

流体との相互作用を考慮した
毛髪 of 塑性変形シミュレーション

筑波大学
図書館情報メディア研究科
2020年3月
出村 佑史

目次

第 1 章	序論	1
1.1	背景と目的	1
1.2	関連研究	2
1.2.1	毛髪弾塑性変形	2
1.2.2	毛髪の相互作用	3
1.3	本論文の構成	5
第 2 章	提案手法の概要	7
2.1	シミュレーション全体の構成	7
2.2	位置ベース法	9
2.2.1	力学ベース法と位置ベース法	9
2.2.2	位置ベース法の理論	9
第 3 章	毛髪の弾塑性変形シミュレーション	13
3.1	毛髪形状と離散化	13
3.2	弾性変形シミュレーション	15
3.2.1	伸びと剪断の制約条件	15
3.2.2	曲げとねじれの制約条件	17
3.3	塑性変形シミュレーション	18
3.3.1	側鎖結合の種類	18
3.3.2	側鎖結合の影響の適用	19
3.3.3	側鎖結合が保持する形状の更新	20
3.3.4	結合率の変動	21
第 4 章	流体シミュレーションによる相互作用	24
4.1	流体シミュレーション	24
4.1.1	流体の支配方程式	24
4.1.2	SPH を用いた流体の離散化	25
4.1.3	空気粒子と毛髪粒子	27
4.2	PBF による流体シミュレーション	28
4.2.1	密度の制約条件	28
4.2.2	XSPH 人工粘性	29
4.2.3	熱拡散	30
4.3	渦を利用した毛髪のなびきの制御	31
4.3.1	Vorticity Confinement による渦の強調	32
4.3.2	ウェーブレット解析を用いた特定の大きさの渦の検出	33

第 5 章	実装	35
5.1	並列化	35
5.1.1	ホストメモリとデバイスメモリ	35
5.1.2	CUDA カーネルによる問題の並列処理	37
5.2	近傍粒子探索	38
5.3	シミュレーションの安定化	40
5.3.1	静止密度の初期値	40
5.3.2	方向を表すベクトルから姿勢を表す四元数の生成	40
5.4	レンダリング	41
第 6 章	結果	43
6.1	シーン 1	43
6.2	シーン 2	44
6.3	シーン 3	45
6.4	シーン 4	46
6.5	考察	47
6.5.1	相互作用を考慮した毛髪の変形	47
6.5.2	渦の制御による毛髪のなびき運動	48
第 7 章	結論	63
	謝辞	65
	参考文献	66
付 録 A	四元数を用いた計算	69

目 次

1.1	パーマネントウェーブを施術した毛髪	1
1.2	四元数を用いたロッドシミュレーション	3
1.3	濡れた毛髪の形状と乾いた毛髪の形状を設定し水分量によって変形させている様子	4
1.4	側鎖結合を考慮した毛髪の塑性変形シミュレーション	4
1.5	SPH を用いて毛髪同士と毛髪と空気の相互作用を統一的に計算した手法	5
2.1	シミュレーションの1ステップの流れ	8
2.2	力学ベース法の計算方法	10
2.3	位置ベース法の計算方法	10
3.1	Cosserat ロッドモデルによる毛髪	13
3.2	毛髪の離散化	14
3.3	離散化した毛髪の Darboux ベクトル	15
3.4	離散化した毛髪の伸び・剪断	16
3.5	側鎖結合によって塑性変形が発生する流れ	19
3.6	再結合することで側鎖結合が現在の形状を記憶する概念図	20
3.7	温度と水素結合の結合率の関係を表す関数 $f(T)$	22
4.1	格子法と粒子法による離散化	25
4.2	近傍粒子と重み関数による物理量の離散化	26
4.3	空気粒子と毛髪粒子	27
4.4	熱拡散と対流	30
4.5	毛髪が風になびくアニメーションを作成する手法の一例	31
4.6	毛髪と空気の相互作用で発生した渦を伴う乱流	32
4.7	Vorticity Confinement による渦の強調	33
4.8	1次元のメキシカンハットウェーブレット ($\alpha = 1$)	34
5.1	ホストとデバイス間のメモリ転送	35
5.2	静的配列を用いた各計算点・粒子のメモリ管理	36
5.3	CUDA カーネルの実行	37
5.4	各辺での並列化で起こる計算点位置修正の競合	38
5.5	等間隔格子による近傍探索手法	39
5.6	初期形状における毛髪粒子の密度分布	41
5.7	Kajiya-Kai モデルによる異方性反射計算	42
5.8	Kajiya-Kai モデルのレンダリング結果	42
6.1	温度のカラーマップ	44
6.2	シーン1：ストレートヘア ($\epsilon_{\text{hair}}^{\text{visc}} = 1$) 既存手法	49

6.3	シーン 1 : ストレートヘア ($\epsilon_{\text{hair}}^{\text{visc}} = 1, \alpha = 0.006[\text{m}]$)	50
6.4	シーン 1 : カールヘア ($\epsilon_{\text{hair}}^{\text{visc}} = 1, \alpha = 0.006[\text{m}]$)	51
6.5	シーン 1 : 乱流シミュレーションの比較	52
6.6	シーン 1 : 粘性の違いの比較	52
6.7	シーン 2 : 温風を 8 秒間あて続けた場合の温度変化	53
6.8	シーン 2 : 温風を 6 秒間あてたあと冷風を 2 秒間あてた場合の温度変化	54
6.9	シーン 2 : 温風を 6 秒間あてたあと冷風を 2 秒間あてた場合のレンダリング結果	55
6.10	シーン 2 : 風と温度による塑性変形の比較	56
6.11	シーン 3 : レンダリング結果	57
6.12	シーン 3 : 温度変化	58
6.13	シーン 3 : 元の形状とシミュレーション後の形状の比較	59
6.14	シーン 4 : レンダリング結果	60
6.15	シーン 4 : 温度変化	61
6.16	シーン 4 : カール形状の比較	62

表 目 次

1.1	本論文で用いる記号の一覧	6
5.1	ソート後のグリッドハッシュと粒子インデックス	39
5.2	各グリッドハッシュの start と end	39
6.1	実行環境	43
6.2	シーン 1 共通のパラメータ設定	44
6.3	シーン 1 の計算時間	44
6.4	シーン 2 のパラメータ設定	45
6.5	シーン 3 のパラメータ設定	46
6.6	シーン 4 のパラメータ設定	47

第1章 序論

1.1 背景と目的

近年の映画やテレビ，ゲームなどエンタテインメント分野において，コンピュータグラフィックス (CG) による映像表現は欠かせない技術であり，日々進化を続けている．従来は CG アニメーションのクオリティはアーティストの技量と作業量に強く依存していたが，物理現象をコンピュータで再現することができる物理シミュレーションが応用されるようになり，アーティストの負担を軽減しつつ，高品質なアニメーションを効率よく生成することが可能になった．さらに，高速なシミュレーション手法を用いることでリアルタイムで計算が可能になり，即時にアニメーションを生成することができるため，ゲームのようにインタラクティブに CG アニメーションが変動するアプリケーションでも幅広く応用されている．毛髪もまた物理シミュレーションの計算対象であり，人間のように毛髪を有するキャラクターのアニメーションを生成するための重要な役割を担っている．しかしながら，毛髪は本数が非常に多く動き方も複雑であるため，盛んに研究が行われているもののまだ多数の解決すべき点がある．

毛髪の基本的な動きとして，まず曲げやねじれなどの変形が発生すると元の形状に戻ろうとする弾性変形がある．ほとんどのシミュレーションでは計算の複雑化を避けるために毛髪を弾性体として扱っている．しかし一方で，実際の毛髪では弾性変形だけではなく，元の形状に完全には戻らなくなる塑性変形が発生する場合がある．寝癖やパーマネントウェーブ (図 1.1)，ヘアドライヤーやヘアアイロンによるスタイリングなどが主な例として挙げられる．水分や温度の変動によって毛髪の性質が後天的に変動し，日常的に塑性変形が起きている．映画やゲームのシーンでもキャラクターの毛髪が水に浸ることが多々あり，より現実的な毛髪のアニメーションを生成するためにはこの塑性変形を考慮することも必要である．



図 1.1: パーマネントウェーブを施した毛髪

毛髪の主成分はケラチンと呼ばれるタンパク質であり、その内部ではケラチンを構成するポリペプチドを繋ぐ無数の主鎖結合と側鎖結合が存在している。水や整髪料による吸着や固定のような外的な力の影響を除くと、この側鎖結合が変化することが毛髪の塑性変形の原因と考えられている [1]。身近な例では、ヘアアイロンは熱によって側鎖結合を切断し毛髪を柔らかくした後冷やすことで側鎖を再結合させて毛髪の形状を変化させており、ヘアドライヤーは水分によって切断された側鎖結合を乾かすことによって再結合させることで毛髪形状を変形させている。このように毛髪の塑性変形には温度や風、水分が深く関わっている。CG アニメーションのために塑性変形を扱った従来の研究 [2] では、この側鎖結合を考慮しているもののその結合率の変動は手動で設定する必要があり、周囲の流体との相互作用や温度の影響を考慮することができなかった。

また、毛髪は非常に軽量であるためヘアドライヤーの温風のような空気の流れでもその挙動が大きく変化し、側鎖結合に影響する。毛髪と空気の相互作用によって毛髪が風になびく現象はその典型的な例であり、CG アニメーションのシーンでも頻出する。従来の流体シミュレーションで空気との相互作用を計算した手法 [3] では、流体の非圧縮性を満たすことができておらず、空気の流れに発生する乱流の影響については考慮していない。近年は三角関数や航空力学を用いた高速なシミュレーション手法 [4][5] が提案されているが、それらの手法では空気が毛髪に与える影響はシミュレーションできているが毛髪が空気の流れを乱す現象は再現できない。

本論文では、外部流体と毛髪の相互作用を考慮した毛髪の塑性変形の過程を流体シミュレーションを用いることで計算する手法を提案する。提案手法では毛髪の弾塑性変形と流体運動を、安定して高速な計算が可能である位置ベース法 (Position Based Dynamics)[6] を用いて計算し、毛髪同士の相互作用と毛髪と流体の相互作用を単一の流体シミュレーションで統一的に計算することで、流体シミュレーションの計算量低下とシミュレーションの安定性を実現する。そして、毛髪の塑性変形計算では毛髪内部の側鎖結合を考慮し、その結合率を流体との相互作用から計算することで、従来の手法では再現できなかった外部流体の影響を考慮する。さらに、空気と毛髪が相互作用することによって発生する乱流を制御することで、ユーザによる風になびくアニメーションの制御を可能にする。提案手法は計算の並列化に適した粒子法を用いているため、実験のために作成したシミュレータではグラフィクス用のプロセッサである GPU 上での汎用計算の並列化を可能とする GPGPU を用いることで、高速なシミュレーションを実現した。本研究の応用例として映画やゲームのシーンにおけるキャラクタの毛髪の映像表現として使用できるほか、パーマメントウェーブ施術後の毛髪形状の 3D モデリング手法や、美容師や理容師が訓練するためのシミュレータ [7] の精度向上などが考えられる。

1.2 関連研究

1.2.1 毛髪の弾塑性変形

CG アニメーション分野では毛髪を有するキャラクタが頻出するため、毛髪の動きのシミュレーションについてはこれまで数多くの手法が提案されてきた [8]。毛髪は本数が膨大であるため、シミュレーションするにあたり最初に問題となるのは計算の高速性である。Müllerらは、従来の力学ベース手法とは異なり幾何学的な制約条件によって物体の位置を動かす手法である位置ベース法 [6] を提案し、毛髪のようにロッド形状の物体を高速にシミュレーションすることを可能にした [9]。しかしながら、質点位置だけでは毛髪の曲げとねじれに対する弾性変形を表現することはできなかった。そこで梅谷らは、新たに曲率と捩率を保

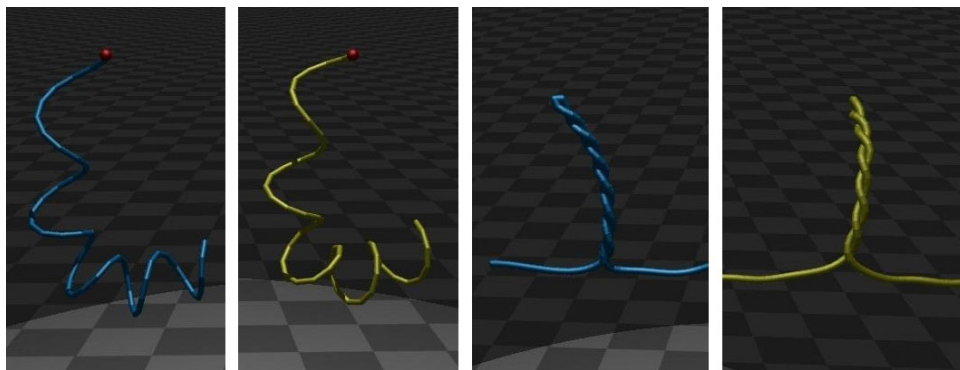


図 1.2: 四元数を用いたロッドシミュレーション [11]

持するためのゴーストポイントを追加することで、位置ベース法での曲げとねじれに対する弾性変形のシミュレーションを実現した [10]. Kugelstadt らは梅谷らの手法を元に、ゴーストポイントの代わりに四元数 (Quaternion) を用いることでより高速にシミュレーションを行うことを可能にしている [11](図 1.2). 従来の位置ベース法では計算の反復回数で弾性を表す物性値が変動するという問題点があり、それを解決して反復回数に依らず物性値を設定することが可能な手法 [12] も提案されているが、本研究では実装が比較的容易で計算の並列化にも適している Kugelstadt らの手法 [11] を用いる.

毛髪の塑性変形を扱う手法についても様々な手法が提案されている. 林本らは美容師や理容師の訓練用シミュレータ [7] を提案し、ヘアカットだけでなくヘアワックスを模倣した変形をユーザが設定できるインターフェースを開発した. 一方で Rungjiratananon らの提案した水に濡れた毛髪のシミュレーション手法 [13] では、濡れている状態の毛髪の形状と乾いている状態の毛髪の形状をあらかじめユーザに定義しておくことで、水分量の変動に伴う毛髪の塑性変形を再現した (図 1.3). しかしこれらの手法では変形の度合いや変形後の形状をユーザに決定させているため、ユーザの技量への依存が大きく、自然に発生した塑性変形を再現することができない. そこで、我々は卒業研究において毛髪を構成するケラチン内に存在する側鎖結合に注目し、側鎖結合の変動を近似的にモデル化することで、寝癖やパーマントウェーブの基本的なメカニズムを再現する手法 [2](図 1.4) を提案した. しかし、側鎖結合の結合率は手動で変更されることが前提であり、周囲の流体との相互作用によって起こる結合率の変動は考慮できていない.

1.2.2 毛髪の相互作用

毛髪の動きを計算するにあたり、欠かせない要素の一つに相互作用計算がある. 例えば、毛髪同士が衝突しあうことで髪全体にボリュームが生まれる. 毛髪のベタつき感や複雑な絡み合いを表現するためには摩擦力の考慮が不可欠であり、毛髪表面に水が付着することによっても質量の変化や水の表面張力による吸着が発生する. 従来の研究では、水による抗力・表面張力・伝播を再現した手法 [13][14] や、毛髪同士の衝突や摩擦力を高解像度で計算した手法 [15][16] などが提案されている.

一方で、毛髪と相互作用を行う身近な流体として空気がある. 空気もまた、風によってなびく毛髪の動きを再現する上で欠かせない流体である. 風によってなびいている毛髪の CG アニメーションを生成するために、航空力学を用いて風が毛髪に与える力をシミュレーションする手法 [5] や、三角関数を用いて波打っている毛髪の動きを制御する手法 [4] が提案さ



図 1.3: 濡れた毛髪の形状と乾いた毛髪の形状を設定し水分量によって変形させている様子 [13]



図 1.4: 側鎖結合を考慮した毛髪の塑性変形シミュレーション [2]

れた．これらの手法は毛髪のなびき運動を非常に高速に計算することが可能であり，高速生が重要視されるリアルタイムアプリケーションでは有用性が高い．しかしながら，毛髪が空気に与える影響を考慮していないため，毛髪によって空気の流れが乱されて発生する風力の減衰や渦の動きを再現することができない．

毛髪同士の相互作用を計算しつつ外部流体との相互作用も考慮することは膨大な計算量を伴う．Hadap らは毛髪同士が衝突して髪型にボリュームが生まれる現象が非圧縮性流体の体積保存と似ていることから，流体シミュレーションによって毛髪同士の相互作用を計算する手法を提案した [3](図 1.5)．毛髪を連続体としてモデル化することで，毛髪同士の相互作用と空気のような流体との相互作用を単一の流体シミュレーションで同時に計算することができる．これにより複数の手法が混在することによるシミュレーションの不安定化や，システムの複雑化による計算量の増加といった問題点を抑制した．Hadap らの手法では流体シミュレーションに粒子法的一种である SPH(Smoothed Particle Hydrodynamics)[17] を用いていたが，粒子法と格子法のハイブリッド手法である FLIP(Fluid Implicit Particles) や MPM(Material Point Method) を拡張した手法 [18][16] など，異なる流体シミュレーション手法を用いて毛髪の相互作用を計算する手法が提案されている．

以上の手法では空気との相互作用において毛髪が空気の流れを乱すことによって発生する渦を伴う乱流の動きは考慮していない．本研究では粒子法を用いて毛髪の相互作用を計算する手法である Hadap らの手法を拡張する．流体シミュレーションには安定して高速に非圧縮性を満たす流体のシミュレーションを可能とする PBF(Position Based Fluids)[19] を使用し，乱流シミュレーション [20] を組み合わせることで空気の流れの中に発生した渦の制御を可能とする．



図 1.5: SPH を用いて毛髪同士と毛髪と空気の相互作用を統一的に計算した手法 [3]

1.3 本論文の構成

本論文で提案する手法は，毛髪の弾塑性変形シミュレーションと流体シミュレーションによる相互作用シミュレーションの二つに大きく分けられる．そこで，まず2章で提案手法によるシミュレーション全体の構成と，シミュレーションを行う上で基礎となる技術である位置ベース法について詳しく説明する．そして，3章で毛髪の弾塑性変形計算を行う手法について述べ，4章で毛髪同士の相互作用および毛髪と外部流体との相互作用を流体シミュレーションを用いて計算する手法について述べる．5章では並列計算を可能とする GPU を用いた提案手法の実装について説明する．6章において実際の実験の環境・結果について示し，最終的に7章で本研究の結論を述べる．

本論文で用いる記号の一覧を表 1.1 に示す．粒子や計算点などのインデックスは添え字 i, j, k を用いて表し，特に対象の粒子が毛髪粒子のみである場合は hair ，空気粒子のみである場合は air を付ける．また，初期状態を表す値には添え字 orig ，外力が加わっていない静止状態で維持しようとする値を rest ，前のステップの状態を表す値には添え字 prev を付ける．なお，本論文では四元数を用いている．四元数 q の共役四元数は \bar{q} ，スカラー部とベクトル部をそれぞれ $\Re(q), \Im(q)$ で表す．スカラー部 \Re とベクトル部 \Im をそれぞれ要素に持つ四元数は $(\Re|\Im)$ と表記する．

表 1.1: 本論文で用いる記号の一覧

C, C	位置ベース法の制約条件 (関数)
λ	位置ベース法の位置修正に用いるスケーリング係数
e_1, e_2, e_3	3次元のグローバル座標系の基底ベクトル
d_1, d_2, d_3	空間曲線のマテリアルフレームの基底ベクトル
L	長さが L^{\max} の空間曲線の始点からの弧長 $L \in [0, L^{\max}]$
t	時間
g	重力加速度
n	計算点の数
V	体積
F	力
a	加速度
v	速度
p	位置
m	質量
w	重み係数
l	計算点間の距離
q	空間曲線の姿勢を表す四元数
ω	角速度
R	ベクトルを四元数で回転させる関数
Ω^*	毛髪形状 (曲率・振率) を表す四元数 (ベクトル部が Darboux ベクトル)
Ω	毛髪形状の Darboux ベクトル $\Omega = \Im(\Omega^*)$
s	Ω^*, Ω, Ω のための符号係数
K	毛髪の弾性を表す係数 $K \in [0, 1]$
B	側鎖結合の種類集合
b	側鎖結合の結合率 $b \in [0, 1]$
M	毛髪の水分含有率 $M \in [0, 1]$
f	温度と水素結合の結合率の関数
W	粒子法のカーネル関数
h	カーネル関数 W の影響半径
ϕ	任意の物理量
ρ	流体の密度
P	流体の圧力
μ	流体の粘性係数
T	温度
κ	熱伝導係数
c	比熱容量
ζ	流体の渦度
N	渦渡場のノルムの傾きを表す単位ベクトル
\hat{v}	流体速度の周波数成分
\hat{e}	渦のエネルギー値
ϵ	調整用の任意設定パラメータ
ψ	ウェーブレット関数
α	抽出する渦のスケール (ウェーブレットの拡大縮小量)
β	ウェーブレットの平行移動量ベクトル
I	反射光の光の強度
A	光の反射・光沢度を表す係数
θ	光の入射角・反射角を表す角度
max, min	それぞれ引数のうち大きな値を返す関数と小さな値を返す関数
r, γ, η	式を整理するための変数
Ω, e_3, r, γ	それぞれ四元数 $(0 \Omega), (0 e_3), (0 r), (0 \gamma)$

第2章 提案手法の概要

本章では提案手法である毛髪シミュレーション全体の構成を 2.1 節で述べたあと、提案手法を説明する上で基礎となる位置ベース法の説明を 2.2 節で行う。

2.1 シミュレーション全体の構成

本節では本研究で用いるシミュレーション全体の構成について述べる。提案するシミュレーション手法は、毛髪の塑性変形をシミュレーションするための手法と流体との相互作用をシミュレーションする手法の二つに分けられる。

毛髪は曲げやねじれのような変形に対して元の形状に戻るように動くロッド形状の弾性体としてふるまう性質を持っている。この弾性変形を実現するために本研究では四元数 (Quaternion) を用いて高速にロッド形状の弾性変形を計算する手法 [11] を用いる。また、毛髪には寝癖やパーマメントウェーブのように塑性変形が発生する。毛髪の塑性変形は毛髪を構成するケラチン内部に無数に存在する側鎖結合が切断と再結合を繰り返すことが原因と考えられており、提案手法ではその基本的なメカニズムを近似的にモデル化した手法 [2] を用いる。側鎖結合の結合率の変動は本物の毛髪を使用した実験データを参考に毛髪の水分含有量と温度を用いて計算する。本研究では実験データから推測される値を利用するが、結合率と水分含有量や温度との関係は毛髪に個人差があるためユーザが自由に設定することも可能である。

本研究では毛髪と相互作用する最も身近な外部流体として空気に焦点をあててシミュレーションを行う。相互作用計算では毛髪同士の相互作用と毛髪と外部流体 (空気や水など) の相互作用に大きく分けることができる。毛髪同士の相互作用と毛髪と空気の相互作用の処理をそれぞれ分割して異なる手法で計算すると、計算量が増大しシミュレーションが複雑化してしまう。そのため、提案手法では流体と毛髪をどちらも連続体としてモデル化することで、一度の流体シミュレーションで統一的に毛髪同士と毛髪と空気の相互作用を計算する手法 [3] を用いる。流体シミュレーションには PBF [19] を使用した。PBF は粒子で流体をモデル化する粒子法的一种であり、非圧縮性を満たす流体を安定してシミュレーションすることが可能である。しかしながら、シミュレーション空間を粒子で満たすと膨大な計算量が要求されるため、本研究では風が発生している空間のみ粒子を配置した上で空気粒子には重力を加えないことで計算量を抑制する。

毛髪は非常に軽量であるため、風の影響によってその挙動が大きく変化する。本研究では現実的な毛髪のなびき表現を実現するためには空気の流れて発生する渦の考慮が必要であると仮定し、この渦を表現・制御するために乱流シミュレーションを行う。従来の乱流シミュレーションの一つに、コンピュータシミュレーションの数値的散逸によって失われた乱流の動きを人工的に力を加えることで渦を強調し修正する手法である Vorticity Confinement [20] がある。提案手法では流体に発生している渦からウェーブレット解析を用いて特定の大きさの渦を検出することで、ユーザが指定した任意の大きさの渦を Vorticity Confinement によって強調することで乱流を制御する。

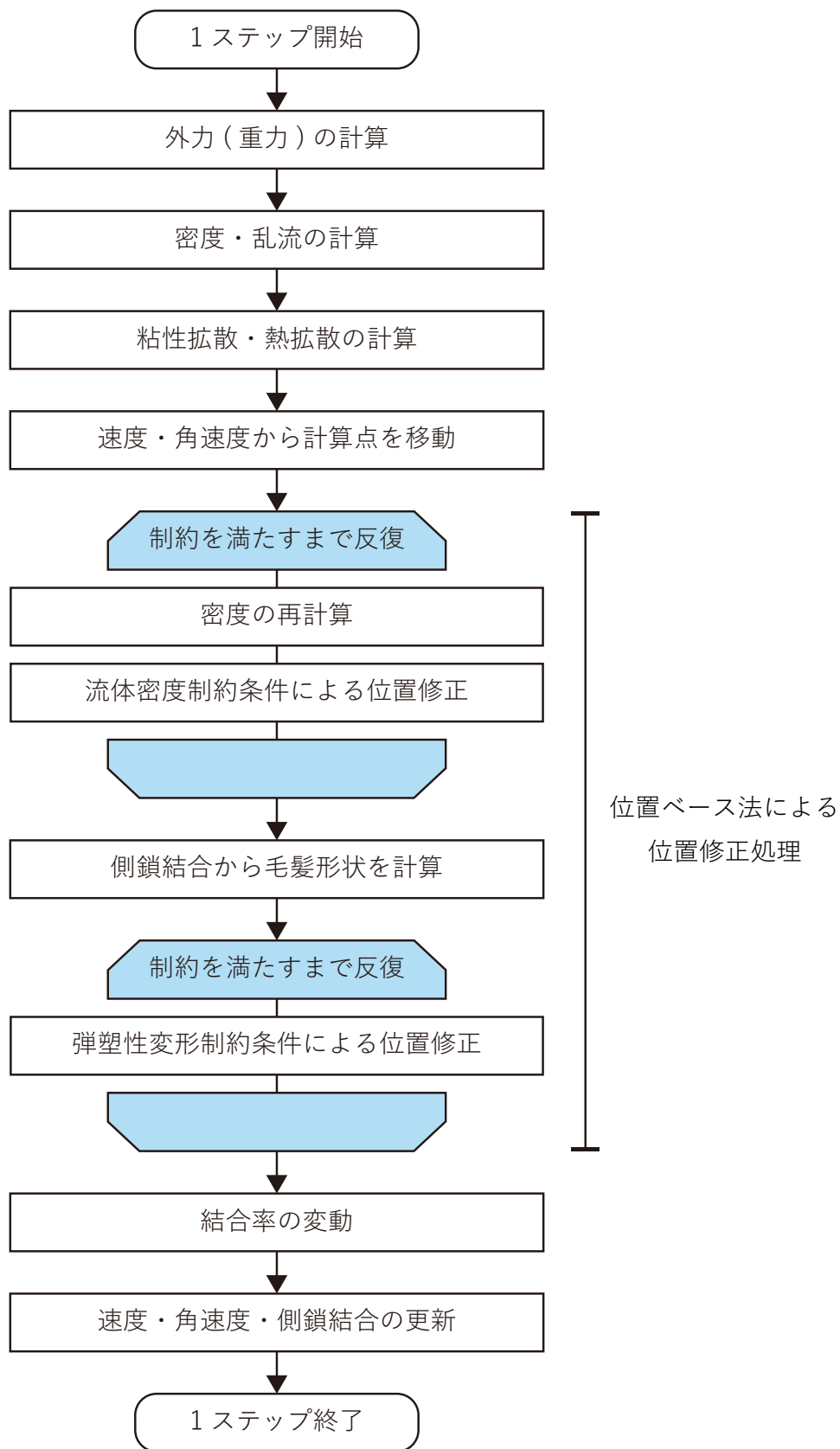


図 2.1: シミュレーションの 1 ステップの流れ

シミュレーションの1ステップ内での処理の流れを図 2.1 に示す。まず重力などの外力を加えた後、流体密度を計算し、渦を制御するための力を加える (4.3 節)。次に流体の粘性拡散・熱拡散を計算し (4.2.2 項, 4.2.3 項)、これまでの計算で求められた速度と角速度を用いて計算点を移動させる。移動後の計算点の位置を、PBF の密度制約条件 (4.2.1 項) と弾塑性変形の制約条件 (3.2.1 項, 3.2.2 項) によって修正する。最後に、熱拡散による温度変化から側鎖結合の結合率を変化させ (3.3.4 項)、速度や角速度とともに側鎖結合を更新する (3.3.3 項)。なお、毛髪弾塑性変形計算と流体の圧力項計算は安定したシミュレーションが可能な位置ベース法に基づいて行う。位置ベース法に関する詳細は次の 2.2 節で述べる。

2.2 位置ベース法

2.2.1 力学ベース法と位置ベース法

位置ベース法 (Position Based Dynamics) [6] は高い制御性と安定性を併せ持つ物理シミュレーション手法である。従来の力学ベース法では、ニュートンの運動方程式に基づいて計算された加速度から物体の位置を求めている。力学ベース法をコンピュータで離散的にシミュレーションを行う場合、シミュレーションステップ間の時間幅を Δt とすると図 2.2 に示す流れでシミュレーションが進む。このとき物体運動の支配方程式に従って加速度 \mathbf{a} 、速度 \mathbf{v} 、位置 \mathbf{p} と順番に時間積分を繰り返しているため、最終的に CG 描画で必要となる位置 \mathbf{p} の値には誤差が蓄積してシミュレーションが不安定になる場合がある。

