

# 放送大学大学院 中間報告

情報学プログラム 佐野宏行

平成 27 年 12 月 21 日

## 1 はじめに

本書では 2015 年 10 月時点の研究活動の中間報告を行う。作成した研究計画と最も重要な関連研究である Visual Simulation of Smoke[3] の調査内容、湯気のシミュレーションのモデルについて記載する。

## 2 研究計画

### 2.1 背景・目的

近年、3D プリンタ、プロジェクションマッピングやバーチャルリアリティ技術の普及より、コンピュータグラフィックス (CG) は重要な技術になっている。これらの 3D 技術が身近になることで、現実世界に近いリアルな画像やアニメーションの作成技術もさらに重要になると予想される。CG の研究においては水の表現のような流体シミュレーションに関する研究が盛んに行われているが水の特徴である三態変化を扱う研究はほとんど行われていない。特に水が液体から気体へ相転移する際に発生する、湯気の表現手法は確立していない。湯気の表現が CG により可能になることで、従来の CG では作成できない湯気が発生する温泉、暖かい料理といったシーンを表現可能となり、より CG の表現能力を上げることができると考えられる。これより本研究では CG によるフォトリアルな湯気のシミュレーションを行うことを目的とする。

### 2.2 先行研究

湯気の発生に関する表現に特化した CG による流体シミュレーションの研究は行われていない。湯気に近い表現としては煙 [3]、雲 [1][13][12]、火山噴煙 [14][15]、水煙 [17] のシミュレーションの研究がある。水の蒸発、凝固、凝結といった性質を包括的に扱う研究 [23] はあるが湯気に関しては言及されない。[7] では湯気 (Steam) の表現の言及はあるが水蒸気量を考慮した表現は行われていない。流体力学では複数の相が混じり合う物質は混相流 (multi phase flow)、湯気のように気体と液体が混合する物体の動きは気液二相流として研究が行われているが CG の作成を目的とはしていない。

### 2.3 関連研究

実用的な CG による流体シミュレーションは Stam による StableFluid 法 [21] により行われた。これは格子法による差分化を行い、移流項をバックトレースすることにより解くセミラグランジュ法を取り入れた手法で大きな時間幅でも安定的である。この手法は煙 [3]、水 [5]、火 [16]、爆発 [4] のような CG による流体シミュレーションの基礎となった。

StableFluid 法は一次の精度のため数値拡散が強く現れる。Kim ら [10] は BFECC 法により高精度な移流を提案した。BFECC 法はバクトレース時にフォワードトレースにより求めた誤差を修正することで 高精度に移流を行う方法である。Selle ら [20] は MacCormack 法により BFECC 法のコストを減らした。Zhu と Bridson[22] は格子法とパーティクル法の両方を用いた FLIP 法によりさらに数値拡散を減らした。

## 2.4 湯気の発生原理と特徴

湯気は水蒸気が温度の低い場所で冷えて凝結し、水滴となって白くなったものである。湯気は水蒸気が外気と触れた際に温度が低下し、水蒸気量が飽和水蒸気量を上回った場合、大気中の細かい塵を核として凝結することで発生する。この際、発生する湯気の粒子は光の波長と同程度、もしくは少し大きめの大きさとなり、どの光の波長も均等に散乱されるミー散乱という現象が起こり白く見える。発生した湯気は空気よりも軽いことにより浮力が発生して上昇し、やがて空気中に蒸発して消滅する。

湯気の特徴を湯気に似た動きを持つ煙との対比により説明する。まず湯気は水蒸気が液体に変化してから蒸発するため、可視化できる時間が一時的で温度や湿度による条件の違いによって発生の仕方が異なる。また映像として利用されるシーンを想定すると飲み物や風呂など湯気の発生元としては液体のように流動性を持ったもので、湯気が発生元の物体の表面温度も一様ではないことから、物体から不規則に湯気が発生する。細かい点では湯気の粒子の大きさは  $10\sim 100\mu\text{m}$  と言われており条件によっては人間の肉眼でも粒子を確認することが可能である。本研究ではこれらの湯気の特徴を元に湯気のモデル化を行う。

## 2.5 湯気のモデル化とシミュレーション

CG による湯気のシミュレーションは既存の CG の表現方法を組み合わせる方針とする。具体的には流れのモデリングは煙をベースとし、湯気の発生と消滅のモデリングは雲の表現をベースとする。

