

気流を考慮した昼花火のシミュレーション

筑波大学

人間総合科学学術院人間総合科学研究群

情報学学位プログラム

2022年03月

矢花 明莉

気流を考慮した昼花火のシミュレーション

Daytime Fireworks Simulation with Airflow

氏名：矢花 明莉
Yabana Akari

近年、映画やゲームなどのエンタテインメントコンテンツにおいて、コンピュータグラフィクス (CG) 技術による表現が多く用いられている。その中でも、煙、水、炎など多岐にわたる自然現象を再現する上で物理シミュレーション技術は欠かせないものとなっている。花火はその対象の1つであり、夜花火のCGはゲームやインタラクティブコンテンツ等で用いられている。CG向けだけでなく、火薬の並びや打ち上げの挙動を物理シミュレーションを用いて精密に計算することで、実際の花火大会で打ち上げを行う前に花火師が試し打ちとしてシミュレーションを行うためにも用いられる。本研究では、花火の一種である昼花火を物理シミュレーションを用いて再現する手法を提案する。昼花火は昼に打ち上げる花火の総称であり、CG分野での研究は少ない。昼花火のCG作成技術の提案により、新しいエンタテインメントの提案、夜花火と同様に花火師の試し打ちに活用できると考えられる。

昼花火にはいくつか種類があるが、本研究では「煙竜」と呼ばれる昼花火を対象とする。煙竜は、パラシュートに発煙筒をつけたものを打ち上げ、煙がパラシュートの落下とともに描く螺旋状の軌跡を鑑賞する花火である。提案手法では、煙竜をパラシュートと煙のシミュレーションの組み合わせで再現する。パラシュートのシミュレーションでは、屋根部分を航空力学に基づいたパラシュートモデル、荷物部分を自由落下する質点としてモデル化し、その間をつなぐ紐をバネ、ダンパを用いて表現する。煙は格子法による流体シミュレーションによって再現する。煙竜では煙がパラシュートの動きに沿って長い螺旋の軌跡を描くため、シミュレーションの範囲が広くなり、結果として計算解像度を高くできないという問題が発生する。本研究では、広範囲のシミュレーションを高解像度かつ高速に実行する方法として、パラシュートの周りは密な格子、離れた部分は疎な格子を用いる Overlapping grid という手法を導入する。これらのシミュレーションを組み合わせ、パラシュートの荷物部分から煙を噴出することで、煙竜の物理シミュレーションを実現した。

提案手法ではパラシュートによって煙が遮られる点については考慮できていない。また、打ち上げ部分の再現や、GPU実装によるシミュレーション高速化が今後の課題として挙げられる。

主研究指導教員：藤澤 誠
副研究指導教員：三河 正彦

目次

第1章 序論	1
1.1 背景	1
1.1.1 昼花火について	1
1.2 目的	2
1.3 関連研究	3
1.3.1 パラシュートシミュレーション	3
1.3.2 流体シミュレーション	3
第2章 提案手法	5
第3章 パラシュートシミュレーション	8
3.1 パラシュートモデルの概要	8
3.2 パラシュートの支配方程式	8
3.2.1 キャノピーの支配方程式	8
3.2.2 ペイロードの支配方程式	11
3.3 抗力とピッチングモーメント	11
3.3.1 抗力	11
3.3.2 ピッチングモーメント	13
3.4 バネ・質点モデル	14
第4章 流体シミュレーション	16
4.1 格子法によるシミュレーション	16
4.1.1 流体の支配方程式	16
4.1.2 ナビエ・ストークス方程式の数値解法	16
4.2 Overlapping grid による高速化	19
4.2.1 境界伝達	21
4.2.2 内部伝達	22
4.2.3 高解像度化	22
4.3 レンダリング	23
第5章 結果	24
5.1 煙のパラメータ調整による変化	24
5.2 C_n の値による軌跡の変化	27
5.3 Overlapping grid による高速な高解像度化	30
5.4 複数の煙竜があるシーン	32

第6章 考察	35
6.1 煙のパラメータ調整による変化	35
6.2 C_n の値による軌跡の変化	35
6.3 Overlapping grid による高速な高解像度化	36
6.4 複数の煙竜があるシーン	37
第7章 結論	38
謝辞	39
参考文献	40

目 次

1.1	煙柳 (石川靖子さん撮影)	2
1.2	煙竜 (石川靖子さん撮影)	2
1.3	ラムエア型パラシュート	3
1.4	円形パラシュート	3
2.1	本研究の概要	5
2.2	提案手法の全体像	6
3.1	パラシュートの構成	8
3.2	提案するモデル	8
3.3	円形パラシュートモデル [1]	9
3.4	キャノピーにかかる力	10
3.5	迎角 α_{sp}	12
3.6	抗力係数 C_D の実験による計測データ	12
3.7	ピッチングモーメント係数 C_m の実験による計測データ	13
3.8	C_n の値による軌跡の変化	14
3.9	バネ, ダンパによる力	15
4.1	スタガード格子	17
4.2	セミ・ラグランジュ法によるバクトレース	18
4.3	ローカルグリッドとグローバルグリッド	20
4.4	煙シミュレーションの流れ	21
4.5	境界伝達	22
4.6	内部伝達	22
4.7	最終グリッドの生成	23
4.8	スライスを用いたボリュームレンダリング	23
5.1	渦度 $\epsilon = 20.0$, 煙の拡散係数 $\nu_s = 0.0005$ のシミュレーション結果	25
5.2	渦度 $\epsilon = 0.0$, 煙の拡散係数 $\nu_s = 0.0005$ のシミュレーション結果	26
5.3	渦度 $\epsilon = 20.0$, 煙の拡散係数 $\nu_s = 0.0$ のシミュレーション結果	27
5.4	$C_n = 2.7$ のシミュレーション結果	28
5.5	$C_n = 4.2$ のシミュレーション結果	29
5.6	高解像度でシミュレーションした結果	30
5.7	Overlapping grid を用いた結果	32
5.8	複数の煙竜 (画像提供:秋田県大仙市)	33
5.9	2つの煙竜があるシーン	34
6.1	ペイロード付近拡大図 (400 ステップ)	35
6.2	ペイロード付近拡大図 (450 ステップ)	36

6.3	ペイロードから離れた部分の拡大図 (450 ステップ)	37
-----	---------------------------------------	----

表 目 次

2.1 記号一覧	7
5.1 開発・実行環境	24

第1章 序論

1.1 背景

近年, 映画やゲームなどのエンタテインメントコンテンツにおいて, コンピュータグラフィクス (CG) 技術による表現が多く用いられている. 特に, 煙, 炎, 水などの自然現象は複雑であり, アーティストが手作業で CG を作成する場合には, その仕上がりがアーティストの作業量や技量に依存してしまう. このような問題を解決するため, 物理シミュレーション技術を用いて自然現象をコンピュータによる自動的な計算によって再現する研究が数多く行われている.

物理シミュレーションの対象となる自然現象は多岐にわたるが, 本研究では打ち上げ花火に着目する. 打ち上げ花火の中でも, 夜花火は夜空に火薬を打ち上げ, 光や音などの物理現象を楽しむコンテンツであり, その CG はゲームやインタラクティブコンテンツ等で用いられることから, 多くの研究がなされている [2][3]. また, 火薬の並びや打ち上げの挙動を物理シミュレーションを用いて精密に計算することで, 実際の花火大会で打ち上げを行う前に花火師が試し打ちとしてシミュレーションを行うためにも用いられる. 一方で, 夜花火と比べて知名度は低いが, 昼に打ち上げる昼花火も存在する. 昼花火は音や煙を楽しむコンテンツであり, 現在は少数の花火大会でしか開催されておらず, また CG 分野においてもその研究は少ないが, 夜のシーンだけでなく, 昼のシーンでも使える表現として, 物理シミュレーションで再現できれば, 新しい表現の可能性を広げることができる.

1.1.1 昼花火について

昼花火は大別すると音物, 割物, 吊り物, 細工小物の 4 種類に分けられる. 音物には大きな爆発音が一度鳴る号砲や, 複数回鳴る段雷, 万雷といった種類があり, 運動会の告知などに使われる. 割物と吊り物は, 煙の入った玉を打ち上げて鑑賞する花火であり, 大曲の花火大会をはじめとしたいくつかの花火大会で見られる [4]. 割物には煙が菊のように広がる煙菊, 柳のように煙が垂れて落ちる煙柳 (図 1.1), 吊り物にはパラシュートに発煙筒をつけて落下させる煙竜 (図 1.2) という種類がある. また, それぞれ煙に色をつけたものを彩煙菊, 彩煙柳, 彩煙竜とも呼ぶ. 細工小物は, 動物などの形をした袋状の薄紙を打ち上げる花火である. 細工小物に関しては, 電線に引っかかったの事故や, 子供が追いかけることによる交通事故の危険性があるため, 現在はあまり見られない.



図 1.1: 煙柳 (石川靖子さん撮影)

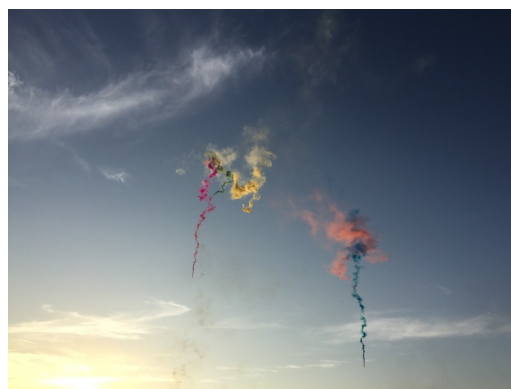


図 1.2: 煙竜 (石川靖子さん撮影)

1.2 目的

本研究の目的は、物理シミュレーションを用いて昼花火の吊り物の一種である煙竜を再現することである。CG分野において昼花火を対象とした研究は少なく、中でも煙竜の再現を目的とした研究は我々の調べた限りではまだない。煙竜は花火大会で昼花火が行われる際に頻出であり、また、パラシュートの設定によって見た目の制御がある程度可能であることから、シミュレーションを用いた実際の制作補助にも有用であると考えられる。煙竜は他の昼花火と比べても、気流の変化に影響されやすいため、本研究では、煙竜を周りの気流も考慮して再現する手法を提案する。本提案手法は、ゲームやインタラクティブアートにおける新たなエンタテインメントの提案や、花火師が花火大会前にシミュレーションで試し打ちをする場合に活用できると考えられる。

煙竜は、パラシュートに発煙筒をつけたものを打ち上げ、パラシュートのゆっくりとした落下とともに煙が螺旋の軌跡を描く様子を鑑賞する花火である。パラシュートが風に揺られ、また、煙が風で広がることで独特な煙の形を作り出し、それを楽しむものである。本研究では、煙竜をパラシュートのシミュレーションと煙のシミュレーションを組み合わせることにより再現する。パラシュートのシミュレーションでは、屋根部分を航空力学に基づいたパラシュートモデル、荷物部分を自由落下する質点としてモデル化し、その間をつなぐ紐をバネ、ダンパを用いて表現する。特に、煙竜で特徴的な螺旋を描くような動きは、パラシュートと荷物部分を繋ぐ複数本の紐の長さや密接に関係している。紐の長さに違いがある場合に、パラシュート自体が自転する力が発生し、その力により螺旋を描く軌跡が得られることから、紐の長さの違いによってかかる力も再現する。煙は、格子法を用いた流体シミュレーションによって再現する。煙竜では煙は落下する軌跡を残す役割を持つため、軌跡に合わせるシミュレーション範囲を広く取る必要がある。しかし、これは計算解像度の低下をまねく。そこで、広範囲のシミュレーションを高速に行う Overlapping grid という手法 [5] を用いる。Overlapping grid では、煙の発生源の周りは細かい格子、離れた部分は粗い格子を用いることで、高速かつ高精細なシミュレーションを可能にしている。

1.3 関連研究

1.3.1 パラシュートシミュレーション

パラシュートには、断面が翼のような形状をしているラムエア型パラシュート (図 1.3) と、円形パラシュート (図 1.4) の 2 種類がある。ラムエア型パラシュートは操縦可能のためエアスポーツ等で利用されるが、屋根部分が潰れやすいという特徴がある。ラムエア型パラシュートの数値計算モデルは訓練用シミュレータ開発のために提案されている [6]。円形パラシュートは操縦できず、風の流れて従って移動するが、屋根部分は潰れにくく耐久性に優れているという特徴がある。本研究の対象である煙竜では、円形パラシュートが用いられている。

円形パラシュートをシミュレーションするため、様々な方法が提案されている。有限要素法を用いてパラシュートの屋根部分で発生する複雑な気流を再現する手法 [7][8] では、パラシュートの屋根の形状変化を考慮した精密なシミュレーションができています。しかし、計算時間やメモリ消費量が大きい点が問題である。Guglieri[9] や Tang ら [10] は、屋根部分を剛体とみなしたモデルを使用することで、精度は落ちるが、シミュレーションの計算コストを低くしている。Dobrokhodov ら [1] は、円形パラシュートを実際のパラシュートの動きから得られる計測データを用いてシミュレーションしている。この研究では、実際のパラシュートの軌跡とシミュレーション結果を比較し、その有効性を確認している。本研究では、Dobrokhodov ら [1] が用いている手法をパラシュートの屋根部分のモデルとして利用する。また、パラシュートの紐部分は Guglieri[9] の手法と同様にバネ・質点モデルを用いて表現する。

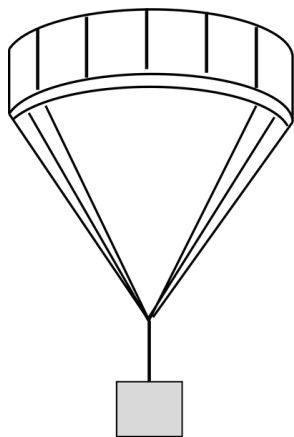


図 1.3: ラムエア型パラシュート

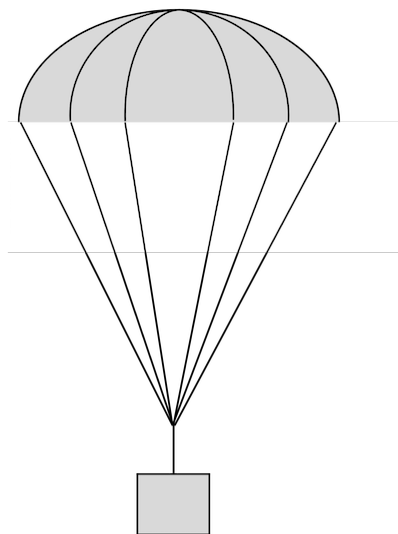


図 1.4: 円形パラシュート

1.3.2 流体シミュレーション

流体の動きを物理シミュレーションで再現するためには、流体の支配方程式であるナビエ・ストークス方程式を離散化して解く必要がある。離散化手法は大別して格子法と粒子法に分けられる。それぞれ、格子法はシミュレーション空間を格子で区切り、各格子に速度や密度などの物理量を定義して計算する方法、粒子法は流体を粒子の集合で表現し、空間内を自由に動き回る粒子を用いて計算を行う方法である。煙のように流体の境界が不明確な現象では、格子法が多く用いられる。Foster ら [11] は、格子法を用いた 3 次元の流体シミュレーション

