

CG による湯気のシミュレーションと可視化

放送大学大学院 情報学プログラム M1

佐野 宏行

平成 28 年 2 月 16 日

1 はじめに

本書では 2016 年 2 月時点の研究活動の中間報告を行う。

1.1 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。このため CG の分野において湯気の表現はクリエイターによる熟練した技術と経験が必要であった。

本研究では CG による湯気のシミュレーションと可視化を行うことを目的とする。具体的には温度、水蒸気量といった湯気の状態に影響するパラメータを変化させることにより容易に CG による湯気の表現を可能とする。これにより従来の CG では作成できない温度、水蒸気量を考慮した温泉、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。

1.2 関連研究

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 [2][14][13]、火山噴煙 [15][16]、水煙 [18]、水滴と水泡 [12] 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 [22] はあるが湯気に関しては考慮されない。[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。

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 [20] により行われた。これは格子法による差分法を行い、移流項をバ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 [17]、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 BFEC 法により高精度な移流を提案した。BFEC 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 [19] は MacCormack 法により BFEC 法のコストを減らした。Zhu と Bridson[21] 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。

湯気の表現に必要な水の液体と気体間の相転移、温度の熱拡散、蒸気分子の拡散を CG に適用した研究がある。水の液体と気体間の相転移のモデルは宮崎ら [14][13] により飽和水蒸気量を利用するモデルが提案された。温度の熱拡散のモデルは Foster ら [7]、宮崎ら [13]、蒸気分子の拡散のモデルは宮崎ら [13] により CG に適用された。

1.3 提案手法の概要

本論文では湯気のシミュレーションと可視化を実際の湯気の発生原理を元にしたモデルと手法を提案する。気体のモデルは Fedkiw ら [3] の手法に基づき非粘性、非圧縮の流体を仮定する。水の液体と気体間の相転移のモデルは宮崎ら [14][13] による飽和水蒸気量を利用するモデルとする。対流に必要な浮力はブシネスク近似による誤差を減らすため周囲の温度との差により計算を行う手法を提案する。熱、水蒸気移動は流体の速度による移動に加えて、熱拡散、分子拡散を考慮したモデルを提案する。

2 湯気のシミュレーション

2.1 湯気の発生原理

湯気の発生と消滅のプロセスの詳細を記載する。ここでは常温の空気中に存在する液体の水の液面上から湯気が発生している状態を対象とする。この時、水温は空気の温度よりも高いが沸騰はしていない状態とする。

まず水が液面上から液体が気体へ蒸発し、蒸発した水蒸気は分子拡散により周囲へ拡散する。同時に水の温度は空気よりも高いため温度勾配により熱拡散が起こり熱が水から空気へ移動する。空気の温度が上昇すると空気の密度が周囲の空気より低くなることにより浮力が発生する。浮力により空気へ上方向の力が働き、対流が発生する。この対流により空気に含まれる水蒸気、熱が移動する。空気中に含まれる水蒸気量が飽和水蒸気量を超えた場合に凝結することで水滴が発生する。凝結は大気中の細かい塵を核として行われる。この水滴の粒子に対して光が当たった際に散乱が起こり白く見えることで湯気として認識される。凝結により発生する水滴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ミー散乱という現象が発生する。ミー散乱は粒子のサイズが大きくなるにつれて前方の指向性が高くなり側方、後方への散乱は弱くなる。空気中に含まれる水蒸気量が飽和水蒸気量を下回った場合、水滴が空気中に蒸発し湯気が消滅する。

2.2 シミュレーションモデル

湯気速度 $v = (u, v, w)$ は非粘性、非圧縮のオイラーの運動方程式 (10,11) によって与えられる。

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

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

B は浮力、 f は風などによる外力を表す。浮力 B は式 (17) で定義する。

$$B = (k_b \Delta T - g q_c) z \quad (3)$$

k_b は浮力の係数、 g は重力の係数、 q_c は湯気の濃度、 z は上方向のベクトルである。湯気が発生する底面周辺のように温度変化が大きい場合、近似による誤差が大きくなる。このため周辺の温度の差分から上向きの浮力の計算を行う。

湯気の密度 q_c と水蒸気 q_v の密度は次式で定義する。

$$\frac{\partial q_c}{\partial t} = -(v \cdot \nabla)q_c + C_c \quad (4)$$

$$\frac{\partial q_v}{\partial t} = D_v \nabla^2 q_v - (v \cdot \nabla)q_v - C_c + S_v \quad (5)$$