流れのモデリングは [21][3] の手法を用いる。この手法では流体の支配方程式としてナビエーストークス方程式、連続の式を用いる。また流体シミュレーションの手法としては空間をグリッドで分割するデータ構造を用いる格子法を用いるが、この場合、数値拡散により小さな渦が消えてしまうため渦を追加する渦度強制法 (vorticity confinement) という手法を用いる。

湯気の発生と消滅のモデリングは [12] の手法を参考とする。これは大気流体をモデルとして雲の生成モデルを偏微分方程式により定義している。湯気の生成は雲とはスケールが異なるが基本的なメカニズムは変わらないため、雲と同様のモデルを採用する。ただし環境温度が空間の高さにより低下するなど大気特有の変数についてはモデルから削除する。湯気が発生元のシミュレーションは、[13] の手法を参考として湯気発生元の表面からランダム性を持つ濃度の水蒸気を追加する。この際、CG においてランダムで自然な地形の生成等に利用するノイズであるパーリンノイズを利用する。これにより水などの液体から湯気発生する場合の不規則な湯気の表現、料理のように温度分布が一様でない物体からの湯気の表現が可能になると考えられる。

## 2.6 評価方法

本研究は過去の研究事例が無い状況であるため過去事例との比較が困難となっている。CG は映画などの映像に利用されることを目的としているため、見た目として正しく表現されていることが求められ、必ずしも物理的に正しいことは求められない。そのため CG において新規の表現手法を提案する場合の評価として定性的なものとなり定量的な測定は困難である。

実際の研究事例から新規の表現手法にどのような評価手法が取られているかを確認した。CGにおいて初めて水 [6] や雲 [8] がシミュレートされた例では出力された絵とその説明のみとなっている。火山噴煙 [14] やプロミネンス [9] が初めてシミュレートされた場合は実際の写真と出力された絵を並べるにより比較を行っている。

上記より評価の手法としては定量的な評価ではなく、出力された絵の説明を行い、可能であれば実際の写真との比較により行うこととする。

## 2.7 発展的研究

計画に記載した湯気のモデル化手法は 10 年以上前の古典的な手法を組み合わせ、改変したものである。現在では湯気に関連のある新たな手法があり、これらを利用することでよりリアリティのある表現が可能となることが考えられる。湯気の発生源のシミュレーションと煙の乱流のシミュレーションで用いる渦度強制法 (Vorticity Confinement) にウェーブレットノイズを用いた手法 [11] を利用することでリアルな湯気の発生と高解像度な乱流を生成することができる。また湯気の発生と消滅のモデルは雲のモデルの利用を予定しており、スケールが異なるため十分な結果を得られない可能性がある。そのため流体力学の混相流のシミュレーションの分野で用いられる手法を取り入れることも考えられる。例えば、[14] では火山噴煙のシミュレーションに 2 流体モデルを利用している。水滴の表現としては水煙の表現で用いられる手法 [17] の利用が検討できる。また湯気の発生箇所としては水分が含まれる液体状の流体であるシーンが多い。液体状の流体から発生する湯気をシミュレートすることにより、さらに表現力を高められると考えられる。湯気の発生元のシミュレーションはパーリンノイズを利用することを想定しているが、柔軟に流体を表現することができる FLIP 法 [22] を用いることも検討したい。

湯気の表現としては現実を再現した場合に必ずしも効果的な湯気の表現とならない。例えば、CG を利用せずに湯気を含む写真の撮影する際、ドライアイス等で外気の温度を下げることで効果的な湯気を生成する技法がある。このように効果的な湯気を生成するためにユーザが自由に湯気をデザインできるモデリングツールの開発が応用例として考えられる。また、GPU を利用した高速化・リアルタイム化、バーチャルリアリティ技術への適用、湯気だけでなく物体へのしみ込みや凝結といった水のさらなる表現力の向上に取り組むことも考えられる。

