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

- 部分空間法：流体シミュレーションの高速化手法
- 部分空間法の前処理の計算負荷が膨大
- 提案手法による計算負荷の軽減と，シミュレーションの変化を評価

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
- 物理的な正確さ

CG 分野

- 流体の映像の生成に利用
- 計算負荷の軽減，流体の挙動が制御しやすさ
- 物理的な正確さよりも，それらしさ

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

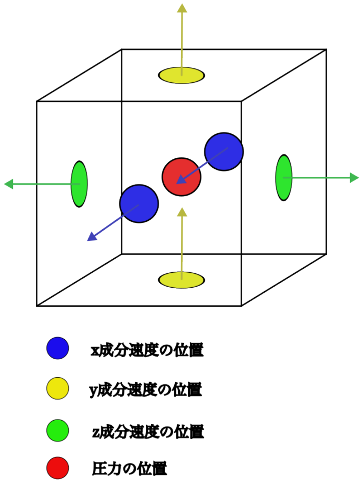
ナビエ・ストークス方程式の離散化

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

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

スタッガード格子

- 速度と圧力の配置位置を工夫し，計算の安定性を向上
- 非直交格子への適用が困難



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

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

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

$$\begin{aligned}
\boldsymbol{u}_0 &= \boldsymbol{u}(t) - \Delta t\boldsymbol{f} \\
\boldsymbol{u}_1(\boldsymbol{x}) &= \boldsymbol{u}_0(\boldsymbol{x}) - \Delta t(\boldsymbol{u}_0(\boldsymbol{x})\cdot\nabla)\boldsymbol{u}_0(\boldsymbol{x}) \\
\boldsymbol{u}_2 &= \boldsymbol{u}_1 - \Delta t\nu\nabla^2\boldsymbol{u}_1 \\
\boldsymbol{u}(t+\Delta t) = \boldsymbol{u}_3 &= \boldsymbol{u}_2 - \Delta t\frac{1}{\rho}\nabla\boldsymbol{p}
\end{aligned}$$

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

$$
\begin{aligned}
\boldsymbol{u}_0 &= \boldsymbol{u}(t) - \Delta t \boldsymbol{f} \\
\boldsymbol{u}_1(\boldsymbol{x}) &= \boldsymbol{u}_0(\boldsymbol{x} - \Delta t \boldsymbol{u}_0) && (1) \\
\boldsymbol{u}_2 &= \boldsymbol{V}\boldsymbol{u}_1 \\
\boldsymbol{b} &= \boldsymbol{W}\boldsymbol{u}_2 && (2) \\
\boldsymbol{p} &= \boldsymbol{A}^{-1}\boldsymbol{b} && (3) \\
\boldsymbol{u}(t+\Delta t) = \boldsymbol{u}_3 &= \boldsymbol{u}_2 - \boldsymbol{Y}\boldsymbol{p}
\end{aligned}
$$

- 式 1 は semi-Lagrangian 法 [3]，式 2，3 はコリンの射影法 [1]
- $\boldsymbol{V}$，$\boldsymbol{A}$ は離散ラプラシアン行列を用いて ∇^2 を離散化
- $\boldsymbol{W}$，$\boldsymbol{Y}$ は勾配演算子 ∇ を離散化

研究背景

先行研究

Visual Simulaton of Smoke [4] [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
- 計算量は空間解像度に依存

計算機実験

- OS:Apple M4 pro，メモリ:64GB
- 解像度 128^3，200 フレーム
- シミュレーション 1800ms/f
- レンダリング 240ms/f

部分空間法 (subspace method)

- 部分空間に射影し，計算負荷を軽減する手法
- 空間解像度を保ったまま高速化可能
- 前処理の計算負荷が大きい

概要

- シミュレーションの前処理として，$\boldsymbol{A}^{\top}\boldsymbol{A} = \boldsymbol{I}$ を満たし，$\boldsymbol{x} = \boldsymbol{A}\tilde{\boldsymbol{x}}$ が成り立つような，$n \times r$ 行列 $\boldsymbol{A}$ を考える

$$\tilde{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$$

- $\tilde{\boldsymbol{x}}$ を用いてシミュレーションを行い，$\boldsymbol{x} = \boldsymbol{A}^{T}\tilde{\boldsymbol{x}}$ を用いて $\boldsymbol{x}$ を求める
- $\tilde{\boldsymbol{u}}$ は，流体の非圧縮条件 $\nabla \cdot \tilde{\boldsymbol{u}} = 0$ を満たしていなければならない
- 本研究の高速化の対象は，性質の良い直交基底 $\boldsymbol{A}$ の計算

研究背景

部分空間法

特異値分解を用いた直交基底の計算

- 既存のシミュレーションの時系列データを T 個用いて，行列 $\boldsymbol{A}$ を作成する手法
- 空間解像度を n とする．一般に，$n \gg T \gg r$
- 時刻 t における速度ベクトル $\boldsymbol{u}_t$ を用いて，$3(n+1) \times T$ 行列 $\boldsymbol{S}$ を定義する

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_{T-1} \end{bmatrix}$$

- $\boldsymbol{S}$ を以下のように特異値分解して得られるユニタリ行列 $\boldsymbol{U}$ を $\boldsymbol{A}$ とする