手法を提案した。しかし、この手法ではタイムステップ幅の大きさが CFL 条件を満たす必要があり、タイムステップ幅が大きい場合にシミュレーションが発散してしまう。Stam[12] は、ナビエ・ストークス方程式の一部を解く際にセミ・ラグランジュ法を用いることで、タイムステップ幅を大きくしても安定したシミュレーションをする手法を提案した。格子法によるシミュレーションでは、ナビエ・ストークス方程式を離散化して解く際に、数値拡散によって細かな流体の動きが失われてしまう。これを解決するために、Fedkiw ら [13] は、細かな流れの特徴を追加する Vorticity confinement を提案した。

これらの手法を用いたシミュレーションでは、広い空間を高精細にシミュレーションしたい場合に高解像度な格子を用いる必要があり、結果として計算時間が増大してしまう。そこで、高速に高精細なシミュレーションをするための方法が提案されている。低精細かつ高速なシミュレーションをベースにして、後处理的に高精細な煙を追加する方法として、ウェーブレット法を用いた手法 [14] や、パッチベース合成を用いた手法 [15] が提案されている。また、格子の大きさを適用的に変化させることでシミュレーションを高速化する手法も提案されている。Losasso ら [16] は水と煙のシミュレーションにおいて、八分木構造を用いて流体の動きが複雑な部分は格子を細かくすることで、高速に詳細な見た目を表現した。Setaluri ら [17] は、SPGrid と呼ばれるグリッド構造を用いることでメモリ使用量の少ない実装を実現した。Dobashi ら [5] は、移動する剛体と煙のシミュレーションにおいて、剛体の周りには細かい格子、離れた部分は粗い格子を用いる手法を提案した。本研究では移動するパラシュートを煙の発生源としているため、Dobashi ら [5] の手法を用いてシミュレーションを高速化する。

第2章 提案手法

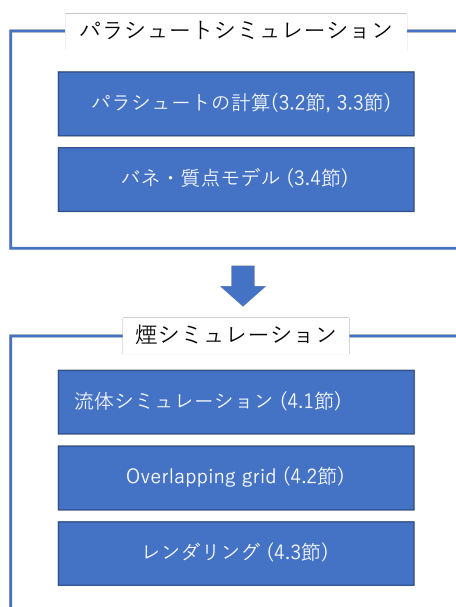


図 2.1: 本研究の概要

本研究で提案する手法の概要を図 2.1 に示す. 本研究の対象である煙竜は, 発煙筒をつけたパラシュートが落下する現象である. この再現のため, 本研究ではパラシュートのシミュレーションと煙のシミュレーションを組み合わせた手法を提案する (図 2.2). シミュレーションはパラシュート, 煙の順で行う.

パラシュートのシミュレーションでは, パラシュートを屋根部分, 荷物部分に分けてモデル化し, 間を繋ぐ紐をバネ・質点モデルを用いて表現する. 提案手法では, パラシュートの屋根部分は航空力学的な力を考慮したパラシュートモデル, 荷物部分は自由落下する質点として, 1 ステップごとにそれぞれにかかる力を計算し, そこからパラシュートの速度を算出し, 位置を更新する.

煙は, 流体の支配方程式であるナビエ・ストークス方程式を格子法を用いて解くことによりシミュレーションする. ナビエ・ストークス方程式の移流項を解く際にはセミ・ラグランジュ法, 乱流の表現には Vorticity confinement[13] を用いる. また, 本研究では広範囲のシミュレーションが必要であるため, Overlapping grid[5] を用いて計算を高速化する. パラシュートのシミュレーションで求めた荷物部分の位置を煙の発生源として, 1 ステップごとに各格子点の速度や密度, 圧力の値を更新し, 煙の密度を用いてボリュームレンダリングすることで煙竜による煙を CG として表現する.

表 2.1 に本論文中で用いる記号の一覧を示す. 以降ではベクトルを太字で表し, パラシュートと流体の速度, 位置ベクトルを区別するために, パラシュートの場合は大文字, 流体は小

文字で表現する.

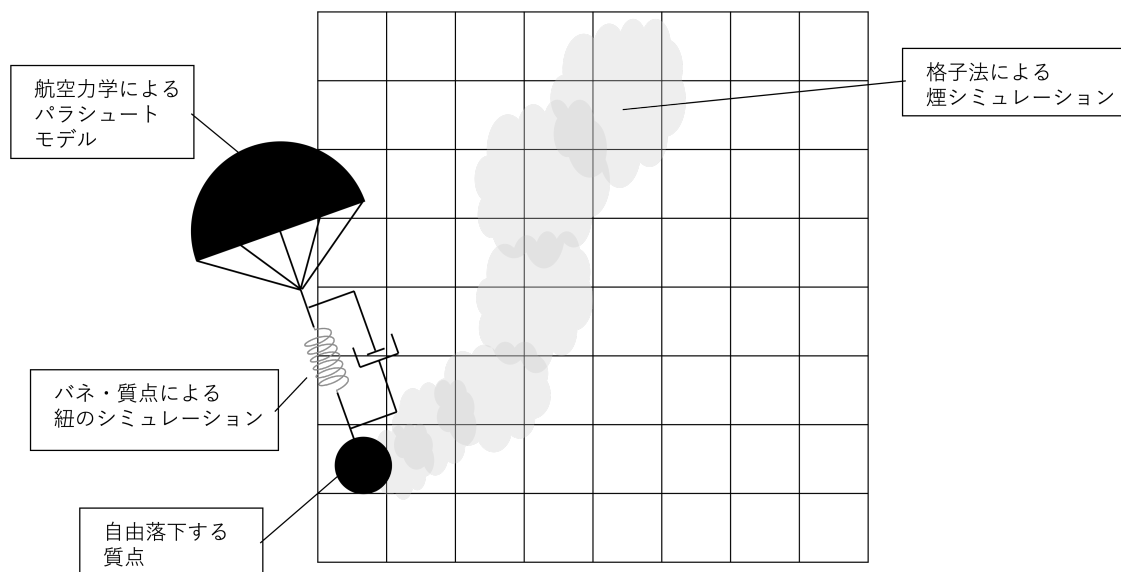


図 2.2: 提案手法の全体像

表 2.1: 記号一覧

F_{canopy}	キャノピーにかかる力
F_{aero}	抗力
$F_{payload}$	ペイロードにかかる力
$F_{tension}$	引張力
$F_{damping}$	ダンパによる減衰力
F_{spring}	バネの力
G_p	パラシュートの物体座標系から見た重力
G	重力
M_{canopy}	キャノピーのモーメント
M_{aero}	ピッチングモーメント
$V = [u, v, w]^T$	キャノピーの速度
$\Omega = [p, q, r]^T$	キャノピーの角速度
ϕ, θ, ψ	オイラー角
X	キャノピーの位置
V_p	ペイロードの速度
X_p	ペイロードの位置
$V_a = [u_a, v_a, w_a]^T$	対気速度
W	風ベクトル
I	慣性モーメント
I_{xx}, I_{yy}, I_{zz}	慣性テンソルの対角成分
m	パラシュートの質量
m_a	キャノピーの半球内に滞留する空気質量
m_p	ペイロードの質量
$\alpha_{11}, \alpha_{33}, \alpha_{44}, \alpha_{66}, \alpha_{15}$	仮想質量
$k_{11}, k_{33}, k_{44}, k_{66}, k_{15}, \gamma$	仮想質量に関する定数係数
z_G	パラシュートの重心
z_P	パラシュートの圧力中心
R_0	通常時のキャノピーの半径
R_p	膨張したキャノピーの半径
C_D	抗力係数
C_m	ピッチングモーメント係数
C_n	M_{aero} の yaw 成分
S_0	キャノピーの投影面積
α_{sp}	迎角
α	迎角の y 軸周りの成分
β	迎角の x 軸周りの成分
R	グローバル座標系から物体座標系への変換行列
l	キャノピー, ペイロード間の初期距離
k	バネ定数
k_d	減衰係数
t	時間
u	流体の速度
u^*, u^{**}, u^{***}	流体の中間速度
u^l	ローカルグリッドの速度
u^g	グローバルグリッドの速度
x	流体の計算点位置
c	ローカルグリッドの中央位置
ν	動粘性係数
ν_s	煙の拡散係数
ρ	流体の密度
ρ_s	煙の密度
p	圧力
T	温度
$T_{ambient}$	周囲の環境温度
E_r	放射熱エネルギー
a	熱拡散率
λ	熱伝導率
c_p	熱容量
h	グリッド幅
g	重力加速度
β_{buoy}	浮力係数
F_{ext}	外力
F_{vort}	渦を強調する力
ω	渦度
ϵ	渦度を調整するパラメータ
S	煙の発生源

第3章 パラシュートシミュレーション

3.1 パラシュートモデルの概要

煙竜で使われる円形パラシュートの模式図を図 3.1 に示す。パラシュートの屋根部分はキャノピー、荷物部分はペイロード、キャノピーから出る複数本の紐はサスペンションライン、サスペンションラインを束ねている部分はライザーと呼ばれており、本研究でも以降はこのように表記する。

提案手法では、キャノピーには航空力学的な力である抗力やピッチングモーメントを考慮した円形パラシュートモデル [1] を用いる。ペイロードは自由落下する質点としてモデル化し、簡略化のためサスペンションラインとライザーは同一として扱い、バネ・質点モデルを用いて表現する (図 3.2)。

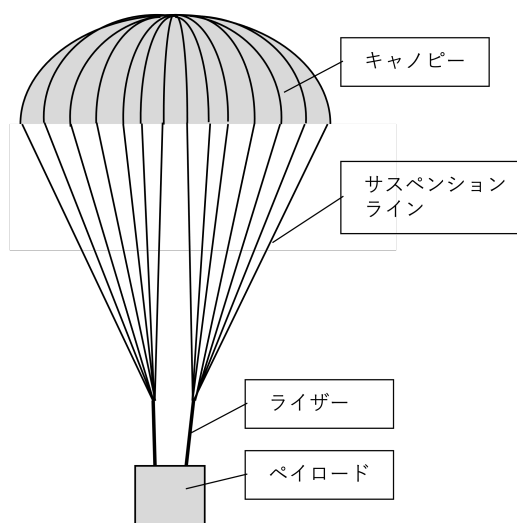


図 3.1: パラシュートの構成

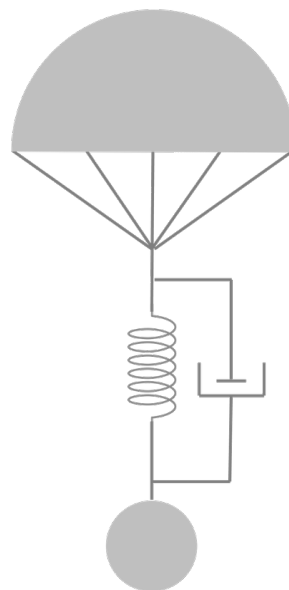


図 3.2: 提案するモデル

3.2 パラシュートの支配方程式

3.2.1 キャンopiesの支配方程式

本研究ではキャノピーを円形パラシュートモデル [1] を用いて 6 自由度の剛体として扱う。物体座標系から見た 6 自由度の剛体の一般的な支配方程式を以下に示す。

$$\mathbf{F}_{canopy} = m\dot{\mathbf{V}} + m(\boldsymbol{\Omega} \times \mathbf{V}) \quad (3.1)$$

$$\mathbf{M}_{canopy} = \mathbf{I}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{I}\boldsymbol{\Omega} \quad (3.2)$$

ここで, m は質量, I は慣性モーメント, \mathbf{V} は速度, $\boldsymbol{\Omega}$ は角速度である.

速度 \mathbf{V} , 角速度 $\boldsymbol{\Omega}$ は以下のように表される.

$$\mathbf{V} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (3.3)$$

$$\boldsymbol{\Omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.4)$$

Dobrokhodov らのモデル [1] では, 式 (3.1)(3.2) に仮想質量を考慮し, 図 3.3 の CG の位置を重心, O の位置を原点として考えることで, 以下をパラシュートの支配方程式としている.

$$\mathbf{F}_{canopy} = \begin{bmatrix} (m + \alpha_{11})(\dot{u} - vr) + (m + \alpha_{33})wq + (mz_G + \alpha_{15})(\dot{q} + rp) \\ (m + \alpha_{11})(\dot{v} - ur) - (m + \alpha_{33})wp - (mz_G + \alpha_{15})(\dot{p} + qr) \\ (m + \alpha_{33})\dot{w} - (m + \alpha_{11})(uq - vp) - (mz_G + \alpha_{15})(p^2 + q^2) \end{bmatrix} \quad (3.5)$$

$$\mathbf{M}_{canopy} = \begin{bmatrix} (I_{xx} + \alpha_{44})\dot{p} - (mz_G + \alpha_{15})(\dot{v} - wp + ur) - (I_{yy} + \alpha_{44} - I_{zz} - \alpha_{66})qr + (\alpha_{33} - \alpha_{11})vw \\ (I_{yy} + \alpha_{44})\dot{q} - (mz_G + \alpha_{15})(\dot{u} + wq - vr) + (I_{yy} + \alpha_{44} - I_{zz} - \alpha_{66})pr - (\alpha_{33} - \alpha_{11})uw \\ (I_{zz} + \alpha_{66})\dot{r} + (I_{yy} - I_{xx})pq \end{bmatrix} \quad (3.6)$$

z_G はパラシュートの重心, I_{xx}, I_{yy}, I_{zz} は慣性テンソルの対角成分であり, Dobrokhodov らのモデル [1] に示されている値を用いた.