位置ベース法では力学ベース法と異なり、図 2.3 に示すように幾何学的な制約条件を用いて直接位置を求めている。そのため誤差の蓄積は発生せず、安定かつ高い制御性を持つことが特徴である。幾何学的な制約条件とは、シミュレーションする物体の位置に注目したときにその位置が満たしているべき条件のことである。例えば破断や変形を考慮しない単純な振り子運動をシミュレーションする場合は、振り子の始端と終端の距離が一定であるという制約条件を考えることができる。一方で幾何学的な手法である位置ベース法は物理的とは言えないのではないかという議論も存在するが、そのシミュレーション結果の現実性の高さや安定性・制御性などの観点において CG 分野では有用性が高く、様々な応用がされている [21]。次の 2.2.2 項において位置ベース法の詳しい理論について説明する。

2.2.2 位置ベース法の理論

位置ベース法では移動する物体の計算点の位置 \mathbf{p} に対して制約条件 $C(\mathbf{p})$ が与えられたとき、式 (2.1) を満たすように位置 \mathbf{p} を修正する。

$$C(\mathbf{p}) = 0 \quad (2.1)$$

なお本研究では扱わないが、 $C(\mathbf{p}) \geq 0$ を満たすように位置を修正する制約も存在する。今回求めたい位置の修正量を $\Delta \mathbf{p}$ とすると、修正後の計算点の位置は $\mathbf{p} + \Delta \mathbf{p}$ と表すことができる。位置修正後には制約条件を満たしている必要があるため、式 (2.2) が得られる。

$$C(\mathbf{p} + \Delta \mathbf{p}) = 0 \quad (2.2)$$

式 (2.2) をテイラー展開し、2 次以降の項を無視すると式 (2.3) に近似することができる。

$$C(\mathbf{p} + \Delta \mathbf{p}) \approx C(\mathbf{p}) + \nabla_{\mathbf{p}} C(\mathbf{p}) \cdot \Delta \mathbf{p} = 0 \quad (2.3)$$

1 ステップ目

2 ステップ目

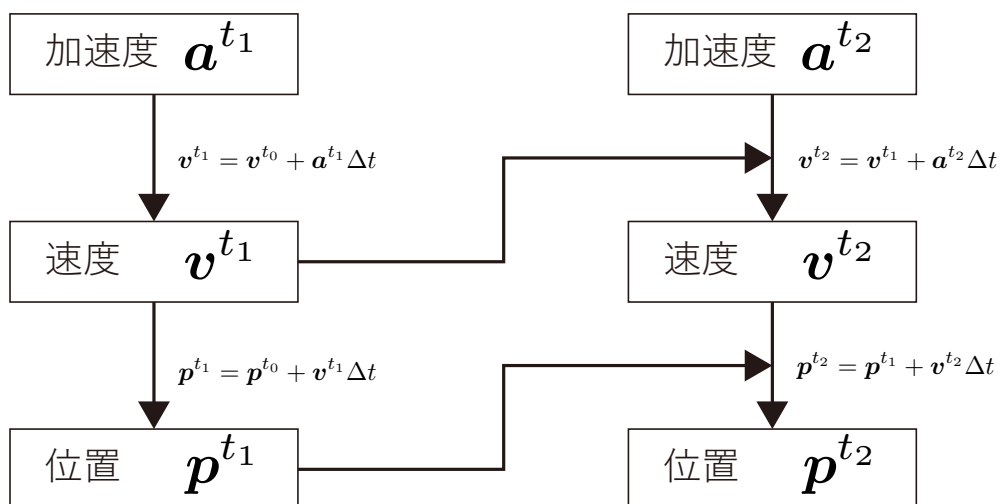


図 2.2: 力学ベース法の計算方法

1 ステップ目

2 ステップ目

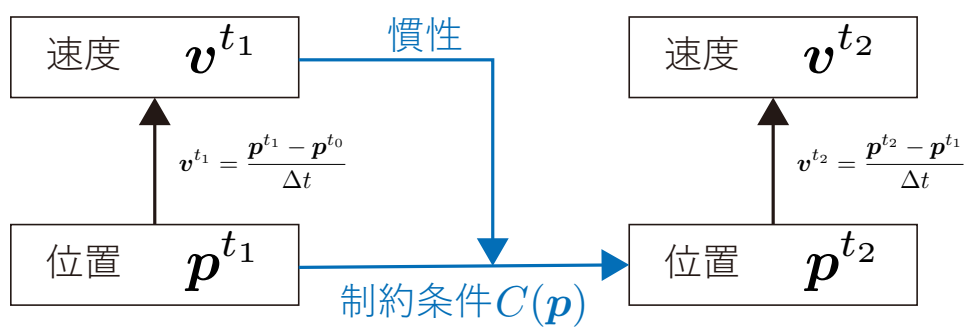


図 2.3: 位置ベース法の計算方法

次に、運動量の保存と角運動量の保存を満たす位置修正を考える．制約条件 $C(\mathbf{p})$ が物体内部の状態に対する制約とすると、 $C(\mathbf{p})$ は物体全体の平行移動や回転運動とは独立した条件となる．逆に物体全体の平行移動や回転運動では $C(\mathbf{p})$ の値は変動しないということなので、 $C(\mathbf{p})$ の空間変化 $\nabla_{\mathbf{p}} C(\mathbf{p})$ はこれらの運動変化に対して直交している．そのため $\nabla_{\mathbf{p}} C(\mathbf{p})$ の方向に位置を修正すると運動量と角運動量が保存される． $\Delta \mathbf{p}$ の向きを $\nabla_{\mathbf{p}} C(\mathbf{p})$ に制限すると以下の式 (2.4) が得られる．

$$\Delta \mathbf{p} = \lambda \nabla_{\mathbf{p}} C(\mathbf{p}) \quad (2.4)$$

ここで、 λ は未知のスケーリング係数である．式 (2.4) を式 (2.3) に代入することで λ の値を計算できる．

$$C(\mathbf{p}) + \nabla_{\mathbf{p}} C(\mathbf{p}) \cdot \lambda \nabla_{\mathbf{p}} C(\mathbf{p}) = 0 \quad (2.5)$$

$$\lambda = -\frac{C(\mathbf{p})}{\|\nabla_{\mathbf{p}} C(\mathbf{p})\|^2} \quad (2.6)$$

$C(\mathbf{p})$ と $\nabla_{\mathbf{p}} C(\mathbf{p})$ は設定された制約条件からその値が求められるため、以下の式 (2.7) で $\Delta \mathbf{p}$ も求めることができる．

$$\Delta \mathbf{p} = -\frac{C(\mathbf{p})}{\|\nabla_{\mathbf{p}} C(\mathbf{p})\|^2} \nabla_{\mathbf{p}} C(\mathbf{p}) \quad (2.7)$$

この考え方を複数の計算点があった場合に拡張する． $\mathbf{p}_1, \dots, \mathbf{p}_n$ の計算点に対して制約条件 $C(\mathbf{p}_1, \dots, \mathbf{p}_n)$ が与えられたとする．このとき計算点 \mathbf{p}_i の修正式は、以下のように導出される．

$$\Delta \mathbf{p}_i = \lambda_i \nabla_{\mathbf{p}_i} C_i(\mathbf{p}_1, \dots, \mathbf{p}_n) \quad (2.8)$$

$$\lambda_i = -\frac{C_i(\mathbf{p}_1, \dots, \mathbf{p}_n)}{\sum_j \|\nabla_{\mathbf{p}_j} C_i(\mathbf{p}_1, \dots, \mathbf{p}_n)\|^2} \quad (2.9)$$

各計算点 $\mathbf{p}_1, \dots, \mathbf{p}_n$ の質量が一定ではない場合はその重みを計算する必要がある．計算点 \mathbf{p}_i の質量の逆数を重み $w_i = \frac{1}{m_i}$ とした場合、求められる修正式を式 (2.10)(2.11) に示す．

$$\Delta \mathbf{p}_i = \lambda_i w_i \nabla_{\mathbf{p}_i} C_i(\mathbf{p}_1, \dots, \mathbf{p}_n) \quad (2.10)$$

$$\lambda_i = -\frac{C_i(\mathbf{p}_1, \dots, \mathbf{p}_n)}{\sum_j w_j \|\nabla_{\mathbf{p}_j} C_i(\mathbf{p}_1, \dots, \mathbf{p}_n)\|^2} \quad (2.11)$$

なお、複数の計算点が存在する場合は一度の修正で制約条件を満たすことが難しい．位置ベース法では図 2.1 に示したように、制約条件が十分満たされるまで位置修正処理を反復する．位置修正後は、1 ステップ前の位置 \mathbf{p}^{prev} からの移動量を元に以下の式で速度 \mathbf{v} を更新する．

$$\mathbf{v} \leftarrow \frac{\mathbf{p} - \mathbf{p}^{\text{prev}}}{\Delta t} \quad (2.12)$$

制約条件の一例

ここでは簡単な制約と、制約条件によって導出される位置修正式の例を示す．計算点 \mathbf{p}_1 と計算点 \mathbf{p}_2 の距離が一定である距離制約を考える． \mathbf{p}_1 と \mathbf{p}_2 の間の距離は $\|\mathbf{p}_2 - \mathbf{p}_1\|$ である．初期状態の頂点間の距離を l とすると、制約条件 C^{stretch} を式 (2.13) のように定義できる．

$$C^{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2) = \|\mathbf{p}_2 - \mathbf{p}_1\| - l = 0 \quad (2.13)$$

また、それぞれの計算点 $\mathbf{p}_1, \mathbf{p}_2$ で制約条件を偏微分して、計算点における制約条件の空間変化を求める．

$$\nabla_{\mathbf{p}_1} C^{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2) = -\frac{\mathbf{p}_2 - \mathbf{p}_1}{\|\mathbf{p}_2 - \mathbf{p}_1\|} \quad (2.14)$$

$$\nabla_{\mathbf{p}_2} C^{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2) = +\frac{\mathbf{p}_2 - \mathbf{p}_1}{\|\mathbf{p}_2 - \mathbf{p}_1\|} \quad (2.15)$$

式 (2.11) より、それぞれのスケール係数 λ^{stretch} は以下のように求まる．

$$\lambda^{\text{stretch}} = -\frac{1}{w_1 + w_2} (\|\mathbf{p}_2 - \mathbf{p}_1\| - l) \quad (2.16)$$

以上より導出される制約条件 C^{stretch} の修正式を以下に示す．

$$\Delta \mathbf{p}_1 = +\frac{w_1}{w_1 + w_2} (\|\mathbf{p}_2 - \mathbf{p}_1\| - l) \frac{\mathbf{p}_2 - \mathbf{p}_1}{\|\mathbf{p}_2 - \mathbf{p}_1\|} \quad (2.17)$$

$$\Delta \mathbf{p}_2 = -\frac{w_2}{w_1 + w_2} (\|\mathbf{p}_2 - \mathbf{p}_1\| - l) \frac{\mathbf{p}_2 - \mathbf{p}_1}{\|\mathbf{p}_2 - \mathbf{p}_1\|} \quad (2.18)$$

ここで、 w_1, w_2 はそれぞれ $\mathbf{p}_1, \mathbf{p}_2$ の質量の逆数である．

第3章 毛髪の弾塑性変形シミュレーション

本章では毛髪の弾塑性変形をシミュレーションする手法について述べる．3.1 節では毛髪表現に用いる Cosserat ロッドモデルとその離散化手法について説明し，3.2 節で弾性変形，3.3 節で塑性変形の計算について説明する．本章を含めて本研究では3次元の回転計算に四元数 (Quaternion) を用いているが，四元数の計算方法については付録 A に示す．なお，本章で説明する弾塑性変形シミュレーションは2.2 節で述べた位置ベース法を用いている．

3.1 毛髪形状と離散化

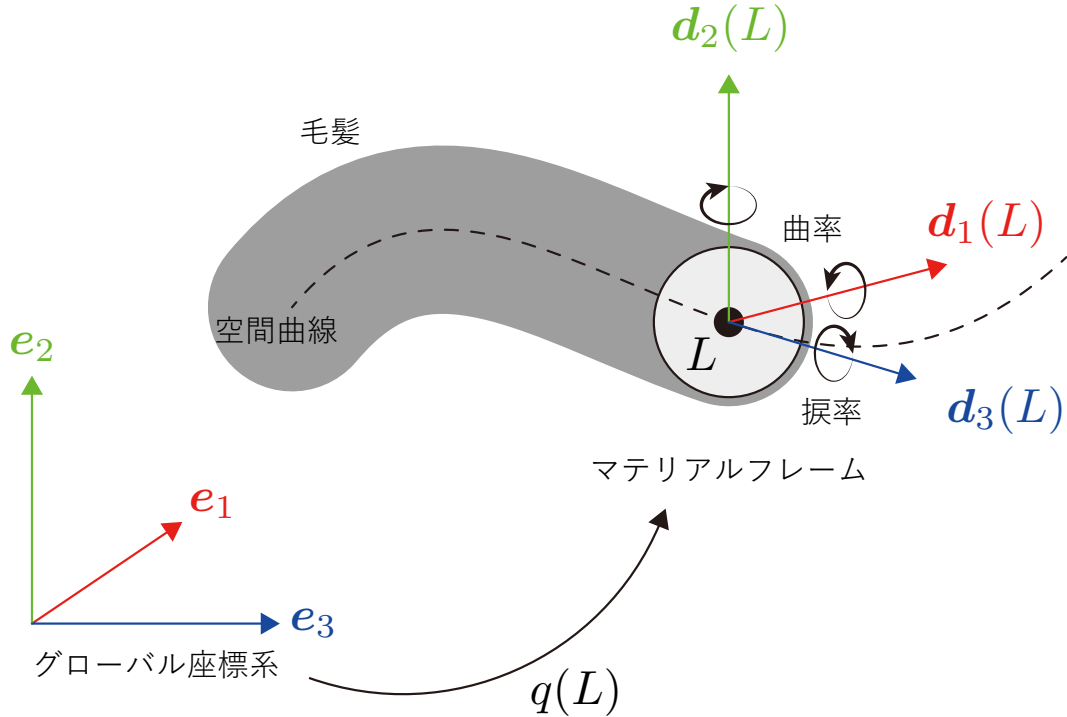


図 3.1: Cosserat ロッドモデルによる毛髪

本節では毛髪のモデル化に使用する Cosserat ロッドモデルとその離散化について述べる．毛髪は極めて細い (約 $0.08[\text{mm}]$) ロッド状の物体であり，太さを無視すると3次元空間上の連続的な空間曲線としてとらえることが可能である．Cosserat ロッドモデルでは空間曲線上の始点からの弧長を $L \in [0, L^{\max}]$ としたとき， L を変数とした曲線の位置を，関数 $\mathbf{p}(L)$ を定義することで表している．しかし曲線の位置情報だけではロッドがねじれたときの捩率を表すことができないため，図 3.1 に示すように L において局所座標系 (マテリアルフレーム) を定義している．マテリアルフレームは3次元空間におけるグローバル座標系を回転し

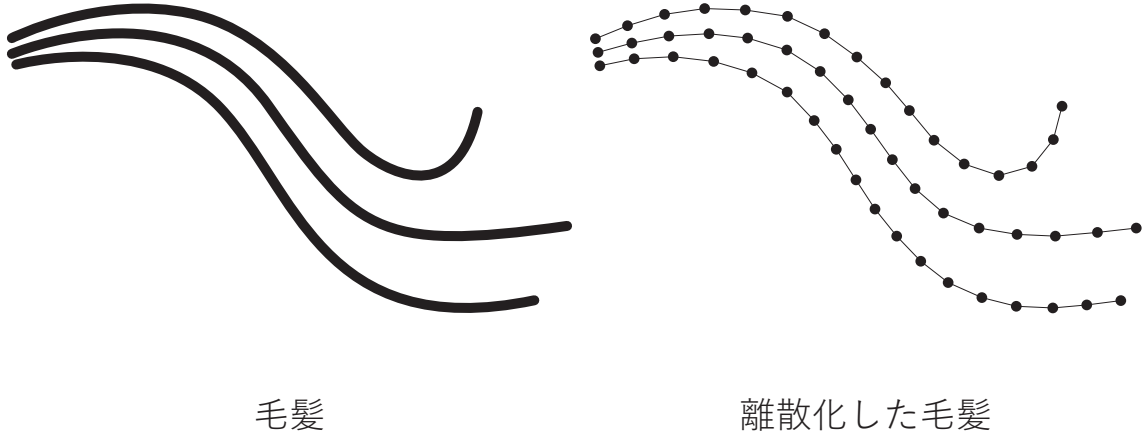


図 3.2: 毛髪の離散化

たものであり、以下の式を用いて求めることができる。

$$\{\mathbf{d}_1(L), \mathbf{d}_2(L), \mathbf{d}_3(L)\} = \{\mathbf{R}(\mathbf{e}_1, q(L)), \mathbf{R}(\mathbf{e}_2, q(L)), \mathbf{R}(\mathbf{e}_3, q(L))\} \quad (3.1)$$

$$\mathbf{R}(\mathbf{r}, q) = \Im(qr\bar{q}) \quad (3.2)$$

ここで、 $\{\mathbf{d}_1(L), \mathbf{d}_2(L), \mathbf{d}_3(L)\}$ は L におけるマテリアルフレームを構成する基底ベクトルの集合、四元数 $q(L)$ は L における曲線の姿勢を表す回転であり、 $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ はシミュレーション空間のグローバル座標系を構成する基底ベクトル、 \mathbf{R} はベクトルを四元数で回転させる関数である。また、 \bar{q} は q の共役四元数、 \Im は四元数のベクトル部を意味しており、四元数をスカラー部 \Re とベクトル部 \Im を用いて $(\Re|\Im)$ と表記すると、 r は $(0|\mathbf{r})$ となる四元数である。

マテリアルフレームは L の値によって変化していくが、その変化量は空間曲線の曲率と捩率、すなわち毛髪の形状 (どれだけ曲がっていてねじれているのか) を意味する。そして曲率と捩率をそれぞれ曲線の従法線ベクトルと接ベクトルにかけて足すことで定義される曲線の変化量を Darboux ベクトルと呼ぶ。本研究では 3 次元の回転を四元数で表すことにより計算を高速化した手法 [11] を用いており、以下に示す式でマテリアルフレームの姿勢を表す四元数 q から Darboux ベクトルを計算する。

$$\mathbf{\Omega} = \Im(\mathbf{\Omega}^*) \quad (3.3)$$

$$\mathbf{\Omega}^* = 2\bar{q} \frac{dq}{dL} \quad (3.4)$$

ここで、 $\mathbf{\Omega}$ は q における Darboux ベクトルである。求めた $\mathbf{\Omega}$ は各マテリアルフレームに対応する Darboux ベクトルであるので、 $\mathbf{\Omega}$ の各要素はそれぞれ $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ 軸に関する姿勢の変化量となる。

ここまでは連続した曲線に対するマテリアルフレームを考えてきたが、実際にコンピュータ上でシミュレーションを行う場合は有限のデータしか扱うことができない。そこで図 3.2 右に示すように有限個の計算点で空間曲線を離散化する。毛髪の接線方向の姿勢を定義するためには二つの計算点情報が必要であるため、姿勢を表す四元数 q は毛髪上の各計算点の間に格納する。この姿勢情報 q から近似的に計算点上の Darboux ベクトルを計算することができる。図 3.3 に示すように、計算点 \mathbf{p} の周囲に距離が $l_{q_1 q_2}$ 離れた姿勢 q_1 と q_2 があつた場合、 \mathbf{p} における姿勢は $\frac{1}{2}(q_1 + q_2)$ と線形補間できる。また同様に姿勢の変化量を $\frac{1}{l_{q_1 q_2}}(q_2 - q_1)$

と近似すると，式 (3.3)(3.4) より \mathbf{p} における Darboux ベクトル Ω を計算する以下の式が得られる．

$$\Omega = \Im(\Omega^*) \quad (3.5)$$

$$\Omega^* \approx \frac{1}{l_{q_1 q_2}}(\bar{q}_1 + \bar{q}_2)(q_2 - q_1) \quad (3.6)$$

$$= \frac{2}{l_{q_1 q_2}}\bar{q}_1 q_2 \quad (3.7)$$

なお毛髪の始点と終点に位置する計算点には Darboux ベクトルを定義しない．上記で離散化された位置・姿勢・Darboux ベクトルを用いることで，次の 3.2 節で述べる制約条件が定義できる．

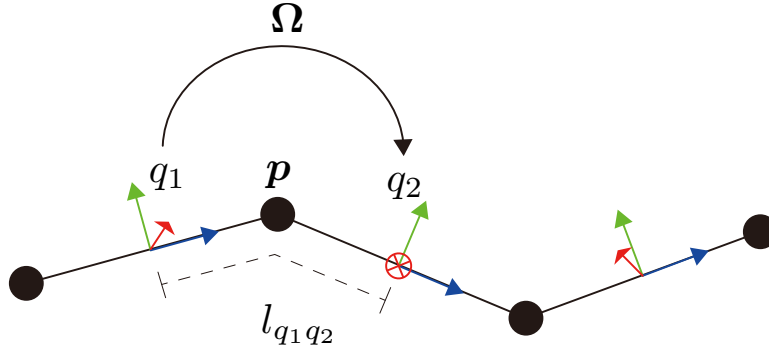


図 3.3: 離散化した毛髪の Darboux ベクトル

3.2 弾性変形シミュレーション

まずは毛髪の弾性変形運動をシミュレーションするため，3.1 節で離散化した毛髪に対して 2.2 節で述べた位置ベース法の制約条件を適用する．本節では各制約条件と，その制約を満たすための修正式について述べる．

2.2.2 項で挙げた制約条件の例では，式 (2.12) によって位置修正後に速度 \mathbf{v} を更新していた．本節で述べる制約条件では，位置 \mathbf{p} だけではなく毛髪を構成する辺の姿勢 q も含んでいるため，速度に加えて角速度 ω も更新する．角速度を更新する式を以下に示す．

$$\omega \leftarrow \Im\left(\frac{2q\bar{q}^{\text{prev}}}{\Delta t}\right) \quad (3.8)$$

ここで， q^{prev} は 1 ステップ前の姿勢である．

3.2.1 伸びと剪断の制約条件

毛髪は接線方向に対する繋がりが非常に強く，ハサミで切断したり重量物を吊り下げたりしない限り伸びや剪断が起きることはない．ここでは伸びと剪断が発生しないような制約条件を定義する．

まずは毛髪の伸びについて考える．図 3.4 左に示すように，有限個の計算点で離散化された毛髪では毛髪が伸びると計算点間の距離 (辺の長さ) が変動するため，距離を一定に保つ

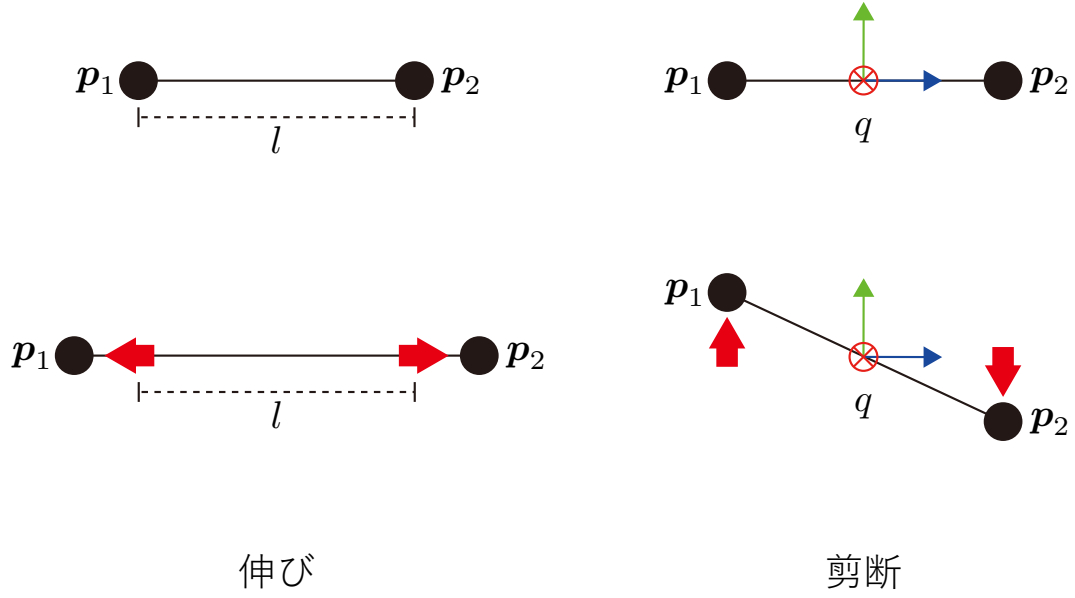


図 3.4: 離散化した毛髪の伸び・剪断

制約が考えられる．これは 2.2.2 項で挙げた制約の例と同様に設定できるため，式 (3.9) の制約条件 C^{stretch} が得られる．

$$C^{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2) = \|\mathbf{p}_2 - \mathbf{p}_1\| - l = 0 \quad (3.9)$$

ここで， l は計算点間の距離である．

次に毛髪の剪断について考える．図 3.4 右に示すように，実際の計算点同士の位置関係が計算点間の姿勢 q と一致しない場合は剪断が起きていると考えられる．計算点位置から毛髪の接線方向を示す単位ベクトルは $\frac{\mathbf{p}_2 - \mathbf{p}_1}{\|\mathbf{p}_2 - \mathbf{p}_1\|}$ であり，本研究では毛髪の接線方向をマテリアルフレームの第 3 要素 \mathbf{d}_3 として定義しているため，剪断に関する制約条件 C^{shear} が式 (3.10) のように定義できる．

$$C^{\text{shear}}(\mathbf{p}_1, \mathbf{p}_2, q) = \frac{\mathbf{p}_2 - \mathbf{p}_1}{\|\mathbf{p}_2 - \mathbf{p}_1\|} - \mathbf{R}(\mathbf{e}_3, q) = \mathbf{0} \quad (3.10)$$

ここで， $\mathbf{R}(\mathbf{e}_3, q)$ は基底座標系の第 3 要素 \mathbf{e}_3 を姿勢を表す四元数 q で回転させたベクトルであり， q が示すマテリアルフレームの第 3 要素 \mathbf{d}_3 と同義である．

伸びの制約と剪断の制約は式 (3.9) を式 (3.10) に代入してまとめることができる．まとめた制約条件 $C^{\text{stretch-shear}}$ を式 (3.11) に示す．

$$C^{\text{stretch-shear}}(\mathbf{p}_1, \mathbf{p}_2, q) = \frac{\mathbf{p}_2 - \mathbf{p}_1}{l} - \mathbf{R}(\mathbf{e}_3, q) = \mathbf{0} \quad (3.11)$$

2.2.2 項の位置ベース法の理論に基づき，式 (2.10)(2.11) に式 (3.11) を代入することで，伸びと剪断に関する制約条件 $C^{\text{stretch-shear}}$ を満たすための位置修正式が以下のように導出さ

れる.

$$\Delta \mathbf{p}_1 = + \frac{w_{\mathbf{p}_1} l}{w_{\mathbf{p}_1} + w_{\mathbf{p}_2} + 4w_q l^2} \gamma \quad (3.12)$$

$$\Delta \mathbf{p}_2 = - \frac{w_{\mathbf{p}_2} l}{w_{\mathbf{p}_1} + w_{\mathbf{p}_2} + 4w_q l^2} \gamma \quad (3.13)$$

$$\Delta q = + \frac{w_q l^2}{w_{\mathbf{p}_1} + w_{\mathbf{p}_2} + 4w_q l^2} \gamma q \bar{e}_3 \quad (3.14)$$

$$\gamma = \frac{\mathbf{p}_2 - \mathbf{p}_1}{l} - \mathbf{R}(\mathbf{e}_3, q) \quad (3.15)$$

四元数をスカラー部 \Re とベクトル部 \Im を用いて $(\Re|\Im)$ と表記すると, $\mathbf{e}_3 = (0|\mathbf{e}_3)$ であり, $\gamma = (0|\gamma)$ である. また, $w_{\mathbf{p}_1}, w_{\mathbf{p}_2}, w_q$ はそれぞれ $\mathbf{p}_1, \mathbf{p}_2, q$ の重み (質量の逆数) である. 姿勢 q は本来質量を持たないが, 重みを定義し姿勢の変化しやすさを表すパラメータとして扱うことで計算を近似し簡略化している. 本研究では毛髪内部の密度が一定であるとして, $\mathbf{p}_1, \mathbf{p}_2$ の質量を l , q の質量を $0.005l$ とした.