## 2.8 スケジュール

現状は流体力学の深い理解、流体シミュレーションの実装の経験はない。そのため既存のソフトウェア、ソースの利用は行わず、スクラッチからの流体シミュレーションからの実装を行う方針とする。湯気の流れは煙の表現をベースに検討しているため Visual Simulation of Smoke[1] の実装を行う (8 月)。この際、データ構造はやや異なるが Stable Fluids[14] によるシンプルなソース、研究者により公開される流体シミュレーションのソースを参考とする。その後、湯気の動きの物理モデルを雲のモデルをベースとして作成し (9 月)、作成したモデルをベースに実装を行う (10 月)。さらに湯気の発生源のシミュレーションを行い (11 月)、その後パラメータの見直し等の改良を行う (12 月)。利用する言語、ツール、フレームワークとしては物理シミュレーションの計算部分は C/C++、レンダリング部分は外部のレンダリングソフトウェア (POV-Ray, PBRT 等) を用いる。また簡易的なモデルの表示確認には OpenGL を用いることとする。また関連研究とともに公開されるソースサンプルは必要に応じて活用を検討する。上記作業を修士課程中に行うこととする。括弧内のスケジュールはベストケースのスケジュールとして記載したものであり、進捗により随時スケジュールは見直す。もし括弧内のスケジュール通りに完了した場合は、以下に記載したさらに発展的な研究に取り組むこととする。

### 3 Visual Simulation of Smoke 解説

Visual Simulation of Smoke[3] を解説する。物理シミュレーションの実装に必要なアルゴリズムを中心に説明する。

#### 3.1 概要

煙のようなガスをシミュレートするためのモデルを提案する。煙の速度を非圧縮性のオイラーの運動方程式でモデル化する。非圧縮性のオイラーの運動方程式はセミラグランジュ法と圧力のポアソン方程式を用いて解く。数値拡散を減らす渦度強制 (Vorticity Confinement) の手法を提供する。このモデルは安定的、高速かつ数値拡散を起こすことがない。

#### 3.2 方程式

気体のモデルとして非粘性、非圧縮、粘度が一定の気体を仮定する。粘性、圧縮性は気体のシミュレーションでは無視できる。煙の速度  $v = (u, v, w)$  は非圧縮のオイラーの運動方程式によって与えられる。

$$\nabla \cdot v = 0 \quad (1)$$

$$\frac{\partial v}{\partial t} = -(v \cdot \nabla)v - \nabla p + f \quad (2)$$

式 (12) は質量保存、式 (13) は運動量保存を表す。 $p$  は気体の圧力、 $f$  は外力を表す。式 (13) の右辺は左から移流項、圧縮項、外力項と呼ぶ。

これらの式を 2 段階で解く。はじめに式 (13) の左辺で時間刻み幅  $\Delta t$  を利用し、右辺の圧縮項をなくした式 (3) を解いて中間速度場  $u^*$  を求める。

$$\frac{v^* - v}{\Delta t} = -(v \cdot \nabla)v + f \quad (3)$$

次に中間速度場  $u^*$  をプロジェクション法によって非圧縮性を満たすようにする。これはノイマン境界条件にしたがった以下のポアソン方程式から圧力を計算することに等しい。

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot v^* \quad (4)$$

式 (4) の導出をする。まず最終的な速度場を  $v$  とした場合に  $\frac{v - v^*}{\Delta t} = -\nabla p$  となる。これに両辺の発散をとる ( $\nabla$  を掛ける) と  $\frac{\nabla \cdot v - \nabla \cdot v^*}{\Delta t} = -\nabla^2 p$  となる。式 (12)  $\nabla \cdot v = 0$  から  $-\frac{\nabla \cdot v^*}{\Delta t} = -\nabla^2 p$  となりこれを式変形することで式 (4) が導出できる。

中間速度場から圧力の勾配を引くと非圧縮性を満たす速度となる。

$$v = v^* - \Delta t \nabla p \quad (5)$$

式 (5) は  $\frac{v - v^*}{\Delta t} = -\nabla p$  の両辺に  $\Delta t$  を掛けて式変形することで導出できる。

