

# 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

情報工学専攻 須之内 俊樹  
Toshiki SUNOUCHI

### 概要

本研究では部分空間法の前処理に対して、計算時間を高速化する手法を提案する。部分空間法を適用することで、シミュレーションは大幅に高速化することができるが、前処理に膨大な計算時間がかかることが課題である。

そこで本研究は、前処理をシミュレーション時間に対して分割し、前処理全体の計算時間を削減する手法を提案する。計算機実験の結果、前処理のボトルネックとなっている基底計算のための特異値分解の高速化を確認し、前処理の高速化に有効であると示すことができた。

**キーワード:** 流体シミュレーション, 部分空間法, スナップショット固有直交分解, 立体求積法。

### 1 はじめに

流体シミュレーションは、工学およびコンピュータグラフィックスの分野において広く利用されている。コンピュータグラフィックスの分野では、現実的な水や煙のアニメーションを生成し、映像作品やゲームの演出に活用されている。コンピュータグラフィックスにおいては、物理的に厳密なシミュレーションよりも、視覚的に自然な動きを高速に計算することが求められる。

近年、より高解像度の流体シミュレーションが求められる中で、計算負荷の増大が課題となっている。そのため、高速化手法の研究が活発に進められており、特に大規模並列計算技術の適用が重要視されている。多くの高速化手法が GPU 上での実装を前提として検討されている。

計算負荷の削減手法の一つとして、前処理によりシミュレーション計算に使用する行列の次元を削減する部分空間法が存在する。部分空間法を適用することで、シミュレーション計算そのものは高速化されるが、前処理に要する計算負荷やメモリ使用量が大きくなるという課題がある。本研究では、部分空間法の計算負荷を削減することにより、前処理に要する時間の短縮を図ることを目的とする。

### 2 関連研究

#### 2.1 煙の流体シミュレーション

下記の式 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。式 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、第四項を外力項と呼ぶ。コンピュータグラフィックスにおける気体の外力には重力のほか、温度差による対流を発生させる力や、気体の渦運動を生成する力などが含まれ、これらを適用することで、気体の自然な挙動を再現することが可能となる。式 2 は流体の非圧縮性条件であり、煙のシミュレーションにおいて流速が著しく大きい場合を除き、利用することが可能である。

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

$$\nabla \cdot \mathbf{u}(t) = 0 \quad (2)$$

コンピュータグラフィックスにおいては、ナビエ・ストークス方程式の数値解法の中でも、全ての項を個別に分割して解く手法としてフラクショナルステップ法 [1] が一般的に用いられている。フラクショナルステップ法は、中間子  $\mathbf{u}_0$  から  $\mathbf{u}_3$  を用いて、ナビエ・ストークス方程式を以下のように分割する。

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

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

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

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

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

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

フラクショナルステップ法の分割において、

流速に従って煙の密度を計算し、ボリュームレンダリングによって可視化することで、煙のアニメーションを作成することができる。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して、描画対象の内側のデータも出力画像に反映させる。

## 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。部分空間上におけるシミュレーションでは、元のシミュレーションよりも  $\Delta t$  を小さい値に設定することで、シミュレーションの時間変化をより詳細に追跡することが可能である。

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  $\tilde{\mathbf{x}}$  に変換する、 $n \times r$  直交行列  $\mathbf{A}$  を考える。直交行列の性質から、以下の 2 式が成り立つ。

$$\tilde{\mathbf{x}} = \mathbf{A}^T \mathbf{x} \quad \mathbf{x} = \mathbf{A} \tilde{\mathbf{x}}$$

また、 $\tilde{\mathbf{x}}' = \mathbf{A}^T \mathbf{x}'$  を満たす  $n$  次元ベクトル  $\mathbf{x}'$ 、 $r$  次元ベクトル  $\tilde{\mathbf{x}}'$  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。

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

以上の 4 式を用いて、行列  $\mathbf{F}$  と  $\tilde{\mathbf{F}}$  の間の以下の関係式が得られる。

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

## 3 スナップショット分割による高速化

### 3.1 フラクショナルステップ法への部分空間法の適用

$$\begin{aligned} \mathbf{U}_2^T \mathbf{u}_2 &= (\mathbf{U}_2^T \mathbf{V} \mathbf{U}_1) \mathbf{U}_1^T \mathbf{u}_1 & \tilde{\mathbf{u}}_2 &= \tilde{\mathbf{V}} \tilde{\mathbf{u}}_1 \\ \mathbf{P}^T \mathbf{b} &= (\mathbf{P}^T \mathbf{W} \mathbf{U}_2) \mathbf{U}_2^T \mathbf{u}_2 & \tilde{\mathbf{b}} &= \tilde{\mathbf{W}} \tilde{\mathbf{u}}_2 \\ \mathbf{P}^T \mathbf{p} &= (\mathbf{P}^T \mathbf{A}^{-1} \mathbf{P}) \mathbf{P}^T \mathbf{b} & \tilde{\mathbf{p}} &= \tilde{\mathbf{A}} \tilde{\mathbf{b}} \\ \mathbf{U}_3^T \mathbf{u}_3 &= \mathbf{U}_2^T \mathbf{u}_2 - (\mathbf{U}_3^T \mathbf{Y} \mathbf{P}) \mathbf{P}^T \mathbf{p} & \tilde{\mathbf{u}}_3 &= \tilde{\mathbf{u}}_2 - \tilde{\mathbf{Y}} \tilde{\mathbf{p}} \end{aligned}$$

### 3.2 部分空間法の高速化手法の提案

部分空間法の前処理において、基底の計算と行列の射影の計算量はスナップショットの 2 乗に比例し、前処理の大多数の時間を占めている。本節では部分空間法の前処理の高速化を達成するため、スナップショットを  $d$  個に分割する手法を提案する。