D_v は水蒸気の分子拡散係数、 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である。

分子拡散係数 D_v は次式で表され。これはアインシュタイン・ストークスの式より温度に依存する。

$$D_v = D_0 T \quad (6)$$

ここで D_0 は分子拡散係数を決定するためのパラメータである。

相転移によって発生する湯気の量 C_c は次式で表される。

$$C_c = \alpha(q_v - q_s) \quad (7)$$

ここで α は相転移率である。

q_s は飽和水蒸気密度を表し、次式で与えられる。

$$q_s = \min \left(S_a \exp \left(\frac{-S_b}{T + S_c} \right), q_v + q_c \right) \quad (8)$$

ここで S_a, S_b, S_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。

$$\frac{\partial T}{\partial t} = D_t \nabla^2 T - (v \cdot \nabla)T + Q C_c + S_T \quad (9)$$

ここで D_t 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し、発生量の分布はユーザにより定義する。

2.3 実装

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温度 T 、水蒸気密度 q_v 、湯気の密度 q_c を割り付ける。初期状態として水蒸気源と熱源が存在する部分には水蒸気密度と温度の固定値を割り付けられる。全体の処理の流れとしては温度による浮力の計算、水蒸気と温度の拡散の計算、水蒸気と温度および湯気の移流の計算、圧力と質量保存則の効果の計算を繰り返す。浮力は計算対象の格子の温度と周囲の格子の温度の差から計算を行う。水蒸気と温度の拡散の計算は拡散方程式の数値解法を用いる。水蒸気と温度および湯気の移流の計算、圧力と質量保存則の効果の計算には5章に記載した Fedkiw ら [3] の手法を用いる。

3 結果

湯気のシミュレーションの結果を図2に示す。実装では底面に一様な温度、水蒸気量を定義した結果である。レンダリングには [1] にある独自のレンダリングモジュールを用いた。提案した手法を用いることで湯気の発生と消滅に近い映像を出力できることを確認した。