気温  $T$  と煙の密度  $\rho$  は煙の速度に沿って運ばれることを仮定する。

$$\frac{\partial T}{\partial t} = -(v \cdot \nabla)T \quad (6)$$

$$\frac{\partial \rho}{\partial t} = -(v \cdot \nabla)\rho \quad (7)$$

濃い煙は重力によって下に落ち、熱い気体は浮力によって上に上がる。

$$f_{buoy} = -\alpha \rho z + \beta(T - T_{amb})z \quad (8)$$

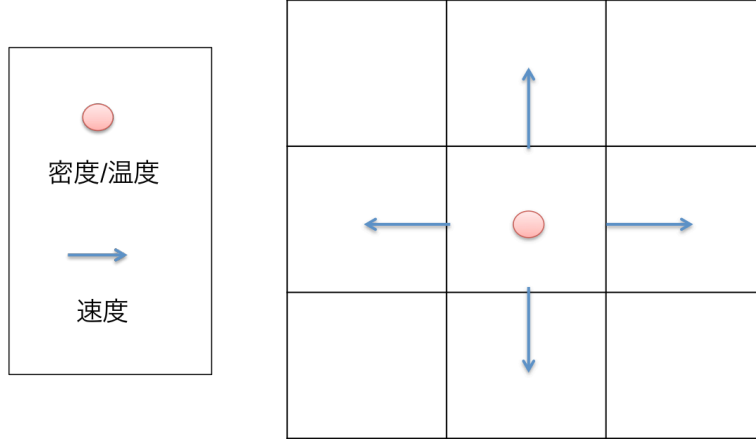


図 1: 空間の差分化

$z = (0, 0, 1)$  は上の垂直方向のベクトル、 $T_{amb}$  は大気環境温度、 $\alpha$  と  $\beta$  は正の定数となる。

数値拡散によって細かい渦の挙動が失われる。渦度強制により検出した渦の情報を元に速度場に速度を追加することで渦の消失を防ぐ。最初に渦  $\omega$  を取得する。

$$\omega = \nabla \times v \quad (9)$$

低い渦濃度から高い渦濃度へのベクトルを正規化する。

$$N = \frac{\eta}{|\eta|} (\eta = \nabla|\omega|) \quad (10)$$

以下より追加する外力を求める。

$$f_{conf} = \epsilon h(N \times \omega) \quad (11)$$

$\epsilon > 0$  は追加する渦のスケール、 $h$  は格子幅を表す。

### 3.3 実装

空間の差分化は同一の大きさの格子に分割する格子法 (図 1) を用いる。格子の中心には煙の密度、温度、外力が定義されるのに対して、格子の面には速度が定義される。この格子スタガード格子と呼ぶ。スタガード格子は通常の格子と比べて人工的な散逸を減らす。

処理全体の流れ (図 2) は流体の速度の計算を行い、計算した速度から煙の密度の計算を行うことを繰り返す。流体の速度はオイラーの運動方程式 (13) を外力項、移流項、圧縮項の計算結果を足すことにより計算する。

すべての物理量を格納する格子を 2 つ用意する。時間刻み幅  $\Delta t$  により修正された格子を、もう一つの格子に更新する。

はじめに外力から速度を更新する。外力にはユーザからの与えられる力、式 (14) に定義される浮力、式 (11) に定義される渦度強制による力がある。

次に式 (3) の移流項をセミラグランジュ法によって解く。セミラグランジュ法とは速度場をバックトレースすることで求める方法である。新しい速度はバックトレースした点にある速度場から補完する。バックトレースした点が格子の外の場合がある。この場合、単純にバックトレースする経路を境界面で切り取る。

補間にエルミート補間を利用した場合、オーバーシュートが起こる。提案するキュービク補間は単調でオーバーシュートしない。

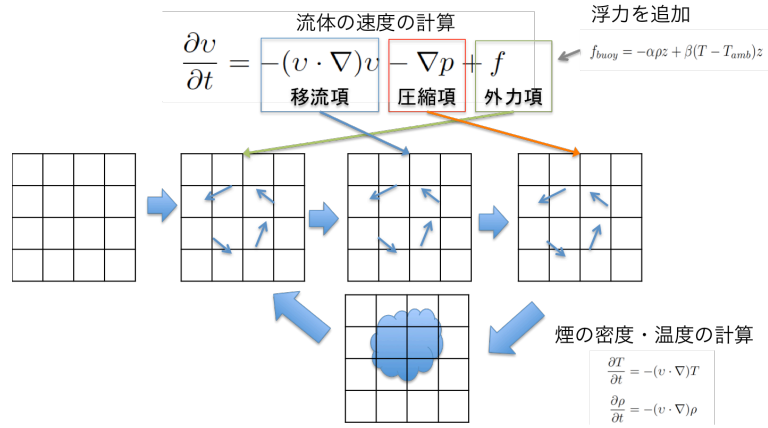


図 2: 処理全体の流れ

最後に速度場を質量保存則に従うようにする。圧力を求めるためポアソン方程式 (4) を解く。この方程式の差分化の結果は疎な連立一次方程式になる。法線方向の圧力勾配が 0 のノイマン境界条件を使う。連立方程式は共役勾配法 (CG 法) が実装が容易で収束性質が良い。収束を良くするため不完全コレスキー分解を用いる。これは標準的な方法である。圧力を求めた後、速度から圧力の勾配を引く。