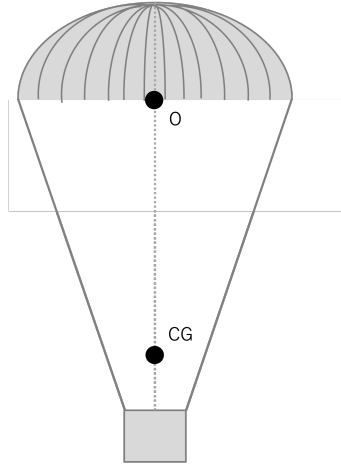


図 3.3: 円形パラシュートモデル [1]

$\mathbf{F}_{canopy}, \mathbf{M}_{canopy}$ はそれぞれ図 3.4 に示すキャノピーにかかる力とモーメントであり, 以下のように表せる.

$$\mathbf{F}_{canopy} = \mathbf{F}_{aero} + m\mathbf{G}_p \quad (3.7)$$

$$\mathbf{M}_{canopy} = \mathbf{M}_{aero} \quad (3.8)$$

$\mathbf{F}_{aero}, \mathbf{M}_{aero}$ は航空力学的な力である抗力とピッチングモーメントであり, 3.3 節で詳しく説明する. \mathbf{G}_p はパラシュートの物体座標系から見た重力である.

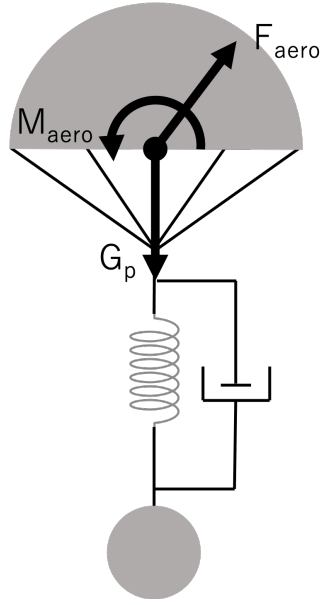


図 3.4: キャンopiesにかかる力

$\alpha_{11}, \alpha_{33}, \alpha_{44}, \alpha_{66}, \alpha_{15}$ は, パラシュートのような大きな物体が流体中を運動するとき, 周りの流体も同時に動くために発生する仮想質量である. 仮想質量は, キャンopiesの半球内に滞留する空気の質量 m_a , パラシュートの圧力中心 z_P , 膨張したキャンopiesの半径 R_p を用いて以下のように表される.

$$m_a = \frac{2}{3}\pi\rho R_p^3\gamma \quad (3.9)$$

$$\alpha_{11} = k_{11}m_a \quad (3.10)$$

$$\alpha_{33} = k_{33}m_a \quad (3.11)$$

$$\alpha_{44} = \frac{1}{5}k_{44}m_a R_p^2(1 + \gamma^2) \quad (3.12)$$

$$\alpha_{66} = \frac{2}{5}k_{66}m_a R_p^2 \quad (3.13)$$

$$\alpha_{15} = k_{15}m_a z_P \quad (3.14)$$

ここで, ρ は流体の密度である. $k_{11}, k_{33}, k_{44}, k_{66}, k_{15}, \gamma$ は実験によって決められた定数係数であり, 本研究では先行研究と同様に $k_{11} = 0.5, k_{33} = 1.0, k_{44} = 0.24, k_{66} = 0, k_{15} = 0.75, \gamma = 0.82$ としている.

式 (3.7) により \mathbf{F}_{canopy} を求め, それを式 (3.5) に代入し, 加速度 $\dot{\mathbf{V}} = (\dot{u}, \dot{v}, \dot{w})^T$ を求める. 加速度 $\dot{\mathbf{V}}$ から, 前進オイラー法を用いて以下のように速度 \mathbf{V} を更新する. 時間を t とすると,

$$\mathbf{V}^{n+1} = \mathbf{V}^n + \dot{\mathbf{V}}\Delta t \quad (3.15)$$

各ステップでの速度 \mathbf{V} から, 前進オイラー法を用いて位置を更新する. 位置を \mathbf{X} とすると,

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \mathbf{V}\Delta t \quad (3.16)$$

また, 角速度 $\boldsymbol{\Omega}$ から, 以下の式に従ってキャンopiesの姿勢を決めるオイラー角 ϕ, θ, ψ を更新

する.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & \sin \phi \\ 0 & \sin \phi (\frac{1}{\cos \theta}) & \cos \phi (\frac{1}{\cos \theta}) \end{bmatrix} \boldsymbol{\Omega} \quad (3.17)$$

3.2.2 ペイロードの支配方程式

本研究ではペイロードを自由落下する質点としてモデル化する. ペイロードの支配方程式を以下に示す.

$$\mathbf{F}_{payload} = m_p \dot{\mathbf{V}}_p \quad (3.18)$$

ここで, m_p はペイロードの質量, \mathbf{V}_p はペイロードの速度である. $\mathbf{F}_{payload}$ はペイロードにかかる力であり, ここでは重力 $m_p \mathbf{G}$ である. ペイロードについてもキャノピーと同様に, 式 (3.18) から式 (3.15) を用いて各ステップでの速度 \mathbf{V}_p を計算し, 式 (3.16) を用いて位置 \mathbf{X}_p を更新する.

3.3 抗力とピッチングモーメント

本節では, キャノピーにかかる力のうち, 航空力学的な力である抗力 \mathbf{F}_{aero} と, ピッチングモーメント M_{aero} について説明する.

3.3.1 抗力

抗力は, 流体中を運動する物体が流れから受ける, 速度と逆方向の力である. 抗力 \mathbf{F}_{aero} は, 抗力係数 $C_D(\alpha_{sp})$, 流体の密度 ρ , キャノピーの投影面積 S_0 , 対気速度 \mathbf{V}_a を用いて以下のよう表される.

$$\mathbf{F}_{aero} = 0.5 C_D(\alpha_{sp}) \rho \|\mathbf{V}_a\| S_0 \left(\frac{\mathbf{V}_a}{\|\mathbf{V}_a\|} \right) \quad (3.19)$$

対気速度 \mathbf{V}_a は, グローバル座標系から物体座標系への変換行列を R , 風ベクトルを \mathbf{W} とすると, パラシュートの速度 \mathbf{V} を用いて以下のように表せる.

$$\mathbf{V}_a = \mathbf{V} + R\mathbf{W} \quad (3.20)$$

抗力係数 $C_D(\alpha_{sp})$ は, 図 3.5 に示すパラシュートの鉛直下方向と速度ベクトル \mathbf{V} がなす角 α_{sp} によって値が決まる. この角度 α_{sp} は迎角と呼ばれる. 迎角 α_{sp} は式 (3.21) に示す対気速度 \mathbf{V}_a の成分を用いて, 式 (3.22) のように計算する.

$$\mathbf{V}_a = \begin{bmatrix} u_a \\ v_a \\ w_a \end{bmatrix} \quad (3.21)$$

$$\alpha_{sp} = \cos^{-1} \left(\frac{w_a}{\sqrt{u_a^2 + v_a^2 + w_a^2}} \right) \quad (3.22)$$

迎角と抗力係数の対応関係は実験的に求めることができ [1], 本研究では図 3.6 に示す実験データを用いて計算している.

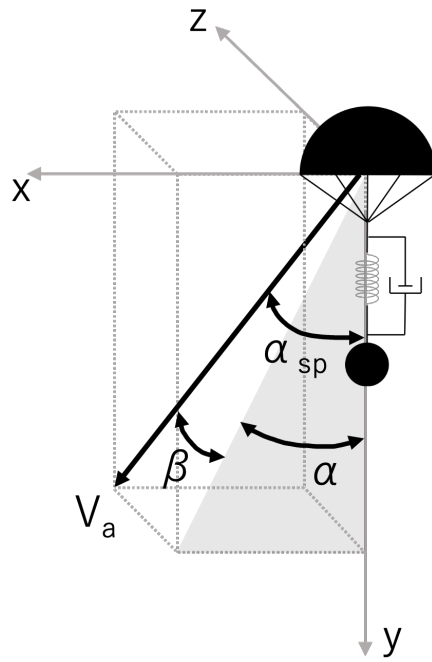


図 3.5: 迎角 α_{sp}

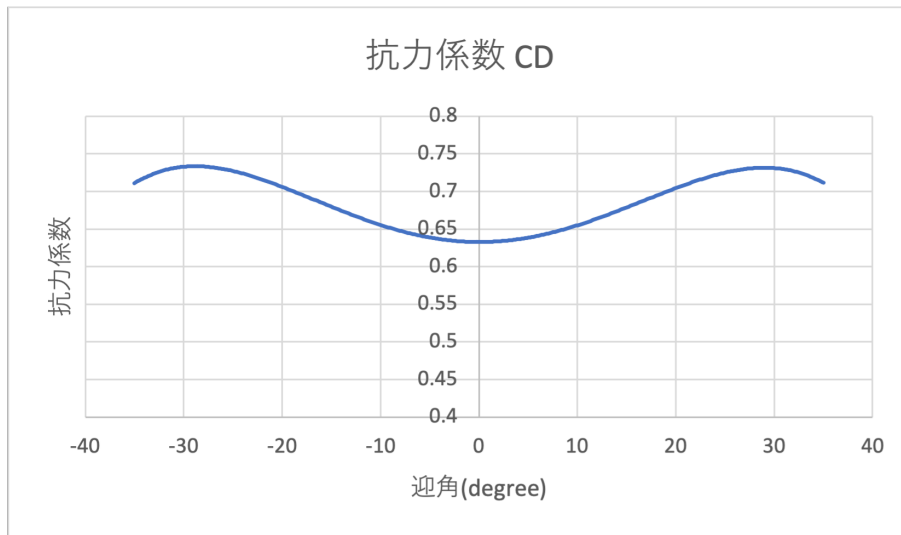


図 3.6: 抗力係数 C_D の実験による計測データ

3.3.2 ピッチングモーメント

ピッチングモーメント M_{aero} は流れから受けるモーメントであり, R_0 をキャノピーの通常時の半径とすると以下の式で表せる.

$$M_{aero} = \rho \|\mathbf{V}_a\| S_0 R_0 \begin{bmatrix} C_m(\beta) \\ C_m(\alpha) \\ C_n \end{bmatrix} \quad (3.23)$$

ここで, ピッチングモーメント M_{aero} の roll 成分, pitch 成分に関わる C_m はピッチングモーメント係数と呼ばれる係数であり, 抗力係数と同様に実験的に求められている値である. 本研究では図 3.7 に示す値を用いている. 図 3.5 に示す迎角の y 軸周りの成分 α , x 軸周りの成分 β は以下のように求められる.

$$\alpha = \tan^{-1}\left(\frac{u_a}{w_a}\right) \quad (3.24)$$

$$\beta = \tan^{-1}\left(\frac{v_a}{\sqrt{u_a^2 + w_a^2}}\right) \quad (3.25)$$

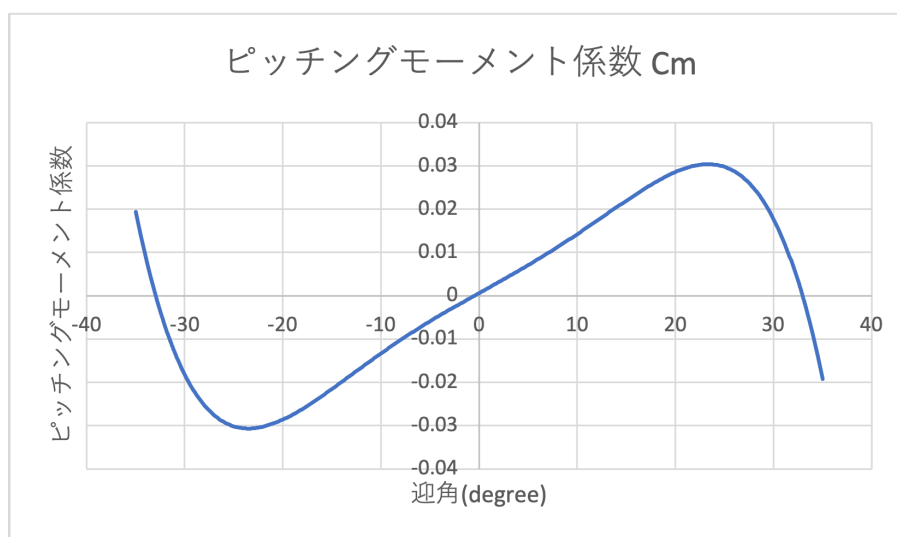


図 3.7: ピッチングモーメント係数 C_m の実験による計測データ

ピッチングモーメント M_{aero} の yaw 成分は, パラシュートが z 軸周りに回転する力を示す. この力が働くことで, パラシュートの軌跡が螺旋を描く様子を再現することができる.

円形パラシュートのキャノピー, ペイロード間は, 複数本のサスペンションライン, ライザーが繋いでいる. この長さが全て等しい時は, $C_n = 0$ であり, ピッチングモーメント M_{aero} の yaw 成分は 0 になる. 従って, この場合はパラシュートは螺旋の軌跡を描かず, 直線的に落下する. 一方で, 複数本のサスペンションライン, ライザーの長さの違いがある場合には, $C_n \neq 0$ となり, その値によってパラシュートの軌跡が変化する. 図 3.8 に $(1.0, 0.0, 0.0)$ の風がある場合の C_n の値と軌跡の変化の様子を示す.