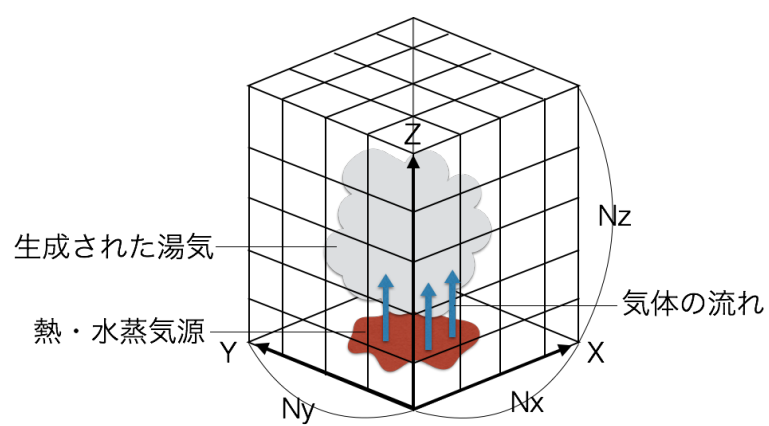


図 1: 湯気のシミュレーション空間

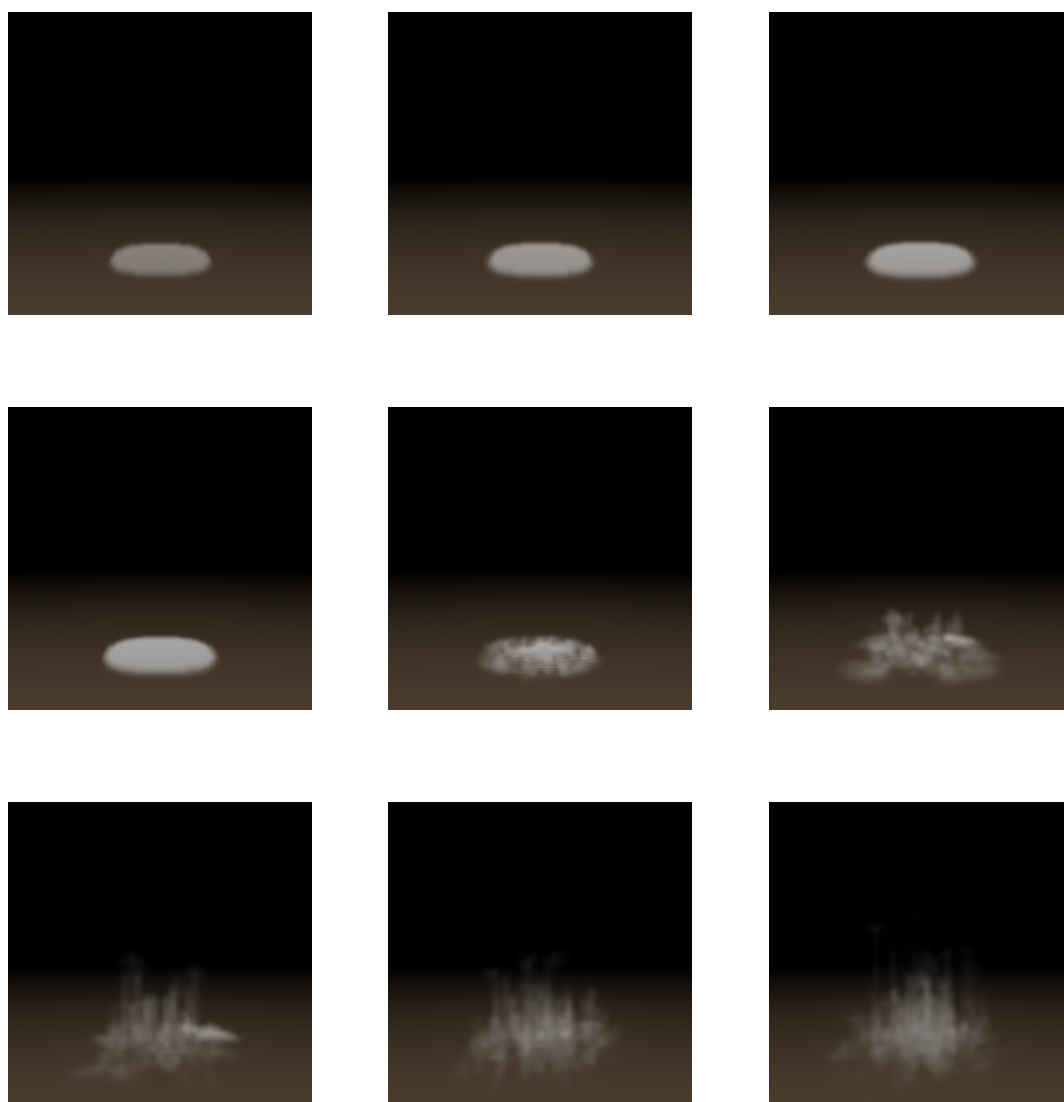


図 2: 湯気のシミュレーション結果

4 今後の研究計画

4.1 研究の方針

本論文で提案したモデルの場合、湯気らしい映像を得ることができたがパラメータにより得られる映像がかなり異なる。そのため湯気らしいパラメータについて調査と調整を行う。レンダリングには簡易的な方法を用いているため POV-Ray, PBRT, Mitsuba などのレンダリングソフトウェアを用いてレンダリングを行う。これによりリアリティのある映像を得ることができると考えられる。また拡散方程式の解法として現在は陽解法を用いているが、不安定になる場合があるため陰解法による解法にも対応を行う。上記が完了した場合、下記に記載の発展的研究も実施する。

4.2 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、[15] 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 [18] の利用が検討できる。また湯気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 [21] 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。

4.3 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 [15] 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ことで比較を行っている。

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。

5 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。

5.1 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。

5.2 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。

$$\nabla \cdot v = 0 \quad (10)$$

$$\frac{\partial v}{\partial t} = -(v \cdot \nabla)v - \nabla p + f \quad (11)$$

式 (10) は質量保存、式 (11) は運動量保存を表す。 p は気体の圧力、 f は外力を表す。式 (11) 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 (11) の左辺で時間刻み幅 Δt を利用し、右辺の圧縮項をなくした式 (12) を解いて中間速度場 u^* を求める。

$$\frac{v^* - v}{\Delta t} = -(v \cdot \nabla)v + f \quad (12)$$

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。

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot v^* \quad (13)$$

式 (13) 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 (∇ を掛ける) と $\frac{\nabla \cdot v - \nabla \cdot v^*}{\Delta t} = -\nabla^2 p$ となる。式 (10) $\nabla \cdot v = 0$ 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 (13) 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。

$$v = v^* - \Delta t \nabla p \quad (14)$$

式 (14) は $\frac{v - v^*}{\Delta t} = -\nabla p$ の両辺に Δt 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 ρ は煙の速度に沿って運ばれることを仮定する。