速度が更新されたら、セミラグランジュ法を再び用いて気温と煙の密度を移流する。

## 4 湯気のシミュレーション

湯気のシミュレーションモデルの気体のモデルは [3] の煙のシミュレーションの手法に基づき非粘性、非圧縮の気体を仮定する。相転移を考慮した湯気のモデルは [12][2] で提案される雲のシミュレーションのモデルから乾燥断熱減率を無視したものとする。乾燥断熱は空気塊が上昇することで周囲の気圧が低下して膨張することにより温度が下がる現象であるが、温度変化の割合は 100m の上昇に対して 1℃であるため、湯気の表現においては無視できると仮定する。湯気の表現においては熱源からのゆるやかな温度変化、水蒸気量の変化が必要のため温度と水蒸気の密度には拡散項を追加した。

湯気の数値  $v = (u, v, w)$  は非粘性、非圧縮のオイラーの運動方程式 (12,13) によって与えられる。シミュレーション空間は  $N_x \times N_y \times N_z$  の格子に分割し各格子点に水蒸気密度  $q_v$ 、湯気の密度  $q_c$ 、温度  $T$  を割り付ける。

$$\nabla \cdot v = 0 \quad (12)$$

$$\frac{\partial v}{\partial t} = -(v \cdot \nabla)v - \nabla p + B + f \quad (13)$$

$B$  は浮力、 $f$  は風などによる外力を表す。浮力  $B$  は式 (14) で定義する。

$$B = k_b \frac{T - T_{amb}}{T_{amb}} z - g q_c z \quad (14)$$

$k_b$  は浮力の係数、 $g$  は重力の係数、 $T_{amb}$  は環境温度、 $q_c$  は湯気の濃度、 $z$  は上方向のベクトルである。

湯気の密度  $q_c$  と水蒸気  $q_v$  の密度は次式で定義する。

$$\frac{\partial q_c}{\partial t} = -(v \cdot \nabla)q_c + C_c \quad (15)$$

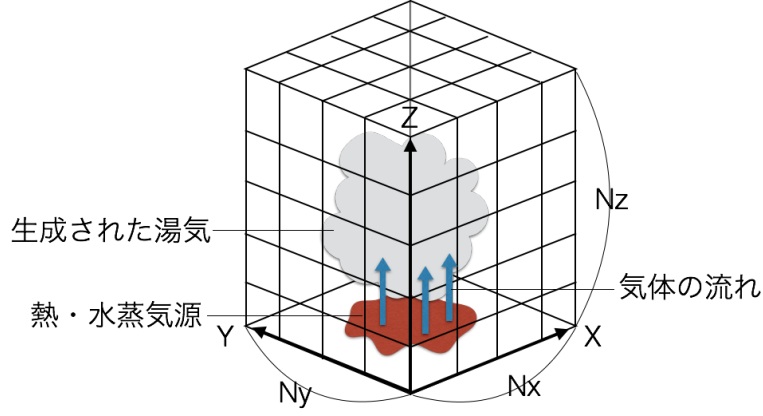


図 3: 湯気のシミュレーション空間

$$\frac{\partial q_v}{\partial t} = D \nabla^2 q_v - (v \cdot \nabla) q_v - C_c + S_v \quad (16)$$

$C_c$  は相転移によって発生する湯気の量、 $S_v$  は水蒸気源から水蒸気の供給量、 $\alpha$  は相転移率、 $D$  は水蒸気の分子拡散係数である。

$$C_c = \alpha(q_v - q_s) \quad (17)$$

$q_s$  は飽和水蒸気密度を表し、次式で与えられる。

$$q_s = \min \left( A \exp \left( \frac{-B}{T+C} \right), q_v + q_c \right) \quad (18)$$

ここで  $A, B, C$  は飽和水蒸気密度を決定するためのパラメータである。温度  $T$  は以下のように表される。

$$\frac{\partial T}{\partial t} = a \nabla^2 T - (v \cdot \nabla) T + Q C_c + S_T \quad (19)$$

ここで  $a$  は熱拡散率、 $Q$  は潜熱係数を表す。右辺第一項は熱拡散、第二項は熱対流、第三項は相転移による潜熱、第四項は外部の熱源からの熱量を表す。水蒸気源と熱源は空間の底面から発生する。底面上の水蒸気と熱の発生量の分布はパーリンノイズ [18][19] を用いる。

