煙シミュレーションのための部分空間法の高速化

須之内　俊樹

中央大学理工学研究科　情報工学専攻
形状情報処理研究室　23N8100018B

2025 年 2 月 21 日

概要

- 流体と接する製品の設計・製造
- 物理的に正確なシミュレーション

研究背景

流体シミュレーション

工業分野

- 流体と接する製品の設計・製造
- 物理的に正確なシミュレーションが重要視

CG 分野

- 流体の映像の生成に利用.
- 計算負荷を少なくすること，流体の挙動が制御しやすいことが重要視.
- 物理的な正確さよりも，それらしさを重要視.

流体シミュレーションの課題

- 希薄な流体や，水飛沫は手法によっては再現できない.
- 近年は高品質な映像が求められ，計算負荷が大きい.

研究背景

流体シミュレーションの数理モデル

ナビエ・ストークス方程式

$$\frac{\partial}{\partial t}\boldsymbol{u} = -(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}$$

$$\nabla\cdot\boldsymbol{u} = 0$$

- $\boldsymbol{u}$，$\boldsymbol{f}$: 位置 $\boldsymbol{x}$ での流体の速度，外力
- p，ρ，ν: 位置 $\boldsymbol{x}$ での流体の圧力，密度，粘性
- $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$

研究背景

流体シミュレーションの数理モデル

部分段階法

$$\frac{\partial}{\partial t}\boldsymbol{u} = -(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}$$

$$\frac{\partial}{\partial t}\boldsymbol{u} = -(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}$$

研究背景

流体シミュレーションの数理モデル

部分段階法

$$\frac{\partial}{\partial t}\boldsymbol{u} = -(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}$$

この式を，中間子 $\boldsymbol{u}_0$ から $\boldsymbol{u}_3$ を用いて，以下のように各項ごとに分割して計算する．

$$\boldsymbol{u}_0 = \boldsymbol{u}(t) - \Delta t\boldsymbol{f} \tag{1}$$

$$\boldsymbol{u}_1(\boldsymbol{x}) = \boldsymbol{u}_0(\boldsymbol{x}) - \Delta t(\boldsymbol{u}_0(\boldsymbol{x})\cdot\nabla)\boldsymbol{u}_0(\boldsymbol{x}) \tag{2}$$

$$\boldsymbol{u}_2 = \boldsymbol{u}_1 - \Delta t\nu\nabla^2\boldsymbol{u}_1 \tag{3}$$

$$\boldsymbol{u}(t+\Delta t) = \boldsymbol{u}_3 = \boldsymbol{u}_2 - \Delta t\frac{1}{\rho}\nabla\boldsymbol{p} \tag{4}$$

研究背景

流体シミュレーションの数理モデル

部分段階法

$$\frac{\partial}{\partial t}\boldsymbol{u} = -(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}$$

$$\boldsymbol{u}_0 = \boldsymbol{u}(t) - \Delta t\boldsymbol{f}$$

$$\boldsymbol{u}_1(\boldsymbol{x}) = \boldsymbol{u}_0(\boldsymbol{x} - \Delta t\boldsymbol{u}_0)$$

$$\boldsymbol{u}_2 = \boldsymbol{V}\boldsymbol{u}_1$$

$$\boldsymbol{b} = \boldsymbol{W}\boldsymbol{u}_2$$

$$\boldsymbol{p} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

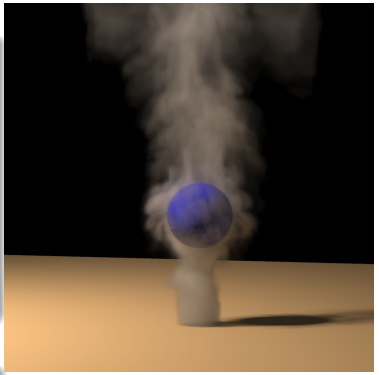
$$\boldsymbol{u}_3 = \boldsymbol{u}_2 - \boldsymbol{Y}\boldsymbol{p}$$

研究背景

先行研究

Visual Simulaton of Smoke [1] [Fedkiw et al. 2001]

- CG において広く用いられる煙のオフラインシミュレーション手法．
- 密度を計算した後，ボリュームレンダリングを用いて描画．
- 陰解法による計算の安定性．
- 計算量は空間解像度に依存．



手法の課題

近年の高品質な映像に不向き

研究背景

先行研究

ボリュームレンダリング

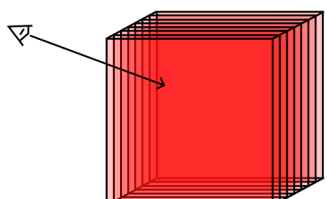
- 3 次元空間の全てのボクセル値を可視化画像に反映させるレンダリング手法．
- 透明度を考慮したレンダリングに用いられる．
- サーフェスレンダリングは物体表面のみ可視化される．