部分空間法の基底の計算に用いるスナップショットを  $d$  分割することによって、基底 1 つの計算負荷  $O(nm^2)$  が  $\frac{1}{d^2}$  に削減される。アルゴリズム全体では  $d$  回計算す

るため、計算負荷は  $\frac{1}{d}$  になることが期待できる。行列の射影計算も分割した基底ごとに行う。空間計算量のボトルネックは分割前のスナップショット行列であり、分割前と変わらない。しかし、特異値分解する行列のサイズが  $\frac{1}{d}$  になることで、GPU を用いた大規模高速計算と併用する場合、GPU メモリの削減が期待できる。空間計算量を GPU が一度に扱うことができる値まで削減することで、並列計算による高速化が期待できる一方で、特異値分解の回数が  $d$  倍になることで、CPU-GPU 間のデータ通信の時間が増加することに注意が必要である。

## 4 計算機実験

本研究では、c++ と線形代数ライブラリ Eigen を用いて、文献 [2] の煙の流体シミュレーションを行うソフトウェアを実装した。立方体形状のシミュレーション空間の底面中心から、煙が立ち上る様子をシミュレーションし、そのデータを元に部分空間法による次元削減を行った。また、OpenGL を用いてスライスペースのボリュームレンダリングを行い、実験結果を可視化するソフトウェアを実装した。これによって得られたシミュレーション結果と、提案手法の計算負荷と削減後のデータの精度について調査を行った。計算負荷の評価は、部分空間法の基底を計算する前処理と行列の射影について実行時間を計測した。データの精度は非線形項の部分空間への射影手法 [4] を参考に、L2 誤差を用いて評価を行った。ここで L2 誤差は次元削減前後のベクトル  $\mathbf{u}$  と  $\tilde{\mathbf{u}}$  について、以下のように表される。

$$L_2 = \frac{\|\mathbf{u} - \tilde{\mathbf{u}}\|_2}{\|\mathbf{u}\|_2}$$

表 1 基底計算における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 <sup>3</sup>	23.428	14.564	8.981	3.780
128 <sup>3</sup>	190.541	118.489	69.615	33.427

表 2 行列の射影における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 <sup>3</sup>	3.514	2.823	0.846	0.666
128 <sup>3</sup>	5.206	3.061	3.050	3.975

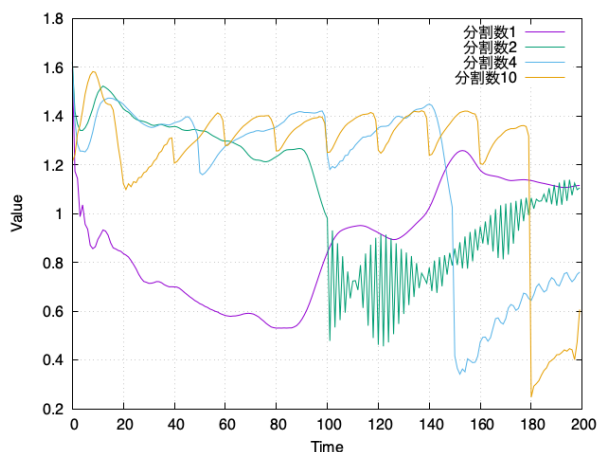


図 1 解像度  $64^3$  における  $L_2$  誤差

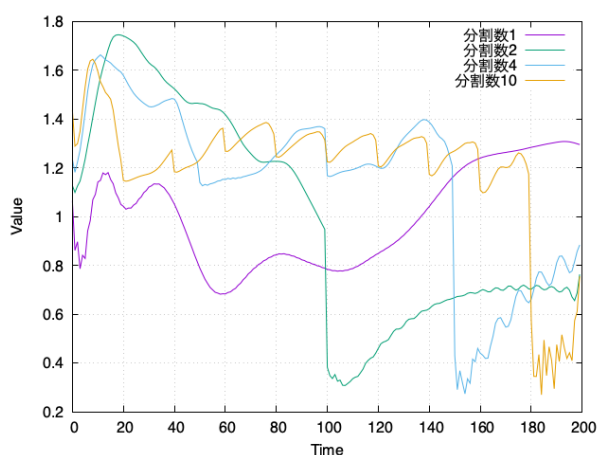


図 2 解像度  $128^3$  における  $L_2$  誤差

- [3] J. Stam. Stable Fluids. In *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, pages 121–128, 1999.
- [4] A. Treuille, A. Lewis, and Z. Popovic. Model Reduction for Real-time Fluids. *ACM Transactions on Graphics*, 25(3) : 826–834, 2006.
- [5] T. Kim, J. Delaney. Subspace Fluid Re-Simulation. *ACM Transactions on Graphics*, 32(4): 62 : 1–62 : 9, 2013.

## 5 考察

## 6 終わりに

流体シミュレーションにおける部分空間法の前処理にかかる計算時間に対し、スナップショットを分割することによって高速化する手法を提案した。特に基底計算については分割数に応じた高速化を達成し、前処理全体の高速化を達成した。スナップショット分割によるシミュレーションの結果に関して、 $L_2$  誤差では差が認められたものの、見た目の上では問題ない範囲であった。

## 参考文献

- [1] A. J. Chorin. Numerical Solution of the Navier-Stokes Equations. *Mathematics of Computation*, 22(104): 745–762, 1968.
- [2] R. Fedkiew, J. Stam, H. Jensen. Visual simulation of smoke. In *Proceedings of SIGGRAPH 01*, 15–22, 2001.