3.2.2 曲げとねじれの制約条件

毛髪は曲げやねじれといった変形に対して元の形状に戻ろうとする内力が発生するため, 弾性体としてふるまう. この弾性変形を位置ベース法では, 初期形状と現在の形状が一致するように制約をかけることで実現する. ただし 2.2.2 項で説明したように, 制約条件は全体の回転と平行移動とは独立している. よって制約で発生する変形もこれらとは独立した運動であるため, 基底座標系からの絶対的な毛髪の姿勢を表す q をそのまま使うことはできない. そこで 3.1 節で述べた, 毛髪を表す線分要素間での曲がり量・ねじれ量を表す Darboux ベクトルを使用する. Darboux ベクトルを用いて毛髪が維持する形状を Ω^{rest} , 現在の形状を Ω とすると, 現在の毛髪の変形量は $\Omega - \Omega^{\text{rest}}$ と計算され, この値を $\mathbf{0}$ にする制約を定義することができる. また, 四元数を用いて回転を表す場合は四元数 q と $-q$ は同じ回転を表すため, 実際の制約条件では修正の方向が逆回転とならないように符号係数 s を導入して修正量が少ない回転方向を選択する. 以下に曲げとねじれに対する制約条件 $\mathbf{C}^{\text{bend-twist}}$ を示す.

$$\mathbf{C}^{\text{bend-twist}}(q_1, q_2) = \Im(\Omega^* - s\Omega^{\text{rest}}) = \Omega - s\Omega^{\text{rest}} = \mathbf{0} \quad (3.16)$$

$$s = \begin{cases} +1 & \text{if } \|\Omega^* - \Omega^{\text{rest}}\|^2 < \|\Omega^* + \Omega^{\text{rest}}\|^2 \\ -1 & \text{if } \|\Omega^* - \Omega^{\text{rest}}\|^2 > \|\Omega^* + \Omega^{\text{rest}}\|^2 \end{cases} \quad (3.17)$$

ここで, q_1, q_2 はそれぞれ隣り合った辺が持つ姿勢情報, $\Omega, \Omega^{\text{rest}}$ はそれぞれ毛髪の現在の形状と維持しようとする形状を表す Darboux ベクトルであり, 辺と辺の間の計算点で定義される. 四元数 Ω^* はスカラー部を含めた Darboux ベクトルである ($\Omega = \Im(\Omega^*)$). 今回のように離散的に空間曲線を扱う場合は Ω^* のスカラー部が 0 にならない場合がある. 式 (3.17) で符号係数 s を計算するとき Ω で比較を行っても問題ない場合もあるが, 曲線が大きく変形した際にシミュレーションが不安定になったため, 本研究では式 (3.17) においてスカラー部を残した Ω^* を用いて比較を行う. 式 (3.7) にあるように本来 Darboux ベクトルを求めるには $\frac{2}{l_{q_1 q_2}}$ をかける必要があるが, $\|\Omega - s\Omega^{\text{rest}}\|$ を最小化する場合係数を無視することができるため, 実際には以下の式で Ω^* を計算する.

$$\Omega^* = \bar{q}_1 q_2 \quad (3.18)$$

伸びと剪断の制約と同様に，ここまでの制約条件から曲げとねじれに関する修正式を求めると以下のように導出される．

$$\Delta q_1 = +\frac{Kw_{q_1}}{w_{q_1} + w_{q_2}}q_2(\Omega - s\Omega^{\text{rest}}) \quad (3.19)$$

$$\Delta q_2 = -\frac{Kw_{q_2}}{w_{q_1} + w_{q_2}}q_1(\Omega - s\Omega^{\text{rest}}) \quad (3.20)$$

$$s = \begin{cases} +1 & \text{if } \|\Omega^* - \Omega^{\text{rest}}\|^2 < \|\Omega^* + \Omega^{\text{rest}}\|^2 \\ -1 & \text{if } \|\Omega^* - \Omega^{\text{rest}}\|^2 > \|\Omega^* + \Omega^{\text{rest}}\|^2 \end{cases} \quad (3.21)$$

ここで， $K \in [0, 1]$ は毛髪の弾性を表す係数， w_{q_1}, w_{q_2} はそれぞれ q_1, q_2 の重み（質量の逆数）， Ω はスカラー部を 0，ベクトル部を Darboux ベクトル Ω とした四元数 $(0|\Omega)$ である．本研究では頂点間の距離が場所によって異なる場合を考慮するため， q_1, q_2 の質量をそれぞれの姿勢が位置する辺の長さに設定した．弾性を表す係数 K は，0 に近づけるとより柔らかい毛髪となり 1 に近づけるとより固い毛髪となるが，位置修正の反復回数によって弾性が変化するため，ヤング率のような物性値と直接の因果関係はない．一方でこの問題を解決した手法 [12] も提案されているが，本研究では修正式の制御性の高さやシンプルさから Kugelstadt らの手法 [11] を使用している．

3.3 塑性変形シミュレーション

3.2 節までで毛髪の弾性変形について述べた．しかしながら，毛髪は弾性変形だけでなく塑性変形もする．毛髪の塑性変形は，毛髪表面に付着している液体などの外力を除くと毛髪内部に存在する無数の側鎖結合が原因だと考えられている [1]．ここでは，3.2 節までの弾性変形シミュレーションに，この側鎖結合を考慮した塑性変形を組み込む手法 [2] と，側鎖結合の結合率の変動の仕方について述べる．

3.3.1 側鎖結合の種類

毛髪はそのほとんどがケラチンと呼ばれるタンパク質である．ケラチンはアミノ酸がペプチド結合によって繋がれたポリペプチドによって構成されており，内部には無数の主鎖結合（ペプチド結合）と側鎖結合が存在している [1]．毛髪の塑性変形は図 3.5 に示すように，側鎖結合が一度切断されてから毛髪が変形し，その後に再結合することで変形後の形状を記憶することで発生する（図 3.5 は簡素化した側鎖の動きのイメージであり実際はこのように一列に並んでいない）．例えば寝癖は側鎖結合の一種である水素結合が切断・再結合を繰り返すことで発生する現象だと考えられており，ヘアドライヤーやヘアアイロンを用いた髪型のスタイリングもこの水素結合の変化を利用している．また，パーマメントウェーブのように変形を長期にわたり維持する場合は，パーマメントウェーブ剤によって人為的にジスルフィド結合などより強固な結合を切断し変形させている．厳密には毛髪の断面形状や結合の分布，キューティクル（毛髪の外皮部）のダメージなど側鎖結合以外にも様々な塑性変形を起こす要因が考えられるが，身近で起こる毛髪の塑性変形を再現するにはこの側鎖結合の影響を考慮することが重要である．本研究では側鎖結合による塑性変形を位置ベース法に組み込んだ手法 [2] を用いる．以下に毛髪内部に存在する主な側鎖結合の種類を挙げる．

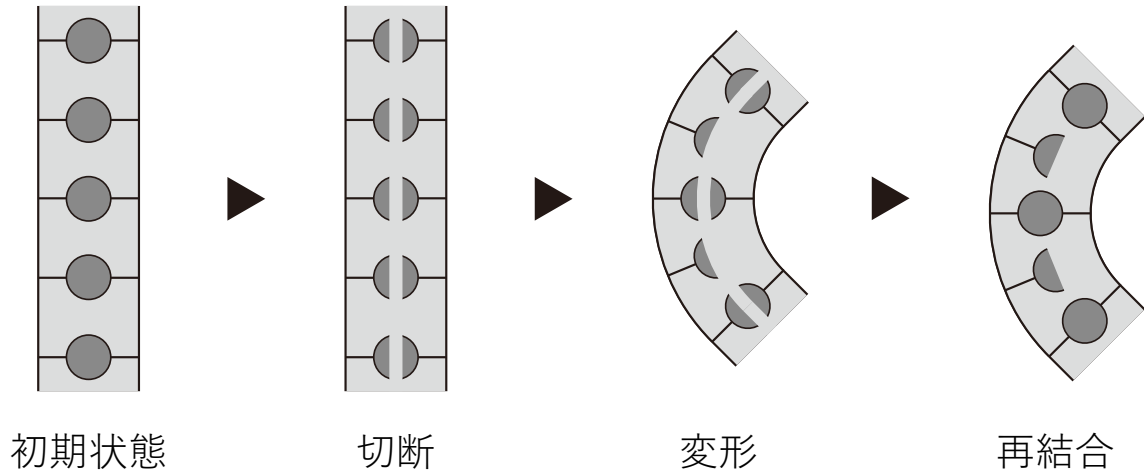


図 3.5: 側鎖結合によって塑性変形が発生する流れ

水素結合

最も弱い結合で、水や熱の影響で切断・再結合する。毛髪を濡らしたり温めると柔らかくなる現象は主にこの水素結合が原因である。

イオン結合

電氣的に結びついた結合であり、水素イオン指数 pH が変化することで切断・再結合する。パーマメントウェーブやシャンプー、整髪料などにより変化する場合がある。

ジスルフィド結合

硫黄を含むタンパク質に含まれる非常に強い結合。パーマメントウェーブはこの結合を切断し再結合することで長期間の形状の維持を行っている。

ペプチド結合

ポリペプチドの主鎖結合と同じ結合。基本的に強い力が加わらない限り切断されることはなく、本研究では切れないものとして扱う。

3.3.2 側鎖結合の影響の適用

毛髪内では無数の側鎖結合が複雑に存在しているため、個々の結合についてシミュレーションすることは計算時間やメモリの観点から難しい。そこで今回用いる手法 [2] では側鎖結合の種類ごとに、十分に結合が進んだ状態の側鎖が毛髪に与える影響力 (重み w) と、その状態を 1 とした結合率 $b \in [0, 1]$ を定義した上で近似的にシミュレーションを行う。また、このとき結合率と側鎖が記憶している形状を保とうとする力は線形な関係であると仮定する。例えばある側鎖結合 j が形状 Ω_j を記憶しており、 j の結合率が b_j だった場合、 j によって維持される毛髪の形状 Ω_j^{rest} は以下の式 (3.22) で計算される。

$$\Omega_j^{\text{rest}} = \Omega + b_j(\Omega_j - \Omega) \quad (3.22)$$

ここで、 Ω は現在の毛髪の形状である。複数種の結合が存在している場合はそれぞれの重み w を用いて重み付き平均をとる。結合率が 0 に近づくほど維持される形状が現在の形状へと近づいていき、結果的に弾性が低く柔らかい状態になる。結合率は式 (3.19)(3.20) にあ

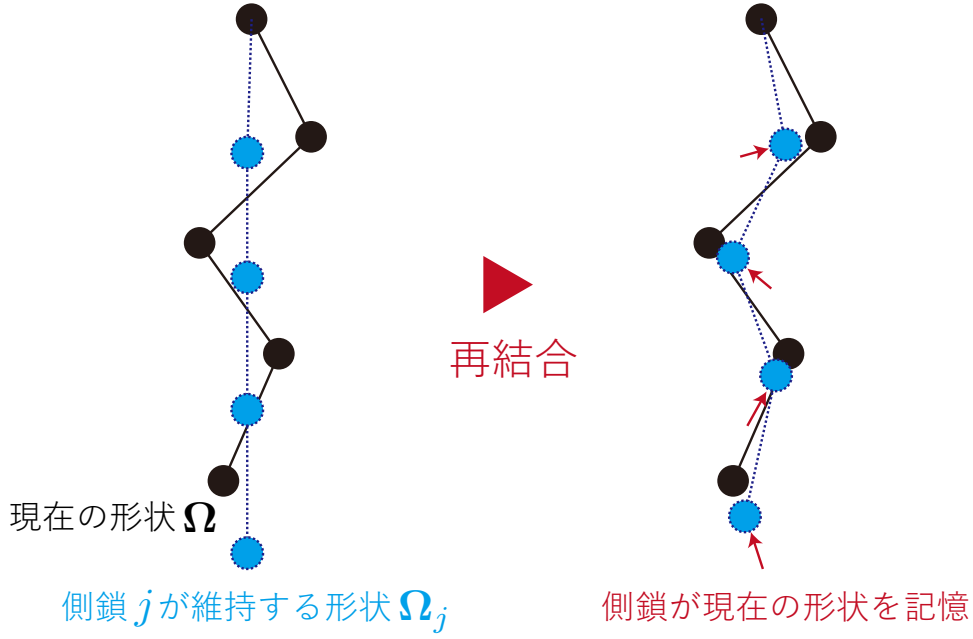


図 3.6: 再結合することで側鎖結合が現在の形状を記憶する概念図

る弾性を表す係数 K と似た働きをするが、 K は結合率が変化しなかった場合の毛髪全体の弾性を制御するパラメータであるのに対して、ここでの結合率は各種結合が切れたことが原因で発生する弾性の変化を再現する。

実際に複数種の結合を用いて毛髪が維持する形状を求める式を以下に示す。

$$\Omega^{*\text{rest}} = \frac{\sum_{j \in B} w_j (\Omega^* + b_j (s \Omega_j^* - \Omega^*))}{\sum_{j \in B} w_j} \quad (3.23)$$

$$s = \begin{cases} +1 & \text{if } \|\Omega^* - \Omega_j^*\|^2 < \|\Omega^* + \Omega_j^*\|^2 \\ -1 & \text{if } \|\Omega^* - \Omega_j^*\|^2 > \|\Omega^* + \Omega_j^*\|^2 \end{cases} \quad (3.24)$$

四元数 Ω^* は 3.2.2 項で用いたものと同様にスカラー部を含めた Darboux ベクトルである ($\Omega = \mathfrak{S}(\Omega^*)$)。曲げ・ねじれ制約の修正式の式 (3.21) でも Ω^* を用いて演算するため、Darboux ベクトルの格納や計算は基本的に Ω^* を用いて行う。この式で Ω^* は現在の毛髪の形状、 w_j, b_j, Ω_j^* はそれぞれ j 種目の側鎖結合の重み、結合率、維持する形状であり、 B は考慮する側鎖結合の集合である。求めた形状 $\Omega^{*\text{rest}}$ は 3.2.2 項で示した曲げ・ねじれ制約の計算式に代入する。なお、 Ω^* の計算には式 (3.18) を用いている。

3.3.3 側鎖結合が保持する形状の更新

側鎖結合は水や熱などの外的な要因で切断・再結合を繰り返している。結合が切断 (結合率が低下) したときは切断された結合が自由に動けるようになり弾性が低下するだけであるが、再結合 (結合率が上昇) すると図 3.6 に示すように、新たに結合した側鎖が現在の形状を記憶する。ある側鎖結合 j において、もともと結合していた割合を b_j 、新たに再結合し

た割合を Δb_j とすると、側鎖が記憶する形状 Ω_j^* を以下の式で更新する．

$$\Omega_j^* \leftarrow \begin{cases} \frac{b_j s \Omega_j^* + \Delta b_j \Omega^*}{b_j + \Delta b_j} & \text{if } \Delta b_j > 0 \\ \Omega_j^* & \text{otherwise} \end{cases} \quad (3.25)$$

$$s = \begin{cases} +1 & \text{if } \|\Omega^* - \Omega_j^*\|^2 < \|\Omega^* + \Omega_j^*\|^2 \\ -1 & \text{if } \|\Omega^* - \Omega_j^*\|^2 > \|\Omega^* + \Omega_j^*\|^2 \end{cases} \quad (3.26)$$

本研究では結合率 b_j の変動を主に温度を元に計算する．次の 3.3.4 項で結合率の計算方法について述べる．

3.3.4 結合率の変動

従来の研究 [2] では側鎖結合の結合率の変動を手動で設定している．例えばヘアドライヤーを模したシーンでは、風が通っていると仮定した位置に存在する毛髪 of 結合率を単純に時間に比例して上昇させており、周囲の毛髪や温度の拡散の影響は考慮していない．しかしながら、実際の毛髪は風に当たる表面側や毛先側は比較的速く乾燥するが、内側や毛根付近の毛髪は乾燥させるのに時間を要する．加えて温度も毛髪全体で一様ではなく結合率の変動も一様ではない．

提案手法では、側鎖結合の中でも最も切断・再結合しやすく、日常的な塑性変形に関係していると考えられる水素結合に焦点をあて、空気との相互作用によって結合率を変動させる．計算には 4.2.3 項で説明する熱拡散の計算によって得られた温度と、本物の毛髪を用いた実験による実験データを参考に近似した関数を用いる．また、各種結合の影響力や温度と結合率の相関はユーザが自由に定義することも可能であり、寝癖のつきやすい毛髪やパーマメントウェーブのかかりにくい毛髪など、性質の違いを表現することが可能である．

水素結合が温度によって変動する主な要因は、熱によって結合が切断される現象と、温度が上がり毛髪内部の水分量が減少したことによって結合が切断される現象に大きく分けられる．以下にそれぞれの現象についての計算方法を述べる．

熱による結合率の変動

毛髪内部に存在する水素結合は、毛髪が熱されたときに切断される．切断された水素結合はその剛性を失い、毛髪が維持する形状は水素結合の影響を受けなくなる．再び毛髪を冷ますと水素結合は再結合し、その時点の毛髪の形状を記憶する．身近な例では、ヘアアイロンはこの原理を応用して髪型をスタイリングしている [1]．ここでは、毛髪の強度 (降伏点) と水素結合の結合率には線形な関係があると仮定し、本物の毛髪を用いた実験から結合率を推定する．

Yu ら [22] は人間の毛髪を用いて実験により温度と毛髪の機械的性質の関係を確かめた．実験は水の中に浸透させた状態の毛髪を用いて行われ、異なる温度帯 (20, 40, 60, 80[°C]) でそれぞれ引張応力を測定した．実験は水の中で行われているため、水分の蒸散による水素結合の変動とは関係なく熱の影響で毛髪の強度が変動していると考えられる．本研究では [22] の実験データから、温度 T [°C] と水素結合の結合率の関係を表す関数 $f(T)$ を最小二乗法を用いて指数関数で近似的に設計する．以下に実際に近似した関数 $f(T)$ を示す．

$$f(T) = 3.83 \exp(-0.092T) + 0.388 \quad (3.27)$$

また、平温時 (20[°C]) の環境における降伏点を 1 に正規化した実験結果のプロットと、式 (3.27) による近似曲線を図 3.7 に示す．厳密にはこの剛性率の変化に関わっているの

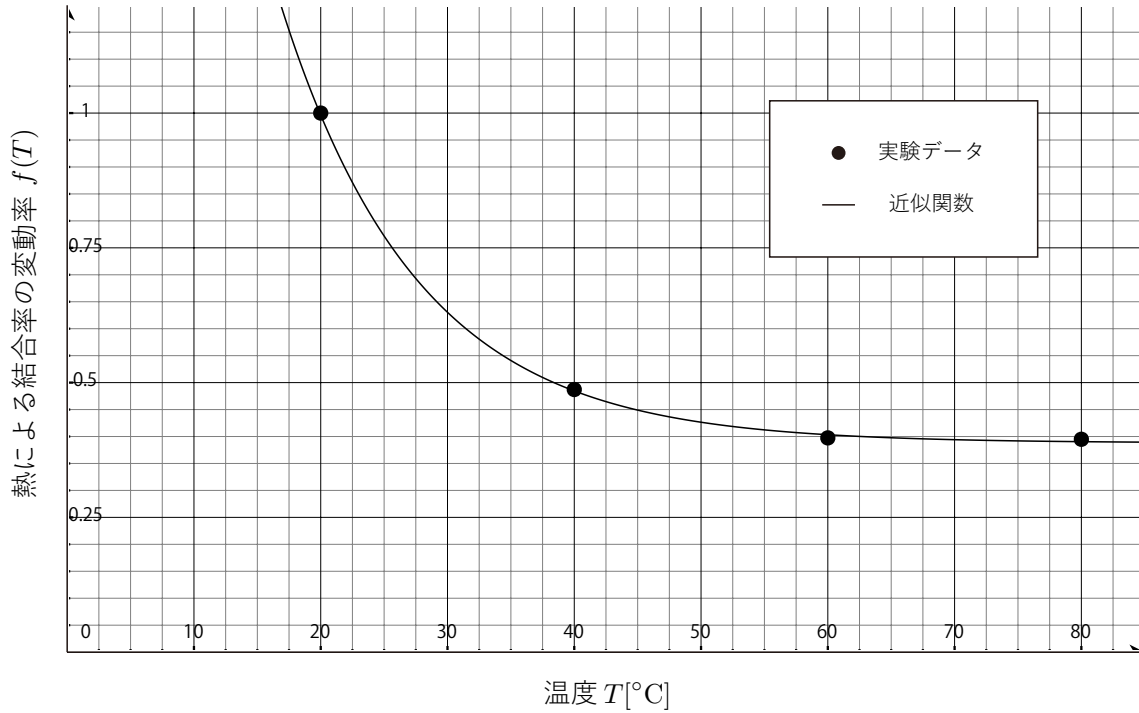


図 3.7: 温度と水素結合の結合率の関係を表す関数 $f(T)$

は水素結合だけではないが，ここでは問題の複雑化を避けるため近似曲線式 (3.27) をそのまま用いるものとする．なお，結合率 b の範囲は $[0, 1]$ であるため， $f(T)$ が $[0, 1]$ の範囲を超えてしまうときはクランプする．

毛髪内水分の蒸散による結合率の変動

毛髪を水で濡らし，ヘアドライヤーなどで乾燥させると寝癖を直すことができる．この方法は Water-Set と呼ばれており [1]，毛髪内部の水分含有量を上下することで水素結合を切断・再結合させる．毛髪が濡れたまま就寝すると寝癖が付きやすいのも同様に，長時間にわたり毛髪内部の水分が空気中に蒸散し乾燥することによって水素結合が再結合されることが原因である．本研究では水分含有量と水素結合の結合率に線形な関係があると仮定して結合率の変化を推定する．

Lee ら [23] は人間の毛髪を一般的なヘアドライヤーと同じ温度帯である $65[^\circ\text{C}]$ で加熱を続ける実験を行った．その結果毛髪の水分含有量が減少し，30 分程度で水分含有量の減少が収まった．実験結果からは水分含有量の減少に伴って乾燥速度が減少していることが確認できるため，水分含有量と加熱時間の関係を指数関数を用いて近似することができる．本研究では毛髪の水分含有率の熱による変動を，指数関数を解とする以下の常微分方程式で近似することを提案する．

$$\frac{dM}{dt} = -\epsilon^{\text{dry}} M \max(0, T - T^{\text{rest}}) \quad (3.28)$$

ここで， $M \in [0, 1]$ は毛髪の水分含有率 (十分乾燥している場合を 0，十分水を浸透させた状態を 1 とする)， ϵ^{dry} は乾燥速度を調整するパラメータ， $T[^\circ\text{C}]$ は現在の温度， T^{rest} は水分の蒸散が少ない状態の温度， \max は引数のうち大きな値を返す関数である．本研究では， T^{rest} を $20[^\circ\text{C}]$ に設定し，短い時間のシーン内で提案手法による挙動の変化

を効果的に示すために ϵ^{dry} を 0.01 とした．Lee らの実験データにフィットするように ϵ^{dry} を設定する場合は 7×10^{-5} 付近の値となる．

求めた M と $f(T)$ から，最終的に水素結合の結合率 b_{hyd} は以下の式で計算される．

$$b_{\text{hyd}} = (1 - M)f(T) \quad (3.29)$$

以上で述べた現象の他にも，周囲の空気の湿度が変動することや，整髪料・パーマメントウェーブ剤・シャンプーなどによって水素結合は切断と再結合を繰り返す．ただし提案手法では毛髪スタイリングに影響が大きい熱と水分の蒸散のみを扱うものとする．

本研究では実験データから近似的に結合率変動の関数を設計したが，毛髪の性質は個人差によって大きく異なる．この場合には関数 $f(T)$ や水分含有率 M の変動式などをユーザーが自由に定義することで対応可能である．

第4章 流体シミュレーションによる相互作用

本章では毛髪 of 相互作用計算を流体シミュレーションを用いて行う手法について述べる．まず，4.1 節では流体を扱う上で重要な Navier-Stokes 方程式について説明し，本研究で扱う数値計算手法について説明する．4.2 節では本研究で用いる流体シミュレーション手法である PBF について詳しく説明する．PBF もまた毛髪 of 弾塑性変形と同様に，位置ベース法 (2.2 節) を用いて非圧縮性流体をシミュレーションする手法である．最後に 4.3 節では，空気と毛髪 of 相互作用計算において空気の流れに発生した渦を検出し，制御する手法について述べる．

4.1 流体シミュレーション

4.1.1 流体の支配方程式

流体の支配方程式には Navier-Stokes 方程式を用いる．本研究では毛髪とその外部流体である空気に焦点をあててシミュレーションを行う．このときシミュレーションを単純化するために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．以下に粘性非圧縮性流体の Navier-Stokes 方程式を示す．

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

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = -\nabla P + \mu \nabla^2 \mathbf{v} + \mathbf{F}^{\text{ext}} \quad (4.2)$$

ここで， \mathbf{v} は速度， ρ は密度， P は圧力， μ は粘性係数， \mathbf{F}^{ext} は外力を表している．式 (4.1) は質量保存式であり，非圧縮性流体では速度場の発散が 0 になる (流入する流体量と流出する流体量が等しくなる) ことを意味している．式 (4.2) は運動量保存式である．左辺の $\frac{\partial \mathbf{v}}{\partial t}$ は流体の加速度， $(\mathbf{v} \cdot \nabla) \mathbf{v}$ は流れに沿って流体が運ばれていくことを表す移流項である．右辺は第 1 項からそれぞれ圧力項，粘性拡散項，外力項である．圧力項は流体が圧力の低い方向に流れていく性質を表す．粘性拡散項は流体の粘性を表しており，粘性係数 μ が高いとクリームのように粘り気が強い流体となる．外力項は重力や浮力などの外力を表している．

Navier-Stokes 方程式の解を解析的に求めることは難しく，数値演算によって近似的に計算する．数値的に解く場合は計算空間の離散化が必要であり，離散化手法は大きくオイラー的手法とラグランジュ的手法の二つに分けられる．オイラー的手法の代表例として格子法がある (図 4.1 左)．格子法は計算空間を格子状に分割し，格子を基準として空間に固定された計算点を用いることで物理量を離散化している．一方でラグランジュ的手法の代表例としては粒子法があげられる (図 4.1 右)．粒子法は流体を粒子の集まりとして離散化し，流体の流れに沿って移動する各粒子を計算点として物理量を離散化している．

本研究では，高速かつ安定に非圧縮性流体をシミュレーション可能で並列化にも適しているという観点から，流体シミュレーションには PBF を用いる．PBF は粒子法 of 一種である SPH を拡張し，圧力項 of 計算を行う代わりに位置ベース法に基づく密度制約をかけた手法であり，4.2 節で詳しく説明する．また次の 4.1.2 項では，PBF を使用するにあたって必要な技術である，SPH による流体 of 離散化手法について述べる．

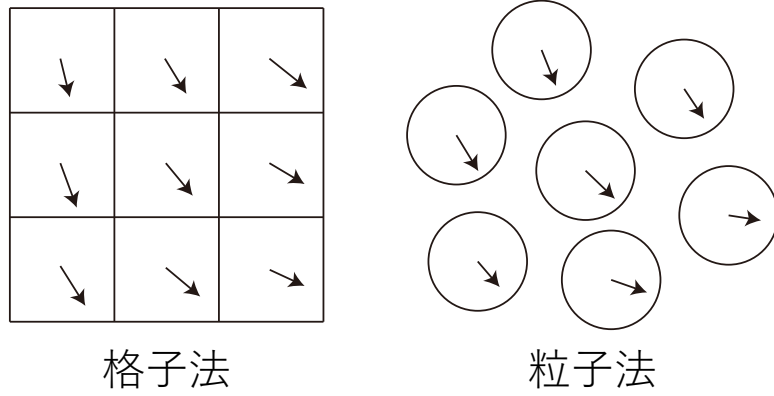


図 4.1: 格子法と粒子法による離散化

4.1.2 SPH を用いた流体の離散化

粒子法はラグランジュ的手法による離散化手法である．空間に計算点を固定して流体を離散化するオイラー的手法とは異なり，流体を流体の流れに合わせて動く粒子の集まりとしてとらえることで離散化する．また，あらかじめ空間に計算点を配置しておく必要がないため計算空間が限定されることはなく，問題を個々の粒子に分割できるため並列化にも適しているという利点がある．粒子法にも様々な種類があるが，ここでは本研究で用いる PBF を使用するにあたって必要な技術である SPH(Smoothed Particle Hydrodynamics)[17] の離散化手法について説明する．

粒子は流れに沿って自由に動き回ることができるため，計算点の位置が規則正しく並んでいない．そこで SPH では図 4.2 に示すように，粒子間距離に応じた重み関数 W を用いて計算対象の周囲に存在する粒子から物理量を計算する．また，各粒子は粒子半径とは別に有効半径 h を持っており， h 内に存在する近傍粒子から影響を受ける．以下に粒子 i の物理量 ϕ_i を求める式を示す．

$$\phi_i = \sum_j \frac{m_j}{\rho_j} \phi_j W(\mathbf{p}_i - \mathbf{p}_j, h) \quad (4.3)$$

ここで， j は近傍粒子 (近傍粒子は計算対象の粒子 i 自身も含む)， m は粒子の質量， ρ は粒子の密度， \mathbf{p} は粒子位置である．式 (4.3) と同様にして物理量の勾配 $\nabla \phi_i$ やラプラシアン $\nabla^2 \phi_i$ は，重み関数の導関数を用いて以下のように求めることができる．

$$\nabla \phi_i = \sum_j \frac{m_j}{\rho_j} \phi_j \nabla W(\mathbf{p}_i - \mathbf{p}_j, h) \quad (4.4)$$

$$\nabla^2 \phi_i = \sum_j \frac{m_j}{\rho_j} \phi_j \nabla^2 W(\mathbf{p}_i - \mathbf{p}_j, h) \quad (4.5)$$

SPH では重み関数 W のことをカーネル関数と呼ぶ．カーネル関数は自由に設定することができるが，原点を中心として単調減少する関数が望ましく，その積分値が 1 になる必要がある ($\int_V W dV = 1$)．また，有効半径 h の範囲外では 0 となるようにカーネル関数を設計する．このように局所的な範囲内の影響だけで物理量が決定されることをコンパクトサポートと呼び，計算量を抑制するためにもこの性質は重要である．Müller らは物理量計算とその勾配，ラプラシアンの計算にそれぞれ異なるカーネル関数を使うことを提案しており [24]，本研究では [24] のカーネル関数を用いる．以下にそれぞれのカーネル関数について説明する．

