束が速くなるため、前処理付き共役勾配法を採用し、Eigen の前処理付き共役勾配法を計算する関数を用いて連立方程式を解く． $\mathbf{A}$  と  $\mathbf{b}$  の各成分は以下のように計算する．

- $\mathbf{A}$ 、 $\mathbf{b}$  を初期化する．この際、 $\mathbf{b}$  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、standergrid の  $(i, j, k)$  に格納されている粒子の数を見て、0なら  $\mathbf{A}$  の  $(iN_yN_z + jN_x + k, iN_yN_z + jN_x + k)$  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 (3.16) 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、 $\mathbf{b}$  の  $(iN_yN_z + jN_x + k)$  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  $iN_yN_z + jN_x + k$  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．

## 3.2 ボリュームデータの可視化手法

ボリュームデータを三次元的な広がりの様子を把握できるようにレンダリングする手法を、ボリュームレンダリングという。描画対象の表面のみをレンダリングするサーフェスレンダリングとは異なり、光の散乱や吸収などの光学的性質を利用する。立方体格子を用いた気体の流体シミュレーションにおいては、一般的にスライスベースのボリュームレンダリングが用いられている。スライスベースのボリュームレンダリングは、シミュレーション空間がボクセル空間として捉えられることを利用し、気体の密度を用いてボクセルの透明度や色を計算する。その後、シミュレーション空間をボクセルの透明度や色をテクスチャとしてマッピングした複数の平面（スライス）に分割し、それらを順次描画することで最終的な画像を生成する。

## 第4章 部分空間法

### 4.1 基底空間構築手法

### 4.2 流体ソルバへの部分空間法の適用

### 4.3 部分空間法の課題

## 第5章 部分空間法的高速化手法の提案

### 5.1 Snapshot の分割による高速化

### 5.2 領域分割による高速化

### 5.3 計算機実験

#### 5.3.1 計算条件

#### 5.3.2 計算結果

### 5.4 課題

## 第6章 結論

$$\frac{\partial}{\partial t} \mathbf{u}(\mathbf{x}, t) = -\mathbf{u}(\mathbf{x}, t) \frac{\partial}{\partial \mathbf{x}} \mathbf{u}(\mathbf{x}, t) \quad (6.1)$$

図・表には通し番号と見出し (caption) を付け、本文中で当該の図・表に言及する。また、単位や目盛を正確に記す。図のタイトルは図の下に、表のタイトルは表の上に書く。

例を図 6.1 と表 6.1 に示す。第 2 章の図には図 2.1, 図 2.2, 図 2.3, …のように番号が振られ、第 2 章の表には表 2.1, 表 2.2, 表 2.3, …のように、図とは独立に番号が振られる。

一例として、和文の著書 [?], 和文の論文誌 [?], 英文の編書 [?], 英文の論文誌 [?], 国際会議 [?], 修士論文 [?], 電子雑誌 [?], Web ページ [?] 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).

図 6.1: 情報工学科のロゴ

表 6.1: 表のタイトル

項目 1	項目 2	項目 3
データ 1	データ 2	データ 3
データ 1	データ 2	データ 3
データ 1	データ 2	データ 3

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 [?] 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．

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森口 昌樹 准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] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *The Physics of Fluids*, 8 (12) : 2182–2189, 1965.
- [2] R. Fedkiew, J. Stam, H. Jensen. Visual simulation of smoke. In *Proceedings of SIGGRAPH 01*, 15–22, 2001.
- [3] J. Stam. Stable Fluids. In *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, pages 121–128, 1999.
- [4] T. Kim, J. Delaney. Subspace Fluid Re-Simulation. *ACM Transactions on Graphics*, 32 (4) : 62:1–62:9, 2013.
- [5] A. Jones, P. Sen, and T. Kim. Compressing fluid subspaces. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 77–84, 2016.
- [6] M. Wicke, M. Stanton, and A. Treuille, Modular Bases for Fluid Dynamics. *ACM Transactions on Graphics*, 39:1–39:8, 2009.

が得られる.