

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 M2

佐野 宏行

平成 28 年 8 月 16 日

1 はじめに

本書では 2016 年 8 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。なお本研究では常温の空気中に存在する沸騰の状態ではない液体の水から湯気が発生している状態を対象とする。

2 既存研究

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 [2], 雲 [1][11][10], 火山噴煙 [12][13], 水煙 [14], 水滴と水泡 [9] 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 [16] はあるが湯気に関しては考慮されない。[4] 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。

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 [11][10] 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 [4], 宮崎ら [10], 蒸気分子の拡散のモデルは宮崎ら [10] により CG に適用された。

本研究では湯気の流体现象の計算に粒子法と格子法の両方を用いる FLIP 法をベースとした手法を採用する。CG において粒子法と格子法を組み合わせることによる手法は雨水 [7], 雪 [15] の表現において適用される。

3 提案法

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と消滅のプロセスをもとに格子法と粒子法を用いた手法を提案する。流体の空気と水蒸気の気体部分は格子法, 湯気の水滴部分は粒子法によって表現を行い, 流体との相対速度から発生する抗力を考慮した湯気の動きを再現する。流体のモデルは Fedkiw ら [2] の手法に基づき非粘性, 非圧縮の流体を仮定する。熱, 水蒸気の発生と移動は熱拡散と分子拡散, 流体の速度による自然対流による移動の双方を考慮する。水滴から水蒸気のための相転移は宮崎ら [11][10] による飽和水蒸気量を利用するモデルを格子法と粒子法を用いる手法へ適用する。

4 湯気のシミュレーション

4.1 湯気の発生と消滅のプロセス

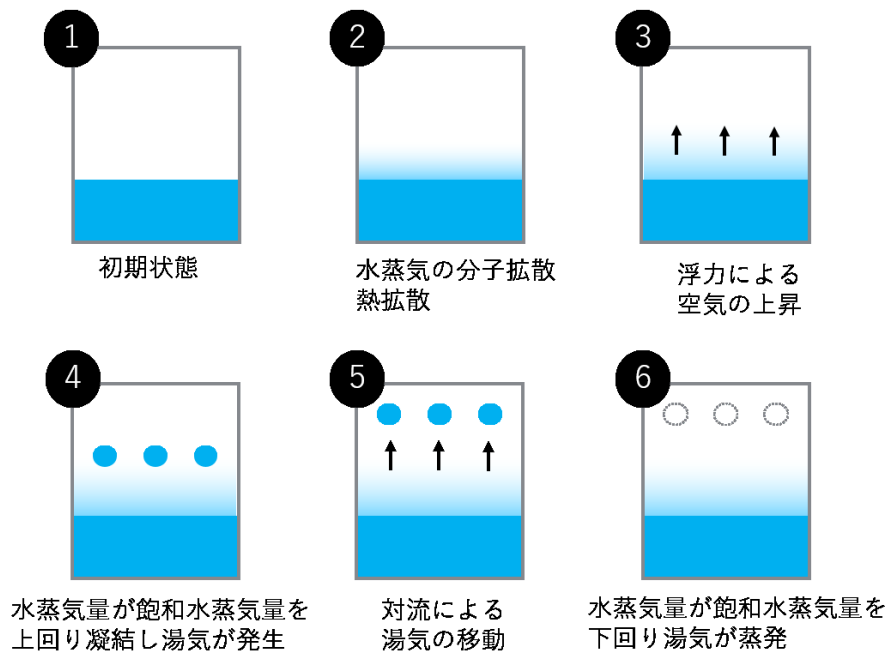


図 1: 湯気の発生と消滅のプロセス

湯気が発生してから消滅するまでのプロセスを図 (1) に記載する。初期状態としては常温の空気内に沸騰状態ではないが空気より高い温度の水があると仮定する。水は水面上から蒸発し, 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。浮力により空気に上方向の力が働き, 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。