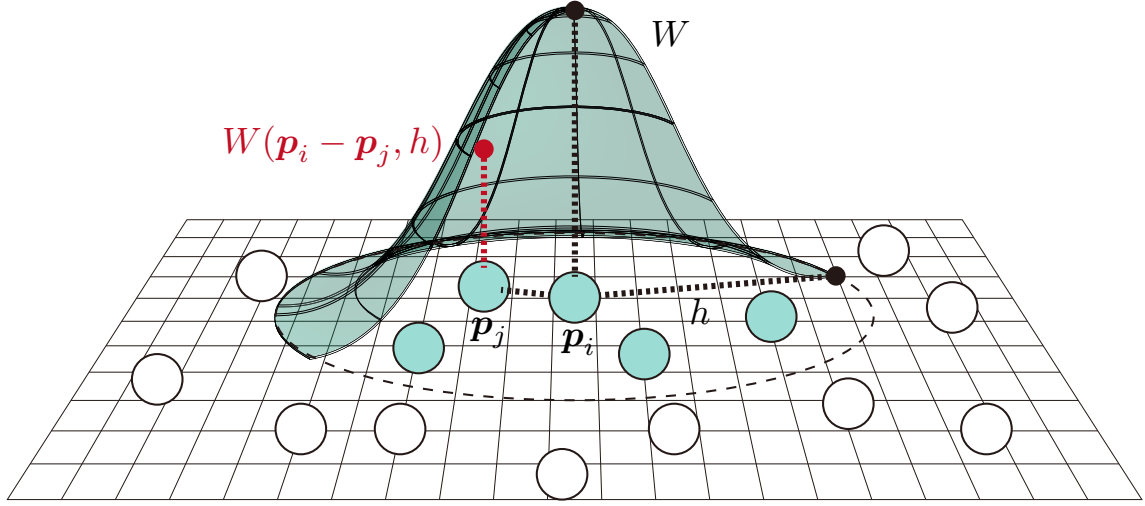


図 4.2: 近傍粒子と重み関数による物理量の離散化

Poly6 カーネル

物理量を計算する際に用いるカーネル関数. \mathbf{r} が入力ベクトルとすると, 関数内には $\|\mathbf{r}\|$ の 2 乗項しか含まれていないため, $\|\mathbf{r}\|$ を求めるために平方根を計算する必要がない. 以下に Poly6 カーネルを示す.

$$W(\mathbf{r}, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - \|\mathbf{r}\|^2)^3 & \text{if } 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \text{otherwise} \end{cases} \quad (4.6)$$

Spiky カーネル

物理量の勾配を計算する際に用いるカーネル関数. Poly6 カーネルでは粒子間距離が 0 となったときの勾配値が 0 になるため, 圧力項の計算で粒子が集まってしまう問題があるが, Spiky カーネルはその問題を解決する. ただし Spiky カーネルは粒子間距離が 0 に近づくと勾配値が発散するため, 計算を安定化するために本研究では距離が 0.001[m] 未満の近傍粒子は無視する. 以下に Spiky カーネルの勾配を示す.

$$\nabla W(\mathbf{r}, h) = -\frac{45}{\pi h^6} \begin{cases} (h - \|\mathbf{r}\|)^2 \frac{\mathbf{r}}{\|\mathbf{r}\|} & \text{if } 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \text{otherwise} \end{cases} \quad (4.7)$$

Viscosity カーネル

物理量のラプラシアンを計算する際に用いるカーネル関数. カーネル関数のラプラシアンが線形かつ常に正の値をとることでシミュレーションの安定性を高めている. 以下に Viscosity カーネルのラプラシアンを示す.

$$\nabla^2 W(\mathbf{r}, h) = \frac{45}{\pi h^6} \begin{cases} (h - \|\mathbf{r}\|) & \text{if } 0 \leq \|\mathbf{r}\| \leq h \\ 0 & \text{otherwise} \end{cases} \quad (4.8)$$

4.1.3 空気粒子と毛髪粒子

毛髪同士の相互作用と毛髪と空気の相互作用は本来であれば異なる物理現象であるが、それぞれを個別に計算しているとシミュレーションが複雑化し計算量も増大する．そこで提案手法では、毛髪同士の相互作用と毛髪と空気の相互作用を一度に計算する手法 [3] を拡張する．本研究で用いる流体シミュレーションは流体を連続体として扱っている．毛髪が衝突しあうことで髪全体のボリュームを維持する現象が非圧縮性流体の体積保存に似ていることから、毛髪も連続体としてモデル化することで毛髪同士の衝突現象も流体シミュレーションによって再現する．

連続体のモデル化には毛髪粒子を用いて行う．毛髪粒子は図 4.3 に示すように、3.1 節で離散化した各計算点上に配置する．毛髪粒子同士が流体シミュレーションによって相互作用を行うことで、毛髪同士の相互作用が計算される．また、空気と毛髪の相互作用計算を行う場合は毛髪粒子の周囲に空気粒子 (図 4.3 の水色で示した粒子) を配置するだけでよい．

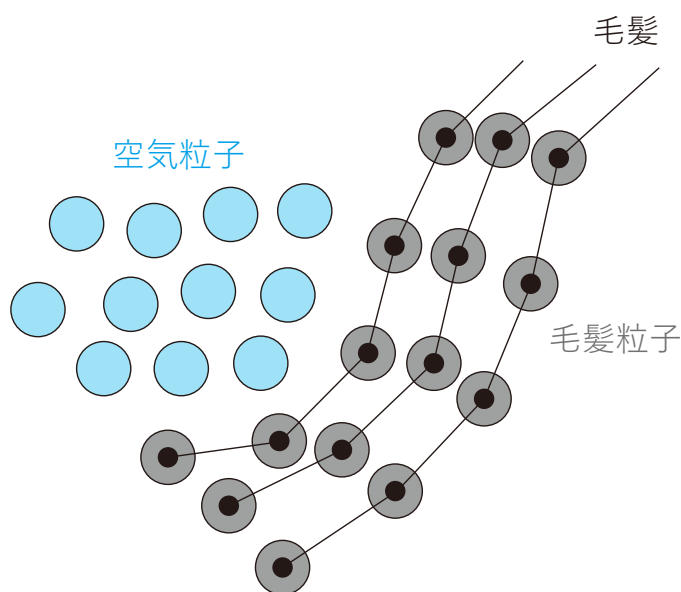


図 4.3: 空気粒子と毛髪粒子

毛髪同士のすり抜けの防止や空気の侵入など、厳密に計算を行う場合は非常に小さな粒子を毛髪曲線状に密に並べ、毛髪同士の間にある小さな隙間に入り込む微小な空気粒子も考慮する必要がある．しかしながら、計算量の観点から毛髪粒子は離散化された計算点上のみに配置し、毛髪粒子は毛髪周辺の空気も含んでいるものとしてシミュレーションを行う．また本論文では、静止密度や粒子質量、有効半径などのパラメータは特に明記しない限り毛髪粒子と空気粒子で同じものを使う．本研究のシーンにおいて、計算点間の距離を $0.01[\text{m}]$ 、粒子の有効半径 h を $0.0093[\text{m}]$ にして実験を行った結果、毛髪のすり抜けは確認できなかった．なお、シミュレーションの高速化のために空気粒子は風が発生している場所のみに配置している．そのため毛髪上の計算点には外力として重力を加えるが、空気粒子には重力は加えないものとする．

4.2 PBF による流体シミュレーション

PBF(Position Based Fluids)[19] は、粒子法である SPH を拡張し、圧力項を計算する代わりに位置ベース法による密度制約をかけることで流体をシミュレーションする手法である。PBF を含む粒子法では粒子 (計算点) 自体が流体の流れに沿って移動していくため、式 (4.2) の移流項を明示的に計算する必要はない。さらに、粒子自体が流体そのものを表しているため質量保存式 (4.1) も計算する必要はない。この性質から、4.1.1 項で述べた Navier-Stokes 方程式を以下のように簡略化することができる。

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla P + \mu \nabla^2 \mathbf{v} + \mathbf{F}^{\text{ext}} \quad (4.9)$$

ここで、 $\frac{D\mathbf{v}}{Dt}$ は移流項を含んでいる速度の実質微分であり、粒子そのものの加速度を表している。

本節では、式 (4.9) の圧力項を解く代わりに用いる制約条件とその修正式を次の 4.2.1 項に、粘性拡散項を扱う計算式を 4.2.2 項に示す。また、Navier-Stokes 方程式の中にはないが、毛髪の変形を扱うために必要となる温度の拡散を計算する手法について 4.2.3 項で説明する。

4.2.1 密度の制約条件

PBF では Navier-Stokes 方程式の圧力項を計算する代わりに、位置ベース法を用いて密度制約条件をかける。流体になにも力が加わっていないときの静止密度を ρ^{rest} とすると、粒子 i の密度 ρ_i を一定に保つ制約条件 C_i^{dens} を以下のように定義できる。

$$C_i^{\text{dens}}(\mathbf{p}_1, \dots, \mathbf{p}_n) = \frac{\rho_i}{\rho^{\text{rest}}} - 1 = 0 \quad (4.10)$$

ここで、 \mathbf{p} は粒子位置、 ρ は現在の密度である。粒子 i の現在の密度 ρ_i を求めるには SPH の物理量計算式 (4.3) を用いて以下のように計算できる。

$$\rho_i = \sum_j m_j W(\mathbf{p}_i - \mathbf{p}_j, h) \quad (4.11)$$

ここで、 j は近傍粒子、 m は粒子質量、 W はカーネル関数、 h は有効半径である。

ここからは 2.2 節で説明した位置ベース法の理論にしたがって制約条件 C_i^{dens} の位置修正式を導出する。PBF ではシミュレーションを安定化させるため、位置ベース法にユーザが定義可能な緩和パラメータ ϵ^{dens} を導入している。これにより、位置ベース法の式 (2.9) から以下のようにスケーリング係数 λ_i^{dens} を求める式が得られる。

$$\lambda_i^{\text{dens}} = -\frac{C_i^{\text{dens}}(\mathbf{p}_1, \dots, \mathbf{p}_n)}{\sum_j \|\nabla_{\mathbf{p}_j} C_i^{\text{dens}}(\mathbf{p}_1, \dots, \mathbf{p}_n)\|^2 + \epsilon^{\text{dens}}} \quad (4.12)$$

式 (4.12) より、 λ_i^{dens} を求めるためには C_i^{dens} と $\nabla_{\mathbf{p}_j} C_i^{\text{dens}}$ を計算する必要がある。 C_i^{dens} は式 (4.10) に式 (4.11) で求めた密度を代入することで計算できる。また、物理量の勾配は単純にカーネル関数の勾配を取ればいいので、 $\nabla_{\mathbf{p}_j} C_i^{\text{dens}}$ は以下の式で計算できる。

$$\nabla_{\mathbf{p}_j} C_i^{\text{dens}}(\mathbf{p}_1, \dots, \mathbf{p}_n) = \frac{1}{\rho^{\text{rest}}} \sum_k m_k \nabla_{\mathbf{p}_j} W(\mathbf{p}_i - \mathbf{p}_k, h) \quad (4.13)$$

ここで、 k は j と同様に近傍粒子のインデックスである。式 (4.13) において、 W は \mathbf{p}_i と \mathbf{p}_k に関する関数であるため、 $j = i$ だった場合は k の値に関係なく W は \mathbf{p}_j を含んだ関数にな

る．一方で $j \neq i$ だった場合， W が \mathbf{p}_j を含んだ関数である条件が $j = k$ のときのみとなる．したがって $j \neq k$ のときに W を \mathbf{p}_j で偏微分すると 0 になる．以上より，式 (4.13) は実際は以下に示す式 (4.14) で計算される．

$$\nabla_{\mathbf{p}_j} C_i^{\text{dens}}(\mathbf{p}_1, \dots, \mathbf{p}_n) = \frac{1}{\rho_{\text{rest}}} \begin{cases} \sum_k m_k \nabla_{\mathbf{p}_j} W(\mathbf{p}_i - \mathbf{p}_k, h) & \text{if } j = i \\ -m_j \nabla_{\mathbf{p}_j} W(\mathbf{p}_i - \mathbf{p}_j, h) & \text{if } j \neq i \end{cases} \quad (4.14)$$

最後に位置ベース法の式 (2.8) から，求めたスケーリング係数を用いて位置修正式を以下のように導出できる．

$$\Delta \mathbf{p}_i = \frac{1}{\rho_{\text{rest}}} \sum_j m_j (\lambda_i^{\text{dens}} + \lambda_j^{\text{dens}}) \nabla W(\mathbf{p}_i - \mathbf{p}_j, h) \quad (4.15)$$

式 (4.11) に示した SPH に基づく粒子の密度推定では，有効半径内の近傍粒子の数が少なすぎると密度が過少評価され，粒子が集まってしまうという問題点がある．この問題を解決するために，密度が低くなったことによって発生した負の値を持つ C^{dens} を 0 にクランプする．また，本研究では緩和パラメータを $\epsilon^{\text{dens}} = 1.0 \times 10^{-6}$ に設定している．

4.2.2 XSPH 人工粘性

流体はその形状を完全に保たないわけではなく，変形に対しての抵抗力を持っている．例えば空気は自在に形状を変化させるが，クリームは空気と比較するとその形状が変形しにくい．この性質を粘性と呼び，クリームと空気と比較するとクリームの方が粘性が高いと言える．粘性を持つ流体を粘性流体，粘性が全くない流体を非粘性流体と呼ぶが，ここでは空気と毛髪を粘性流体として扱っているため，この性質を表す粘性拡散項の計算が必要である．

SPH では粘性拡散項はカーネル関数のラプラシアンを用いて以下のように力を計算する．

$$\mathbf{F}_i^{\text{visc}} = \mu \sum_j \frac{m_j}{\rho_j} (\mathbf{v}_j - \mathbf{v}_i) \nabla^2 W(\mathbf{p}_i - \mathbf{p}_j, h) \quad (4.16)$$

ここで， \mathbf{F}^{visc} は流体の粘性によって働く力， μ は流体の粘性係数， \mathbf{v} は速度である．しかしながら，この手法では粘性係数 μ が大きいとシミュレーションが不安定になるという問題がある．

そこで提案手法では XSPH 人工粘性 [25] を用いて粘性拡散項の計算を行う．XSPH 人工粘性は調整パラメータ $\epsilon^{\text{visc}} \in [0, 1]$ を用いて人工的に粘性をかけ，シミュレーション中に発生したノイズを減衰させる目的で提案された．物理的な内力を計算しているわけではないため粘性係数のような物性値と対応付けることができないが，比較的高い粘性でも安定して計算を行うことが可能である．以下に XSPH 人工粘性による速度更新式を示す．

$$\mathbf{v}_i \leftarrow \mathbf{v}_i + \sum_j \epsilon^{\text{visc}} \frac{m_j}{\rho_j} (\mathbf{v}_j - \mathbf{v}_i) \nabla W(\mathbf{p}_i - \mathbf{p}_j, h) \quad (4.17)$$

$$\epsilon^{\text{visc}} = \min(\epsilon_i^{\text{visc}}, \epsilon_j^{\text{visc}}) \quad (4.18)$$

ここで， ϵ^{visc} は粘性の調整パラメータ， $\min(\epsilon_i^{\text{visc}}, \epsilon_j^{\text{visc}})$ は i と j で粘性パラメータが異なったときに小さい値を使用することを意味している．2.1 節の図 2.1 で示したように XSPH 人工粘性は，重力や 4.3 節で述べる乱流シミュレーションによって加えられた力による速度変化の影響を全て計算した後に行う．

4.2.3 熱拡散

熱は3.3節で述べたように毛髪の変形に強く影響する．そのため提案手法では，流体を通して熱が伝わる現象を計算する．本研究で扱うシーンでは図4.4に示すように，熱拡散と対流によって熱が伝わる．ここでの熱拡散(図4.4左)は，静止している流体内部または

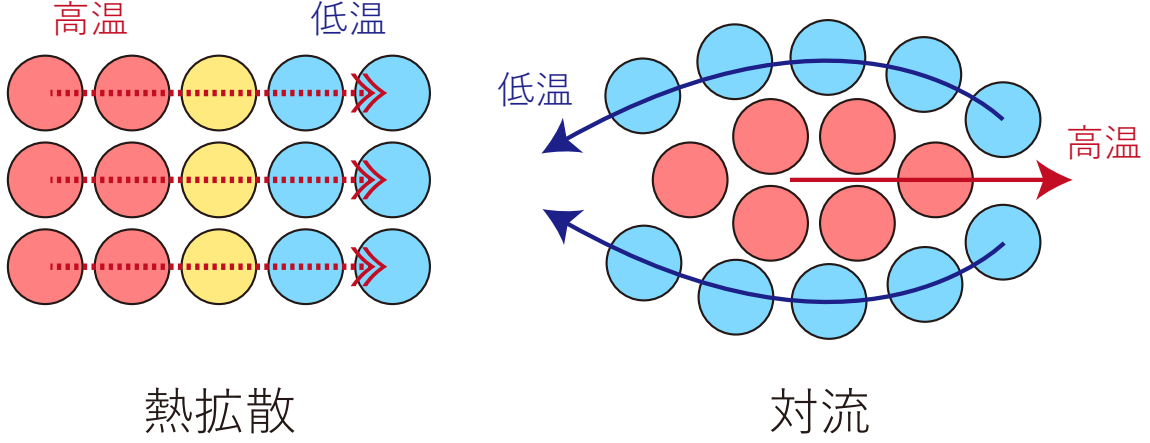


図 4.4: 熱拡散と対流

異なる流体同士が接触しているときに高温側から低温側に向かって熱が伝導する現象のことであり，対流(図4.4右)は熱を持った物質が流体の流れに沿って移流することで熱が伝達される現象のことである．粒子法では各粒子に物理量を持たせて粒子を移流させているため対流を明示的に計算する必要はなく，粒子間で熱が拡散する現象のみ計算すればよい．本研究では熱拡散の計算に，複数種類の流体の熱拡散を考慮可能な Hochstetter らの手法 [26] を用いる．以下に粒子 i において熱拡散による温度変化を計算する式を示す．

$$\frac{dT_i}{dt} = \frac{1}{\rho_i c_i} \sum_j \frac{m_j}{\rho_j} \frac{4\kappa_i \kappa_j}{\kappa_i + \kappa_j} (T_j - T_i) \nabla^2 W(\mathbf{p}_i - \mathbf{p}_j, h) \quad (4.19)$$

ここで， T は温度， c は比熱容量， κ は熱伝導係数である．なお，熱で流体の体積が膨張することによって密度変化や浮力など流体自体の挙動に変化が起きるが，本研究で行うシミュレーションでは扱わない．

4.3 渦を利用した毛髪のなびきの制御

毛髪と空気の相互作用によって起きる動きに、風になびく毛髪の運動がある．毛髪がなびくアニメーションを制作する方法の一例として、図 4.5 に示すように空気の塊のような円形の物体を流れに沿って動かし、この円に沿うように毛髪を変形させていく方法がある [4]．この方法では一度空気の塊によって持ち上げられた毛髪は、空気の塊の通過後に再び重力によって落ちることで上下運動を繰り返す．本研究では、現実世界でこの毛髪の上下運動を発生させている要因を仮定することで、毛髪のなびき運動を制御する手法を提案する．

毛髪は軽量のため風によってその動きが大きく変化するが、空気も同様に軽量であるために毛髪によってその流れが乱される．こうして発生した乱流は図 4.6 に示すように、大小さまざまな渦を形成して毛髪の動きに影響を与える．そこで提案手法では、空気の流れの中で発生した渦がこの上下運動に深く起因していると仮定し、その上で渦を制御することによって毛髪のなびき運動の制御を可能とする．4.3.1 項では渦を制御するために用いる乱流シミュレーション技術について説明し、4.3.2 項ではウェーブレット解析を応用することで任意の大きさの渦を強調する手法について述べる．

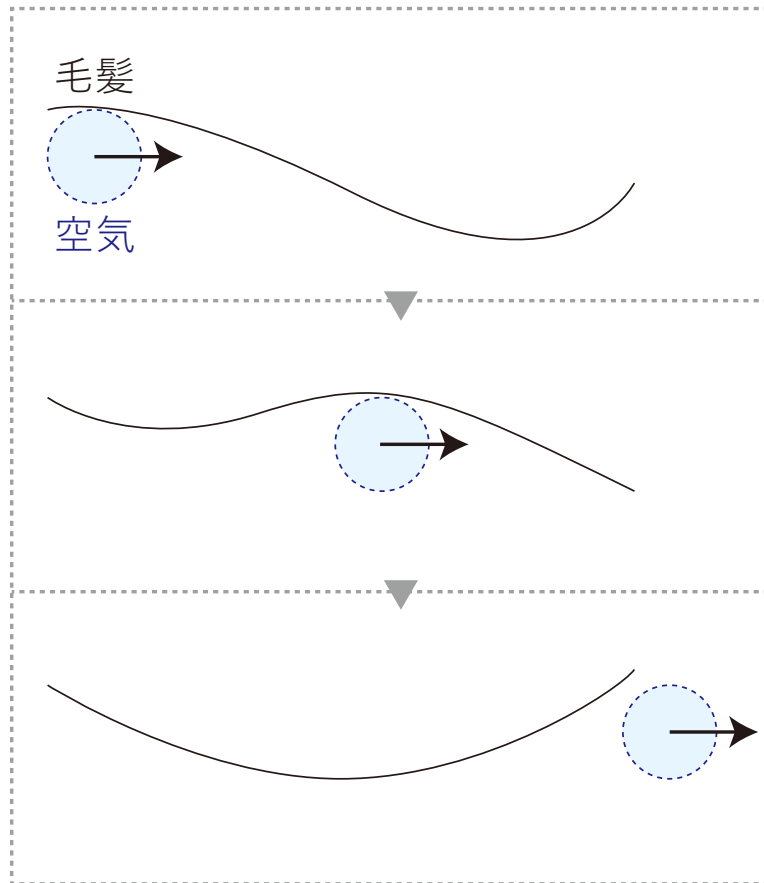


図 4.5: 毛髪が風になびくアニメーションを作成する手法の一例

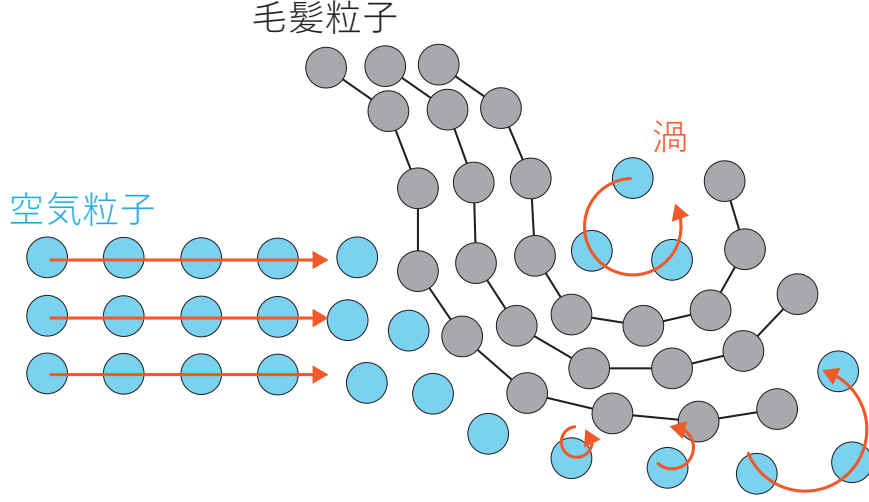


図 4.6: 毛髪と空気の相互作用で発生した渦を伴う乱流

4.3.1 Vorticity Confinement による渦の強調

提案手法では渦を制御するための乱流シミュレーションに、Vorticity Confinement[20] を拡張する。Vorticity Confinement は、シミュレーションの数値的散逸によって失われた渦のエネルギーを人工的に補填するために提案された手法である。ユーザが指定した任意の強さで渦を強調することができる高い制御性が特徴であり、渦を伴う乱流がその見た目に大きく影響を与える煙のシミュレーションや、水中を移動する泡の振動表現などに応用されている。本項ではこの Vorticity Confinement について説明する。

まず、流体の渦度 ζ を計算する。渦度は流体がその流れによって回転している状態を表すベクトルである。渦度のノルム $\|\zeta\|$ は渦の回転速度を表し、ベクトルの向きは渦の回転軸を表している。渦度は流体速度 \mathbf{v} から式 (4.20) を用いて求めることができる。

$$\zeta = \nabla \times \mathbf{v} \quad (4.20)$$

SPH による物理量の離散化 (4.1.2 項) から、粒子 i における渦度 ζ_i は以下の式で計算される。

$$\zeta_i = (\nabla \times \mathbf{v})_i = \sum_j \frac{m_j}{\rho_j} (\mathbf{v}_j - \mathbf{v}_i) \times \nabla \mathbf{p}_j W(\mathbf{p}_i - \mathbf{p}_j, h) \quad (4.21)$$

渦は中心部になるほど回転速度が高くなるため、求めた渦度のノルム $\|\zeta\|$ の勾配をとると渦の中心を示すベクトルが計算できる。以下に粒子 i において渦の中心を示す単位ベクトル \mathbf{N}_i の計算方法を示す。

$$\mathbf{N}_i = \frac{\boldsymbol{\eta}_i}{\|\boldsymbol{\eta}_i\|} \quad (4.22)$$

$$\boldsymbol{\eta}_i = (\nabla \|\zeta\|)_i = \sum_j \frac{m_j}{\rho_j} \|\zeta_j\| \nabla W(\mathbf{p}_i - \mathbf{p}_j, h) \quad (4.23)$$

粒子 i において渦の中心を示す単位ベクトル \mathbf{N}_i と渦度 ζ_i から、渦によって粒子 i が動いている向きを計算することができる。Vorticity Confinement では図 4.7 に示すように、この渦によって動いている向きに人工的に力を加えることによって渦を強調する。最終的に渦を強調するために粒子 i に加える力 $\mathbf{F}_i^{\text{vort}}$ を式 (4.24) に示す。

$$\mathbf{F}_i^{\text{vort}} = \rho_i \epsilon^{\text{vort}} (\mathbf{N}_i \times \zeta_i) \quad (4.24)$$

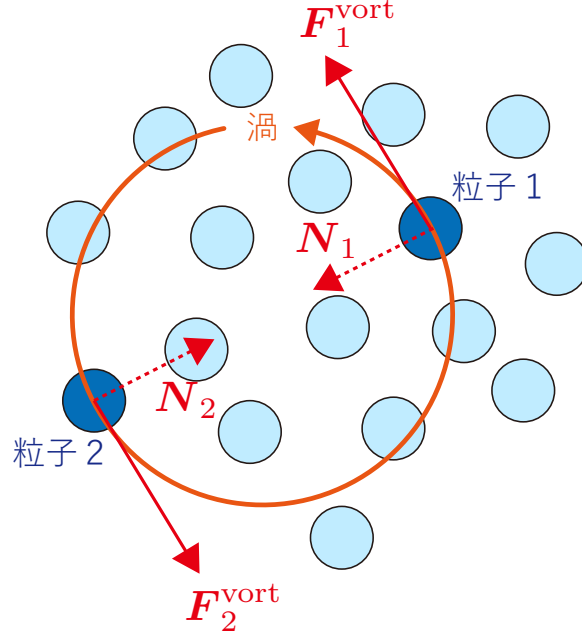


図 4.7: Vorticity Confinement による渦の強調

ここで、 ϵ^{vort} は渦を強調する力を調整するためのパラメータである。

ここまでの Vorticity Confinement では渦の大きさを考慮しておらず、流れの中で発生した大小さまざまな渦を大きさに関わらず全て強調してしまう。そのため制御が難しい問題がある。そこで、提案手法では新たに式 (4.24) に特定の大きさの渦の場所でのみ大きな値を持つエネルギー値 \hat{e} を導入することで、ユーザが指定した特定の大きさの渦を強調することを可能にする。以下に提案手法で用いる乱流シミュレーションの式を示す。

$$\mathbf{F}_i^{\text{vort}} = \rho_i \epsilon^{\text{vort}} \hat{e}_i(\alpha) (\mathbf{N}_i \times \boldsymbol{\zeta}_i) \quad (4.25)$$

ここで、 α は強調する渦の大きさ (渦の半径)、 $\hat{e}_i(\alpha)$ は粒子 i における大きさ α の渦のエネルギー値である。 \hat{e} を計算するために周波数解析手法の一つであるウェーブレット解析を用いる。ウェーブレット解析を用いた計算については次の 4.3.2 項で述べる。

4.3.2 ウェーブレット解析を用いた特定の大きさの渦の検出

4.3.1 項で述べた大きさ α の渦のエネルギー値 $\hat{e}(\alpha)$ をウェーブレット解析により計算する。本研究では渦の大きさを周波数として扱うので、時間的ではなく空間的な周波数特性を抽出する。ウェーブレット解析は周波数解析手法の一つであり、ウェーブレットと呼ばれる局所的な波を平行移動・拡大縮小して足し合わせることで任意の波形を表現する。同じく周波数解析手法であるフーリエ解析と比較すると、ウェーブレット解析は時間的 (本研究の場合は空間的) に定常的ではない波の解析に適している。提案手法では、[27] の粒子法によるウェーブレット変換手法を用いる。

大きさ α の渦を検出するために、まず流体の速度場 \mathbf{v} の周波数 α 成分 $\hat{\mathbf{v}}(\alpha)$ を求める。以下に速度場 \mathbf{v} の連続ウェーブレット変換式を示す。

$$\hat{\mathbf{v}}(\alpha, \beta) = \iiint_{-\infty}^{\infty} \mathbf{v} \bar{\psi}(\alpha, \mathbf{p} - \beta) dx dy dz \quad (4.26)$$

ここで、 α はウェーブレットの拡大縮小量 (検出する渦の大きさ), \mathbf{p} は位置 $(x, y, z)^T$, β はウェーブレットの平行移動量, $\bar{\psi}$ はウェーブレット関数の複素共役である. ただし, 本研究で用いるウェーブレット関数は虚数部を持たないので $\bar{\psi} = \psi$ になる. 周波数 α 成分を抽出するためには, α を固定したときに平行移動 β を連続的に変化させたウェーブレットと速度場の内積を計算する. 提案手法では粒子法によって速度場を離散化しているため, 粒子 i の周囲に存在する近傍粒子 j の速度 \mathbf{v}_j を用いて畳み込み積分を離散的に行う. 以下に粒子 i の位置 \mathbf{p}_i における速度の周波数 α 成分 $\hat{\mathbf{v}}_i(\alpha)$ を計算する式を示す.

$$\hat{\mathbf{v}}_i(\alpha) = \sum_j \mathbf{v}_j \bar{\psi}(\alpha, \mathbf{p}_i - \mathbf{p}_j) \quad (4.27)$$

式 (4.27) で求めた周波数成分 $\hat{\mathbf{v}}_i(\alpha)$ から以下に示す式 (4.28) で粒子 i における渦のエネルギー値 $\hat{e}_i(\alpha)$ を計算する.

$$\hat{e}_i(\alpha) = \frac{1}{2} m_i \|\hat{\mathbf{v}}_i(\alpha)\|^2 \quad (4.28)$$

ここで, m は粒子質量である. 以上より求めたエネルギー値を 4.3.1 項で述べた乱流シミュレーションの式 (4.25) に代入する. なお, 極めて高いエネルギー値を持つ粒子が現れるとシミュレーションが不安定になるため, ユーザが設定した最大のエネルギー値 \hat{e}^{\max} を超えた場合はクランプする. 本研究では $\hat{e}^{\max} = 3$ とした.