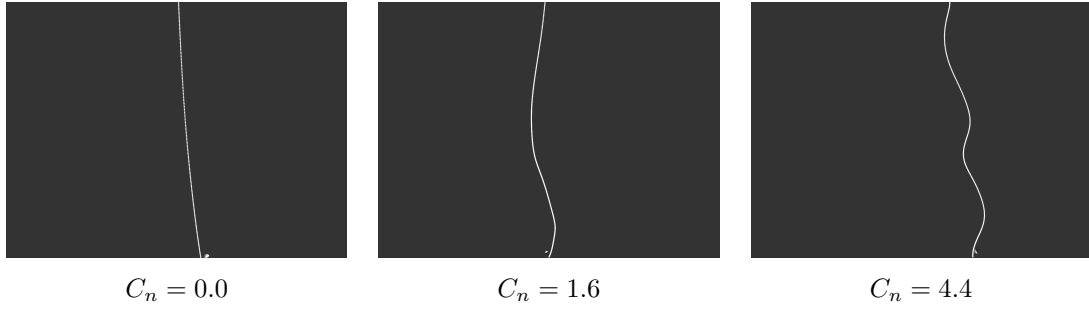


図 3.8: C_n の値による軌跡の変化

3.4 バネ・質点モデル

本研究では、パラシュートのサスペンションラインとライザーを合わせたものを1本の紐として扱い、ペイロードとキャノピーをそれぞれ質点として、紐によるそれぞれの動きをバネ・質点モデルで計算する。バネ・質点モデルでは、質点と質点の間をバネで繋いでいると考える。しかし、バネによる力だけでは質点が振動してしまい、シミュレーションが安定しない。そこで、ダンパによる減衰する力をかけることで、振動を抑えることができる。

シミュレーションでは、まずキャノピー、ペイロードの位置を更新したあとに、バネ、ダンパによる引張力 $\mathbf{F}_{tension}$ を計算する。バネの力を \mathbf{F}_{spring} 、ダンパによる減衰する力を $\mathbf{F}_{damping}$ 、キャノピーの現在の位置を \mathbf{X} 、ペイロードの現在の位置を \mathbf{X}_p とすると、 $\mathbf{F}_{tension}$ は以下の式で表される。

$$\mathbf{F}_{tension} = (\mathbf{F}_{spring} + \mathbf{F}_{damping}) \frac{\mathbf{X}_p - \mathbf{X}}{|\mathbf{X}_p - \mathbf{X}|} \quad (3.26)$$

バネの力 \mathbf{F}_{spring} は、比例定数であるバネ定数を k 、キャノピー、ペイロード間の初期距離を l とすると、フックの法則より

$$\mathbf{F}_{spring} = k(|\mathbf{X}_p - \mathbf{X}| - l) \quad (3.27)$$

と計算される。

ダンパによる減衰力 $\mathbf{F}_{damping}$ は、キャノピーの現在の速度を \mathbf{V} 、ペイロードの現在の速度を \mathbf{V}_p 、減衰係数を k_d とすると

$$\mathbf{F}_{damping} = k_d(\mathbf{V}_p - \mathbf{V}) \quad (3.28)$$

となる。

引張力 $\mathbf{F}_{tension}$ は、キャノピーに $\mathbf{F}_{tension}$ 、ペイロードに $-\mathbf{F}_{tension}$ のように反対向きにかける(図 3.9)。この力から、キャノピー、ペイロードそれぞれの速度を更新し、式 (3.16) を用いて位置を求める。

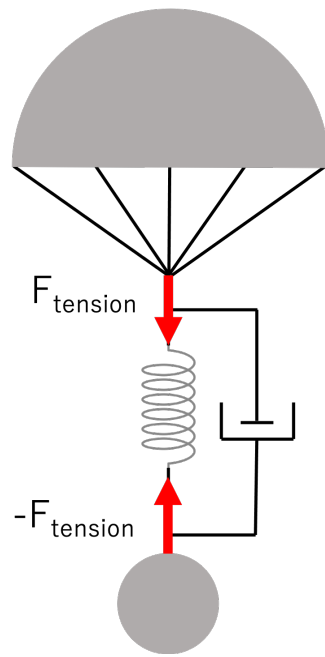


図 3.9: バネ, ダンパによる力

第4章 流体シミュレーション

本章では煙を格子法を用いた流体シミュレーションで再現する手法について述べる。4.1節では流体の動きを表すナビエ・ストークス方程式を、格子法を用いて解く方法について説明する。4.2節では、高速化のための手法である Overlapping grid について説明する。Overlapping grid は格子の取り方を工夫することで高速化する手法であり、粗い格子と細かい格子の2種類で値を交換しながら4.1節で説明する流体シミュレーションをする。最後に、4.3節でシミュレーション結果である煙の密度のボリュームデータを、スライスを用いてボリュームレンダリングする手法について説明する。

4.1 格子法によるシミュレーション

4.1.1 流体の支配方程式

本研究では煙を流体として扱い、流体の支配方程式としてナビエ・ストークス方程式を用いる。ここで、単純化のため空気の流速は音速よりも十分に小さいものとし、非圧縮性流体として扱う。非圧縮の粘性流体に関するナビエ・ストークス方程式を以下に示す。

$$\nabla \cdot \mathbf{u} = 0 \quad (4.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F}_{ext} \quad (4.2)$$

ここで、 \mathbf{u} は流体速度、 ν は動粘性係数、 ρ は流体の密度、 p は圧力、 \mathbf{F}_{ext} は外力である。式(4.1)は非圧縮性流体の質量保存を表す。式(4.2)は運動量保存式であり、右辺第一項は移流項、第二項は圧力項、第三項は粘性拡散項、第四項は外力項である。移流項は、流体が流れに従って移動することを示す。圧力項は、流体が圧力の高いところから低いところに向かって流れることで、圧力を均一にしようとする性質を示す。粘性拡散項は流体の粘性による速度の拡散を示す。外力項では重力や浮力を考慮している。

また、温度に関する支配方程式を以下に示す。

$$\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T = a \nabla^2 T + E_r \quad (4.3)$$

ここで、 T は温度、 E_r は放射熱エネルギー、 $a = \frac{\lambda}{c_p \rho}$ は熱伝導率 λ 、熱容量 c_p と密度 ρ から計算される熱拡散率である。左辺第二項は流体流れによる熱の移流、右辺第一項は熱拡散、第二項は熱放射によるエネルギー損失および外部に放射熱源があったときに受けるエネルギーを表す。

4.1.2 ナビエ・ストークス方程式の数値解法

ナビエ・ストークス方程式をコンピュータで解くためには、物理量や式を離散化する必要がある。本研究では流体の境界が不明確である煙を対象としているため、物理量の離散化手

法として、シミュレーション空間を格子状に分割して、各格子に計算点を固定する格子法を用いる。物理量を各格子に割り当てるとき、速度、密度、圧力、温度を全て格子の中央に定義する方法が簡単である。しかしこの方法では、中心差分で離散化する際に自身の格子の値は使わず、左右の値のみを使うことから、縞状のアーティファクトが発生する。そこで、本研究ではスカラー量である密度、圧力、温度は格子中心に定義し、速度ベクトルは格子の境界上に定義するスタガード格子を用いる。図 4.1 に 2 次元の場合のスタガード格子を示す。このように物理量をずらして配置することで、空間差分において隣合う格子の値がやりとりされるため、アーティファクトを防ぐことができる。

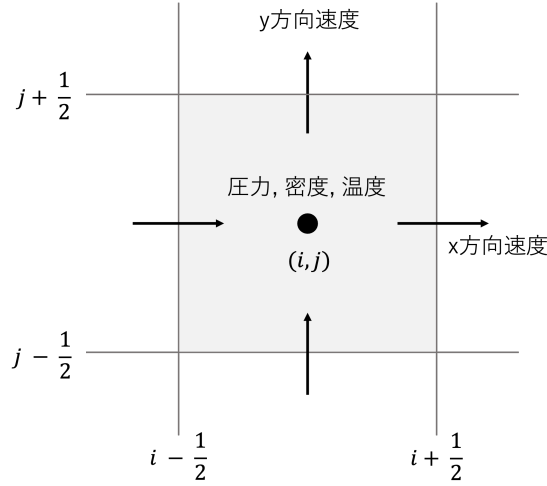


図 4.1: スタガード格子

格子法では、各格子に離散化した物理量を用いて、式 (4.2) を差分化して解く。ある時間 t の速度 \mathbf{u}^n から次の時間 $t + \Delta t$ の速度 \mathbf{u}^{n+1} を求めるため、以下のように式 (4.2) を前進オイラー法で差分化する。

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{F}_{ext} \quad (4.4)$$

右辺を一度に求めようとする解が不安定になるため、各項を段階的に解いていく。以下で外力項、移流項、粘性拡散項、圧力項の順に説明する。

1. 外力項

まず以下の式によって外力項を解き、現在の速度 \mathbf{u}^n から中間速度場 \mathbf{u}^* を求める。

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t \mathbf{F}_{ext}^n \quad (4.5)$$

外力 \mathbf{F}_{ext} は重力と浮力であり、以下の式から計算できる。

$$\mathbf{F}_{ext} = -g\mathbf{y} + \beta_{buoy}(T - T_{ambient})\mathbf{y} \quad (4.6)$$

ここで、 g は重力加速度、 β_{buoy} は浮力係数、 \mathbf{y} は鉛直上方向の単位ベクトル、 T は流体の温度場、 $T_{ambient}$ は周囲の環境温度である。

また、数値拡散によって滑らかになってしまった流れに乱流を追加する方法として、Vorticity confinement[13]を用いる。Vorticity confinement はローカルグリッドのみに

適用する。渦を追加するために、まず以下のように渦度 ω を求める。

$$\omega = \nabla \times \mathbf{u} \quad (4.7)$$

次に、渦を強調するための力 \mathbf{F}_{vort} を以下のように計算する。

$$\mathbf{F}_{vort} = \epsilon h (\mathbf{N} \times \omega) \quad (4.8)$$

ここで、 h はグリッド幅、 ϵ は渦の大きさを調整するためのパラメータである。 \mathbf{N} は以下の式で計算される。

$$\mathbf{N} = \frac{\eta}{|\eta|} \quad (4.9)$$

$$\eta = \nabla |\omega| \quad (4.10)$$

ローカルグリッドの計算の時のみ外力 \mathbf{F}_{ext} に \mathbf{F}_{vort} を追加する。

2. 移流項

移流項 $(\mathbf{u} \cdot \nabla) \mathbf{u}$ の差分化には、セミ・ラグランジュ法を用いる。セミ・ラグランジュ法は、ある点における物理量は、前のステップで別のところにあった物理量が移動してきたものであるとする考え方で移流項を解く方法である。つまり、計算点の位置を \mathbf{x} 、速度を \mathbf{u}^* とすると、 Δt だけバックトレースした位置での物理量 $\phi(\mathbf{x} - \mathbf{u}^* \Delta t, t)$ が次のステップの物理量 $\phi(\mathbf{x}, t + \Delta t)$ となる (図 4.2)。ここで、バックトレースした位置での物理量が必要となるが、この位置が物理量を離散化した格子点と重なるとは限らない。従って、バックトレースした位置での物理量を周囲の値から補間する必要がある。本研究では補間の手法として線形補間を用いているが、線形補間は 1 次精度であるため、物理量の拡散が発生する。より高精度の補間方法を用いることで、この数値拡散を抑えることができる。セミ・ラグランジュ法を用いてナビエ・ストークス方程式の移流項を計算し、中間速度場 \mathbf{u}^{**} を求める式を以下に示す。

$$\mathbf{u}^{**} = \mathbf{u}^*(\mathbf{x} - \mathbf{u}^* \Delta t) \quad (4.11)$$

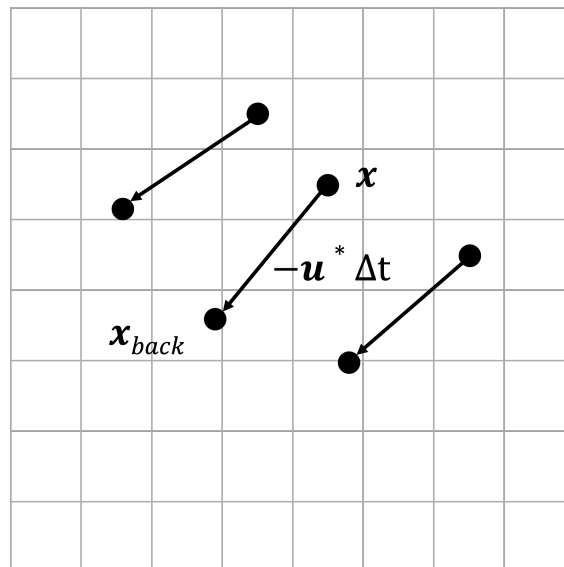


図 4.2: セミ・ラグランジュ法によるバックトレース

3. 粘性拡散項

粘性拡散項 $\nu \nabla^2 \mathbf{u}$ は、安定性のため陰解法によって中間速度場 \mathbf{u}^{**} を用いて以下のよう
に解く.

$$\mathbf{u}^{***} = \mathbf{u}^{**} + \nu \Delta t \nabla^2 \mathbf{u}^{***} \quad (4.12)$$

$\nabla^2 \mathbf{u}$ を離散化し、式を整理すると、 \mathbf{u}^{***} についての線形システムが作られる. 本研究ではこれをガウス・ザイデル法を用いて反復計算をすることで解いている. $\nabla^2 \mathbf{u}$ の離散化には中心差分を用いる.

4. 圧力項

最後に圧力項を計算することで、非圧縮性を保つように速度場を修正する. 非圧縮性を満たす圧力 p を算出するために以下の圧力のポアソン方程式を用いる.

$$\nabla^2 p = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}^{***} \quad (4.13)$$

式 (4.13) は圧力項と式 (4.1) の質量保存式から導出される. 式 (4.13) の $\nabla^2 p$ を中心差分で離散化し、得られた線形連立方程式を解くことで圧力 p を求める. 本研究では圧力のポアソン方程式から得られる線形システムの解法として、不完全コレスキー分解付共役勾配法 (ICCG 法) を用いている.

得られた圧力から、以下の式によって最終的な速度場 \mathbf{u}^{n+1} を計算する.

$$\mathbf{u}^{n+1} = \mathbf{u}^{***} - \frac{\Delta t}{\rho} \nabla p \quad (4.14)$$

以上のようにして、ナビエ・ストークス方程式を解いて次の時間 $t + \Delta t$ の速度場 \mathbf{u}^{n+1} を求める. 煙のシミュレーションでは、この速度場を使って煙のすすの密度 ρ_s を移流・拡散させることで、煙の挙動を表現する. ρ_s に関する移流・拡散の方程式を以下に示す.

$$\frac{\partial \rho_s}{\partial t} + (\mathbf{u}^{n+1} \cdot \nabla) \rho_s = \nu_s \nabla^2 \rho_s + S \quad (4.15)$$

ここで、 ν_s は煙の拡散係数である. S は煙の発生源であり、本研究ではパラシュートのペイロード位置を用いる. 式 (4.15) はナビエ・ストークス方程式の移流項、拡散項と同様に、セミ・ラグランジュ法、ガウス・ザイデル法によって解く.

4.2 Overlapping grid による高速化

本研究で扱う煙はパラシュートの動きに沿って長い螺旋の軌跡を描くため、詳細なシミュレーション結果を得るためにはシミュレーション空間全体を高解像度に格子分割して計算する必要があり、膨大な計算時間がかかってしまう. そこで、物体の周囲にだけ定義するグリッド幅が小さいローカルグリッドと計算空間全体で定義するグリッド幅が大きいグローバルグリッドの二種類に分けて流体シミュレーションを行う Overlapping grid という手法 [5] を用いる. 図 4.3 に示すように、ローカルグリッドはペイロードを中心としてパラシュートの動きに合わせて移動する格子、グローバルグリッドは計算空間全体を粗に分割した格子である. ローカルグリッドとグローバルグリッドのそれぞれで 4.1 節で説明した格子法の流体シミュレーションをしながら、速度、圧力などの値のやりとりをする.