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

$$\frac{\partial \rho}{\partial t} = -(v \cdot \nabla)\rho \quad (16)$$

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。

$$f_{buoy} = -\alpha \rho z + \beta(T - T_{amb})z \quad (17)$$

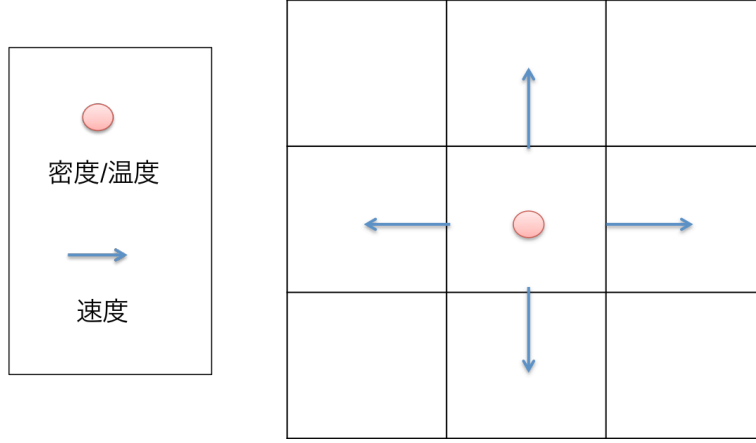


図 3: 空間の差分化

$z = (0, 0, 1)$ は上の垂直方向のベクトル、 T_{amb} は大気環境温度、 α と β 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 ω を取得する。

$$\omega = \nabla \times v \quad (18)$$

低い渦濃度から高い渦濃度へのベクトルを正規化する。

$$N = \frac{\eta}{|\eta|} (\eta = \nabla|\omega|) \quad (19)$$

以下より追加する外力を求める。

$$f_{conf} = \epsilon h(N \times \omega) \quad (20)$$

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

5.3 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 3) 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 4) 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 (11) 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 Δt 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 (17) に定義される浮力、式 (20) に定義される渦度強制による力がある。

次に式 (12) 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。

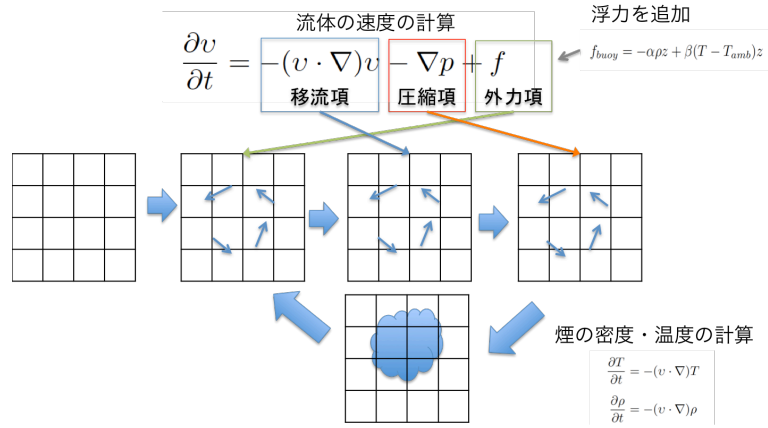


図 4: 処理全体の流れ

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 (13) 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。

参考文献

- [1] smoke3d. <https://code.google.com/archive/p/smoke3d/>.
- [2] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, and Tomoyuki Nishita. A Simple , Efficient Method for Realistic Animation of Clouds. *SIGGRAPH*, 2000.
- [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 BFECC 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] Viorel Mihalef, Dimitris Metaxas, and Mark Sussman. Simulation of two-phase flow with sub-scale droplet and bubble effects. *Computer Graphics Forum*, Vol. 28, No. 2, pp. 229–238, 2009.
- [13] R Miyazaki, Y Dobashi, and T Nishita. Simulation of Cumuliform Clouds Based on Computational Fluid Dynamics. 2002.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] MB Nielsen and Ole Østerby. A Two-Continua Approach to Eulerian Simulation of Water Spray. *ACM Transactions on Graphics*, Vol. 32, No. 4, p. 1, 2013.
- [19] 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.
- [20] Jos Stam. Stable Fluids. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 121–128, 1999.
- [21] Yongning Zhu and Robert Bridson. Animating sand as a fluid. *ACM Transactions on Graphics*, Vol. 24, No. 3, p. 965, 2005.
- [22] 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.