ウェーブレット関数 ψ には 3 次元に拡張したメキシカンハット関数を使用する. 図 4.8 に 1 次元のメキシカンハットウェーブレットを示し, 実際に用いる 3 次元のウェーブレット関数 ψ を式 (4.29) に示す.

$$\psi(\alpha, \mathbf{r}) = \frac{2}{\sqrt{15}\pi^{\frac{3}{4}}\alpha^{\frac{3}{2}}} \left(3 - \frac{\|\mathbf{r}\|^2}{\alpha^2} \right) \exp\left(-\frac{\|\mathbf{r}\|^2}{2\alpha^2}\right) \quad (4.29)$$

ここで, $\frac{2}{\sqrt{15}\pi^{\frac{3}{4}}\alpha^{\frac{3}{2}}}$ はスケール値 α が変動してもエネルギー値を一定にするために正規化する係数である ($\int_V |\psi|^2 dV = 1$). また, ψ はコンパクトサポートな関数ではない. そのため本来であれば全ての計算領域で畳み込みを行う必要があるが, 本研究では計算の高速化のため, 距離が 3α 以上離れた影響力の小さな粒子は計算しないものとした.

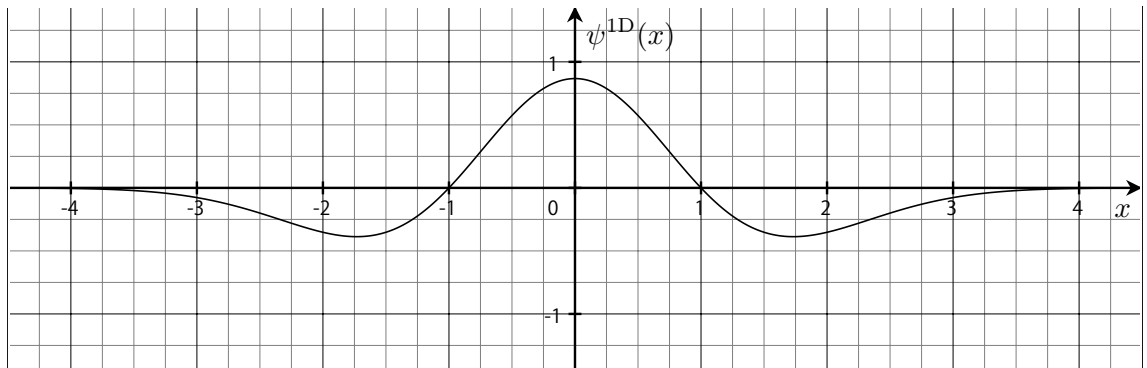


図 4.8: 1 次元のメキシカンハットウェーブレット ($\alpha = 1$)

第5章 実装

本章では、提案手法の実装について説明する．本研究では計算の高速化のため、並列計算が可能な GPU(Graphics Processing Unit) を用いて実装する．GPU はもともとグラフィックス処理を高速に行うために開発された演算装置であるが、その優れた並列処理能力をグラフィックス処理以外のより一般的な問題にも活用する技術 (GPGPU) が近年よく用いられている．GPU は数百から数千もの演算コアを有しており、非常に多数のデータに対して同じ処理をするような演算 (SIMD) を一度に実行する能力を持っている．提案手法で用いている粒子法は、問題をそれぞれの粒子に分割することが容易にできるため、GPU で実装することによる大幅な高速化が期待できる．本研究では GPGPU の実装に NVIDIA CUDA を使用した．

5.1 節では提案手法の並列化について、5.2 節では粒子法における近傍粒子探索を GPU で並列実行するための方法について述べ、5.3 節では本研究においてシミュレーションを安定化するために行った処理を説明する．最後に 5.4 節でレンダリング手法を示す．また本章の説明文では、CPU 側をホスト、GPU 側をデバイスと記述する．

5.1 並列化

5.1.1 ホストメモリとデバイスメモリ

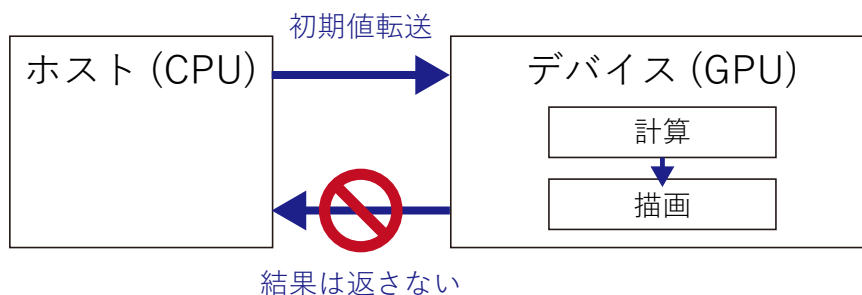


図 5.1: ホストとデバイス間のメモリ転送

ホストとデバイスは異なるメモリ空間 (ホストメモリとデバイスメモリ) を持っており、基本的にそれぞれが他方にアクセスすることはできない．そのため GPU を用いた計算フローでは、まず最初に計算に必要なデータをホストからデバイスに転送し、デバイスが計算を行った後、計算結果をデバイスからホストに転送するという流れでシミュレーションを行う．しかしながら、ホストとデバイス間でのデータのやりとりはレイテンシが大きく、シミュレーション速度の低下に直結する．ただし本研究の場合では、結果画像を画面に出力することが目的であるため求めた計算結果をホストに返す必要はない．そこで図 5.1 に示すように、デバイスで計算した結果は直接レンダリング計算を行うためのシェードプログラムに渡し、ホストに結果は返さない．また、初期化処理で計算に必要なデータを転送した後

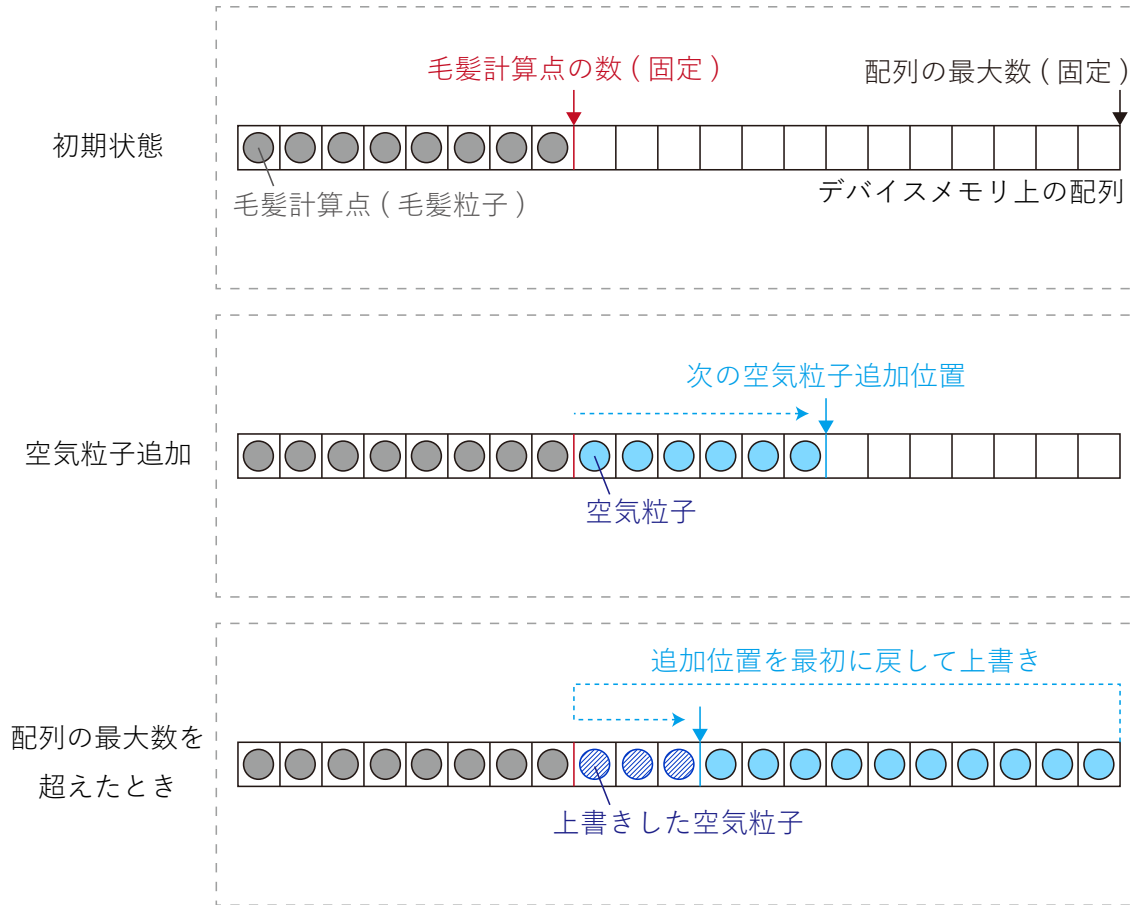


図 5.2: 静的配列を用いた各計算点・粒子のメモリ管理

は、ユーザがシミュレーション途中でパラメータを変更しない限りホストとデバイス間でメモリ転送を行わないものとする。

ここからはデバイスメモリの管理について説明する．本研究では，4.1.3 項で述べたように毛髪の計算点と毛髪粒子は同じ位置に配置しているため，管理する配列をまとめることができる．また，シミュレーション中に毛髪の計算点が増減することは考慮しておらず毛髪分の配列長は固定である．そこで，毛髪粒子・空気粒子を含む全ての計算点をまとめて静的配列で管理する．

図 5.2 に静的配列の管理の概念図を示す．まず，初期化時に十分に大きな静的配列をデバイスメモリに確保し，配列の先頭から毛髪の計算点 (毛髪粒子) の情報を格納する．毛髪計算点の数はシミュレーション中に変動しないことが前提のため，毛髪計算点の使う領域は固定である (図 5.2 上段)．次に空気粒子を追加する．空気粒子は風の発生している場所にしか配置しないため粒子数が変動する．そこで，毛髪計算点の直後から空気粒子の情報を格納し，粒子が増えるごとに次に追加する位置を遷移させていく (図 5.2 中段)．このとき PBF ソルバが処理する粒子の全体数は毛髪粒子数と空気粒子数の和であり，単純に配列の先頭から粒子の全体数分を処理すればよい．また，空気粒子は増え続けると配列の最大数を超える．その場合には次に追加する粒子位置を毛髪粒子の直後の位置に戻し，古い粒子から順番に新しい粒子の情報で上書きする (図 5.2 下段)．以上により，配列が確保するメモリサイズの変動や配列要素の並び替えなどの処理を行わないことによって，計算量が抑制される．

デバイスメモリにはグローバルメモリやシェアードメモリ，レジスタなど様々な種類があ

り、それぞれ制約や性質が異なる．例えば、グローバルメモリはメモリ容量が巨大で共有可能な範囲が広いが、読み書きの速度が遅く計算のボトルネックになりやすい性質を持つ．より高速な計算を可能とするためには、適切なメモリを活用して読み書き速度が遅いグローバルメモリへのアクセスを減らすなどの最適化が必要である．本研究では高速化のための細かなメモリ最適化は行わないが、シミュレーションを通して値が変動しない定数へのアクセスを高速化するためにコンスタントメモリを使用する．コンスタントメモリはデバイスメモリに存在する読み取り専用の領域であり、ホストからデータを転送することによって書き換え可能である．容量が小さくデバイスからの書き換えができない制約があるが、キャッシュが存在することによってグローバルメモリよりも高速に読み取りが行える．

5.1.2 CUDA カーネルによる問題の並列処理

デバイスで並列的に処理される関数のことを CUDA カーネルと呼ぶ．図 5.3 に示すように、CUDA カーネルを起動するとデバイス内で複数のスレッドが立ち上がり、それぞれが異なるデータ（本研究では粒子）に対して同じ処理を実行する．このとき、各スレッドはそれぞれが担当するメモリアドレスを計算し、データを決定するための ID を持っている．

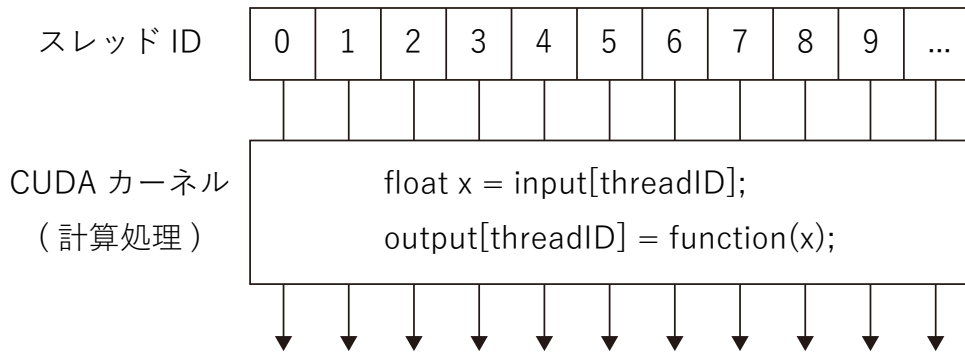


図 5.3: CUDA カーネルの実行

先述したように、粒子法を用いている場合は粒子ごとに問題を分割することができるため、粒子の数だけスレッドを立ち上げて並列的に処理を行えばよい．ただし、4.1.2 項で述べたように、各粒子の計算には近傍粒子の情報が必要となる．例えば、式 (4.21) において粒子 i での渦度を計算するために粒子 j の密度情報が必要だったとすると、渦度の計算を行う前に粒子 j の密度が計算されている必要がある．しかしながら、それぞれのスレッドは非同期で処理が進むため、粒子の計算に必要な他の近傍粒子の情報が既に計算済みであることが保証されない．この問題を防ぐためには、全ての粒子で計算を同期させる必要があるタイミングで CUDA カーネルを一度終了させる．これにより、デバイス内で同期を取ってから次に呼ばれた CUDA カーネルが処理される．

3 章で述べた毛髪弾塑性変形についても、基本的には粒子法と同様に各辺・各 Darboux ベクトルごとにスレッドを立てて処理することができる．ただしスレッドを立てる対象と、計算結果によって位置 \mathbf{p} や姿勢 \mathbf{q} の修正を行う対象が異なることに注意が必要である．ここでは 3.2.1 項で述べた伸びと剪断に対する制約条件の計算を例に説明する．制約条件 $C^{\text{stretch-shear}}$ は、計算点間の辺の伸び・剪断に関する制約なので、各辺で並列化することができる．しかしながら、この制約条件によって修正されるのは辺 q だけではなく、辺を構成している二つの計算点 $\mathbf{p}_1, \mathbf{p}_2$ も含まれる．これによって図 5.4 に示すように、隣りあう辺の修正する計算

点が重複してしまい、後に計算した結果で上書きされシミュレーション結果が不安定になる。この問題を解決するために、本研究では奇数番目の辺と偶数番目の辺に分割し、先に奇数番目の辺を処理した後で偶数番目の辺の処理を行っている。その他の対策では、計算点の情報を格納する配列を二つ用意し、重複した計算点の計算結果をそれぞれの配列に格納した後に計算結果の平均値をとるなど考えられる。しかし本研究ではメモリアクセス効率と収束速度の観点からこの方法は用いない。

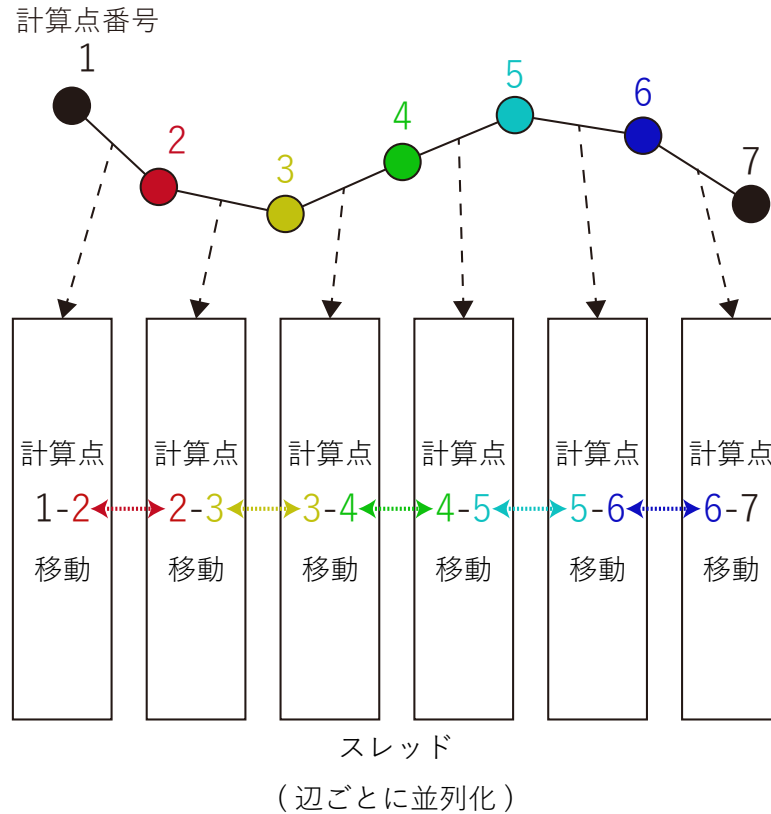


図 5.4: 各辺での並列化で起こる計算点位置修正の競合

5.2 近傍粒子探索

粒子法では近傍に存在する粒子の情報を用いて計算を行うが、シミュレーション空間に存在する粒子全てから全探索していると全粒子数の 2 乗のオーダーで計算量が増加してしまい、膨大な計算コストがかかる。そこで、より効率的に近傍粒子を探索する手法が提案されてきた。本研究では GPU 処理に最適化された近傍探索手法 [28] を用いる。

全体の流れとしては、まずシミュレーション空間を等間隔の格子 (グリッドセル) に区切り、次に各グリッドセル内に存在する粒子を登録する。そして近傍探索を行う際に、自身が存在しているグリッドセルとその隣接したグリッドセルに存在する粒子のみを探索することで高速化する。本研究では格子の間隔を粒子の有効半径 h に設定した。これにより粒子がグリッドセル内のどこにいたとしても、有効半径 h 内の近傍粒子はそのグリッドセルか隣接するグリッドセル (3 次元の場合は計 27 個) の中に存在する。図 5.5 に等間隔格子を用いた近傍探索の概念図を示す。

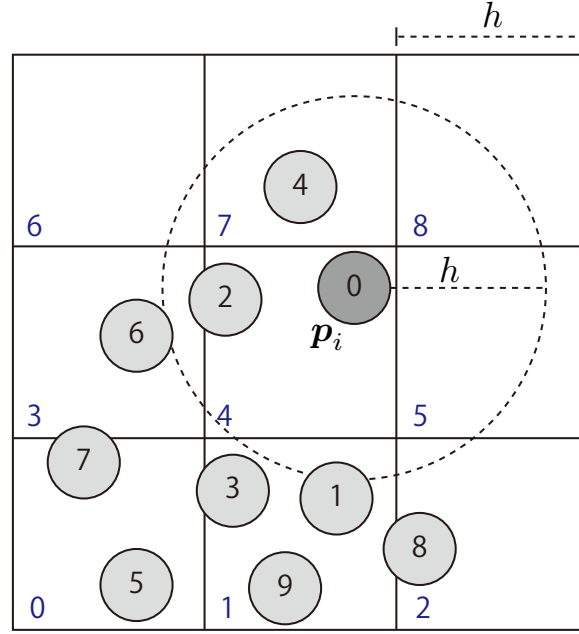


図 5.5: 等間隔格子による近傍探索手法

ここからは図 5.5 に示すようにグリッドセルと粒子が配置されていた場合の近傍探索の具体的な手順を説明する．図 5.5 のグリッドセルの左下の番号は各グリッドセルでユニークなハッシュ値であり，粒子に記述してある番号は粒子のインデックスである．まず最初に，粒子位置を用いて各粒子がどのグリッドセルに属しているか対応付ける．例えば，粒子 0 はグリッドハッシュ値 4 のセルに属しており，粒子 1 はグリッドハッシュ値 1 のセルに属している．対応付けが終わると次はグリッドハッシュをキーとして昇順にソートする．本研究ではソートに NVIDIA Thrust ライブラリを用いることで，GPU で並列にソート処理を行った．ソート後のグリッドハッシュ値 (gridHash) と粒子インデックス (particleIndex) の対応表を以下の表 5.1 に示す．次に，表 5.1 において同じグリッドハッシュが続く領域の開始地点と終了地点をそれぞれ配列 start と end に格納する．例えば，グリッドハッシュが 1 となるのは配列の gridHash[2] から gridHash[4] までなので，start[1] = 2, end[1] = 4 と計算できる．また，粒子が存在しないグリッドセルが存在した場合，粒子が存在していないことを識別できる値を格納しておく．本研究では粒子の存在しないグリッドセルのハッシュ値に -1 を格納している．こうして求められた start と end を以下の表 5.2 に示す．

表 5.1: ソート後のグリッドハッシュと粒子インデックス

gridHash	0	0	1	1	1	2	3	4	4	7
particleIndex	5	7	1	3	9	8	6	0	2	4

表 5.2: 各グリッドハッシュの start と end

gridHash	0	1	2	3	4	5	6	7	8
start	0	2	5	6	7	-1	-1	9	-1
end	1	4	5	6	8	-1	-1	9	-1

最後に自身が属するグリッドセルとその周囲のグリッドセルに属している粒子を探索する。隣接するグリッドセル j に属している粒子を探索するときは、`particleIndex[start[j]...end[j]]` の粒子を探索すればよい。start[j] = -1 の場合はそのグリッドセルを無視する。ここまで説明した処理 (グリッドハッシュ値の計算・ソート・start と end の計算) はすべて並列に処理が可能である。

5.3 シミュレーションの安定化

本節では、シミュレーションを安定に実行するための実装上の工夫について述べる。

5.3.1 静止密度の初期値

4章で述べた流体シミュレーションでは、基本的に同一の物体の静止密度 ρ^{rest} は物体内で一定である。しかしながら、計算点間の距離や初期状態の形状は髪型データにより異なるため、図 5.6 に示すように初期状態において密度分布が一樣ではない場合がほとんどである。そのため一定の静止密度を用いてシミュレーションを行うと、元の髪型を維持することが難しい。初期状態の髪型データで均等に毛髪が配置されている場合は問題にならないが、毛髪データから適切な計算点の分布を再計算する必要があり、解像度が上がると計算コストも増加する。そこで本研究では、最初の 0 ステップ目で一度毛髪粒子の密度場を計算し、その値を静止密度として使用する。その結果、毛髪が密集して毛束となっていた場所はより集まりやすい毛髪となり、逆に疎に配置されていた毛髪は集まりにくい毛髪としてふるまう。これにより初期状態の毛髪形状を維持するように相互作用計算が行われる。以下に粒子 i において、0 ステップ目で初期値として設定した静止密度 ρ_i^{rest} を更新する式を示す。

$$\rho_i^{\text{rest}} = \max(\rho_i^{\text{rest}}, \rho_i^{\text{orig}}) \quad (5.1)$$

ここで、 ρ^{rest} は静止密度、 ρ^{orig} は初期状態の密度、max は引数のうち大きな値を返す関数である。

5.3.2 方向を表すベクトルから姿勢を表す四元数の生成

ある毛髪の初期形状データを読み込むとき、入力情報に姿勢情報 q が含まれていない場合は、毛髪の計算点の初期位置情報から初期状態の姿勢 q を求める必要がある。本研究では 3.1 節で述べたように、グローバル座標系の基底をマテリアルフレームの基底に回転させる四元数 q を毛髪の姿勢として扱っている。毛髪の接線方向はマテリアルフレームの基底 \mathbf{d}_3 で表され、グローバル座標系の基底 \mathbf{e}_3 と \mathbf{d}_3 の情報があれば姿勢 q を逆算することができる (逆算方法は付録 A に示す)。しかしながら、毛髪の接線方向 \mathbf{d}_3 が \mathbf{e}_3 の真逆の方向に近くなると、 q が計算誤差の影響を受けやすくなり、シミュレーションが不安定になる現象が確認された。この問題を解決するために、 \mathbf{e}_3 の向きを真下方向のベクトル $(0, -1, 0)^T$ に設定した。本研究の実験で用いた毛髪形状では、初期状態で真上を向いている毛髪がほとんどなく安定したシミュレーションが可能である。

また、他の解決法として計算の精度を上げるという方法もある。現在は単精度で演算を行っているが、倍精度以上で計算することで誤差の減少が期待できる。ただし、本研究では高速化のために GPU を用いているが、本研究で用いている GPU (NVIDIA GeForce RTX 2080Ti) は倍精度よりも単精度演算性能が優れているため、倍精度以上の精度で計算すると計算速度が大きく低下するという問題がある。

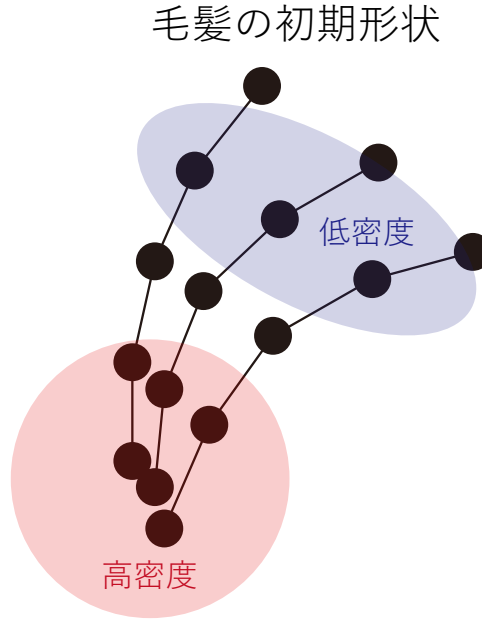


図 5.6: 初期形状における毛髪粒子の密度分布

5.4 レンダリング

空気は透明なものとして扱い描画する必要がないため、本項では毛髪のレンダリング手法について述べる。毛髪は光が入る角度によって特定の方向に強く反射する性質を持っている。この性質は異方性反射と呼ばれており、光が当たった毛髪に天使の輪のような光沢が入って見えるのはこの性質が原因である。本研究では、異方性反射のシェーディングモデルに Kajiya-Kai モデル [29] を用いる。Kajiya-Kai モデルは毛髪の拡散反射成分と異方性鏡面反射成分を光の入射角と視線方向から計算する。以下に拡散反射成分 I^{diff} と異方性鏡面反射成分 I^{spec} の計算式を示す。

$$I^{\text{diff}} = A^{\text{diff}} \sin \theta^{\text{in}} \quad (5.2)$$

$$I^{\text{spec}} = A^{\text{spec}} \cos A^{\text{shin}} (\theta^{\text{in}} - \theta^{\text{out}}) \quad (5.3)$$

ここで、 A^{diff} は拡散反射係数、 A^{spec} は鏡面反射係数、 A^{shin} は光沢度、 $\theta^{\text{in}}, \theta^{\text{out}}$ はそれぞれ図 5.7 に示すように入射光と毛髪の接線がなす角と、視野方向と毛髪の接線がなす角である。13461 本の毛髪を有する高解像度な髪型のアセットデータ¹の Kajiya-Kai モデルによる光沢を図 5.8 に示す。

¹アセットデータ提供：株式会社デジタル・フロンティア

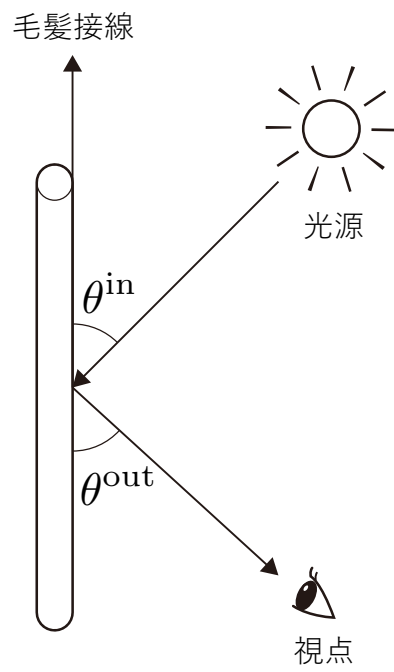


図 5.7: Kajiya-Kai モデルによる異方性反射計算



図 5.8: Kajiya-Kai モデルのレンダリング結果

第6章 結果

本章では，提案手法を実装し複数のシーンで実験を行った結果を示し，その考察を述べる．提案手法の計算結果は全て NVIDIA CUDA によって GPU デバイス上で並列的に処理されたものである．計算時間は運動計算にかかる時間であり，レンダリング処理にかかった時間は含めていない．ただし，5.4 節で述べたレンダリング手法は GLSL で実装することにより非常に高速に処理できるため，実行時間への影響は少ない．実験に用いた実行環境を表 6.1 に示す．