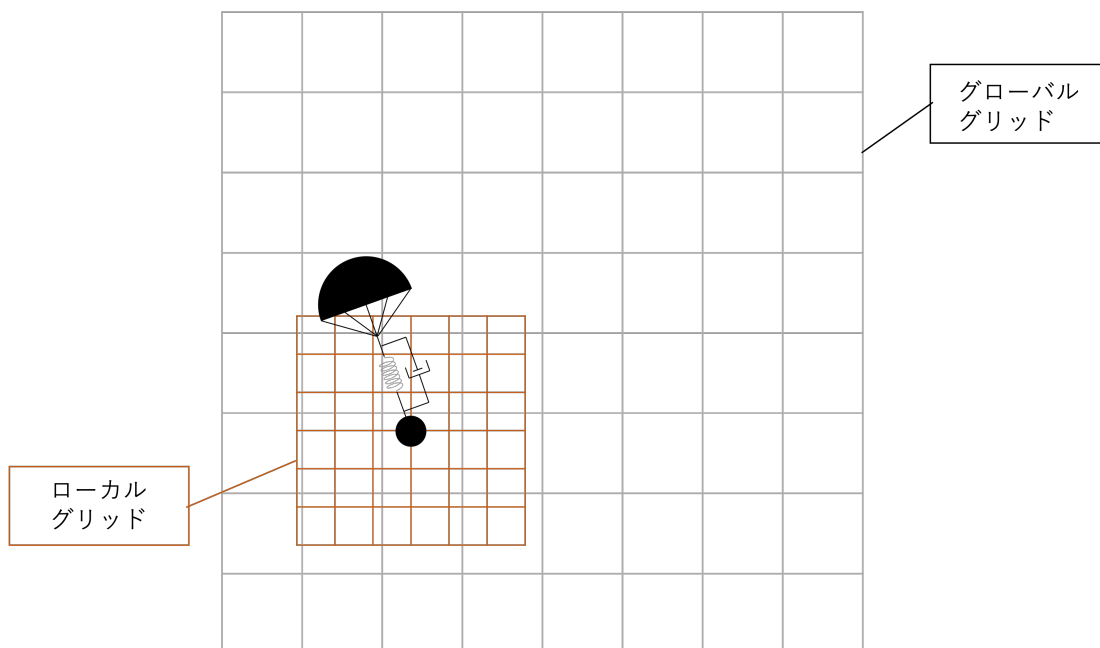


図 4.3: ローカルグリッドとグローバルグリッド

値のやりとりは、ローカルグリッドの境界部分の値をやりとりする境界伝達と、内部の値をやりとりする内部伝達に分けて行う。値のやりとりを含めた全体の計算手順を図 4.4 に示す。図 4.4 では、Overlapping grid の処理にあたる部分を水色で表示している。シミュレーションの順序としては、まず速度の境界伝達を行い、それぞれの格子で圧力以外の項から速度を更新し、中間速度場 u^{***} の内部伝達を行う。次に、圧力の境界伝達を行い、それぞれのグリッドで圧力を計算し、内部伝達を行う。圧力の計算は 4.1.2 節で説明したように反復計算する必要があるため、反復の中で毎回境界伝達、内部伝達をする。圧力が収束したら、圧力項を計算し、速度を更新する。そして最後に高解像度の速度場を生成して、煙密度を移流する。

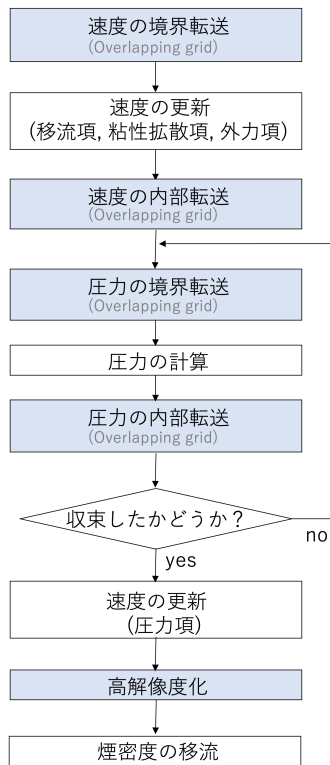


図 4.4: 煙シミュレーションの流れ

4.2.1 境界伝達

境界部分のやりとりを示す境界伝達では、ローカルグリッドの境界部分のグリッドセルに関して、ローカルグリッドの値をグローバルグリッドの値で上書きする (図 4.5). ローカルグリッドはパラシュートと共に動いているため、ローカルグリッドのある点の速度 \mathbf{u}_l に対応するグローバルグリッドの速度 \mathbf{u}_g は、ローカルグリッドの中央位置 \mathbf{c} を用いて以下のように表せる.

$$\mathbf{u}_l = \mathbf{u}_g - \frac{d\mathbf{c}}{dt} \quad (4.16)$$

\mathbf{u}_g はグローバルグリッドの計算点位置と一致しないため、線形補間によって周囲のグリッドから値を補間する. 圧力についても同様に線形補間により値を求める.

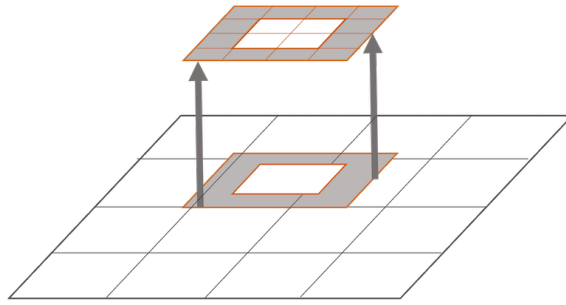


図 4.5: 境界伝達

4.2.2 内部伝達

格子内部のやりとりを示す内部伝達では、ローカルグリッド内部のセルに関して、内部と重なっているグローバルグリッドセルの値をローカルグリッドの値で上書きする (図 4.6). 境界伝達と同様に、ローカルグリッドが移動していることを考慮して、以下のように速度を求める.

$$\mathbf{u}_g = \mathbf{u}_l + \frac{d\mathbf{c}}{dt} \quad (4.17)$$

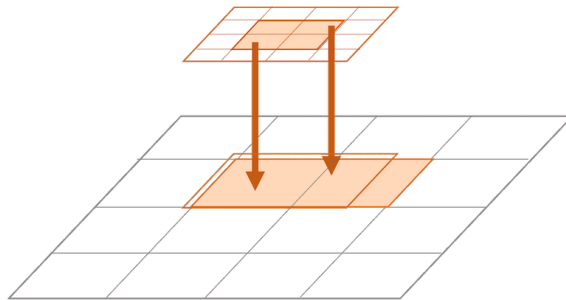


図 4.6: 内部伝達

4.2.3 高解像度化

最終的に得られたグローバルグリッドの速度で煙の密度を移流させると、解像度が低いいため滑らかな見目の煙になってしまう。そこで、グローバルグリッドの速度場を線形補間により高解像度化した最終グリッドを最終速度場として用いる。最終グリッド生成の模式図を図 4.7 に示す。最終グリッドにはローカルグリッドと同じセル幅を用い、全体の大きさはグローバルグリッドと一致させる。グローバルグリッドの速度場を線形補間した後、ローカルグリッドと重なっている所は、ローカルグリッドの値で上書きする。この最終速度場を用いて煙の密度を移流し、4.3 節に示す方法でレンダリングする。最終速度場は式 (4.15) による煙の移流・拡散のみに用い、計算時間のかかる圧力項の計算には使わないため、高解像度化しても全体の計算時間にはほとんど影響しない。

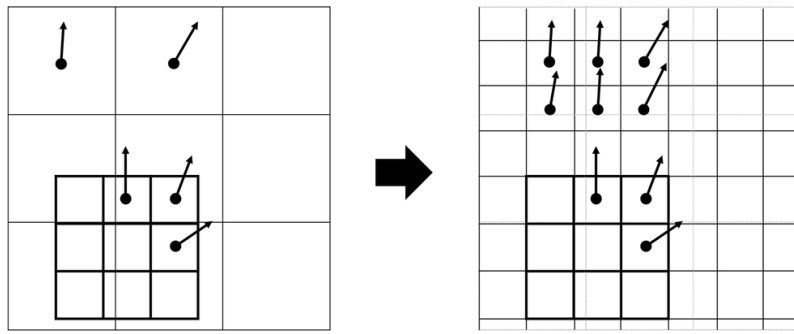


図 4.7: 最終グリッドの生成

4.3 レンダリング

本研究では 4.1 節, 4.2 節で行ったシミュレーションから得られた, 最終グリッドで移流させた煙の密度をボリュームデータとして, ボリュームレンダリングを行う. 各セルの煙の密度を 3 次元のボリュームデータとして 3D テクスチャに格納し, 視点に垂直な一定間隔のスライスを用意する (図 4.8). 各スライスは 3D テクスチャを貼った半透明なポリゴンとして描画し, アルファブレンディングを用いてスライスを遠い方から描画することで, 3 次元の煙がレンダリングできる. この方法では高速にレンダリングすることが可能だが, 煙内部の多重散乱や, 照明の当たり方を考慮できない点が問題である. ボリュームレイキャスティング法やフォトンマッピング法を用いることで, 高速性は劣るが, これらの効果を考慮することができる.

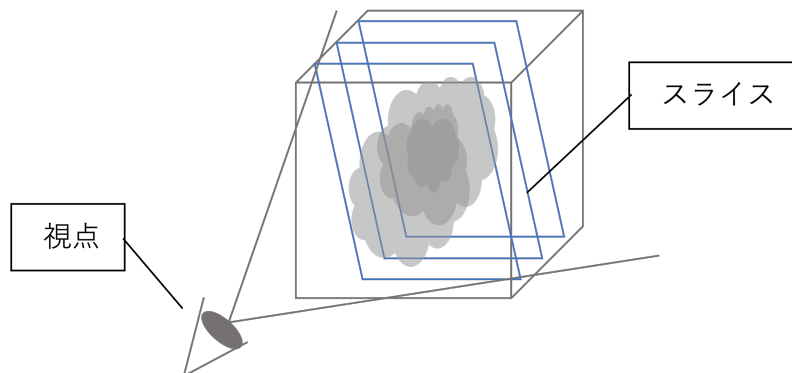


図 4.8: スライスを用いたボリュームレンダリング

第5章 結果

本章では, 提案手法によって昼花火のシミュレーションを行なった結果を示す. 表 5.1 に開発・実行環境を示す.

表 5.1: 開発・実行環境

CPU	Intel Core i7-3630QM 2.4GHz
メインメモリ	16GB
OS	Microsoft Windows 10
開発言語	C++
グラフィックス API	OpenGL

5.1 煙のパラメータ調整による変化

本節では, 煙をシミュレーションする際のパラメータ変化による見た目の違いを示す. $C_n = 4.4$, ローカルグリッド全体の大きさは $1 \times 1 \times 1[m]$, グリッド数は $20 \times 20 \times 20$, グローバルグリッド全体の大きさは $20 \times 20 \times 20[m]$, グリッド数は $40 \times 40 \times 40$, 計算時間は 1 ステップあたり約 140 秒であった. パラメータによる比較の基準として, 渦度 $\epsilon = 20.0$, 煙の拡散係数 $\nu_s = 0.0005$ のシミュレーション結果を図 5.1 に示す. 図 5.1 では上空からパラシュートを落下させた時のカメラの視野に入ったフレームのみを示している (そのため 325 ステップから始まっている).

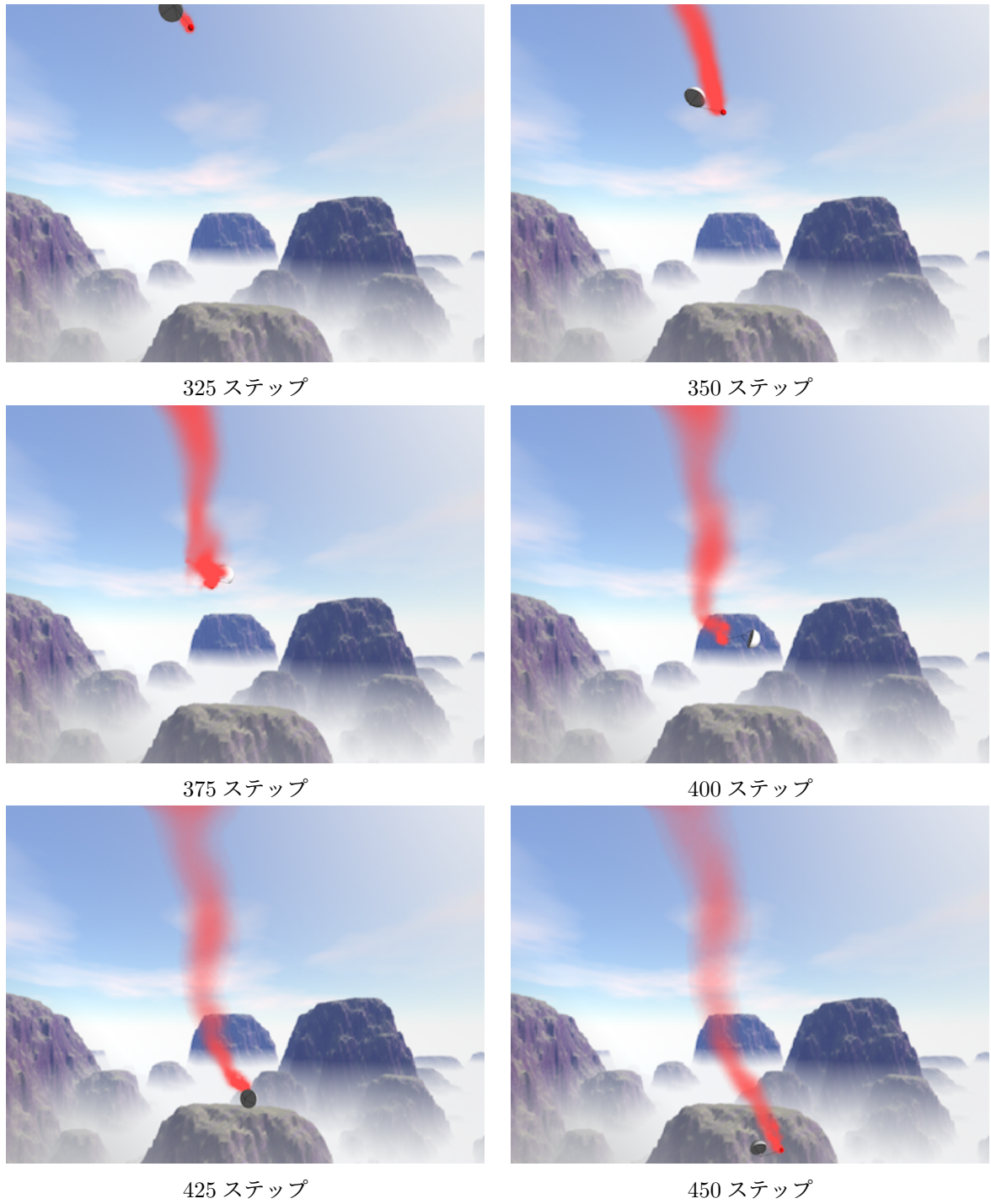


図 5.1: 渦度 $\epsilon = 20.0$, 煙の拡散係数 $\nu_s = 0.0005$ のシミュレーション結果

図 5.1 に示すシミュレーションから拡散係数は変更せず, 渦度 $\epsilon = 0.0$ として, 煙の乱れを少なくした結果を図 5.2 に示す.