4.2 シミュレーションモデル

流体の速度 $v = (u, v, w)$ 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。

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

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

p は圧力, B は浮力, f は風などによる外力を表す。浮力 B はブシネスク近似より式 (19) で定義する。

$$B = k_b \frac{T - T_a}{T_a} z \quad (3)$$

k_b は浮力の係数, T は流体の温度, T_a は環境温度, 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。

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

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

D_v は水蒸気の分子拡散係数, C_s 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。

温度 T は次式で表される。

$$\frac{\partial T}{\partial t} = -(v \cdot \nabla)T + D_t \nabla^2 T + Q C_s + S_T \quad (6)$$

ここで D_t は熱拡散率, Q は潜熱係数を表す。右辺第一項は熱対流, 第二項は熱拡散, 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。

分子拡散係数 D_v は次式で表される。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。

$$D_v = D_0 T \quad (7)$$

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。

相転移によって増減する湯気の量 C_s は次式で表される。

$$C_s = \begin{cases} \alpha(q_v - q_s) & q_v \geq q_s \\ \max(\alpha(q_v - q_s), -q_c) & q_v < q_s \end{cases} \quad (8)$$

$$q_s = \min \left(S_a \exp \left(\frac{-S_b}{T + S_s} \right), q_v + q_c \right) \quad (9)$$

ここで α は相転移率である。 q_s の \min 関数の第一引数は飽和水蒸気密度を表し S_a, S_b, S_s は飽和水蒸気密度を決定するためのパラメータである。 q_s の \min 関数の第二引数で湯気と水蒸気の密度の合計値を指定する。これにより水蒸気と湯気の密度の合計値が飽和水蒸気密度を超えている場合にも湯気の消滅を行う。湯気の消滅は湯気の密度以上行うことはない。

湯気は速度 v_s は流体との相対速度から発生する抗力, 重力を考慮して次式で表される。

$$\frac{dv_s}{dt} = -F_{drag} + mg \quad (10)$$

$$F_{drag} = -C_D(v_s - v)^\beta \quad (11)$$

F_{drag} は抗力, C_D は抗力を決定するためのパラメータ, β はレイノルズ数を表しこれは粒子の半径に依存する.