表 6.1: 実行環境

OS	Microsoft Windows 10 64bit
CPU	Core i7-6700K 4.00GHz
RAM	16GB
GPU	NVIDIA GeForce RTX 2080Ti
VRAM	11GB
Graphics API	OpenGL
開発言語	C++, CUDA, GLSL

6.1 シーン 1

シーン 1 では 4.3 節で述べた渦による流れの制御について実験する．そのために，毛髪の塑性変形は考慮せず，毛髪が風でなびいている様子のみを再現する．毛髪と空気の相互作用によって毛髪がなびいている運動を再現するために，束状に配置した毛髪に冷風を当てる．このシーンでは側鎖結合は全て十分に結合しており，シミュレーション中に切断されることはないものとした．既存手法との比較として，航空力学を用いたモデル [5] による実験も行った（ただし毛髪同士の相互作用は提案手法で計算する）．風の吹く方向は結果画像の左側から右側であり，縦横の幅 0.24[m] の長方形の領域から風が発生する．なお，毛髪が配置されている白い球体は毛根位置の目印として設置しているもので，衝突判定の計算は行っていない．

本シーンで用いる共通のパラメータの設定を以下の表 6.2 に，シミュレーション手法による計算時間の比較を表 6.3 に示す．図 6.2 にはストレートヘアで航空力学を用いたモデル [5] による実験，図 6.3 と図 6.4 には，それぞれストレートヘアとカールヘアで乱流シミュレーションを行った実験 ($\alpha = 0.006[\text{m}]$) のシミュレーション過程を示す．また，異なる大きさの渦を強調した場合に確認できる動きの比較を図 6.5 に示し，異なる粘性値で実験した場合に確認できる動きの比較を図 6.6 に示す．

表 6.2: シーン 1 共通のパラメータ設定

タイムステップ幅 Δt [s]	1/720
重力加速度 g [m/s ²]	9.8
弾塑性変形制約条件の反復回数	10
毛髪本数	550
毛髪の長さ L^{\max} [m]	0.35
毛髪の計算点間距離 l [m]	0.01
毛髪の弾性を表す係数 K	0.003
流体密度制約条件の反復回数	2
空気粒子の最大数	200000
流体の静止密度 ρ^{rest} [kg/m ³]	1.293
流体粒子の質量 m [kg]	1.46×10^{-7}
流体粒子の有効半径 h [m]	0.0093
乱流シミュレーションの調整係数 ϵ^{vort}	0.08
空気粒子の粘性を表す係数 $\epsilon_{\text{air}}^{\text{visc}}$	0.015
風の初期速度 [m/s]	3

表 6.3: シーン 1 の計算時間

シミュレーション手法	既存手法 [5]	乱流制御無	$\alpha = 0.006$ [m]	$\alpha = 0.018$ [m]
1 ステップの計算時間 [ms]	1.6	4.8	9.3	62.5

6.2 シーン 2

シーン 2 では温度変化による塑性変形を再現するため、水素結合が十分に切断されている状態の毛髪 ($b_{\text{hyd}} = 0$) に温風や冷風をあてる実験を行う。変形を分かりやすくするためにシミュレーション空間には筒状の障害物を用意し、衝突により毛髪が曲がるようにした。風の吹く方向は結果画像の右側から左側であり、障害物である筒の右側に存在する高さ 0.12[m]、横幅 0.2[m] の長方形の領域から風が発生する。また、本シーンでは塑性変形のみを検証するために乱流シミュレーションを行わず、毛髪が配置されている白い球体は衝突判定の計算を行っていない。

本シーンで用いるパラメータの設定を以下の表 6.4 に示す。本シーンでの 1 ステップあたりの計算時間は 5.5[ms] である。図 6.7 には、静止状態の毛髪に対して 8 秒間温風をあて続けた場合の温度変化の過程を示す。図 6.8 には、静止状態の毛髪に対して 6 秒間温風をあてたあと、2 秒間冷風をあてた場合の温度変化の過程を示す。図 6.9 は、温風と冷風をあてた図 6.8 に示す実験のレンダリング結果である。風と温度によって発生した塑性変形を確認するため、毛髪の元の形状と、それぞれ異なる時間温風と冷風をあてた後の毛髪の形状の比較を図 6.10 に示す。また、以下の図 6.1 に温度と色の関係を表すカラーマップを示す。

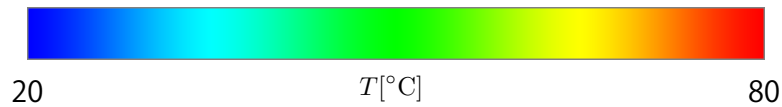


図 6.1: 温度のカラーマップ

表 6.4: シーン 2 のパラメータ設定

タイムステップ幅 $\Delta t[\text{s}]$	1/720
重力加速度 $g[\text{m/s}^2]$	9.8
弾塑性変形制約条件の反復回数	10
毛髪本数	949
毛髪の長さ $L^{\text{max}}[\text{m}]$	0.40
毛髪の計算点間距離 $l[\text{m}]$	0.008
毛髪の弾性を表す係数 K	0.05
初期状態の水素結合結合率 $b_{\text{hyd}}^{\text{orig}}$	0
水素結合の影響力 (重み) w_{hyd}	2/3
水素結合以外の結合の影響力 (重み)	1/3
流体密度制約条件の反復回数	2
空気粒子の最大数	200000
流体の静止密度 $\rho^{\text{rest}}[\text{kg/m}^3]$	1.293
流体粒子の質量 $m[\text{kg}]$	4.33×10^{-8}
流体粒子の有効半径 $h[\text{m}]$	0.0074
空気粒子の粘性を表す係数 $\epsilon_{\text{air}}^{\text{visc}}$	0.015
毛髪粒子の粘性を表す係数 $\epsilon_{\text{hair}}^{\text{visc}}$	1
流体の比熱容量 $c[\text{J/s/m/K}]$	1010
流体の熱伝導係数 $\kappa[\text{J/kg/K}]$	0.024
風の初期速度 $[\text{m/s}]$	3
温風の温度 $[\text{°C}]$	80
冷風の温度 $[\text{°C}]$	20

6.3 シーン 3

ここまでは規則的に整列し、全ての毛髪が同じ長さ・初期形状を持つ髪型データで実験を行ってきたが、シーン 3 では実際に CG アニメーションを作成することを想定した髪型のアセットデータを用いて実験を行う。アセットデータはアーティストにより作成されたものであり¹、各毛髪の長さも一様ではなく現実の人間の毛髪の状態に非常に近い。一方で、毛髪が均等に配置されていないことにより密度分布が一様ではなく、提案手法によるモデル化では初期状態の髪型を保つことは難しい。そのためシミュレーションが不安定となる場合があるが、この問題については 5.3.1 項で述べた手法で解決した。

本シーンでは、塑性変形の計算と同時に乱流シミュレーションも行う。シーン 2 と同様に温風と冷風をあてるが、風の吹く方向は結果画像の左側から右側であり、左側の高さ 0.12[m]、横幅 0.2[m] の長方形の領域から風が発生する。最初の 6 秒間は温風をあて、続いて 2 秒間冷風をあてる。これまでのシーンと同様に白い球体との衝突判定は行っていない。

本シーンで用いるパラメータの設定を以下の表 6.5 に示す。本シーンでの 1 ステップあたりの計算時間は 20.8[ms] である。図 6.11 にシミュレーションのレンダリング結果、図 6.12 にその温度変化を示す。また、シミュレーション前の元形状とシミュレーション後の形状で、斜め前からみた図と真横から見た図のそれぞれの比較を図 6.13 に示す。

¹アセットデータ提供：株式会社デジタル・フロンティア

表 6.5: シーン 3 のパラメータ設定

タイムステップ幅 Δt [s]	1/720
重力加速度 g [m/s ²]	9.8
弾塑性変形制約条件の反復回数	10
毛髪本数	3457
毛髪の長さ L^{\max} [m]	不定 (平均値 約 0.23)
毛髪の計算点間距離 l [m]	不定 (平均値 約 0.01)
毛髪の弾性を表す係数 K	0.03
初期状態の水素結合結合率 $b_{\text{hyd}}^{\text{orig}}$	0
水素結合の影響力 (重み) w_{hyd}	2/3
水素結合以外の結合の影響力 (重み)	1/3
流体密度制約条件の反復回数	2
空気粒子の最大数	200000
流体の静止密度 ρ^{rest} [kg/m ³]	1.293
流体粒子の質量 m [kg]	4.33×10^{-8}
流体粒子の有効半径 h [m]	0.0074
空気粒子の粘性を表す係数 $\epsilon_{\text{air}}^{\text{visc}}$	0.015
毛髪粒子の粘性を表す係数 $\epsilon_{\text{hair}}^{\text{visc}}$	1
流体の比熱容量 c [J/s/m/K]	1010
流体の熱伝導係数 κ [J/kg/K]	0.024
乱流シミュレーションの調整係数 ϵ^{vort}	0.02
強調する渦の大きさ α [m]	0.006
風の初期速度 [m/s]	3
温風の温度 [°C]	80
冷風の温度 [°C]	20

6.4 シーン 4

シーン 4 では、筒状の物体を用いて水素結合のみでカールを作るシーンを再現する．実験で用いる筒の太さは半径が 0.02[m] である．まず，カールしていない毛髪の先端を固定したうえで垂直に立てた筒に巻きつける．巻きつけが終わると画面右側から 6 秒間温風をあて、次に冷風を 4 秒間あてる．風は右側の縦横の幅 0.16[m] の長方形の領域から発生する．風を止めた後は固定されている毛髪を解放し筒を撤去する．また、本シーンでは毛髪が固定されているため乱流による毛髪の挙動の変化がほとんど見られない、そのため乱流シミュレーションは行わず、塑性変形計算だけ行うものとする．

本シーンで用いるパラメータの設定を以下の表 6.6 に示す．本シーンでの 1 ステップあたりの計算時間は 5.1[ms] である．図 6.14 にシミュレーションのレンダリング結果、図 6.15 にその温度変化を示す．図 6.16 には、毛髪の前形状とシミュレーション後の形状、さらに同条件で温風のみを 10 秒間あて続けた場合のシミュレーション後の形状の比較を示す．

表 6.6: シーン 4 のパラメータ設定

タイムステップ幅 $\Delta t[\text{s}]$	1/720
重力加速度 $g[\text{m/s}^2]$	9.8
弾塑性変形制約条件の反復回数	10
毛髪本数	345
毛髪の長さ $L^{\text{max}}[\text{m}]$	0.48
毛髪の計算点間距離 $l[\text{m}]$	0.008
毛髪の弾性を表す係数 K	0.05
初期状態の水素結合結合率 $b_{\text{hyd}}^{\text{orig}}$	0
水素結合の影響力 (重み) w_{hyd}	2/3
水素結合以外の結合の影響力 (重み)	1/3
流体密度制約条件の反復回数	2
空気粒子の最大数	200000
流体の静止密度 $\rho^{\text{rest}}[\text{kg/m}^3]$	1.293
流体粒子の質量 $m[\text{kg}]$	4.33×10^{-8}
流体粒子の有効半径 $h[\text{m}]$	0.0074
空気粒子の粘性を表す係数 $\epsilon_{\text{air}}^{\text{visc}}$	0.015
毛髪粒子の粘性を表す係数 $\epsilon_{\text{hair}}^{\text{visc}}$	1
流体の比熱容量 $c[\text{J/s/m/K}]$	1010
流体の熱伝導係数 $\kappa[\text{J/kg/K}]$	0.024
風の初期速度 $[\text{m/s}]$	3
温風の温度 $[\text{°C}]$	80
冷風の温度 $[\text{°C}]$	20

6.5 考察

6.5.1 相互作用を考慮した毛髪の塑性変形

シーン 2 で行った実験では、風をあてることで毛髪の温度を上下させて水素結合の結合率を変化させた。シーン 2 の温度変化を示した図 6.7 と図 6.8 から確認できるように、風が発生している毛髪の中央部から温度が拡散している。結果として水素結合が再結合し、現在の形状を記憶する塑性変形を確認することができた (図 6.10)。温風のみをあてた場合は水分の蒸散によって塑性変形を起こしたものの、熱により水素結合が切断されているため、まだ塑性変形する可能性がある。一方で冷風を当てた場合は、温度が下がったことで水素結合が再結合し、結合率が高く剛性が高い毛髪となる。またシーン 3 で示したように、現実の毛髪形状に近いアセットデータを用いた実験でも安定したシミュレーションで塑性変形の再現が可能であることを確認した。シーン 4 の実験では、筒状の物体に毛髪を巻きつけ風をあてることで、パーマメントウェーブのようなカール形状を形成する過程を再現することができた。以上の実験結果から、3.3.4 項で述べた熱の影響と水分の蒸散の影響の基本的な性質が再現できていることが確かめられた。

一方で毛髪の個人差を調節するためのパラメータ設定や結合率と温度の関係を表す関数の設計は直感的でなく、ユーザへの負担が大きい問題がある。上記の要因以外にも側鎖結合が変化する要因は存在するが、それらを全てモデル化しパラメータを設定するのはシミュレーションを複雑化させ、ユーザにとって使いにくいシステムとなる。そのため、比較的容

易に計測可能な指標から毛髪の挙動を推定できることが望ましい。そのためには実物を用いた実験との比較による検証が必要である。

なお、本研究のシーンでは筒状の障害物のような剛体との衝突判定を計算している。このとき粒子法では周囲の近傍粒子から物理量を計算しているため、近傍粒子数が少なくなる剛体との境界付近では密度値が過小評価され、粒子が集まってしまう問題がある。この問題を解決するためには境界粒子を用いて剛体との相互作用を計算する手法 [30] など剛体境界の処理に工夫が必要である。

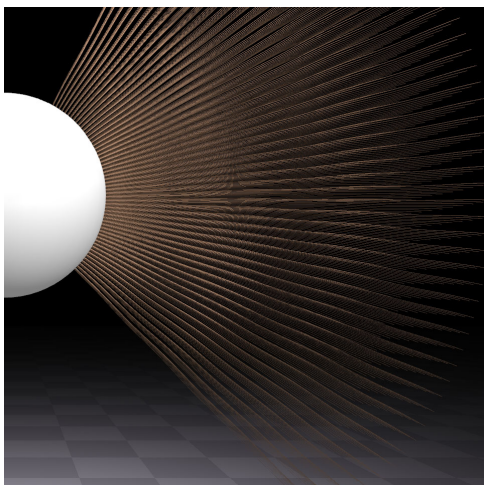
6.5.2 渦の制御による毛髪のなびき運動

シーン 1 では、航空力学を用いた既存手法 [5] と異なるパラメータ設定の提案手法で毛髪が風になびくアニメーションを再現した。航空力学を用いた手法は計算速度が非常に速く、嵐のように風が発生している領域が広いシーンを高速で計算するのに適している。一方で提案手法は、ヘアドライヤーのように局所的に発生する空気の流れとの相互作用を計算するのに適しており、単純な上下運動ではなく毛髪が空気の流れを乱すことによって発生する渦を巻くような運動を再現できている。さらに、ユーザは強調したい渦の大きさと強調の度合いを決めることによって、なびき表現を大きく変えることが可能であることが実験で確かめられた (図 6.5)。粘性を表す係数を調整することによって毛髪のまとまり具合を調整することも可能である (図 6.6)。

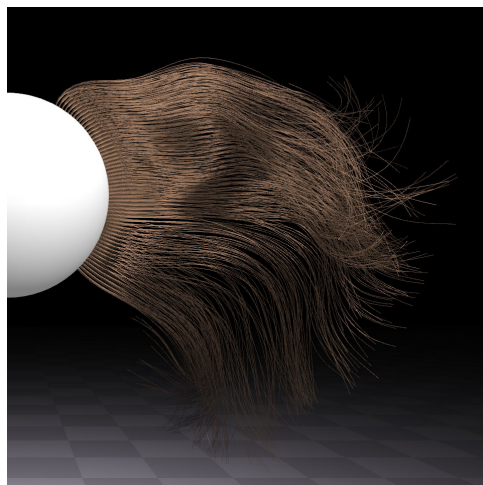
しかしながら、提案手法には様々な問題点がある。提案手法では既に存在している渦を強調することしかできないため、存在していない (もしくは非常にエネルギー値の小さな) 大きさの渦を強調することはできない。それと同時に、強調したくない大きさの渦があったとしても渦の影響を抑制することはできない。より直感的に渦の制御を行うためには、新しい渦を生成するなど追加の手法が必要である。また、毛髪を球形の粒子でモデル化しているため、毛髪の細さやキューティクルの摩擦力が生み出す影響を表現することができない。

計算時間も問題の一つである。表 6.3 に示したように、乱流シミュレーションをした場合に多くの計算時間がかかっており、抽出する渦が巨大化するほどその計算量は増大する。リアルタイムシミュレーションに近づけるためには、より高速な手法・実装方法が求められる。

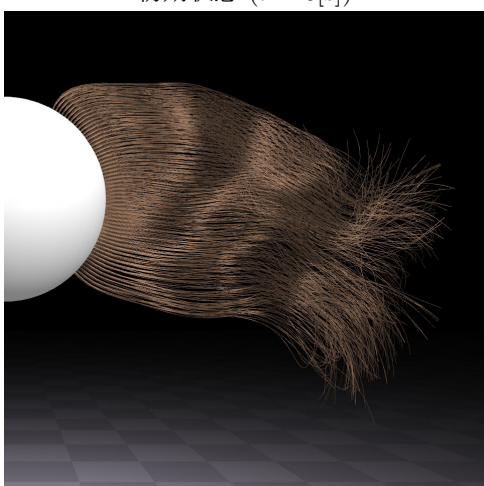
今回は全てのシーンにおいて風が発生している場所にのみ空気粒子を配置した。これにより風が発生していない地点の計算を無視することで計算量を減らすことができるのは粒子法の長所である。しかし、本来は空気の粘性により周囲に存在している速度の遅い粒子の影響を受け風は減衰する。例えばヘアドライヤーの場合は、空気の流れの中心部の流速が最も速くなる。この問題を解決するために計算空間を空気粒子で満たすのは計算時間の観点から難しい。そこで、空気が周囲に存在する中で移流する液体の動きを、多層流体シミュレーションすることなく再現した手法 [31] などが応用できると考えられる。



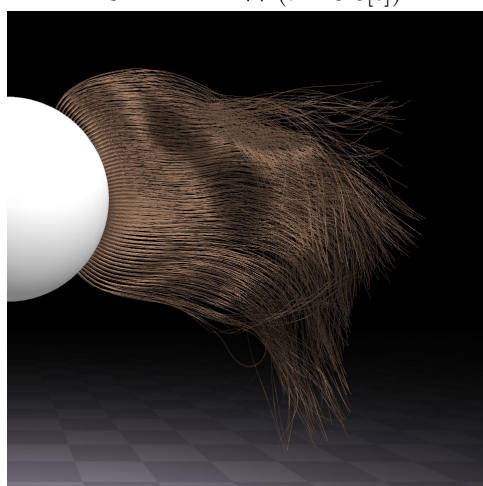
初期状態 ($t = 0[s]$)



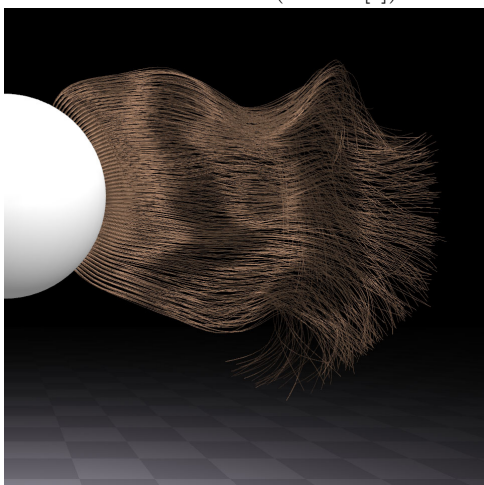
432 フレーム目 ($t = 0.6[s]$)



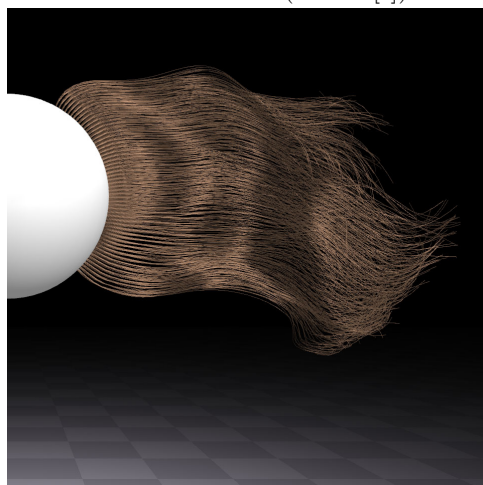
864 フレーム目 ($t = 1.2[s]$)



1296 フレーム目 ($t = 1.8[s]$)

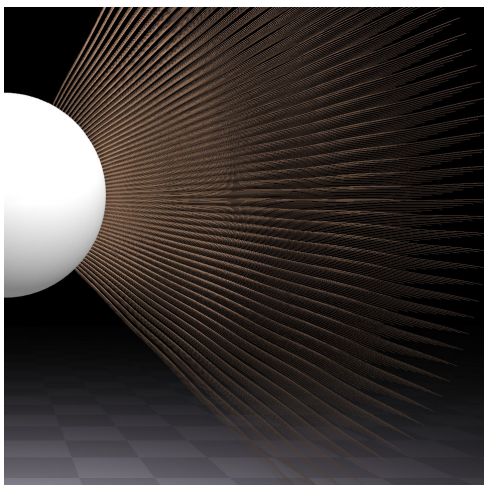


1728 フレーム目 ($t = 2.4[s]$)

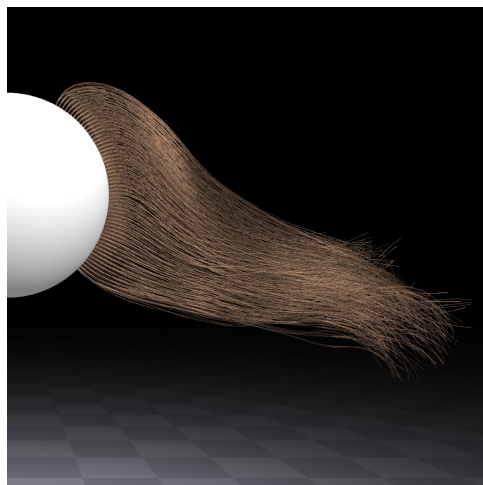


2160 フレーム目 ($t = 3[s]$)

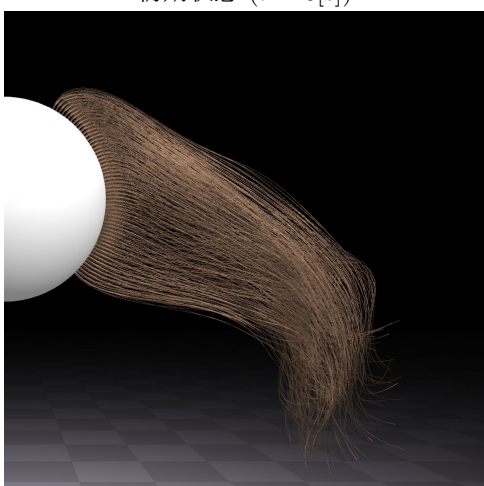
図 6.2: シーン 1 : ストレートヘア ($\epsilon_{\text{hair}}^{\text{visc}} = 1$) 既存手法 [5]



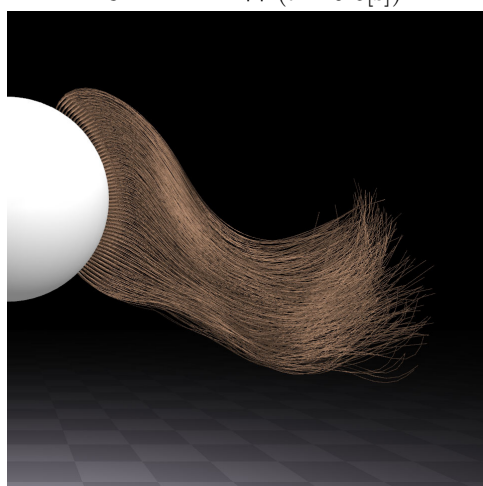
初期状態 ($t = 0[s]$)



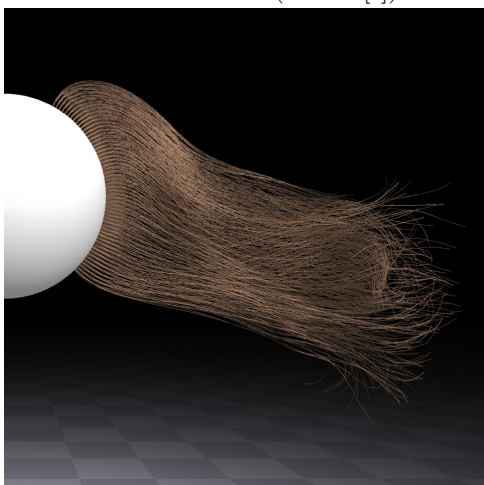
432 フレーム目 ($t = 0.6[s]$)



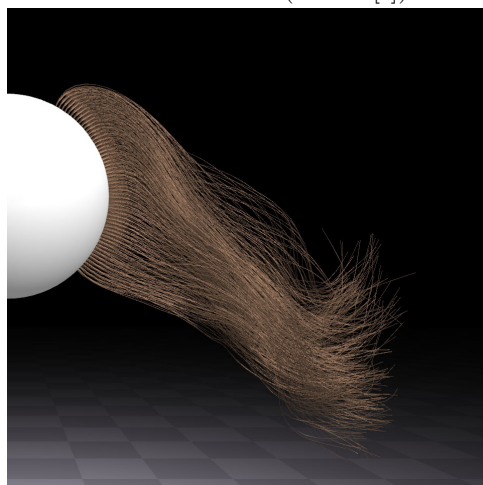
864 フレーム目 ($t = 1.2[s]$)



1296 フレーム目 ($t = 1.8[s]$)

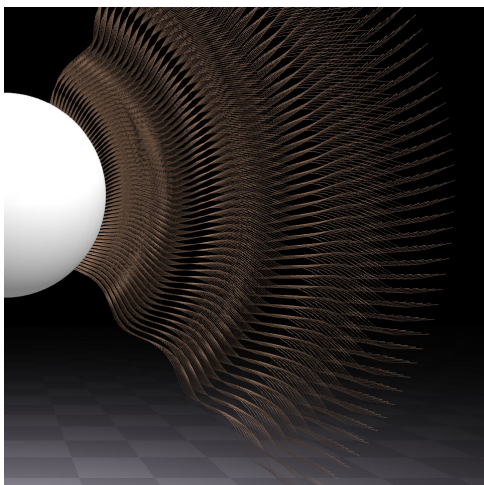


1728 フレーム目 ($t = 2.4[s]$)

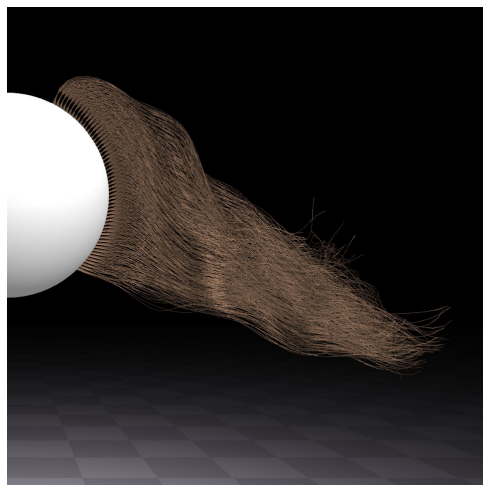


2160 フレーム目 ($t = 3[s]$)

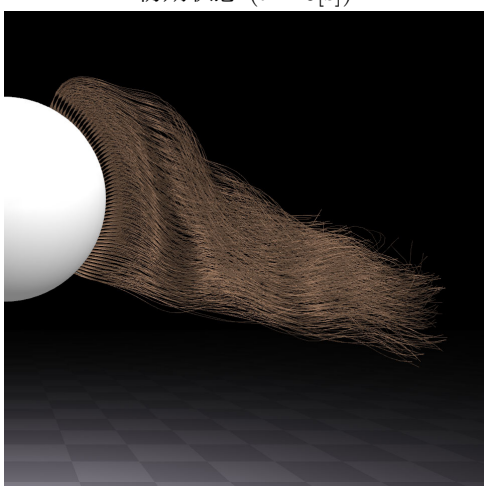
図 6.3: シーン 1 : ストレートヘア ($\epsilon_{\text{hair}}^{\text{visc}} = 1, \alpha = 0.006[m]$)



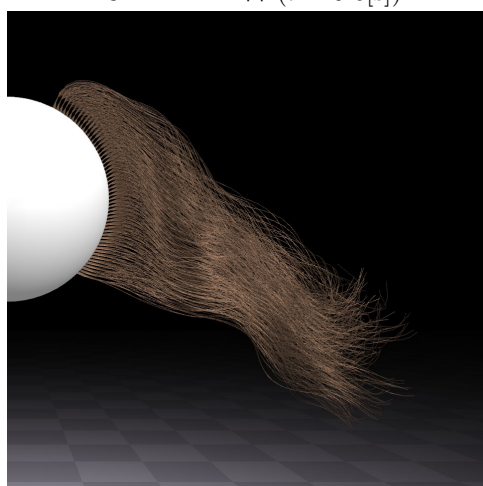
初期状態 ($t = 0[s]$)