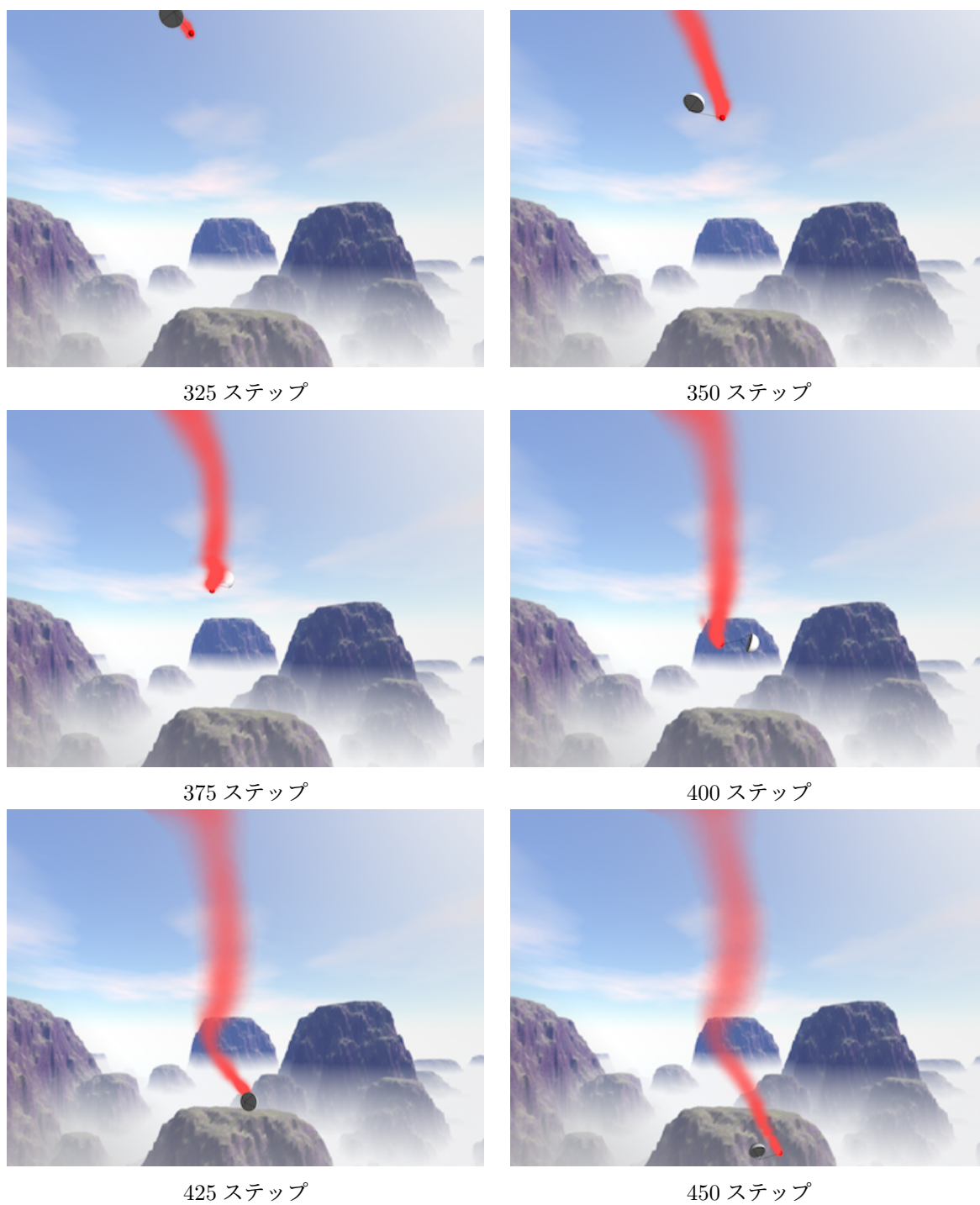


図 5.2: 渦度 $\epsilon = 0.0$, 煙の拡散係数 $\nu_s = 0.0005$ のシミュレーション結果

図 5.1 に示すシミュレーションから渦度は変更せず, 拡散係数 $\nu_s = 0.0$ として, 煙の拡散が起きないようにした場合の結果を図 5.3 に示す.

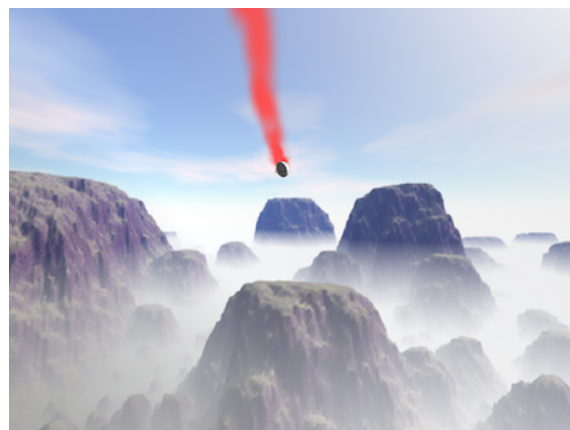
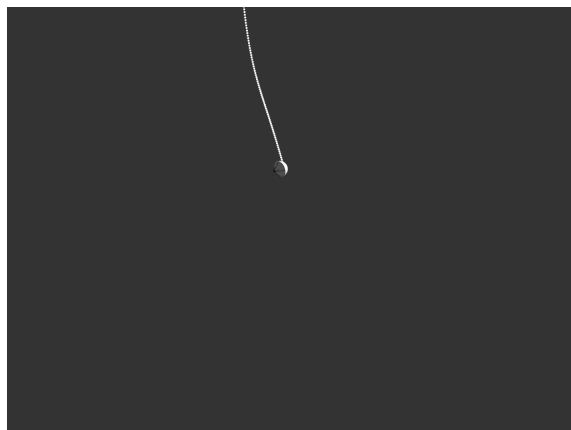


図 5.3: 渦度 $\epsilon = 20.0$, 煙の拡散係数 $\nu_s = 0.0$ のシミュレーション結果

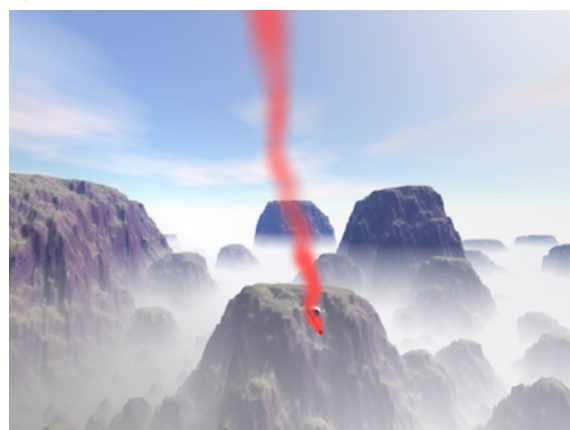
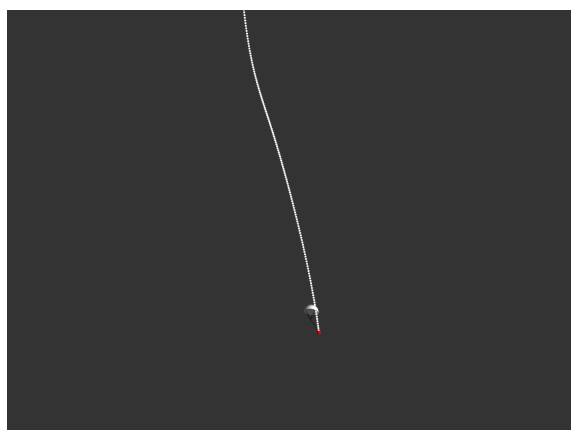
5.2 C_n の値による軌跡の変化

本節では, キャノピーにかかる航空力学的な力に関わる C_n の値を変更した場合のシミュレーション結果を示す. 図 5.4 に $C_n = 2.7$ の場合の結果を示す. 左側はパラシュートのみを

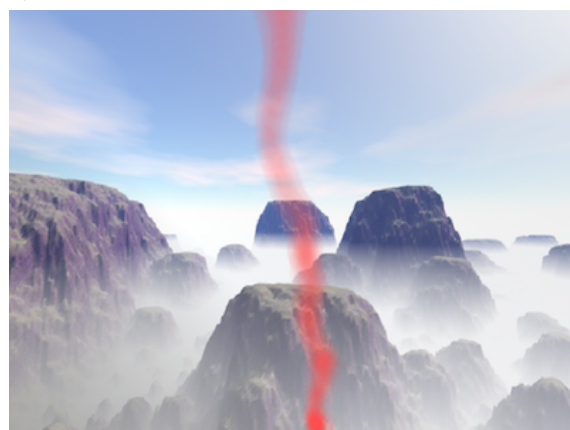
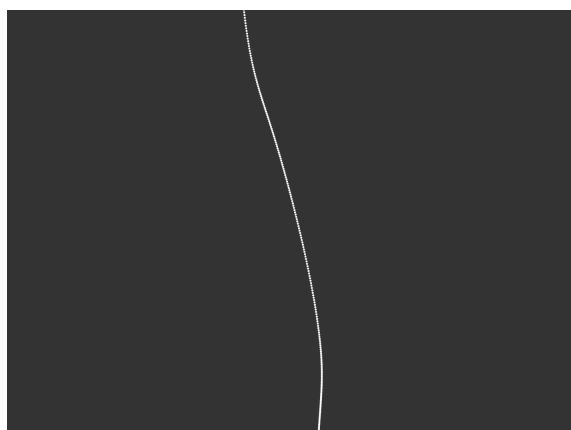
シミュレーションし、軌跡を白線で示した図、右側は軌跡に沿って煙のシミュレーションを行なった結果である。ローカルグリッド、グローバルグリッドの設定は5.1節と同様であり、計算時間は1ステップあたり約140秒であった。また、パラシュートのみのシミュレーションの計算速度は約60fpsであった。



300 ステップ



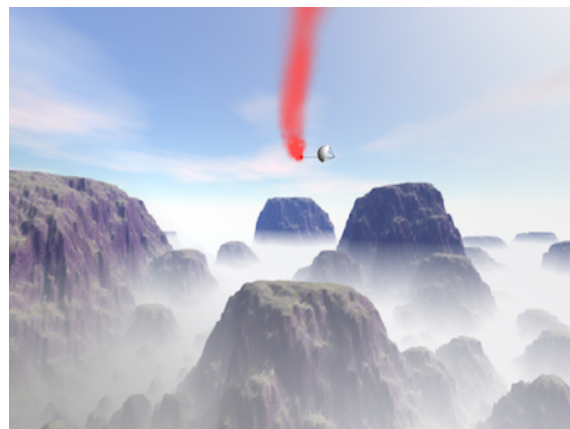
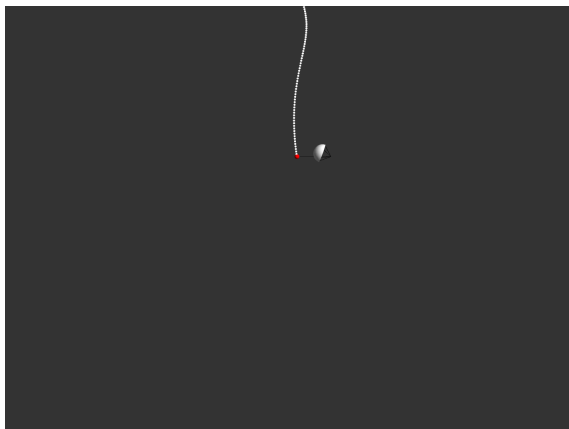
370 ステップ



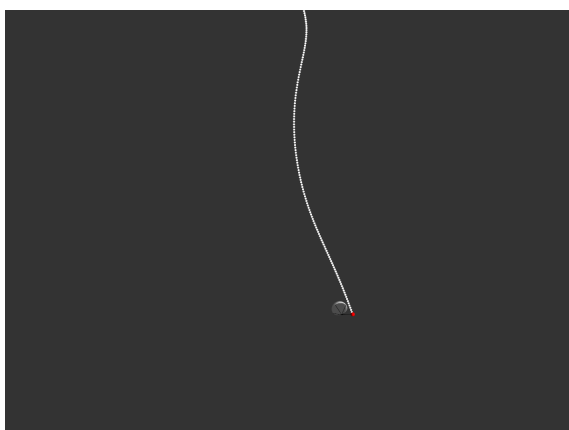
440 ステップ

図 5.4: $C_n = 2.7$ のシミュレーション結果

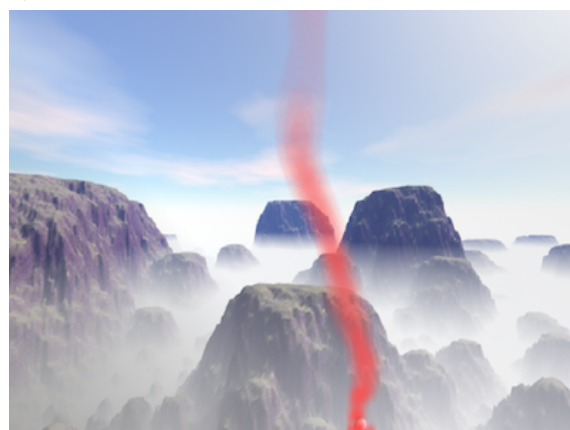
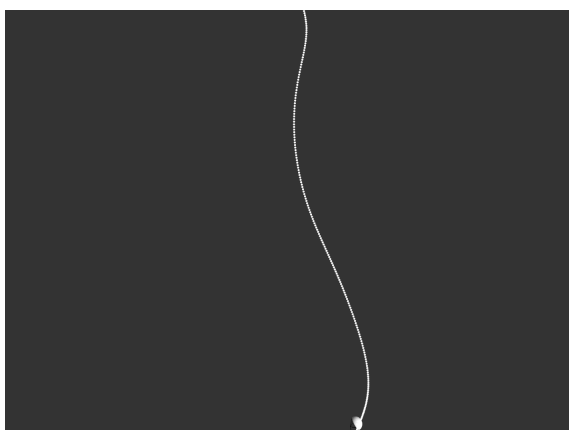
$C_n = 4.2$ の場合の結果を図 5.5 に示す.



300 ステップ



370 ステップ



440 ステップ

図 5.5: $C_n = 4.2$ のシミュレーション結果

5.3 Overlapping grid による高速な高解像度化

本節では, Overlapping grid を用いた場合と, 全て高解像度でシミュレーションした場合の比較を行うための結果を示す. 図 5.6 にグリッド全体の大きさ $10 \times 10 \times 10[m]$, グリッド数 $200 \times 200 \times 200$ で, 全て高解像度でシミュレーションした結果を示す. 計算時間は 1 ステップあたり約 260 秒であった.