4.3 実装

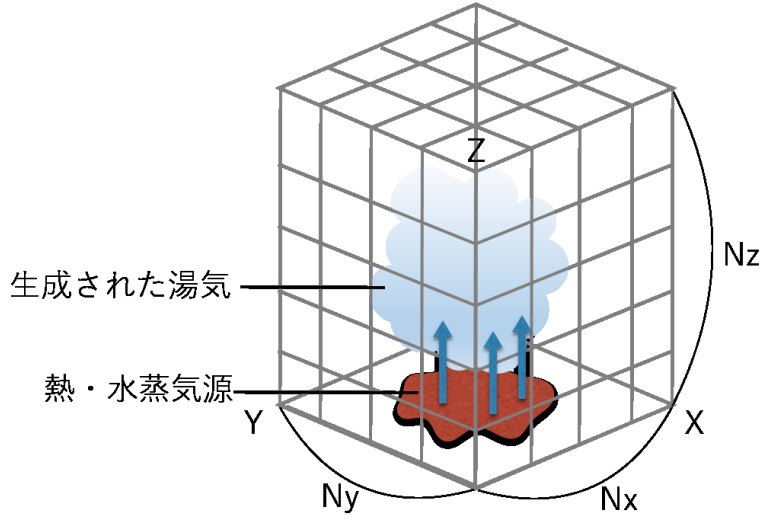


図 2: 湯気のシミュレーション空間

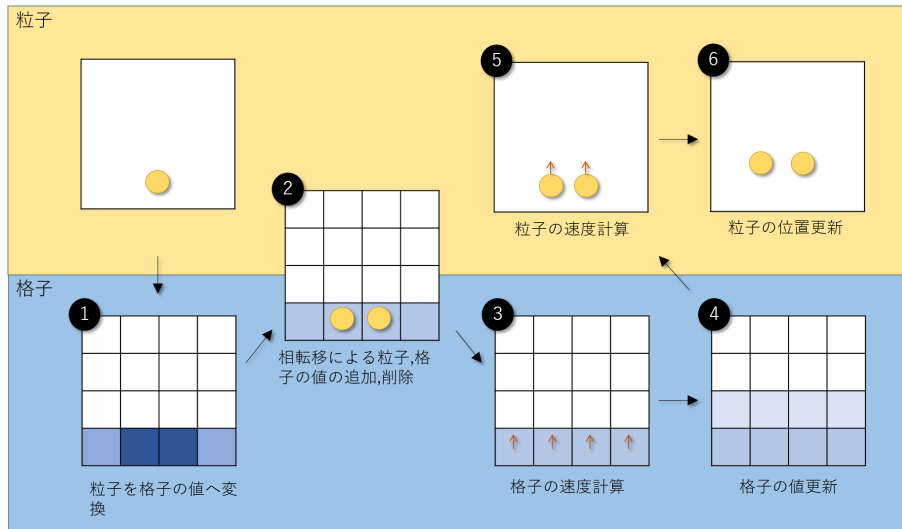


図 3: 提案法の概要

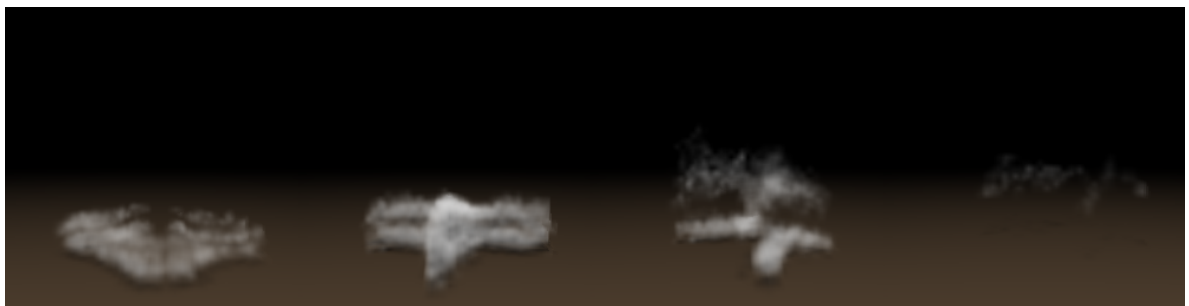
シミュレーション空間は $N_x \times N_y \times N_z$ の格子に分割し各格子点に温度 T , 水蒸気の密度 q_v , 流体の速度 v を割り付ける. 格子には温度, 水蒸気の密度を格子の面に定義するスタガード格子を採用する. 初期状態として水蒸気源と熱源が存在する部分には水蒸気密度と温度の固定値を割り付ける. 水

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。湯気は粒子により表現し粒子には格子空間上の位置、速度、質量が格納される。粒子は質量が一定で粒子同士の衝突、粒子から格子の流体の速度、温度に対する影響はないと仮定する。

図 6 に本シミュレーション空間上での 1 タイムステップ中の処理の流れを示す。

1. 粒子を格子の値へ変換。格子内の湯気の粒子の質量を合計し、すべての格子に対して湯気の密度 q_c を計算する。
2. 相転移による粒子、格子の値の追加、削除。式 (8) の相転移のモデルを用いて相転移によって増減する湯気の量 C_s を求め、湯気と水蒸気量の追加、削除を行う。 $C_s \geq 0$ の場合、格子内に湯気の粒子を追加する。湯気の粒子の位置は格子内のランダムな位置とし、速度は粒子の位置にある速度を格子面に定義される速度から線形補間により求め、質量は一定とする。これを追加した粒子の質量の合計が C_s になるまで処理を続ける。 $C_s < 0$ の場合、格子内の湯気の粒子の質量を削除する。これを削除した粒子の質量の合計が C_s になるまで処理する。湯気の粒子の追加と削除もしくは処理の後、温度へ C_s に依存した潜熱の追加、水蒸気の密度 q_v から C_s の減算、質量が 0 の湯気の粒子の削除処理を行う。
3. 格子の速度計算。非粘性、非圧縮のオイラーの運動方程式 (12,13) を Fedkiw ら [2] の手法に基づき外力、浮力、圧力、移流により流体の速度を計算する。
4. 格子の値更新。温度と水蒸気量を式 (18,5) に基づき計算する。式 (18,5) は共に第一項が対流項、第二項が拡散項となる。対流項はセミラグランジュ法、拡散項は拡散方程式の数値解析を用いる。拡散項の数値解法で本研究で採用した陽解法の場合は CFL 条件によりタイムステップ幅に厳しい制限を課す必要がある。
5. 粒子の速度計算。粒子の速度を式 (10) に基づき計算する。抗力 F_{drag} は格子と粒子の速度の間の相対速度により求める。この計算で用いる格子の速度は粒子の位置にある速度を格子面に定義される速度から線形補間により求める。
6. 粒子の位置更新。粒子の位置を粒子自体の速度を追加することにより求める。