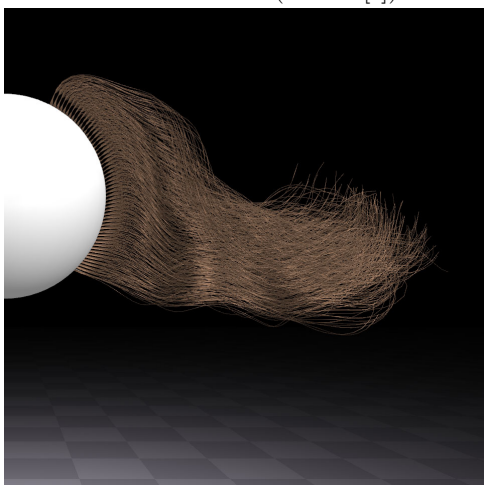
432 フレーム目 ($t = 0.6[s]$)



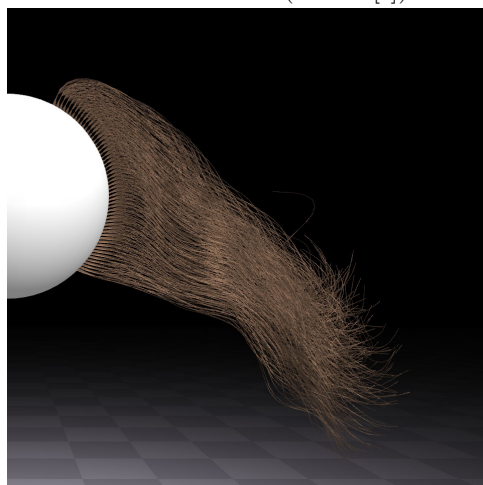
864 フレーム目 ($t = 1.2[s]$)



1296 フレーム目 ($t = 1.8[s]$)

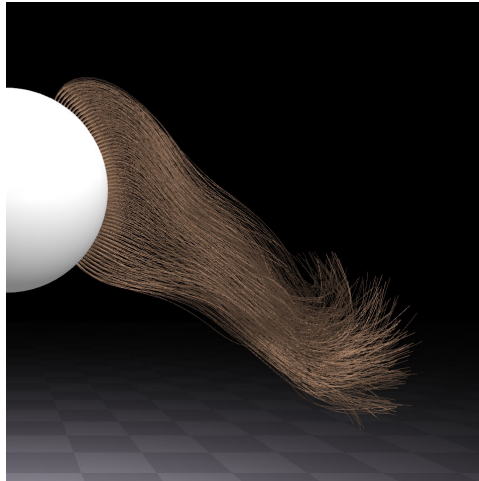


1728 フレーム目 ($t = 2.4[s]$)

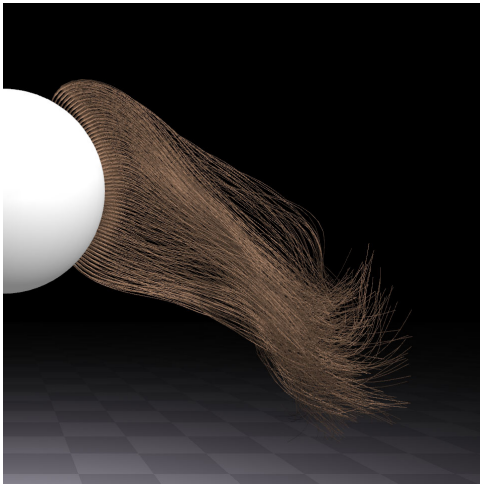


2160 フレーム目 ($t = 3[s]$)

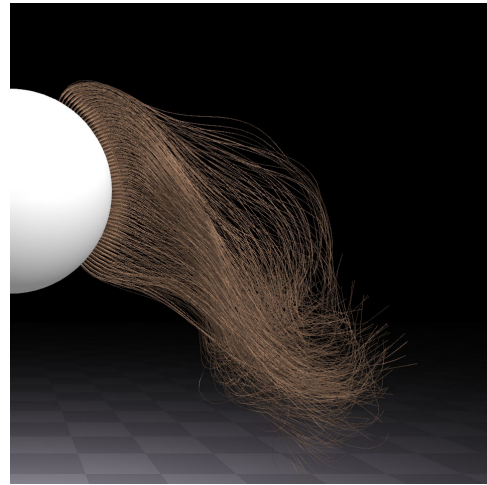
図 6.4: シーン 1 : カールヘア ($\epsilon_{\text{hair}}^{\text{visc}} = 1, \alpha = 0.006[\text{m}]$)



乱流シミュレーション無し

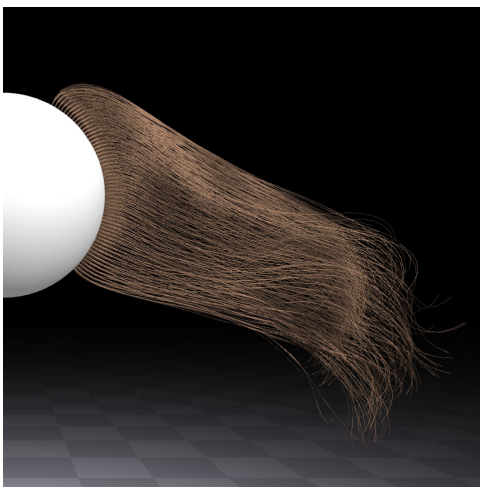


小さな渦を強調 ($\alpha = 0.006[\text{m}]$)

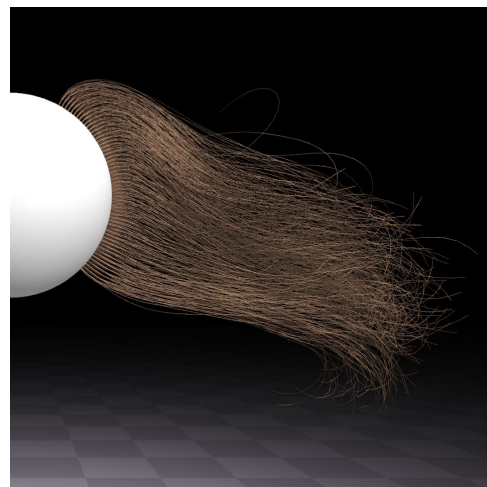


大きな渦を強調 ($\alpha = 0.018[\text{m}]$)

図 6.5: シーン 1 : 乱流シミュレーションの比較



高粘性 ($\epsilon_{\text{hair}}^{\text{visc}} = 1$)



低粘性 ($\epsilon_{\text{hair}}^{\text{visc}} = 0.1$)

図 6.6: シーン 1 : 粘性の違いの比較

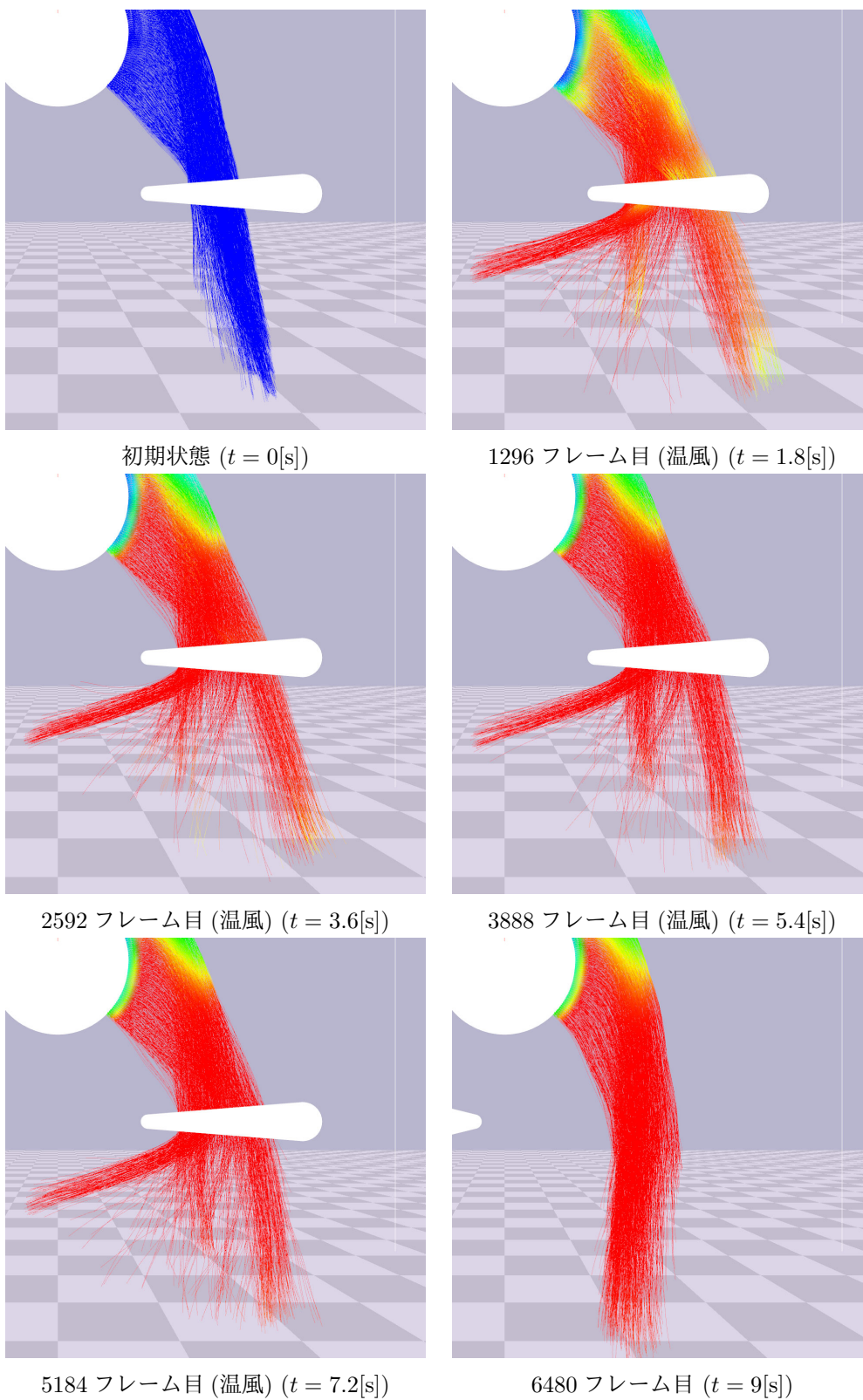


図 6.7: シーン 2 : 温風を 8 秒間あて続けた場合の温度変化

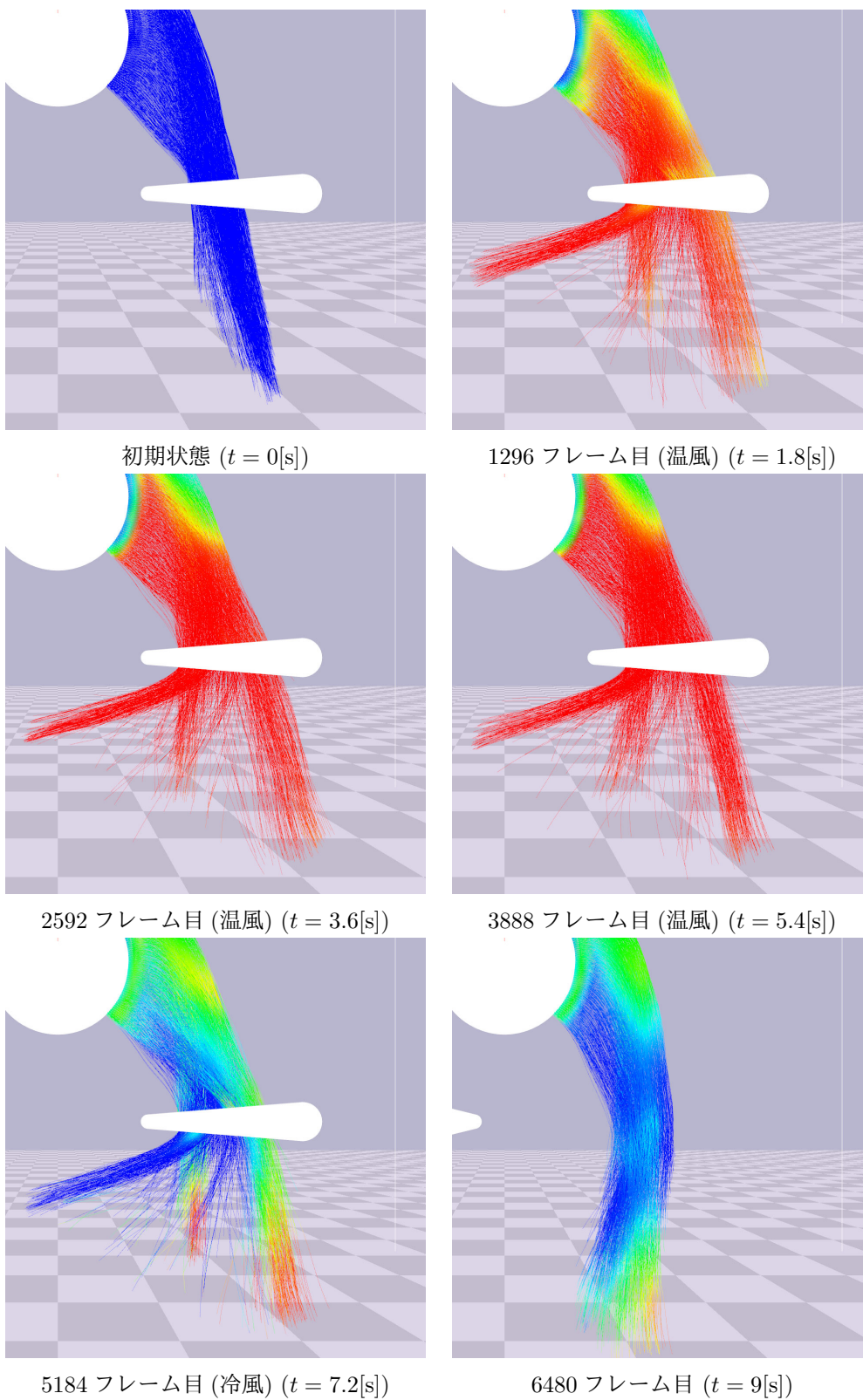


図 6.8: シーン 2: 温風を 6 秒間あてたあと冷風を 2 秒間あてた場合の温度変化

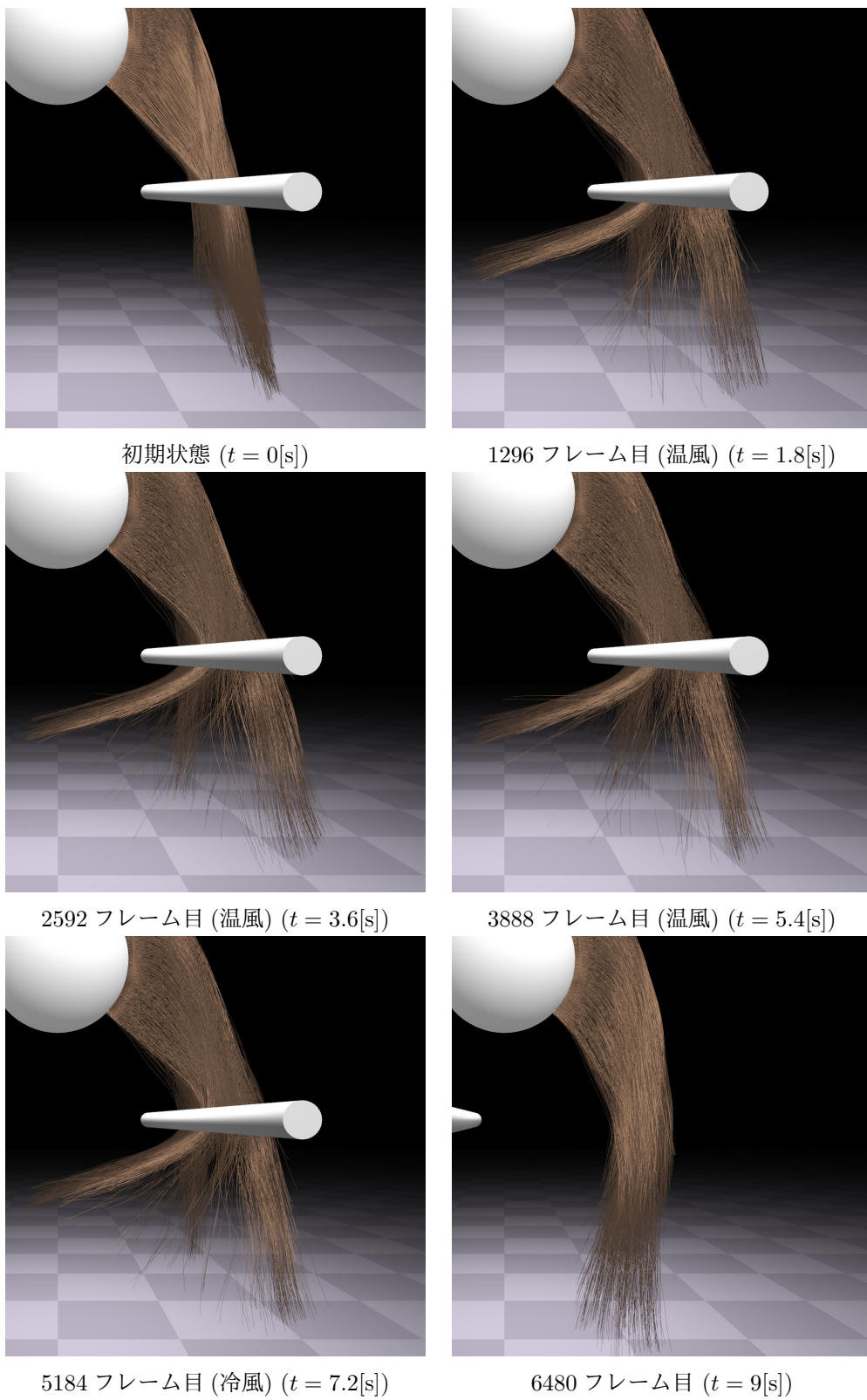
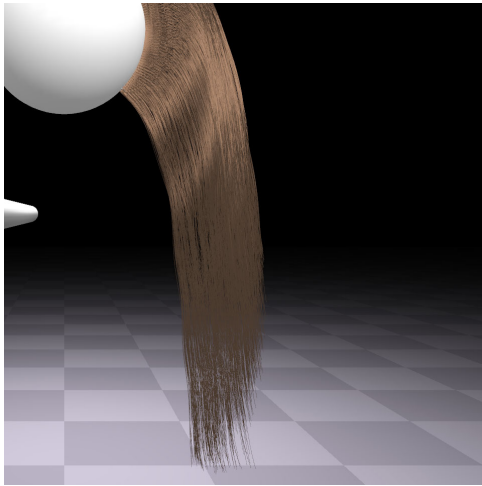
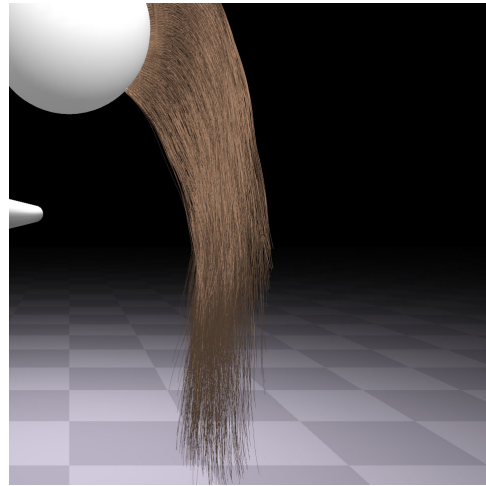


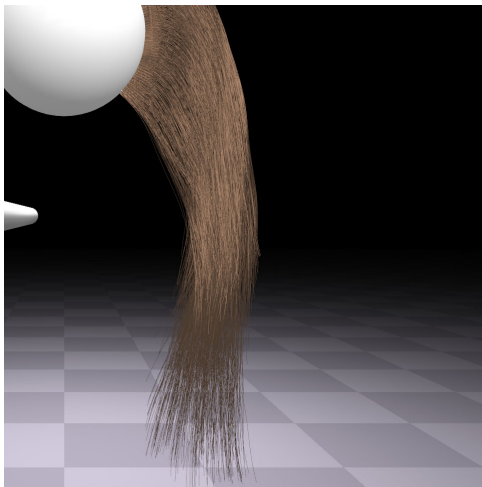
図 6.9: シーン 2 : 温風を 6 秒間あてたあと冷風を 2 秒間あてた場合のレンダリング結果



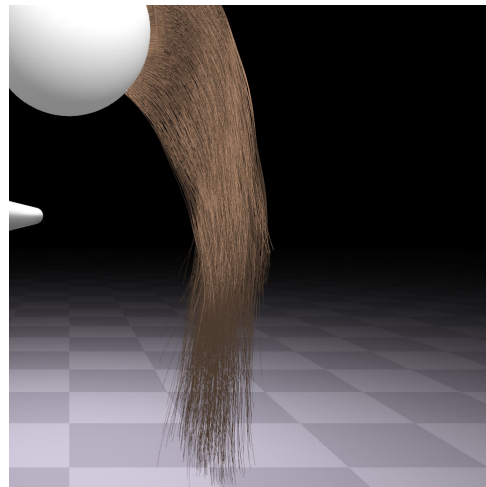
元の形状



温風 8 秒間



温風 6 秒間のあと冷風 2 秒間

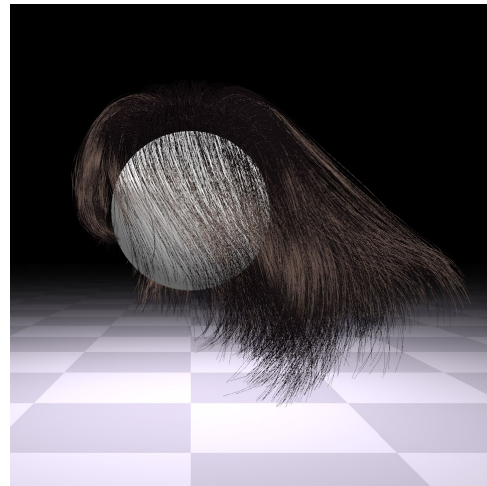


温風 2 秒間のあと冷風 6 秒間

図 6.10: シーン 2 : 風と温度による塑性変形の比較



初期状態 ($t = 0[s]$)



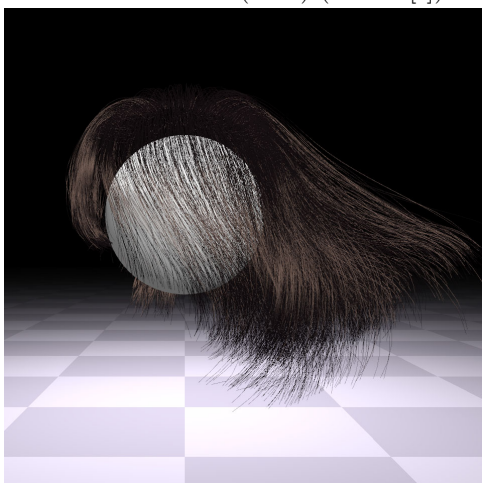
1296 フレーム目 (温風) ($t = 1.8[s]$)



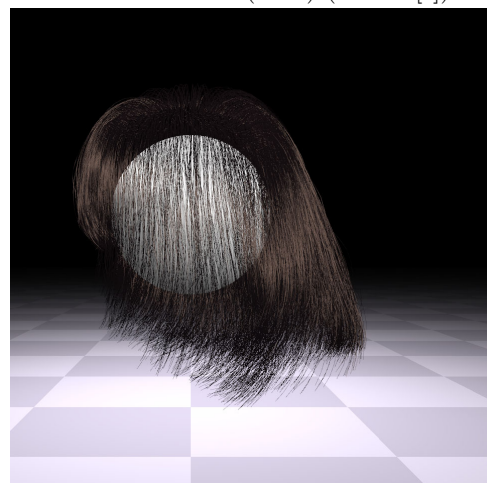
2592 フレーム目 (温風) ($t = 3.6[s]$)



3888 フレーム目 (温風) ($t = 5.4[s]$)

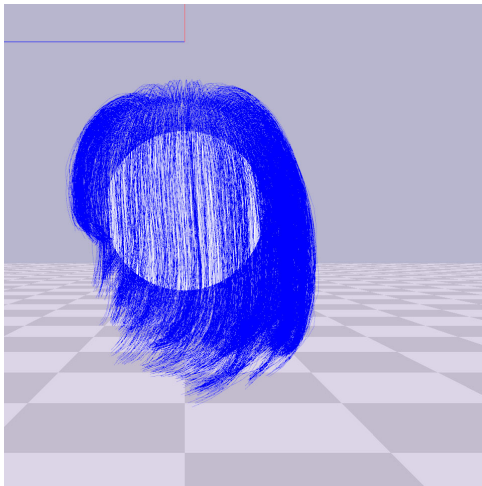


5184 フレーム目 (冷風) ($t = 7.2[s]$)

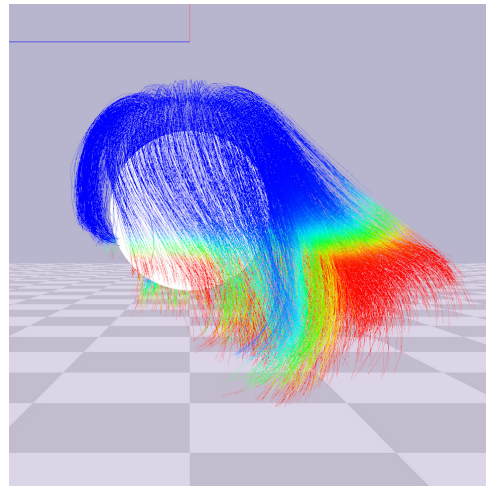


6480 フレーム目 ($t = 9[s]$)

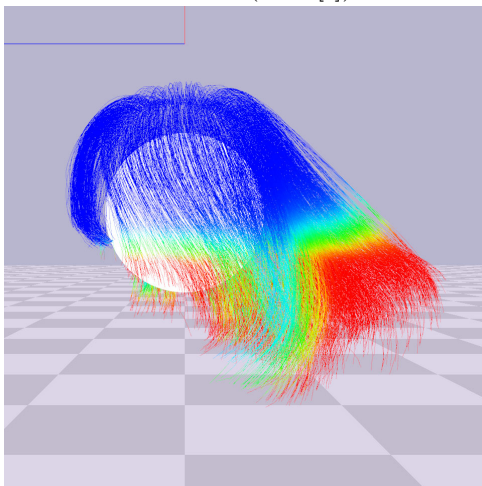
図 6.11: シーン 3 : レンダリング結果



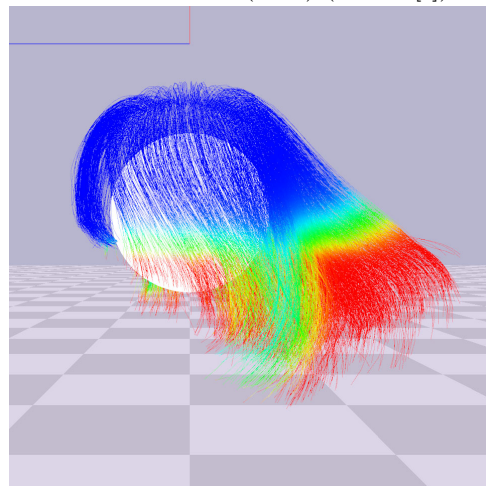
初期状態 ($t = 0[s]$)



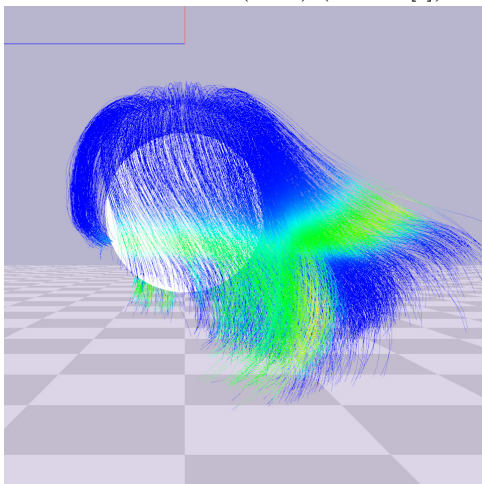
1296 フレーム目 (温風) ($t = 1.8[s]$)



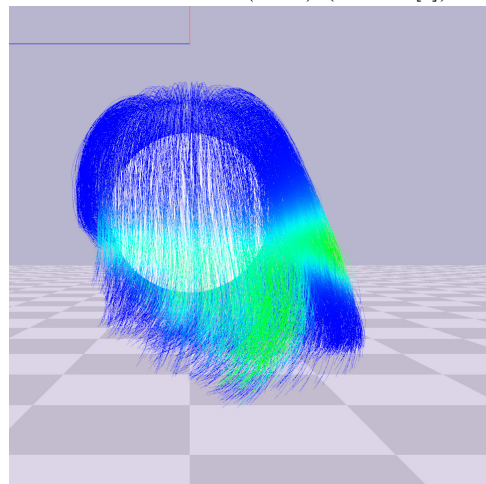
2592 フレーム目 (温風) ($t = 3.6[s]$)



3888 フレーム目 (温風) ($t = 5.4[s]$)

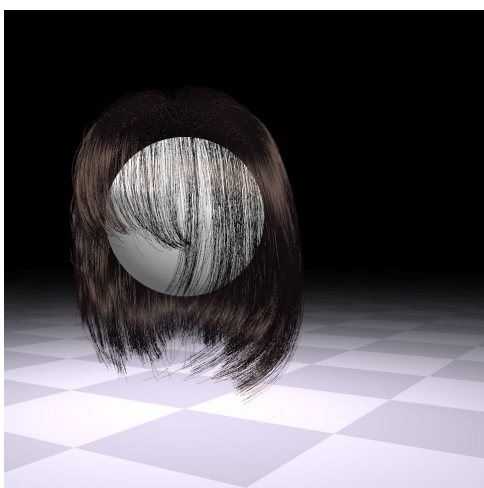


5184 フレーム目 (冷風) ($t = 7.2[s]$)



6480 フレーム目 ($t = 9[s]$)