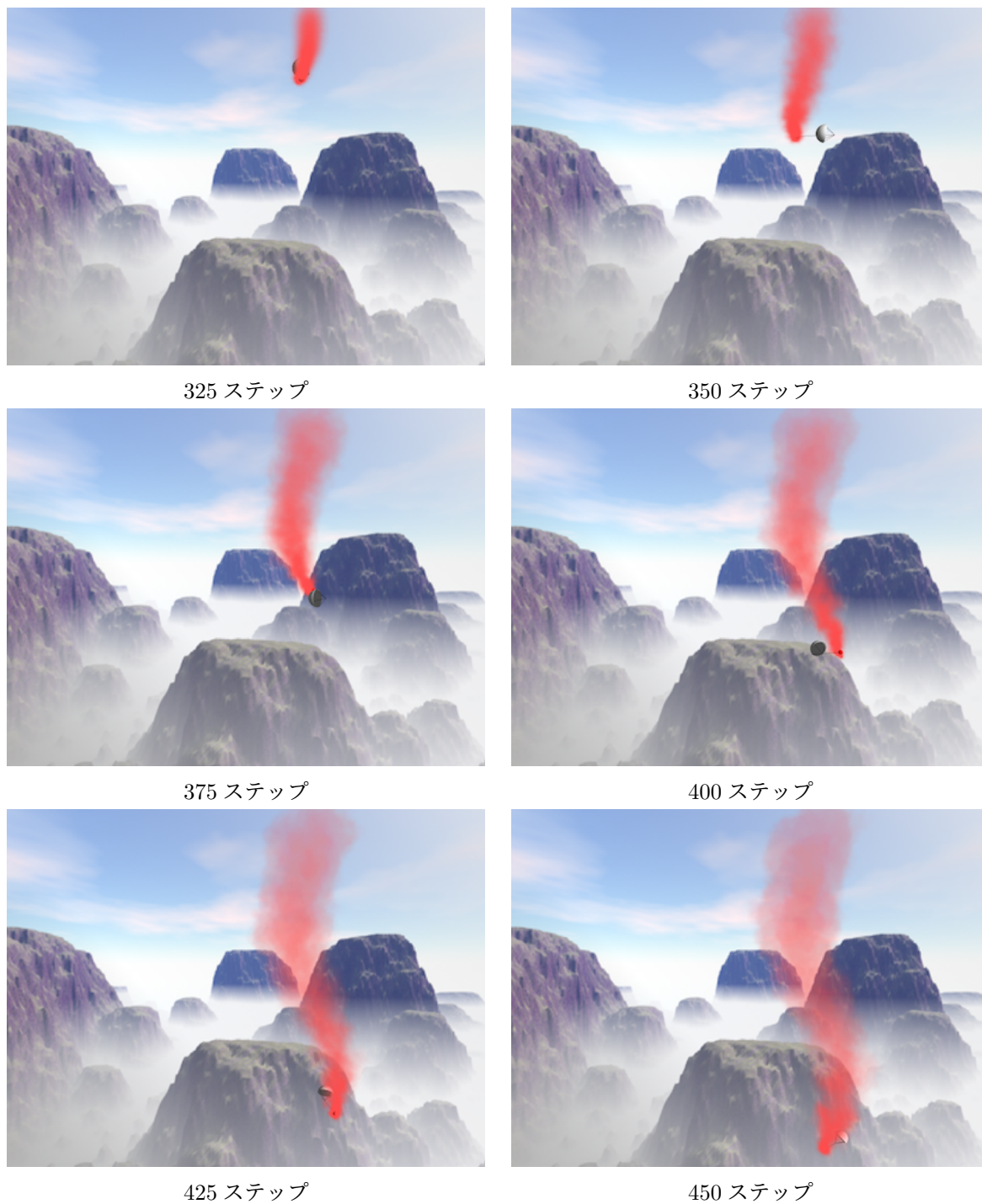
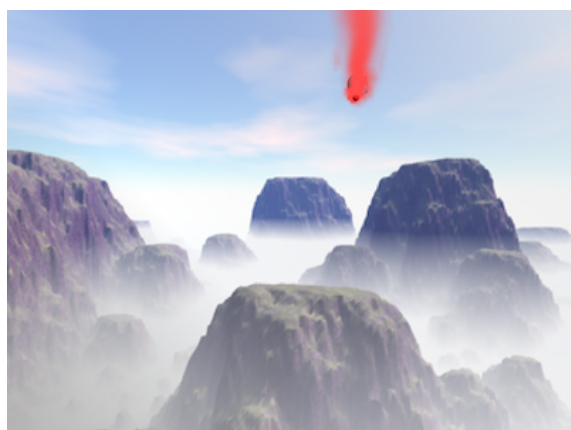
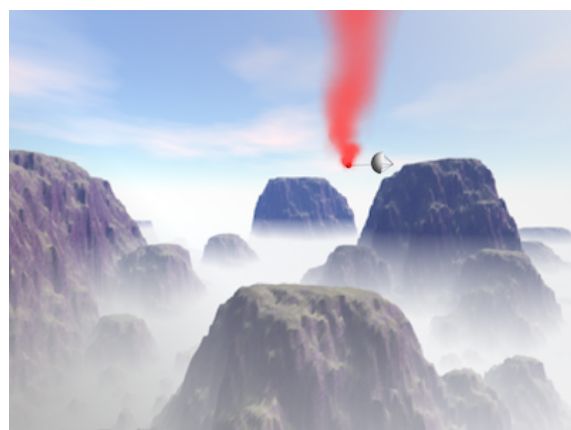


図 5.6: 高解像度でシミュレーションした結果

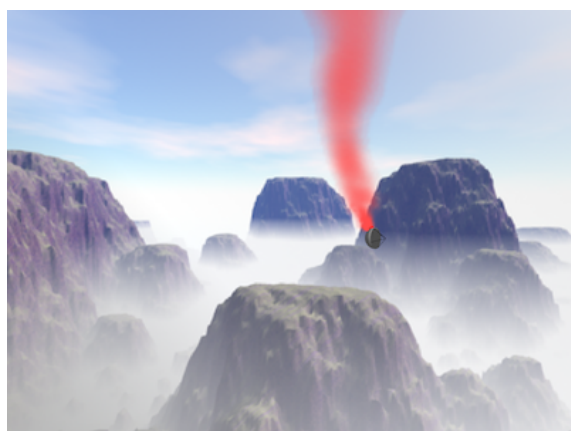
Overlapping grid を用いて, 高速に高解像度化した場合のシミュレーション結果を図 5.7 に示す. ローカルグリッド全体の大きさは $1 \times 1 \times 1[m]$, グリッド数は $20 \times 20 \times 20$, グローバルグリッド全体の大きさは $14 \times 14 \times 14[m]$, グリッド数は $28 \times 28 \times 28$ である. 境界転送, 内部転送でグリッドの境界付近では転送を行わないで良いようにするために, グローバルグリッドのシミュレーション範囲を全て高解像度の場合と比べて大きくとっている. 計算時間は 1 ステップあたり約 42 秒であった.



325 ステップ



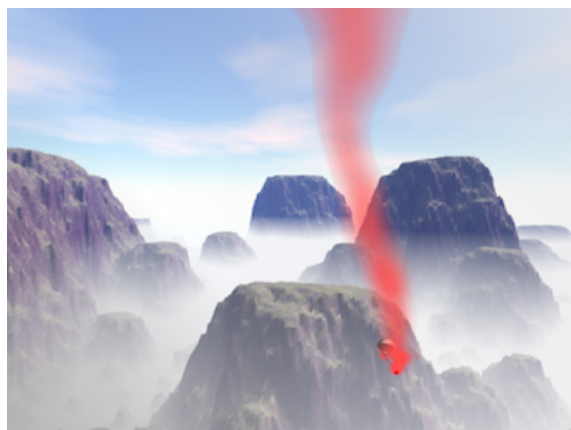
350 ステップ



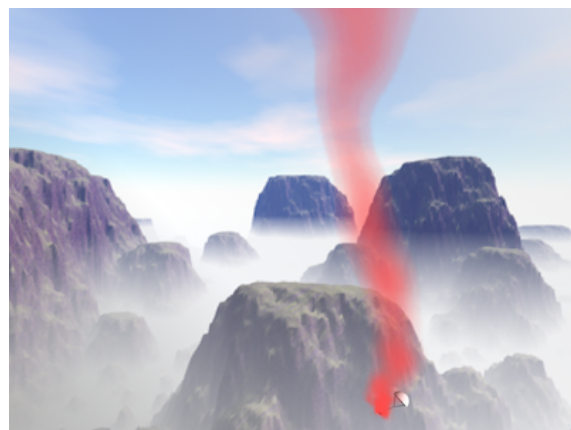
375 ステップ



400 ステップ



425 ステップ



450 ステップ

図 5.7: Overlapping grid を用いた結果

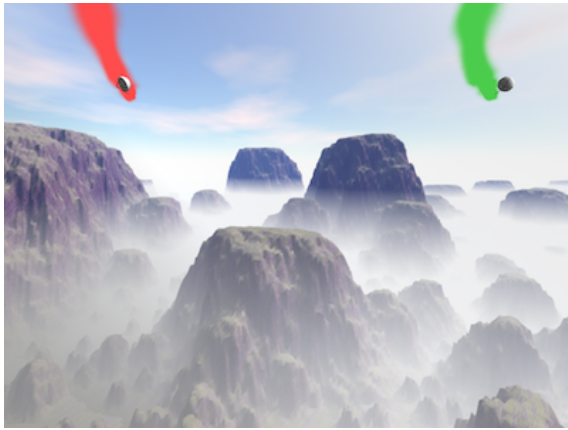
5.4 複数の煙竜があるシーン

実際の花火大会では、図 5.8 のように複数の煙竜を同時にあげるシーンが多く見られる。

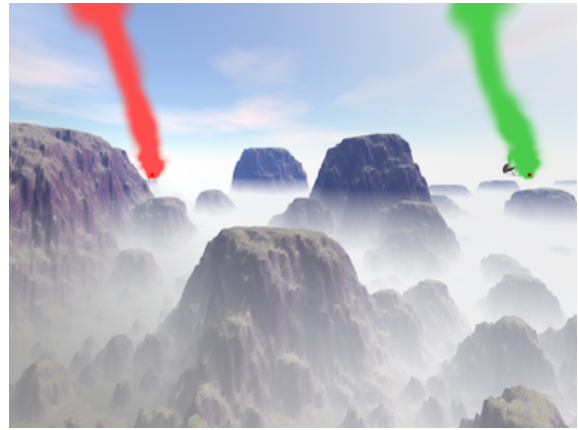


図 5.8: 複数の煙竜 (画像提供:秋田県大仙市)

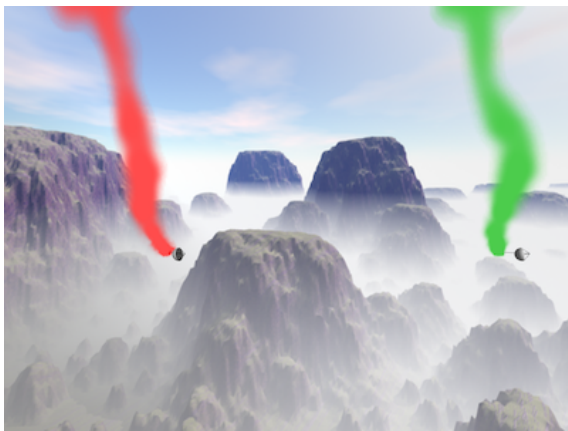
このようなシーンの再現のため、レンダリング時に場所によって色を変えることで、色の違う煙竜が2つある場合のシミュレーションを行なった結果を図 5.9 に示す.



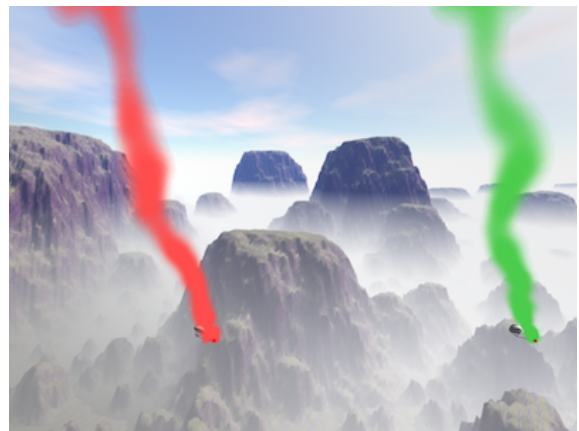
200 ステップ



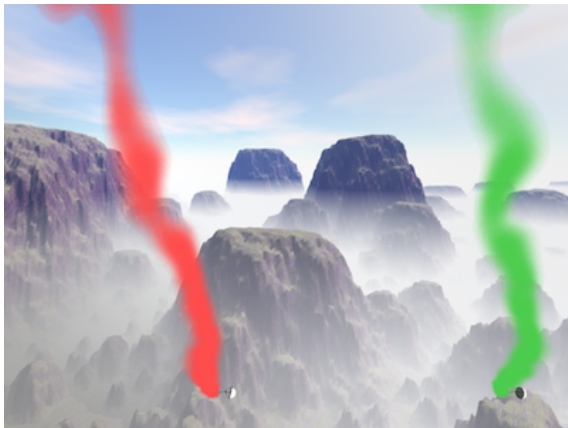
250 ステップ



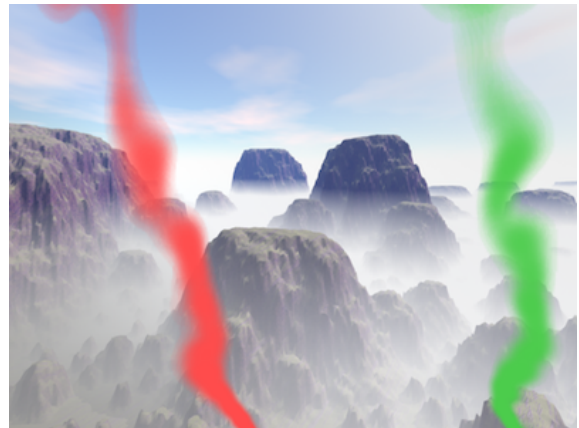
300 ステップ



350 ステップ



400 ステップ



450 ステップ

図 5.9: 2つの煙竜があるシーン

第6章 考察

6.1 煙のパラメータ調整による変化

渦度 ϵ に関して $\epsilon = 20.0$ の場合(図5.1)と $\epsilon = 0.0$ の場合(図5.2)を比較すると、パラシュートのペイロード付近で $\epsilon = 20.0$ の場合の方が乱流による煙の乱れが表れていることが確認できる。図6.1に400ステップのペイロード付近を拡大した様子を示す。Vorticity confinementはローカルグリッドのみに適用しているため、ペイロードから離れた部分に関してはどちらも滑らかな見た目になっているが、 $\epsilon = 0.0$ の場合では煙の太さが一定の部分が多く、精細度が低く見える。