## 5 乱流モデル

(TBD) Wavelet Turbulence などのことを書く。

## 参考文献

- [1] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, and Tomoyuki Nishita. A Simple , Efficient Method for Realistic Animation of Clouds. *SIGGRAPH*, 2000.
- [2] Yoshinori Dobashi, Katsutoshi Kusumoto, Tomoyuki Nishita, and Tsuyoshi Yamamoto. Feed-back control of cumuliform cloud formation based on computational fluid dynamics. *ACM Transactions on Graphics*, Vol. 27, No. 3, p. 1, 2008.
- [3] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pp. 15–22. ACM, 2001.

- [4] Bryan E. Feldman, James F. O'Brien, and Okan Arikan. Animating suspended particle explosions. *ACM Transactions on Graphics*, Vol. 22, No. 3, p. 708, 2003.
- [5] Nick Foster and Ronald Fedkiw. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 23–30. ACM, 2001.
- [6] Nick Foster and Dimitri Metaxas. Realistic Animation of Liquids. *Graphical Models and Image Processing*, Vol. 58, No. 5, pp. 471–483, 1996.
- [7] Nick Foster and Dimitris Metaxas. Modeling the Motion of a Hot, Turbulent Gas. *SIGGRAPH*, 1997.
- [8] Geoffrey Y. Gardner. Visual simulation of clouds. *ACM SIGGRAPH Computer Graphics*, Vol. 19, No. 3, pp. 297–304, 1985.
- [9] Tomokazu Ishikawa and Yonghao Y U E Member. Visual Simulation of Solar Photosphere Based on Magnetohydrodynamics and Quantum Theory.
- [10] ByungMoon Kim, Yingjie Liu, Ignacio Llamas, and Jarek Rossignac. FlowFixer: Using BFEC for fluid simulation. *Natural Phenomena*, pp. 51–56, 2005.
- [11] Theodore Kim, Nils Thürey, Doug James, and Markus Gross. Wavelet turbulence for fluid simulation. *ACM Transactions on Graphics*, Vol. 27, No. 3, p. 1, 2008.
- [12] R Miyazaki, Y Dobashi, and T Nishita. Simulation of Cumuliform Clouds Based on Computational Fluid Dynamics. 2002.
- [13] R. Miyazaki, S. Yoshida, Y. Dobashi, and T. Nishita. A method for modeling clouds based on atmospheric fluid dynamics. *Proceedings Ninth Pacific Conference on Computer Graphics and Applications. Pacific Graphics 2001*, 2001.
- [14] R. Mizuno, Y. Dobashi, Bing-Yu Chen Bing-Yu Chen, and T. Nishita. Physics motivated modeling of volcanic clouds as a two fluids model. *11th Pacific Conference on Computer Graphics and Applications, 2003. Proceedings.*, No. August, pp. 2–6, 2003.
- [15] Ryoichi Mizuno, Yoshinori Dobashi, and Tomoyuki Nishita. Modeling of volcanic clouds using CML. *Journal of Information Science and Engineering*, Vol. 20, No. 2, pp. 219–232, 2004.
- [16] Duc Quang Nguyen, Ronald Fedkiw, and Henrik Wann Jensen. Physically based modeling and animation of fire. *ACM Transactions on Graphics*, Vol. 21, No. 3, 2002.
- [17] MB Nielsen and Ole Ø sterby. A Two-Continua Approach to Eulerian Simulation of Water Spray. *ACM Transactions on Graphics*, Vol. 32, No. 4, p. 1, 2013.
- [18] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, Vol. 19, No. 3, pp. 287–296, July 1985.
- [19] Ken Perlin. Improving noise. *ACM Transactions on Graphics*, Vol. 21, No. 3, pp. 2–3, 2002.
- [20] Andrew Selle, Ronald Fedkiw, Byungmoon Kim, Yingjie Liu, and Jarek Rossignac. An unconditionally stable MacCormack method. *Journal of Scientific Computing*, Vol. 35, No. 2-3, pp. 350–371, 2008.



- [21] Jos Stam. Stable Fluids. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 121–128, 1999.
- [22] Yongning Zhu and Robert Bridson. Animating sand as a fluid. *ACM Transactions on Graphics*, Vol. 24, No. 3, p. 965, 2005.
- [23] 藤澤誠. コンピュータグラフィックスにおける物質の三態シミュレーションに関する研究, 2008.