5 結果



(1)20 タイムステップ後 (2)40 タイムステップ後 (3)60 タイムステップ後 (4)80 タイムステップ後

図 4: 湯気のシミュレーションのレンダリング結果

湯気のシミュレーションのレンダリング結果を図 (4) に示す。格子法と粒子法を用いることで、格子法のみでは実現が困難な水滴の発生位置による微細な湯気の動きを再現した。本レンダリング結果

では湯気の発生分布が大きく2つに分かれている様子を確認した。レンダリングは格子ごとに湯気の密度を合計し、ボリュームレイキャスティング法により行った。シミュレーション空間は $32 \times 32 \times 32$ の格子で、シミュレーションで発生した粒子数は3,000個から4,000個となった。計算時間はIntel Core i5-6200U 2.3GHz のCPUを用いて1タイムステップあたり3秒から4秒となった

6 今後の研究計画

6.1 研究の方針

本研究報告では格子法と粒子法を組み合わせることにより、湯気に近いアニメーションを生成することができた。ただし、現状は試行錯誤的に既存のモデルを組み合わせもしくは改変しているに過ぎないため、今後、本研究で対象としている湯気の定義の明確化、本手法の利用の妥当性、本研究の貢献は何かを明確とすることを念頭に置き、研究を継続する。

現時点の課題としては、パラメータにより得られる映像がかなり異なり、パラメータの調整が困難であることがあげられる。そのため実際の物理現象をもとに湯気らしいパラメータについて調査と調整を行うことが考えられる。また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。

6.2 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 [8] 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、[12] 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 [14] 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。

6.3 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 [3] や雲 [5] 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 [12] やプロミネンス [6] 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、もしくは物理的な実験値との比較により行うこととする。

7 Visual Simulation of Smoke 解説

Visual Simulation of Smoke[2] 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。

7.1 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。

7.2 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。

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

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

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 Δt を利用し、右辺の圧縮項をなくした式 (14) を解いて中間速度場 u^* を求める。

$$\frac{v^* - v}{\Delta t} = -(v \cdot \nabla)v + f \quad (14)$$

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。

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot v^* \quad (15)$$

式 (15) 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 (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 (15) 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.

$$v = v^* - \Delta t \nabla p \quad (16)$$

式 (16) 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.

$$\frac{\partial T}{\partial t} = -(v \cdot \nabla) T \quad (17)$$

$$\frac{\partial \rho}{\partial t} = -(v \cdot \nabla) \rho \quad (18)$$

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.

$$f_{buoy} = -\alpha \rho z + \beta (T - T_{amb}) z \quad (19)$$

$z = (0, 0, 1)$ は上の垂直方向のベクトル, T_{amb} は大気的环境温度, 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.

$$\omega = \nabla \times v \quad (20)$$

低い渦濃度から高い渦濃度へのベクトルを正規化する.

$$N = \frac{\eta}{|\eta|} (\eta = \nabla |\omega|) \quad (21)$$

以下より追加する外力を求める.

$$f_{conf} = \epsilon h (N \times \omega) \quad (22)$$

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

7.3 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 5) 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. この格子を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 6) 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 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 (19) に定義される浮力, 式 (22) に定義される渦度強制による力がある.