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$$

- 先に QR 分解を適用し，サイズを小さくすることができる

研究背景

先行研究

線形項 [2][Treuille et al. 2006]

- 中間子 $\boldsymbol{u}_i$ から得た基底を $\boldsymbol{U}_i$，$\boldsymbol{p}$ から得られた基底を $\boldsymbol{P}$ とする
- $n \times n$ 行列が $r \times r$ 行列に．各ステップが高速化

$$\boldsymbol{U}_2^\top \boldsymbol{u}_2 = (\boldsymbol{U}_2^\top \boldsymbol{V} \boldsymbol{U}_1)\boldsymbol{U}_1^\top \boldsymbol{u}_1 \qquad \widetilde{\boldsymbol{u}}_2 = \widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{u}}_1$$

$$\boldsymbol{P}^\top \boldsymbol{b} = (\boldsymbol{P}^\top \boldsymbol{W} \boldsymbol{U}_2)\boldsymbol{U}_2^\top \boldsymbol{u}_2 \qquad \widetilde{\boldsymbol{b}} = \widetilde{\boldsymbol{W}}\widetilde{\boldsymbol{u}}_2$$

$$\boldsymbol{P}^\top \boldsymbol{p} = (\boldsymbol{P}^\top \boldsymbol{A}^{-1} \boldsymbol{P})\boldsymbol{P}^\top \boldsymbol{b} \qquad \widetilde{\boldsymbol{p}} = \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{b}}$$

$$\boldsymbol{U}_3^\top \boldsymbol{u}_3 = \boldsymbol{U}_2^\top \boldsymbol{u}_2 - (\boldsymbol{U}_3^\top \boldsymbol{Y} \boldsymbol{P})\boldsymbol{P}^\top \boldsymbol{p} \qquad \widetilde{\boldsymbol{u}}_3 = \widetilde{\boldsymbol{u}}_2 - \widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{p}}$$

非線形項

- 立体求積法 [5][Kim et al.2013] を用いる．
- 本研究の高速化の対象ではない

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 $\boldsymbol{A}$ のために 10GB ほど必要
- 削減後の次元の数に制限．$r \leq T$
- GPU による高速化が不可能

前処理の計算時間

- 線傾項：$3(n+1) \times T$ 行列の特異値分解，QR 分解の計算量は $O(nT^2)$
- 非線形項：$O(rTP^3)$

提案手法

スナップショットの分割

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_{T-1} \end{bmatrix}$$

二分割すると，

$$\boldsymbol{S}_\mathbf{0} = \begin{bmatrix} \boldsymbol{u}_0 & \cdots & \boldsymbol{u}_{T/2-1} \end{bmatrix} \qquad \boldsymbol{S}_\mathbf{1} = \begin{bmatrix} \boldsymbol{u}_{T/2} & \cdots & \boldsymbol{u}_{T-1} \end{bmatrix}$$

$O(nT^2)$ の計算負荷が，$O(n(T/2)^2)$ に削減できる

実験結果

Table: 基底計算の解像度と分割数ごとの実行時間 (秒)

解像度	分割なし	分割数 2	分割数 4	分割数 10
64^3	23.43	14.56	8.98	3.78
128^3	190.54	118.49	69.62	33.43

Table: 行列の射影の解像度と分割数ごとの実行時間 (秒)

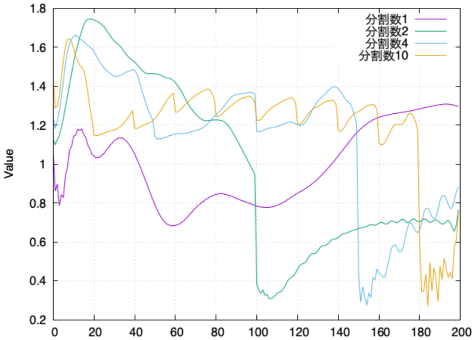
解像度	分割なし	分割数 2	分割数 4	分割数 10
64^3	3.51	2.82	0.85	0.67
128^3	5.21	3.06	3.05	3.98

実験結果

Table: 基底の累積寄与率の最小値と最大値

解像度	分割なし	分割数 2	分割数 4	分割数 10
64^3	0.969886	0.972114	0.979512	0.992557
		0.985917	0.990411	0.996893
128^3	0.940446	0.954735	0.971454	0.988967
		0.96489	0.976306	0.996132

平均誤差 $L_2 = \frac{||\boldsymbol{u} - \widetilde{\boldsymbol{u}}||_2}{||\boldsymbol{u}||_2}$



実験結果

解像度 128^3，200 フレーム

[1] A. J. Chorin. Numerical solution of the navier-stokes equations. *Mathematics of Computation*, 22(104):745–762, 1968.

[2] A. Treuille, A. Lewis, and Z. Popović. Model reduction for real-time fluids. *ACM Transactions on Graphics*, 25(3):826–834, 2006.

[3] J. Stam. Stable fluids. In *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, pages 121–128, 1999.

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

[5] T. Kim and J. Delaney. Subspace fluid re-simulation. *ACM Transactions on Graphics*, 32(4):62:1–62:9, 2013.