Algorithm 1 Axis Alined slice-based Volume Rendering

1: x，y，z 軸の中から，視線方向に近いものを選ぶ．
2: 軸に垂直な平面（スライス）を，アルファブレンディングして描画．

$$\mathrm{I_n}(\boldsymbol{x}) = \alpha_n(\boldsymbol{x})c_n(\boldsymbol{x}) + (1 - \alpha_n(\boldsymbol{x}))\mathrm{I_{n-1}}(\boldsymbol{x})$$

$$\alpha(\boldsymbol{x}) = 1 - exp(-\rho(\boldsymbol{x})\Delta s)$$



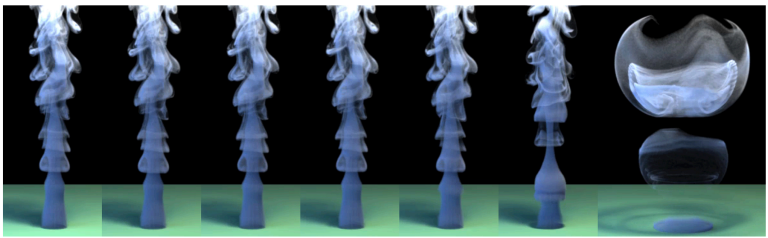
部分空間法 (subspace method)

部分空間法 [2] [Kim et al. 2013]

- 行列の低ランク近似を用いた手法．
- シミュレーションの前処理として，n 次元ベクトル $\boldsymbol{x}$ を r 次元ベクトル $\boldsymbol{y}$ に変換する $n \times r$ 行列 $\mathbf{A}$ を考える．

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

- $\boldsymbol{y}$ を用いてシミュレーションを行い，$\boldsymbol{x} = \mathbf{A}^T\boldsymbol{y}$ を用いて $\boldsymbol{x}$ を求める．
- 空間解像度を保ったまま高速化可能．



部分空間法 (subspace method)

Snapshot 固有直交分解

- 既存のシミュレーションの時系列データを m 個用いて，行列 $\mathbf{A}$ を作成する手法．
- 空間解像度を n とする．一般に，$n \gg m \gg r$
- 時刻 t の速度ベクトル $\boldsymbol{u_t}$ を用いて，$3n \times m$ 行列 $\mathbf{U}$ を定義する．

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

- 行列 $\mathbf{A}$ について，以下のエネルギーを考える．

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

- $\mathbf{A}$ を $\mathbf{U}^T\mathbf{U}$ の固有ベクトルとすることで最小化できる．
- snapshot の数，削減後の次元の数が多いほど，削減前との誤差が少なくなる．

部分空間法の課題

空間計算量

一般に，

$$64^3 \leq n \leq 1024^3$$

$$30 \leq r \leq 150$$

- $n = 256^3$，$r = 50$ のとき，行列 **A** のために 10GB ほど必要．
- 削減後の次元の数に制限．
- GPU による高速化が不可能．

表現の限界

- 既存のシミュレーションと大幅に異なる表現．

改善手法

離散コサイン変換（DCT）を用いた行列の圧縮と展開 [3] [Jones et al. 2016]

- $8 \times 8 \times 8$ 個の格子を 1 ブロックとする．
- 1 ブロックずつ展開し，GPU にデータを送信する．
- 圧縮率が高すぎると視覚的に問題．
- より疎な行列による空間計算量の削減が可能であるとされる．

領域分割 [4] [Wicke et al. 2009]

- シミュレーション空間を分割し，小領域の計算を繰り返す．
- 速度の発散を 0 にするような分割を行う．
- 計算誤差は大きく，計算負荷が増加する．

領域分割後の小領域に，離散コサイン変換を適用．更なる空間計算量の削減．
計算結果の精度が低下する．

実験

流体と熱源を配置し，煙が立ち上る様子をシミュレーションした．
空間解像度 $n = 64^3$
シミュレーション 120ms/f，ボリュームレンダリング 15ms/f

[1] R. Fedkiew, J. stam, H. Jensen. Visual simulation of smoke. In *Proceedings of SIGGRAPH 01*, 15–22, 2001.

[2] T. Kim, J. Delaney. Subspace Fluid Re-Simulation. *ACM Transactions on Graphics*,32, (4): , 62:1–62:9, 2013.

[3] A. Jones, P. Sen, and T. Kim. Compressing fluid subspaces. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 77–84, 2016.

[4] M. Wicke, M. Stanton, and A. Treuille, Modular Bases for Fluid Dynamics. *ACM Transactions on Graphics*, 39:1–39:8, 2009.