次に式 (14) 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.

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 (15) 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

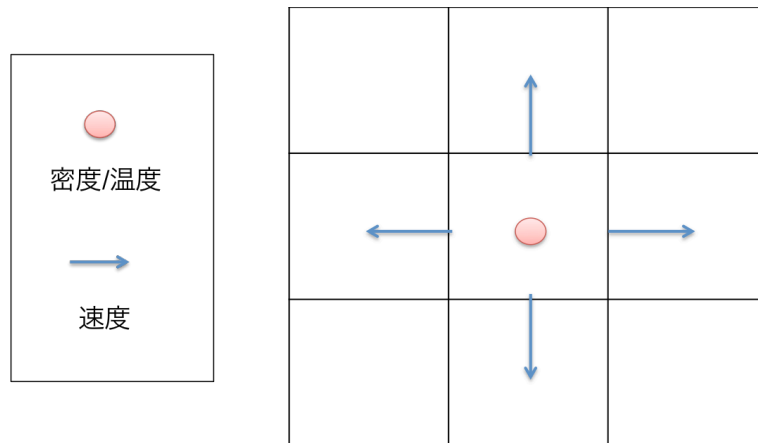


図 5: 空間の差分化

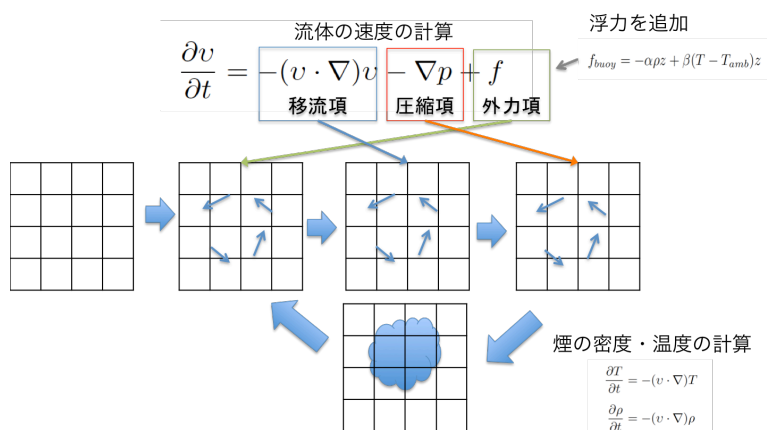


図 6: 処理全体の流れ

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] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, and Tomoyuki Nishita. A Simple , Efficient Method for Realistic Animation of Clouds. *SIGGRAPH*, 2000.
- [2] 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.
- [3] Nick Foster and Dimitri Metaxas. Realistic Animation of Liquids. *Graphical Models and Image Processing*, Vol. 58, No. 5, pp. 471–483, 1996.
- [4] Nick Foster and Dimitris Metaxas. Modeling the Motion of a Hot, Turbulent Gas. *SIGGRAPH*, 1997.
- [5] Geoffrey Y. Gardner. Visual simulation of clouds. *ACM SIGGRAPH Computer Graphics*, Vol. 19, No. 3, pp. 297–304, 1985.
- [6] Tomokazu Ishikawa and Yonghao Y U E Member. Visual Simulation of Solar Photosphere Based on Magnetohydrodynamics and Quantum Theory.
- [7] Tomokazu Ishikawa, Yonghao Y U E Member, Taichi Watanabe Member, and Kei Iwasaki Member. Visual Simulation of Glazed Frost Using Hybrid Heat Calculation. Vol. 3, No. 2, pp. 136–142, 2015.
- [8] 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.
- [9] 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.
- [10] R Miyazaki, Y Dobashi, and T Nishita. Simulation of Cumuliform Clouds Based on Computational Fluid Dynamics. 2002.
- [11] 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.
- [12] 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.
- [13] 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.

- [14] 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.
- [15] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. A material point method for snow simulation. *ACM Transactions on Graphics*, Vol. 32, No. 4, p. 1, 2013.
- [16] 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.