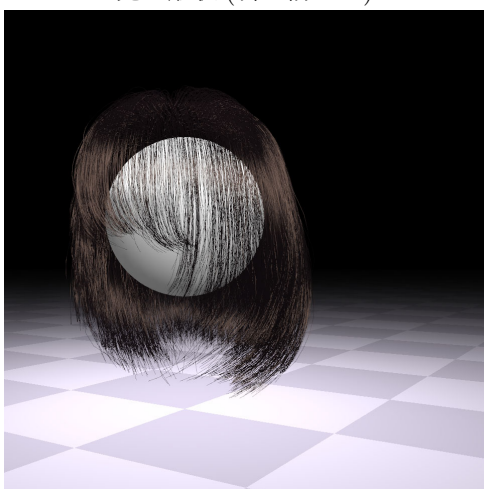
図 6.12: シーン 3 : 温度変化



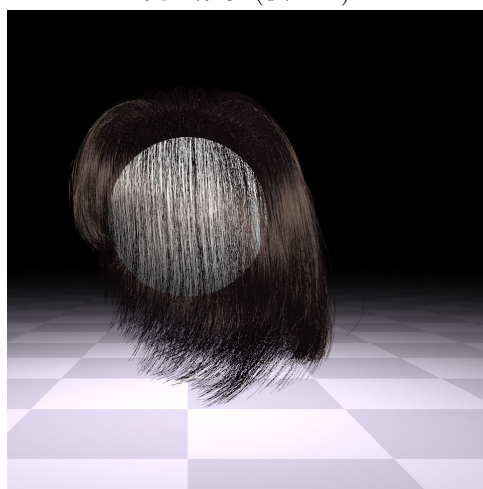
元の形状 (斜め前から)



元の形状 (横から)

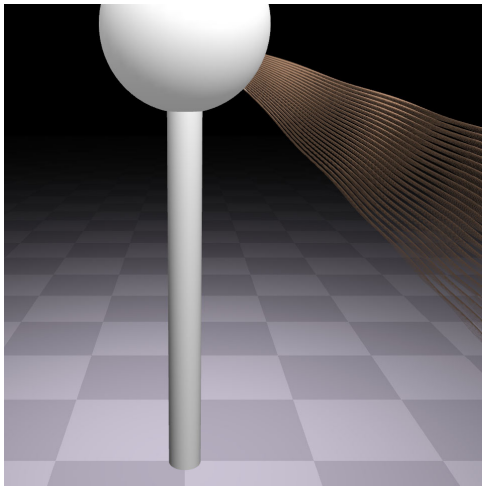


シミュレーション後の形状 (斜め前から)

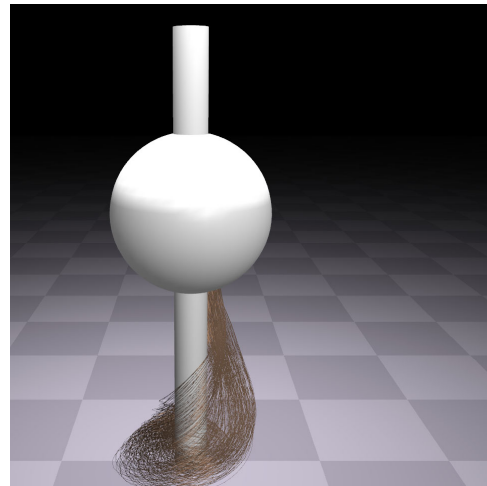


シミュレーション後の形状 (横から)

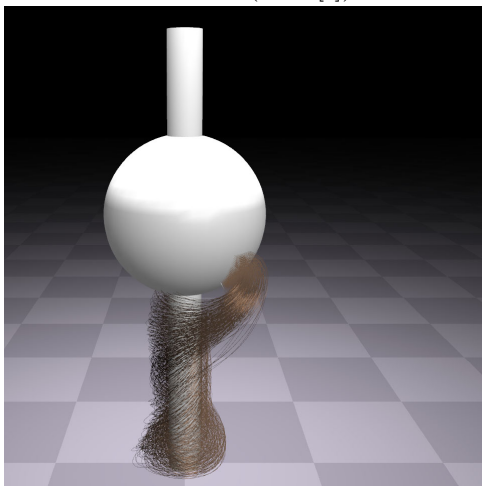
図 6.13: シーン 3 : 元の形状とシミュレーション後の形状の比較



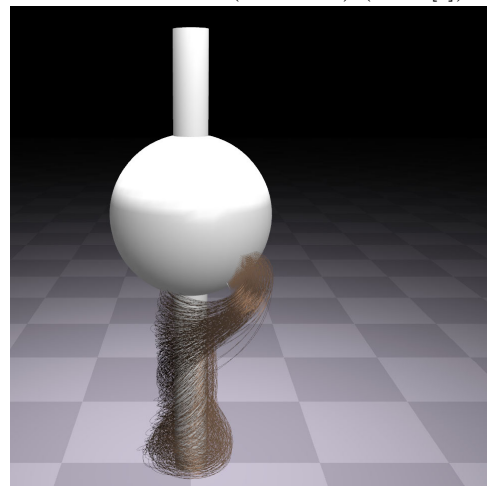
初期状態 ($t = 0[s]$)



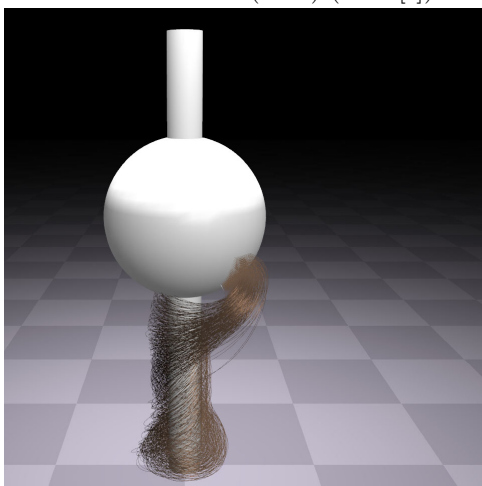
2880 フレーム目 (巻きつけ) ($t = 4[s]$)



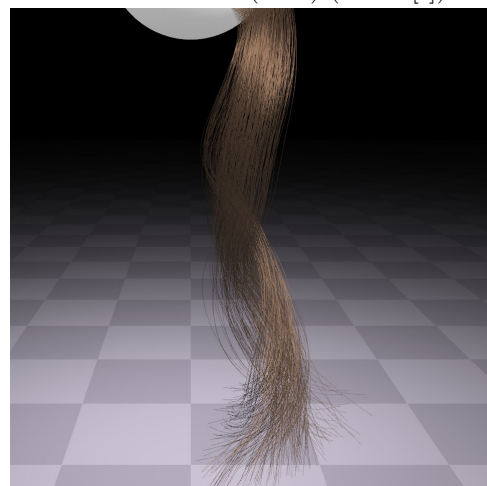
5760 フレーム目 (温風) ($t = 8[s]$)



8640 フレーム目 (温風) ($t = 12[s]$)

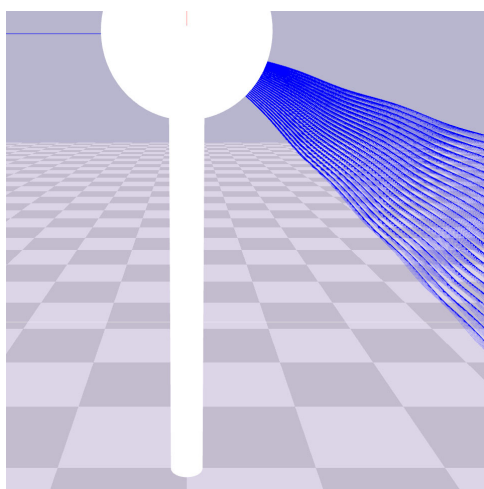


11520 フレーム目 (冷風) ($t = 16[s]$)

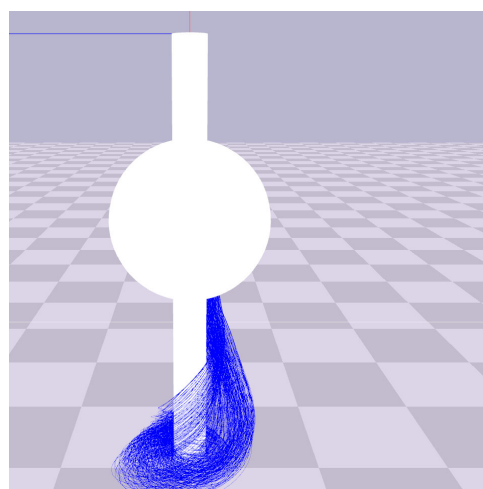


14400 フレーム目 ($t = 20[s]$)

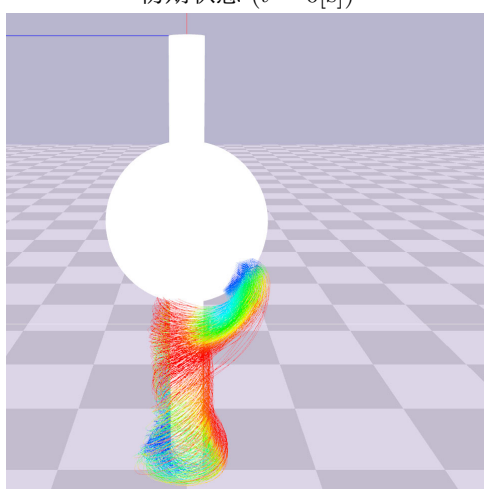
図 6.14: シーン 4 : レンダリング結果



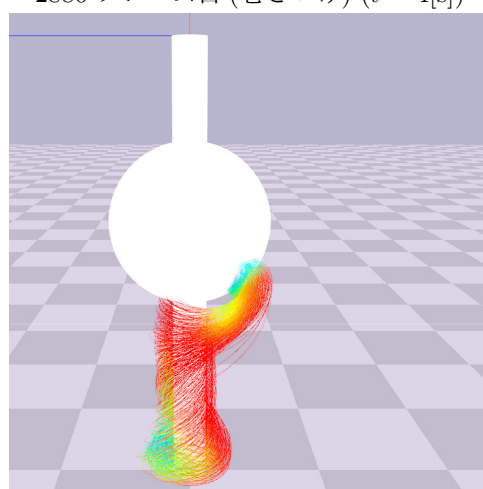
初期状態 ($t = 0[s]$)



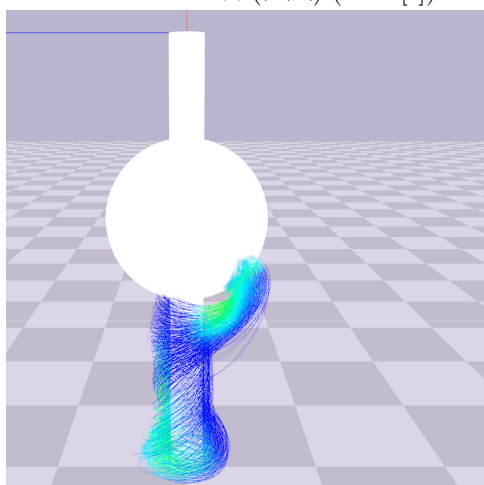
2880 フレーム目 (巻きつけ) ($t = 4[s]$)



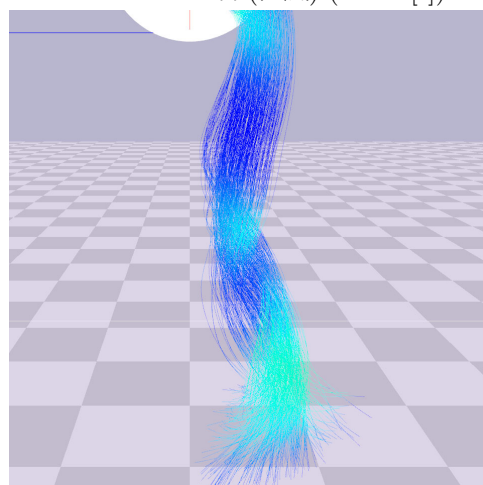
5760 フレーム目 (温風) ($t = 8[s]$)



8640 フレーム目 (温風) ($t = 12[s]$)



11520 フレーム目 (冷風) ($t = 16[s]$)

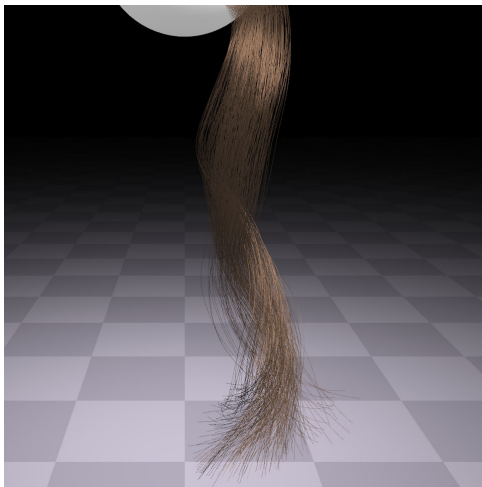


14400 フレーム目 ($t = 20[s]$)

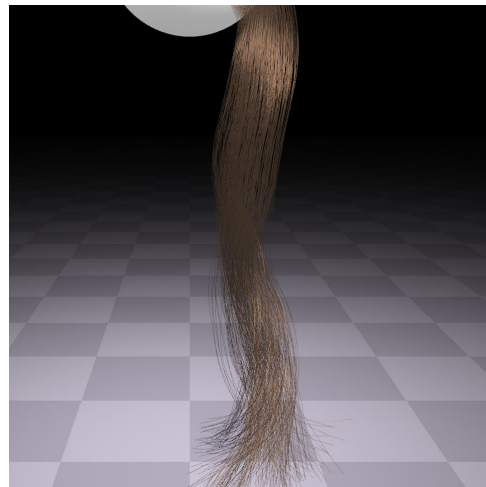
図 6.15: シーン 4 : 温度変化



元の形状



温風 6 秒間のあと冷風 4 秒間



温風のみ 10 秒間

図 6.16: シーン 4 : カール形状の比較

第7章 結論

本研究では、空気と毛髪の相互作用と毛髪内部の側鎖結合から毛髪の塑性変形運動をシミュレーションする手法を提案した。加えて、空気と毛髪の相互作用では、空気の流れが毛髪に乱されることによって生じる渦を制御することによって、毛髪のなびき運動を制御する手法を提案した。実験では、ドライヤーを模した熱風との相互作用計算によって温度が拡散し、それにより毛髪の側鎖結合の結合率が変動することで、塑性変形が発生する工程を再現できることが確かめられた。また、Vorticity Confinement とウェーブレット解析を用いた渦の制御では、抽出する渦の大きさのパラメータを操作することによって、毛髪のなびき運動のふるまいを調整できることが確認できた。ストレートヘアとカールヘアで実験を行ったところ、毛髪の形状によらず提案手法が有効であることが確認でき、粘性を変更することによって毛髪のまとまり具合を調整することが可能であることが確かめられた。

提案手法では相互作用計算に粒子法による流体シミュレーションを用いている。毛髪同士の相互作用計算と毛髪と空気の相互作用計算を単一の流体シミュレーションで同時に処理することによって、シミュレーションシステム全体の複雑化を防ぎ、同時に計算コストも抑制する。さらに、粒子法は問題を個々の粒子に分割できるため並列化に適しており、並列演算が可能な GPU を活用することによって高速な計算を可能とした。提案手法は、美容師・理容師訓練用シミュレータへの精度向上や毛髪形状の新たなモデリング手法、映画・ゲームなどのより現実的な映像表現への応用が期待できる。

一方で、提案手法には様々な問題点が残されている。提案手法では毛髪形状を球形の粒子を用いて離散化しているために、きわめて細い毛髪形状 (0.08[mm] 程度) を表現することができない。本来であれば毛髪間に空いた僅かな隙間に空気が入り込み、空気の移流や温度の拡散が起きているが、提案手法ではそのような空気の侵入を再現することができない。また、毛髪を流体として扱っているために実際の摩擦現象や接触を考慮しておらず、シーンによっては毛髪同士のすり抜けを許してしまったり、摩擦により生じる複雑な絡み合い現象を再現することができない。塑性変形シミュレーションでは、ユーザが設定すべきパラメータが多く、個々の毛髪の性質に対する設定が直感的ではないという問題点がある。ユーザの負担軽減のため、容易に計測できる指標を用いて毛髪のパラメータを設定できることが望ましい。毛髪のなびき運動の制御に用いる乱流シミュレーションでは、既に存在している渦を強調している。そのため存在しない新たな渦を生成することは出来ず、強調の度合いも最初から存在した渦に強く依存するため、より直感的な渦の制御には他の手法の検討も必要である。本研究では高速な演算を可能とする GPU によって実装を行った。しかしながら計算時間の観点では、実際の毛髪量 (約 10 万本) をシミュレーションすることは難しい。特にウェーブレット解析の計算コストが大きく、リアルタイムシミュレーションに近づけるためには更なる高速化の工夫が必要である。

今後の研究としては、これらの問題を解決するほか、本研究では扱わなかった外部流体・側鎖結合の考慮が挙げられる。本研究では外部流体として空気のみ、さらに変動する側鎖結合として水素結合のみを考慮していた。より複雑なシーンのシミュレーションを行うにはこれら以外の要素の計算も必要であり、本研究では考慮していなかった温度による空気の体積変化や温度によらず発生する水分の蒸散現象などを考慮する必要もあると考えられる。また、本物の毛髪を用いて実験を行い、提案手法で設定した仮定を検証することも今後の課題として挙げられる。

謝辞

本研究を進めるにあたり，大変丁寧に御指導をいただきました筑波大学図書館情報メディア系 藤澤誠助教に心から感謝申し上げます．研究発表の練習の場において，研究内容および発表について様々なご助言をいただきました筑波大学図書館情報メディア系 三河正彦准教授に深く感謝申し上げます．髪型アセットデータをご提供いただきました株式会社デジタル・フロンティア 後藤浩之氏，毛髪の写真をご提供いただきました舟木佳奈子氏に厚く御礼申し上げます．そして，日々の研究活動だけでなく日常生活でも支えていただきました，物理ベースコンピュータグラフィクス研究室ならびにソーシャルロボット研究室的の皆様，家族と友人たちに深く感謝いたします．

参考文献

- [1] C. R. Robbins. *The Chemical and Physical Behavior of Human Hair*. Springer-Verlag, 2012.
- [2] 出村佑史, 藤澤誠, 三河正彦. 側鎖結合を考慮した毛髪の塑性変形シミュレーション. 画像電子学会誌, Vol. 47, No. 4, pp. 447–453, 2018.
- [3] S. Hadap and N. M.-Thalmann. Modeling Dynamic Hair as a Continuum. *Computer Graphics Forum*, Vol. 20, No. 3, pp. 329–338, 2001.
- [4] K. Furukawa and S. Nakata. Automatic Generation of the Periodic Hair Motion of 3D Characters for Anime Production. In *Proc. of Entertainment Computing – ICEC 2018*, pp. 289–292, 2018.
- [5] K. Wilson, A. McAdams, H. Leo, and M. Simmons. Simulating Wind Effects on Cloth and Hair in Disney’s Frozen. In *ACM SIGGRAPH 2014 Talks*, pp. 48:1–48:1, 2014.
- [6] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position Based Dynamics. *Journal of Visual Communication and Image Representation*, Vol. 18, No. 2, pp. 109–118, 2007.
- [7] 林本法也, 三谷純, 五十嵐健夫. ヘアカットの教習を支援するための対話的なヘアシミュレータの開発. Visual Computing/グラフィクスと CAD 合同シンポジウム 2010 予稿集 (poster), pp. 35:1–35:6, 2010.
- [8] K. Ward, F. Bertails, T.-Y. Kim, S. R. Marschner, M.-P. Cani, and M. C. Lin. A Survey on Hair Modeling: Styling, Simulation, and Rendering. *IEEE Trans. on Visualization and Computer Graphics*, Vol. 13, No. 2, pp. 213–234, 2007.
- [9] M. Müller, T.-Y. Kim, and N. Chentanez. Fast Simulation of Inextensible Hair and Fur. In *Proc. of 9th Workshop on Virtual Reality Interaction and Physical Simulation*, pp. 39–44, 2012.
- [10] N. Umetani, R. Schmidt, and J. Stam. Position-based Elastic Rods. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 21–30, 2014.
- [11] T. Kugelstadt and E. Schömer. Position and Orientation Based Cosserat Rods. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 169–178, 2016.
- [12] C. Deul, T. Kugelstadt, M. Weiler, and J. Bender. Direct Position-based Solver for Stiff Rods. *Computer Graphics Forum*, Vol. 37, No. 6, pp. 313–324, 2018.
- [13] W. Rungjiratananon, Y. Kanamori, and T. Nishita. Wetting Effects in Hair Simulation. *Computer Graphics Forum*, Vol. 31, No. 7, pp. 1993–2002, 2012.

- [14] Y. Fei, H. T. Maia, C. Batty, C. Zheng, and E. Grinspun. A Multi-scale Model for Simulating Liquid-hair Interactions. *ACM Trans. on Graphics*, Vol. 36, No. 4, pp. 56:1–56:17, 2017.
- [15] D. M. Kaufman, R. Tamstorf, B. Smith, J.-M. Aubry, and E. Grinspun. Adaptive Nonlinearity for Collisions in Complex Rod Assemblies. *ACM Trans. on Graphics*, Vol. 33, No. 4, pp. 123:1–123:12, 2014.
- [16] C. Jiang, T. Gast, and J. Teran. Anisotropic Elastoplasticity for Cloth, Knit and Hair Frictional Contact. *ACM Trans. on Graphics*, Vol. 36, No. 4, pp. 152:1–152:14, 2017.
- [17] J. J. Monaghan. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics*, Vol. 30, No. 1, pp. 543–574, 1992.
- [18] A. McAdams, A. Selle, K. Ward, E. Sifakis, and J. Teran. Detail Preserving Continuum Simulation of Straight Hair. *ACM Trans. on Graphics*, Vol. 28, No. 3, pp. 62:1–62:6, 2009.
- [19] M. Macklin and M. Müller. Position Based Fluids. *ACM Trans. on Graphics*, Vol. 32, No. 4, pp. 104:1–104:12, 2013.
- [20] R. Fedkiw, J. Stam, and H. W. Jensen. Visual Simulation of Smoke. In *Proc. of 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 15–22, 2001.
- [21] J. Bender, M. Müller, M. A. Otaduy, M. Teschner, and M. Macklin. A Survey on Position-based Simulation Methods in Computer Graphics. *Computer Graphics Forum*, Vol. 33, No. 6, pp. 228–251, 2014.
- [22] Y. Yu, W. Yang, B. Wang, and M. A. Meyers. Structure and Mechanical Behavior of Human Hair. *Materials Science and Engineering: C*, Vol. 73, pp. 152–163, 2017.
- [23] Y. Lee, Y.-D. Kim, H.-J. Hyun, L.-Q. Pi, X. Jin, and W.-S. Lee. Hair Shaft Damage from Heat and Drying Time of Hair Dryer. *Annals of dermatology*, Vol. 23, No. 4, pp. 455–462, 2011.
- [24] M. Müller, D. Charypar, and M. Gross. Particle-based Fluid Simulation for Interactive Applications. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 154–159, 2003.
- [25] H. Schechter and R. Bridson. Ghost SPH for Animating Water. *ACM Trans. on Graphics*, Vol. 31, No. 4, pp. 61:1–61:8, 2012.
- [26] H. Hochstetter and A. Kolb. Evaporation and Condensation of SPH-based Fluids. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 3:1–3:9, 2017.
- [27] M. Fujisawa, G. Mimura, T. Amano, J. Miyazaki, and H. Kato. A Fast Simulation Method Using SPH and Wavelet for Sub-particle-scale Turbulent Flow. In *Proc. of Pacific Conference on Computer Graphics and Applications - Short Papers*, pp. 67–72, 2011.

- [28] S. Green. Particle Simulation Using CUDA. <http://developer.download.nvidia.com/assets/cuda/files/particles.pdf>, 2010. Accessed: 2020-01-06.
- [29] J. T. Kajiya and T. L. Kay. Rendering Fur with Three Dimensional Textures. In *Proc. of 16th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 271–280, 1989.
- [30] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner. Versatile Rigid-fluid Coupling for Incompressible SPH. *ACM Trans. on Graphics*, Vol. 31, No. 4, 2012.
- [31] C. Gissler, S. Band, A. Peer, M. Ihmsen, and M. Teschner. Generalized Drag Force for Particle-based Simulations. *Computers & Graphics*, Vol. 69, pp. 1–11, 2017.
- [32] 金谷一朗. 3D-CG プログラマーのためのクォータニオン入門. 工学社, 2004.

付 録 A 四元数を用いた計算

ここでは本研究で用いる四元数の定義と基本的な計算方法を示す．なお，本付録で使用する記号の意味は論文本文での記号用法 (表 1.1) とは異なる．

四元数の定義

四元数 (Quaternion) は四つの成分を含む超複素数の一種である [32]．実数 $a, b, c, d \in \mathbb{R}$ と $i^2 = j^2 = k^2 = ijk = -1$ を満たす虚数単位 $i, j, k \in \mathbb{S}$ を用いると，四元数 q は以下の式によって定義される．

$$q = a + bi + cj + dk \quad (\text{A.1})$$

ここで，式 (A.1) の a をスカラー部 (実数部)， $bi + cj + dk$ をベクトル部 (虚数部) と呼ぶ．

四元数の積・共役四元数・単位四元数・逆四元数

四元数同士は掛け算ができ，その結果もまた四元数となる．四元数 $q_1 = a_1 + b_1i + c_1j + d_1k$ と $q_2 = a_2 + b_2i + c_2j + d_2k$ の積は以下の式で計算できる．

$$\begin{aligned} q = & a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2 \\ & + (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)i \\ & + (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)j \\ & + (a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)k \end{aligned} \quad (\text{A.2})$$

四元数 $q = a + bi + cj + dk$ には複素共役 \bar{q} が定義されており，ベクトル部の符号を反転させて計算することができる ($\bar{q} = a - bi - ci - di$)．また，ノルム $\|q\| = \sqrt{a^2 + b^2 + c^2 + d^2} = 1$ となる四元数を単位四元数と呼ぶ．単位四元数は共役もまた単位四元数であり，単位四元数とその共役の積は 1 になる．さらに，このように積が 1 となる四元数を逆四元数と呼ぶ．

回転を表す四元数

単位四元数は，三次元空間での回転を表すことができる．四元数で回転を扱うことによって，保持する情報量が減り (四元数は 4 要素，回転行列の場合は $3 \times 3 = 9$ 要素)，オイラー角を用いた方法で問題となるジンバルロックが発生しないという利点がある．以下に任意の単位ベクトル $(x, y, z)^T$ を軸として $\theta[\text{rad}]$ 回転させる四元数を計算する式を示す．

$$q = \cos \frac{\theta}{2} + xi \sin \frac{\theta}{2} + yj \sin \frac{\theta}{2} + zk \sin \frac{\theta}{2} \quad (\text{A.3})$$

ベクトルや回転を表す四元数を，回転を表す四元数で回転させることが可能である．ベクトル \mathbf{r} を四元数 q で回転させる式を式 (A.4)，四元数 u を四元数 q で回転させる式を式 (A.5) に示す．

$$\mathbf{r}' = \Im(q\mathbf{r}\bar{q}) \quad (\text{A.4})$$

$$u' = qu \quad (\text{A.5})$$

ここで, u', \mathbf{r}' はそれぞれ回転後の値, r はスカラー部を 0, ベクトル部を \mathbf{r} にした四元数である.

姿勢を表す四元数

グローバル座標系の基底となるベクトルを用意し, マテリアルフレームの基底ベクトルになるように回転させる四元数 q を定義すると, q はそのマテリアルフレームの姿勢を表す. 具体的には式 (A.4) より, グローバル座標系の基底を \mathbf{e} , マテリアルフレームの基底を \mathbf{d} とすると, q は以下の式を満たす.

$$\mathbf{d} = \Im(q\mathbf{e}\bar{q}) \quad (\text{A.6})$$

ここで, \mathbf{e} はスカラー部を 0, ベクトル部を \mathbf{e} にした四元数である.

グローバル座標系の基底 \mathbf{e} とマテリアルフレームの基底 \mathbf{d} から, 姿勢を表す四元数 q を逆算することができる (計算方法は一つではない). ここからは姿勢を表す四元数 q の計算方法について説明する. まず, 式 (A.3) に代入するための回転軸 $(x, y, z)^\top$ と回転角情報 $\cos \frac{\theta}{2}, \sin \frac{\theta}{2}$ を決定する. 基底はそれぞれ単位ベクトルであるため, $\cos \theta$ はベクトルの内積をとることで計算できる. 求めた $\cos \theta$ から以下の式で $\cos \frac{\theta}{2}, \sin \frac{\theta}{2}$ を計算する.

$$\sin \frac{\theta}{2} = \sqrt{\frac{1 - \cos \theta}{2}} \quad (\text{A.7})$$

$$\cos \frac{\theta}{2} = \sqrt{\frac{1 + \cos \theta}{2}} \quad (\text{A.8})$$

$(x, y, z)^\top$ はベクトルの外積と同じ向きであるため, 外積を正規化することで計算できる. 以上を式 (A.3) に代入することで q を計算することができる. ただし, ベクトルが平行 (またはほぼ平行) だった場合は $(x, y, z)^\top$ が一意に定まらないか, 計算誤差により不安定化する. この問題の対策は本文の 5.3.2 項にて述べた通りである.

四元数と角速度

ある姿勢 q が角速度 $\boldsymbol{\omega}$ で動いていたときの姿勢 q の時間変化の計算式を以下に示す.

$$\frac{dq}{dt} = \frac{1}{2}\boldsymbol{\omega}q \quad (\text{A.9})$$

ここで, $\boldsymbol{\omega}$ はスカラー部を 0, ベクトル部を $\boldsymbol{\omega}$ にした四元数である.