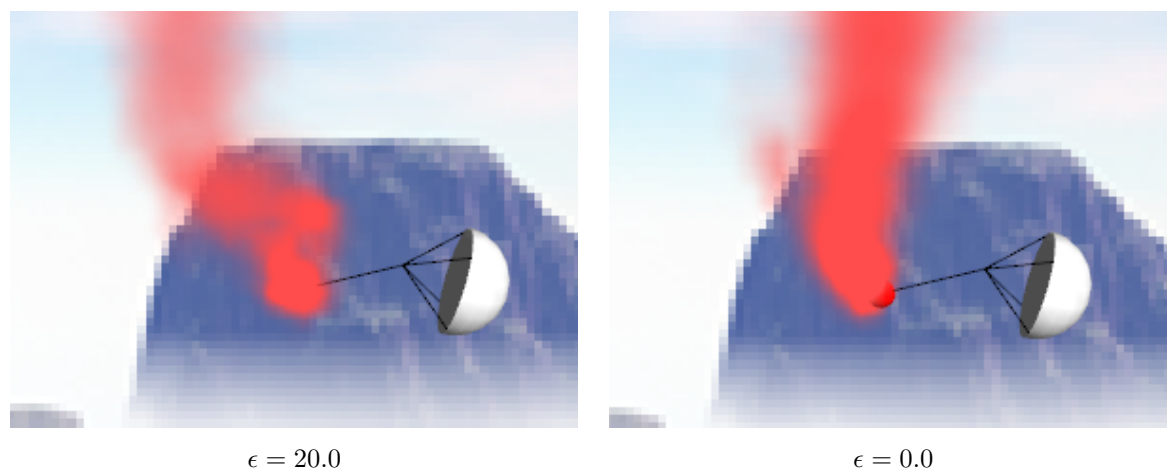


図 6.1: ペイロード付近拡大図 (400 ステップ)

煙の拡散係数 ν_s に関して $\nu_s = 0.0005$ の場合(図5.1)と $\nu_s = 0.0$ の場合(図5.3)を比較すると、 $\nu_s = 0.0$ の場合は煙が全体的に細くなっていることが確認できる。煙竜では特に軌跡の上方で煙が拡散している様子が見られるため、適切な拡散係数を設定し、煙の拡散を考慮することも必要である。また、4.1節で述べたように移流項を計算する際に数値拡散によって煙が消失してしまう場合がある。移流項の計算方法を検討することで、不自然に煙が細くなったり消えてしまったりする現象を抑えられると考える。

6.2 C_n の値による軌跡の変化

パラシュートの軌跡は航空力学的な力の働き方によって変化する。3.3節で述べたように、ピッチングモーメント M_{aero} のyaw成分である C_n の値について、 $C_n = 0$ の時は風の影響のみを受けてパラシュートは直線的に落下し、 $C_n \neq 0$ の時は値が小さいと緩やかな螺旋、値を大きくするに従って細かな螺旋を描いて落下する。図5.4($C_n = 2.7$)と図5.5($C_n = 4.2$)を比

較すると, C_n の値が大きい方が細かな螺旋状の軌跡を描いていることが確認できる. 本研究では C_n をユーザが指定するパラメータとしており, このパラメータを変化させることで様々な軌跡を描くシミュレーションを行うことができる. パラシュートは約 60fps でシミュレーションできていることから, 所望の軌跡がある場合には, パラメータを変更しながらパラシュートのみのシミュレーションを繰り返すことで, 高速にパラメータを決定することができる. また, 花火大会の試し打ちとして利用したい場合には, パラシュートの形状と軌跡の関係を正確にシミュレーションする必要がある. そのため, 複数本のサスペンションライン, ライザーの長さの違いと C_n の値の関係を調査し, サスペンションライン, ライザーの長さの違いから C_n を求めてシミュレーションすることが今後の課題として挙げられる.

6.3 Overlapping grid による高速な高解像度化

Overlapping grid を用いた場合では, 全て高解像度でシミュレーションした場合と比べて計算速度が 6.2 倍高速 (260 秒/ステップから 42 秒/ステップ) になっている. 見た目に関しては, 特にパイロードから離れた部分で, 高解像度でシミュレーションした場合の方が細かな煙の動きが表現できていることが確認できる. しかし, Overlapping grid を用いた場合でもパイロード付近では細かな乱流が表現できており, 高速性を重視したい場合には提案手法の方が有効である. 図 6.2 に 450 ステップのパイロード付近を拡大した様子, 図 6.3 に 450 ステップのパイロードから離れた部分を拡大した様子を示す.

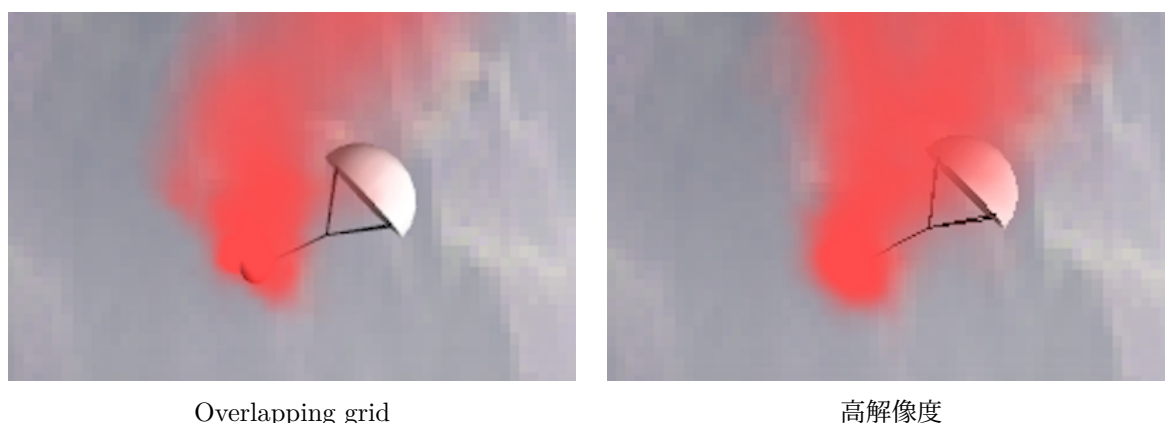
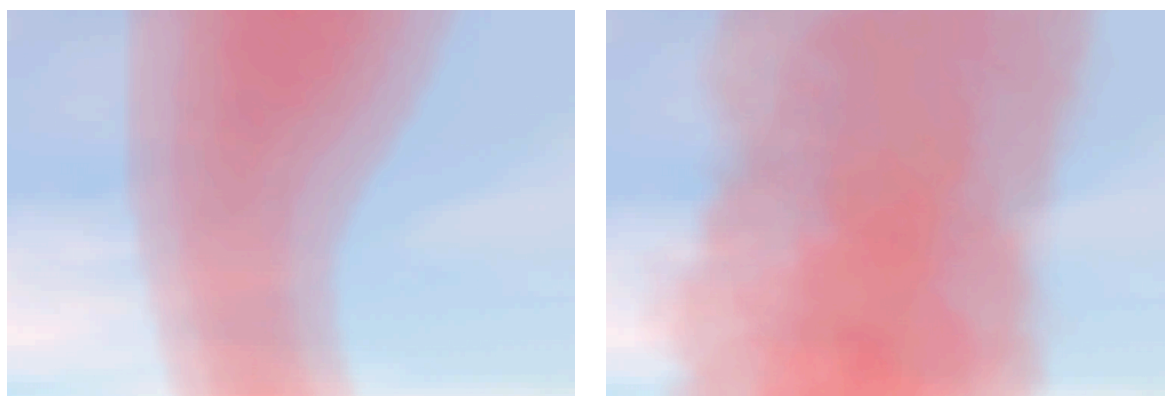


図 6.2: パイロード付近拡大図 (450 ステップ)



Overlapping grid

高解像度

図 6.3: ペイロードから離れた部分の拡大図 (450 ステップ)

6.4 複数の煙竜があるシーン

本提案手法を用いて、実際の花火大会で見られる複数の色の違う煙竜があるシーンを再現できた。2つの煙竜の密度のデータを1つのボリュームデータに格納し、ボリュームレンダリングの際に、場所によって色を変えることでそれぞれ違う色で描画を行なった。この手法では簡単に複数の煙竜を描画できるが、色を場所によって指定しているため、煙竜の場所を調節する必要がある。また、色の混ざり合いは表現することができない。4.4節で示したような他のレンダリング手法を検討することで、このような問題を解決できると考える。

第7章 結論

本論文では、パラシュートのシミュレーションと煙のシミュレーションを組み合わせることにより、昼花火の一種である煙竜を再現する手法を提案した。煙竜は、発煙筒をつけたパラシュートを落下させて、煙が螺旋状の軌跡を描く様子を鑑賞する花火であるため、パラシュートのシミュレーションでは、パラシュートを航空力学的な力を考慮した屋根部分、自由落下する質点である荷物部分としてモデル化し、間を繋ぐ紐をバネ・質点モデルを用いて表現することで、パラシュートが描く螺旋状の軌跡を再現した。また、煙は格子法による流体シミュレーションで再現し、Overlapping grid を用いて広範囲のシミュレーションを高解像度かつ高速に行うことを可能にした。実験では、パラシュートのパラメータ調整による軌跡の変化、煙のパラメータ調整による煙の見た目の変化から、様々な見た目の煙竜を再現できることを確認した。

今後の課題としては、まずパラシュートのシミュレーションに関して、現在は屋根と紐や荷物部分の衝突を考えていないため、屋根を紐や荷物部分が突き抜けてしまう場合がある点が挙げられる。また、本研究では荷物部分を方向を持たない質点としてモデル化しているが、剛体としてモデル化することで、煙が噴出する向きの変化を考慮することができる。煙のシミュレーションに関しては、4.1節で述べたように、移流項の計算の際に高精度な補間法を用いることで数値拡散を抑えることも今後の課題である。また、パラシュートの形状と煙のインタラクション(衝突)を考慮し、さらにレンダリングの際にレイキャスティング法やフォトンマッピング法を用いて煙内部の光の散乱を再現することで、より現実的な煙のCG表現が可能になると考えられる。

以上のように衝突判定を追加したり、レンダリング手法を変えることで、見た目はより現実的になるが、計算時間が増大してしまう。そのためGPUによって処理を並列化することで、シミュレーションを高速化することも今後必要となる。また、試し打ちとしてシミュレーションを利用するためには、本研究では扱っていない花火の打ち上げ部分やパラシュートが開く過程などの再現も必要となると考えられる。

謝辞

本研究を進めるにあたって、終始丁寧にご指導いただいた筑波大学図書館情報メディア系物理ベースコンピュータグラフィックス研究室の藤澤誠准教授に心から感謝申し上げます。合同ゼミにおいて多くのご意見、ご助言を賜りました筑波大学図書館情報メディア系の三河正彦准教授に深く御礼申し上げますとともに、物理ベースコンピュータグラフィックス研究室内のメンバー、ソーシャルロボット研究室のメンバーに感謝致します。

また、論文内で使用した昼花火の写真に関して、突然のお願いにも関わらず快くご提供いただいた石川靖子様、秋田県大仙市花火産業推進課の皆様にも心より御礼申し上げます。

参考文献

- [1] Vladimir N. Dobrokhodov, Oleg A. Yakimenko, and Christopher J. Junge. Six-degree-of-freedom model of a controlled circular parachute. *Journal of Aircraft*, Vol. 40, No. 3, pp. 482–493, 2003.
- [2] Hsuan-Ting Chang, Hsiao-Wei Peng, and Cheng-Hsiu Tsai. Cuda-accelerated rendering of fireworks in nearly ultra high definition videos. In *Proceedings of 2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*, pp. 251–254, 2016.
- [3] Shihui Zhang. Fireworks simulation based on particle system. In *Proceedings of 2009 Second International Conference on Information and Computing Science*, pp. 187–190, 2009.
- [4] 大曲商工会議所. 全国花火競技大会 | 大曲の花火. <https://www.oomagari-hanabi.com>.
- [5] Yoshinori Dobashi, Yasuhiro Matsuda, Tsuyoshi Yamamoto, and Tomoyuki Nishita. A fast simulation method using overlapping grids for interactions between smoke and rigid objects. *Computer Graphics Forum*, Vol. 27, No. 2, pp. 477–486, 2008.
- [6] Sandaruwan Gunasinghe, Kapila Dias, Damitha Sandaruwan, and Maheshya Weerasinghe. A real-time 6dof computational model to simulate ram-air parachute dynamics. *International Journal of Information Technology and Computer Science (IJITCS)*, Vol. 3, pp. 19–27, 2017.
- [7] Kenji Takizawa and Tayfun Tezduyar. Computational methods for parachute fluid–structure interactions. *Archives of Computational Methods in Engineering*, Vol. 19, pp. 125–169, 2012.
- [8] Keith Stein, Richard Benney, Vinay Kalro, Tayfun Tezduyar, John Leonard, and Michael Accorsi. Parachute fluid-structure interactions: 3-d computation. *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, pp. 373–386, 2000.
- [9] Giorgio Guglieri. Parachute-payload system flight dynamics and trajectory simulation. *International Journal of Aerospace Engineering*, Vol. 2012, pp. 1–17, 2012.
- [10] Jianhua Tang and Linfang Qian. Numerical formulation of parachute payload decelerator system and its animation with moving background. In *Proceedings of 2015 International Conference on Virtual Reality and Visualization (ICVRV)*, pp. 201–204, 2015.
- [11] Nick Foster and Dimitris N. Metaxas. Modeling the motion of a hot, turbulent gas. In *Proceedings of SIGGRAPH 97*, pp. 181–188, 1997.
- [12] Jos Stam. Stable fluids. In *Proceedings of SIGGRAPH 1999*, pp. 121–128, 1999.

- [13] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In *Proceedings of SIGGRAPH 2001*, p. 15–22, 2001.
- [14] Theodore Kim, Nils Thürey, Doug James, and Markus Gross. Wavelet turbulence for fluid simulation. *ACM Transactions on Graphics*, Vol. 27, No. 3, p. 1–6, 2008.
- [15] Syuhei Sato, Yoshinori Dobashi, Theodore Kim, and Tomoyuki Nishita. Example-based turbulence style transfer. *ACM Transactions on Graphics*, Vol. 37, No. 4, 2018.
- [16] Frank Losasso, Frédéric Gibou, and Ron Fedkiw. Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics*, Vol. 23, No. 3, p. 457–462, 2004.
- [17] Rajsekhar Setaluri, Mridul Aanjaneya, Sean Bauer, and Eftychios Sifakis. Spgrid: A sparse paged grid structure applied to adaptive smoke simulation. *ACM Transactions on Graphics*, Vol. 33, No. 6, pp. 205:1–205:12, 2014.